OMRON

Sysmac Library

User's Manual
for EtherCAT® G5 Series Library
SYSMAC-XR004

SYSMAC
always in control

W548-E1-05

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 EtherCAT G5 Series 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 EtherCAT G5 Series Library is used to initialize the absolute encoder, back up and restore the parameters for an OMRON G5-series Servo Drive with built-in EtherCAT communications.

You can use this library to reduce manpower of programming when implementing the processing for a Servo Drive.

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

Introduction 1
Features of the Library 1
Intended Audience 1
Applicable Products 1

Manual Structure 2
Special Information 2

CONTENTS 4

Terms and Conditions Agreement 6
Warranty, Limitations of Liability 6
Application Considerations 7
Disclaimers 7

Safety Precautions 8
Definition of Precautionary Information 8
Symbols 8
Cautions 9

Precautions for Correct Use 10

Related Manuals 11

Revision History 14

Procedure to Use Sysmac Libraries 15
Procedure to Use Sysmac Libraries Installed Using the Installer 16
Procedure to Use Sysmac Libraries Uploaded from a CPU Unit or an Industrial PC 20

Common Specifications of Function Blocks 23
Common Variables 24
Precautions 30

Individual Specifications of Function Blocks 31
InitEncoder_G5 32
Backup_G5 40
Restore_G5 55

Appendix 69
Referring to Library Information 70
Referring to Function Block and Function Source Codes 73

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.	

Precautions for Correct Use

Using the Library

- When you use the library, functions or function blocks that are not described in the library manual may be displayed on the Sysmac Studio. Do not use functions or function blocks that are not described in the manual.
- Do not change the names and the contents of any object list.
- Setting the *NoProductChk* input variable to TRUE allows you to execute restore even when the product code in the backup data differs from the product code of the restore destination. Make sure that the restore destination is a G5-series Servo Drive before you attempt to restore the parameters.
- Do not attempt to turn OFF the power supply to the Controller and EtherCAT slaves or stop the EtherCAT communications until the processing for this function block ends normally or ends in an error.
- Execute restore while the Servo is OFF.
- After completion of Restore_G5 function block, cycle the control power supply to the G5-series Servo Drive to apply the restored parameters.

Using Sample Programming

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

Related Manuals

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

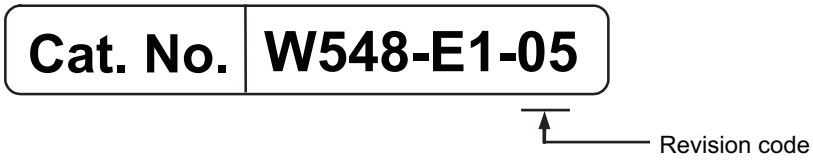
Manual name	Cat. No.	Model numbers	Application	Description
NX-series CPU Unit Hardware User's Manual	W535	NX701-□□□□	Learning the basic specifications of the NX-series NX701 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided	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.

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 2015	Original production
02	December 2015	Corrected mistakes.
03	July 2016	Changed the manual name.
04	November 2016	Changed the manual name.
05	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.

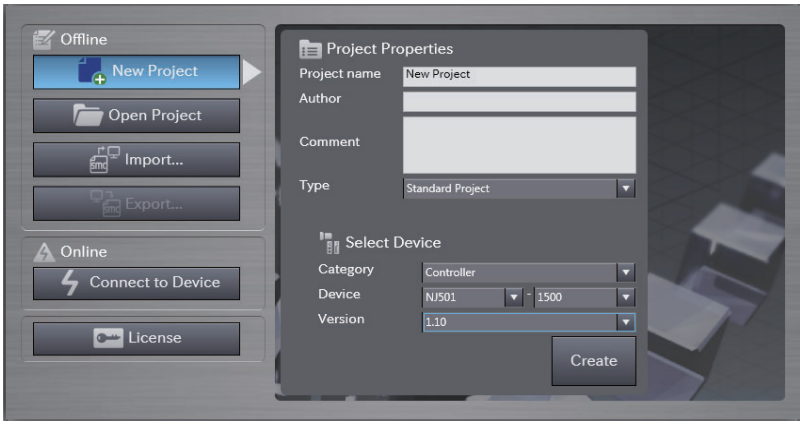
- Using newly installed Sysmac Libraries
- Using upgraded Sysmac Libraries

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.

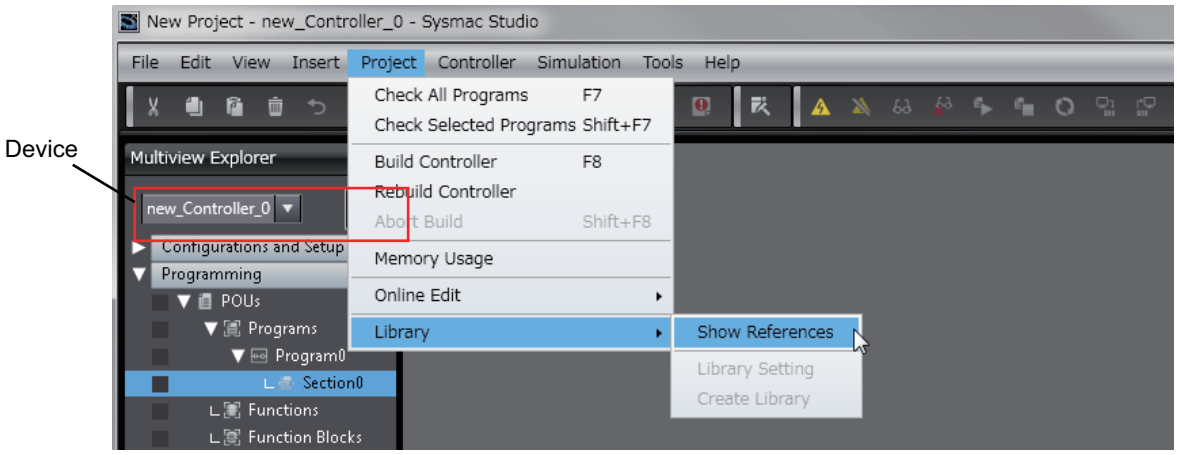


Precautions for Correct Use

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.

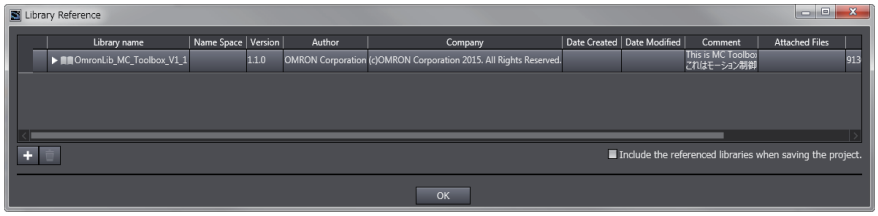
2 Select **Project** – **Library** – **Show References**.



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.



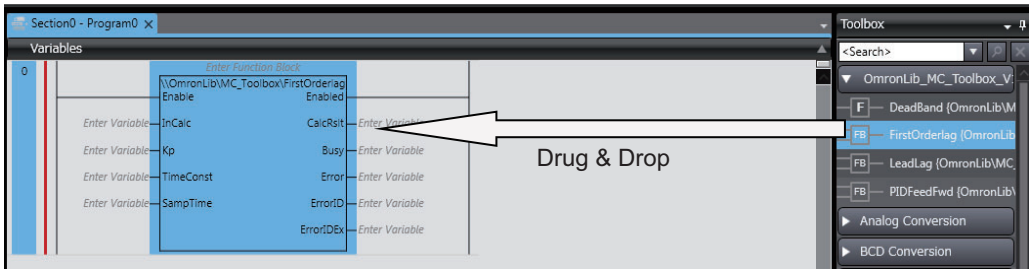
The Sysmac Library file is read into the project.

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

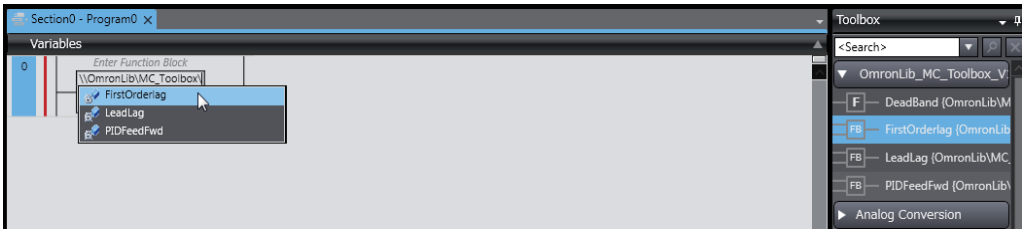
For the procedure for adding and setting libraries in the above screen, refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504).

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

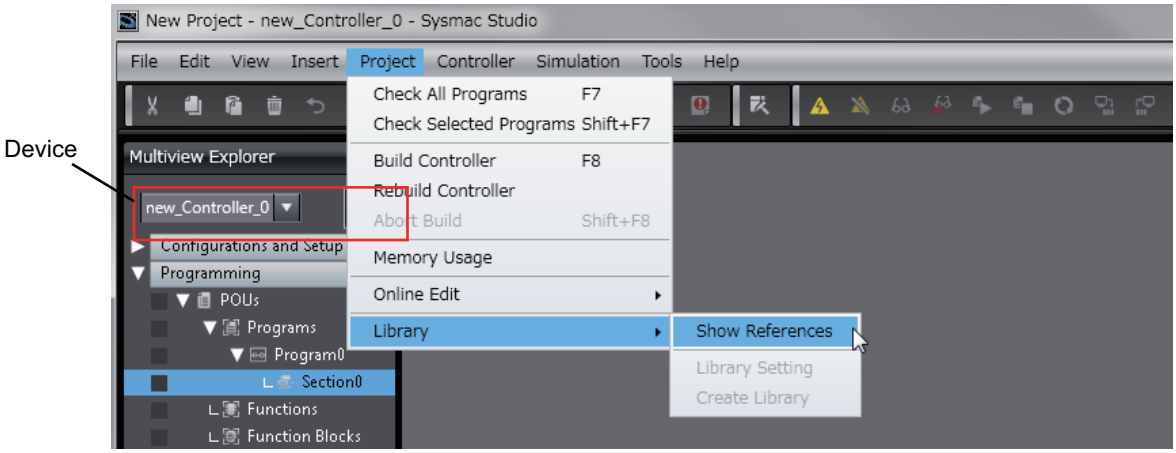
After you upgrade the Sysmac Studio, check all programs and make sure that there is no error of the program check results on the Build Tab Page.

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.

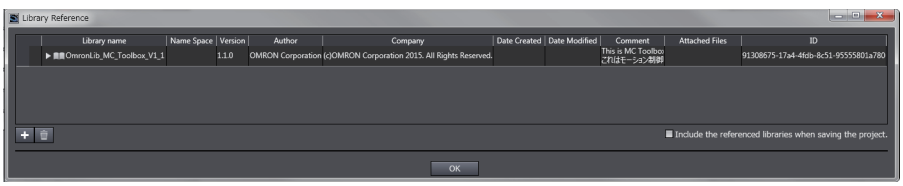
2 Select **Project** – **Library** – **Show References**.



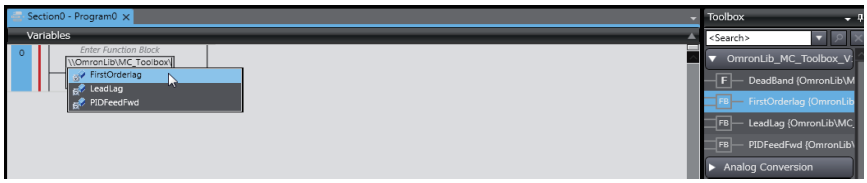
Precautions for Correct Use

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

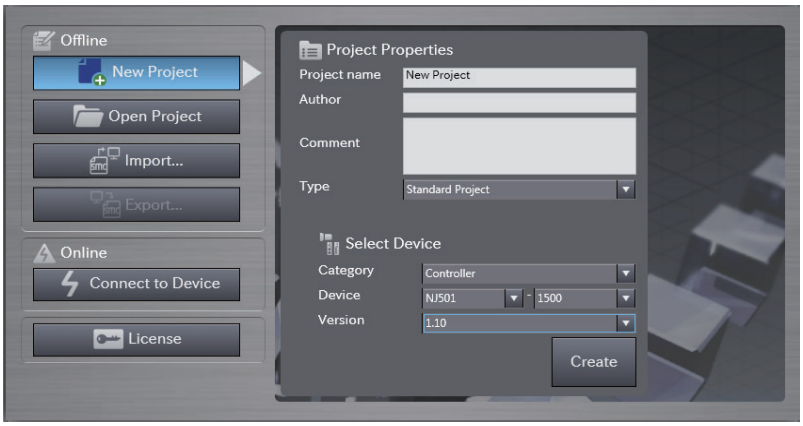
You can use Sysmac Libraries uploaded from a CPU Unit or an Industrial PC to your computer if they are not installed.

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.



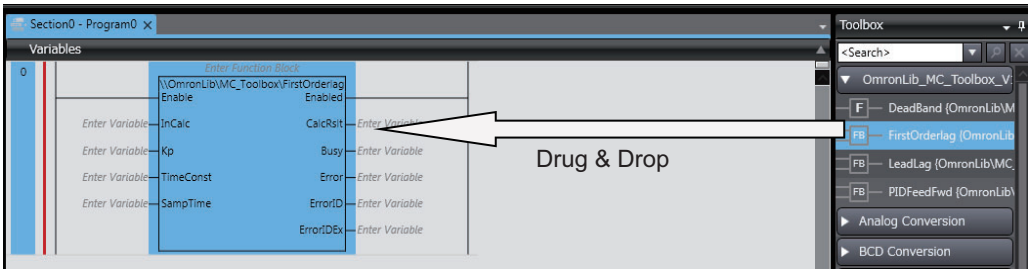
2 Connect the computer to the CPU Unit or the Industrial PC and place it online.

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

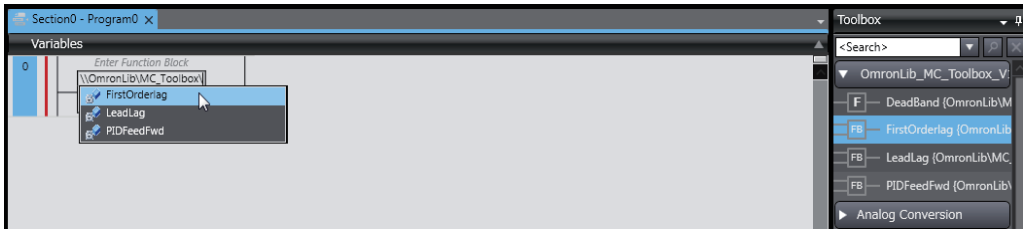
Now, when you select the Ladder Editor or ST Editor, the function blocks and functions included in the Sysmac Library used in the uploaded POUs appear in the Toolbox.

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

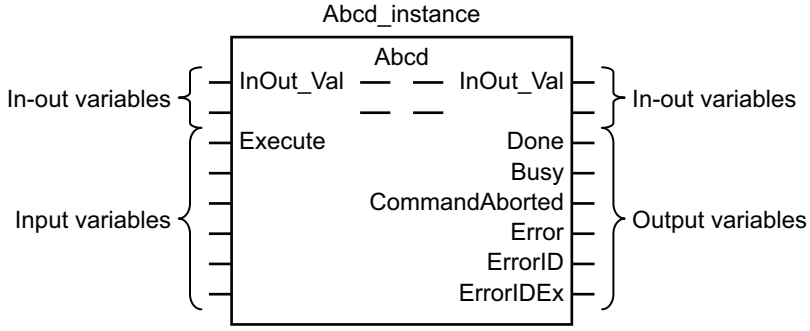
Common input variables and output variables used in functions and function blocks are as follows.

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.

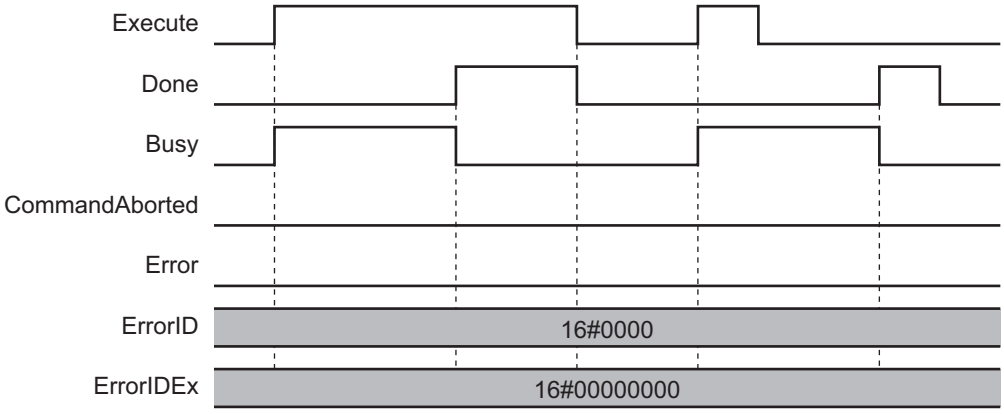


- If *Execute* is TRUE and *Done*, *CommandAborted*, or *Error* changes to TRUE, *Done*, *Command-Aborted*, and *Error* changes to FALSE when *Execute* is changed to FALSE.
- If *Execute* is FALSE and *Done*, *CommandAborted*, or *Error* changes to TRUE, *Done*,*Command-Aborted*, and *Error* changes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

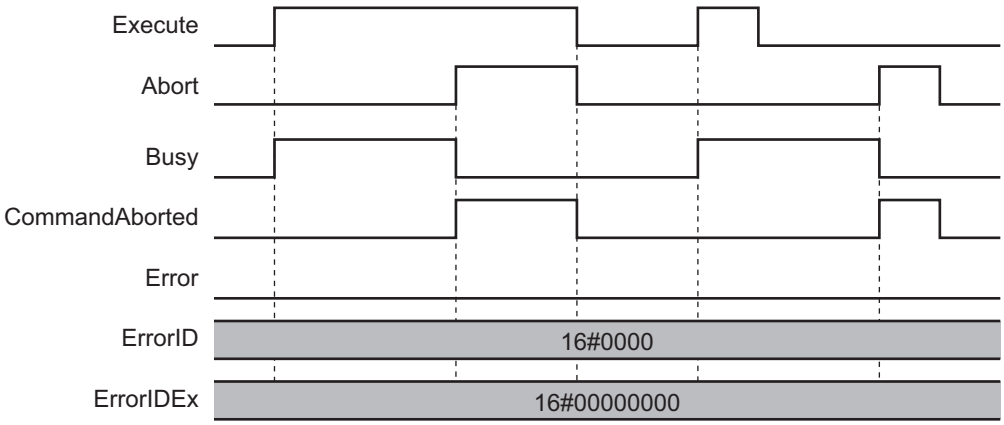
Timing Charts

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

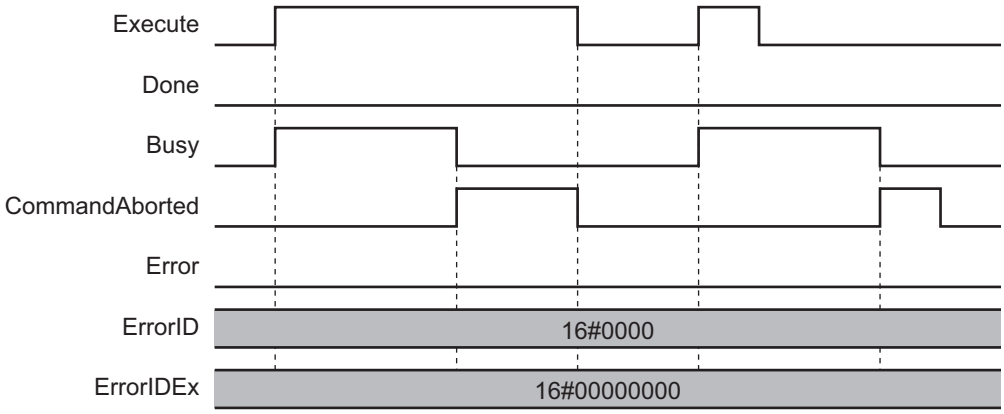
● Normal End



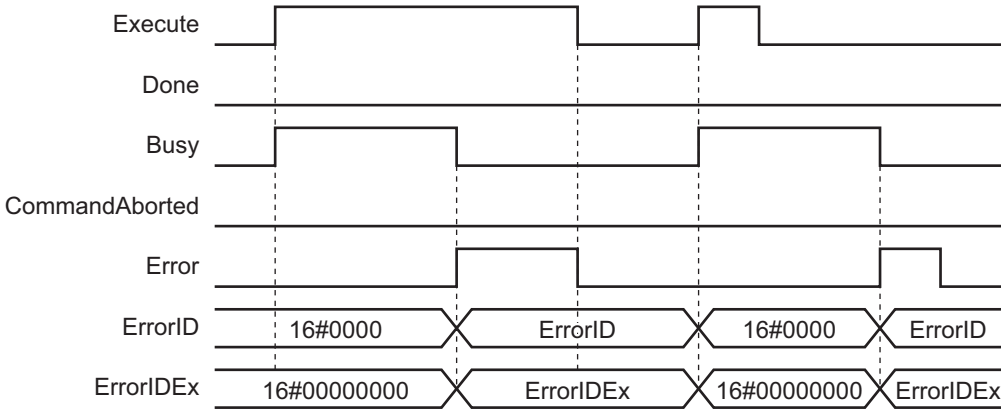
● Canceled Execution



Aborted Execution

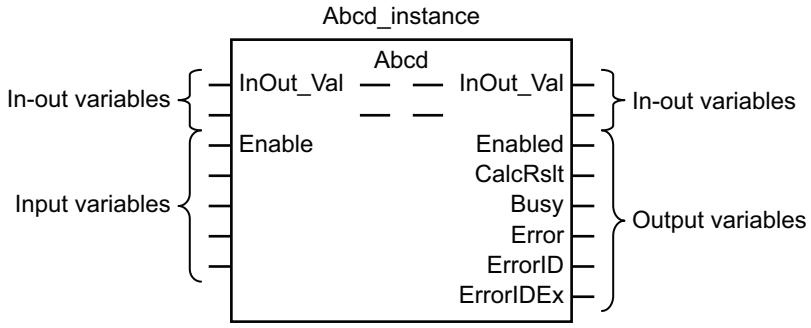


Errors



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.

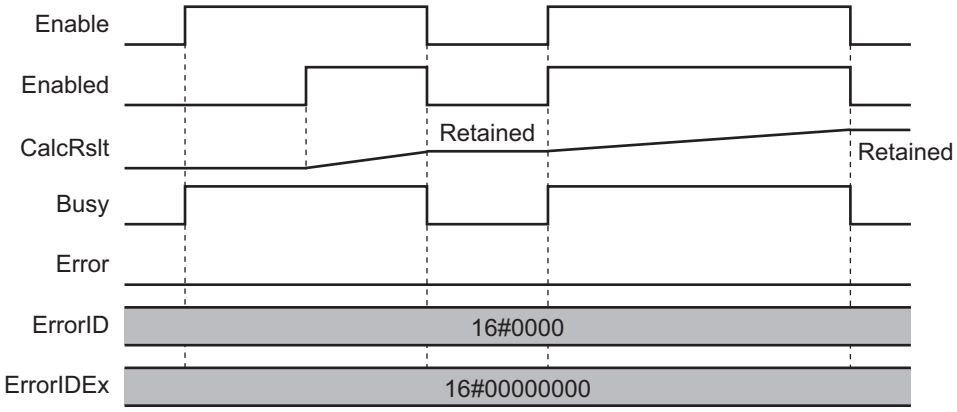


- If an error occurs, the relevant error code and expansion error code are set in *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code). The error codes are retained even after *Error* changes to FALSE, but *ErrorID* is set to 16#0000 and *ErrorIDEx* is set to 16#0000 0000 when *Enable* changes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

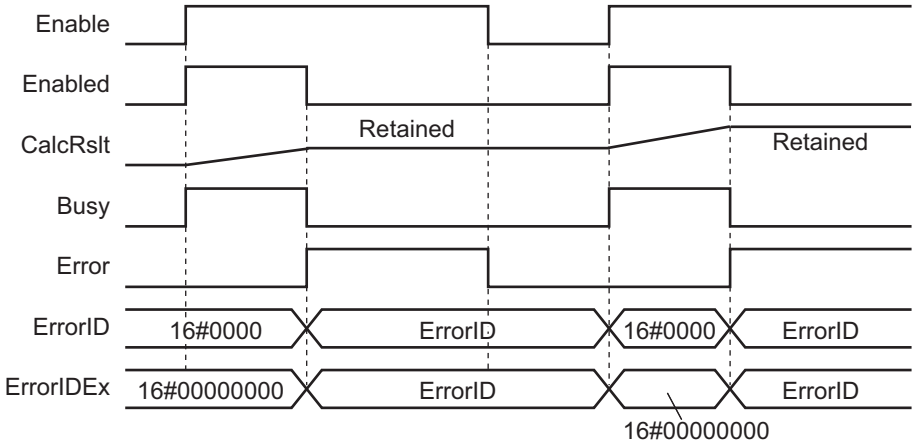
Timing Charts

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.

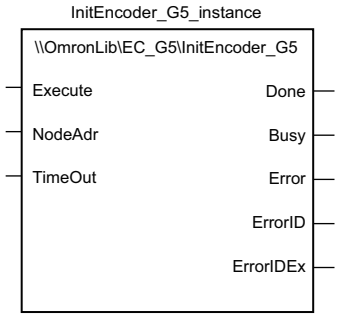
For details on re-execution, refer to the motion control user's manual.

Individual Specifications of Function Blocks

Function block name	Name	Page
InitEncoder_G5	G5-series Servo Drive Absolute Encoder Initialize	P.32
Backup_G5	G5-series Servo Drive Parameter Backup	P.40
Restore_G5	G5-series Servo Drive Parameter Restore	P.55

InitEncoder_G5

The InitEncoder_G5 function block initializes the absolute encoder (clears the multi-rotation counter for the absolute encoder) for a G5-series Servo Drive.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
InitEncoder_G5	G5-series Servo Drive Absolute Encoder Initialize	FB		InitEncoder_G5_instance(Execute, NodeAdr, TimeOut, Done, Busy, Error, ErrorID, ErrorIDEx);

Function Block and Function Information

Item	Description
Library file name	OmronLib_EC_G5_V1_1.slr
Namespace	OmronLib\EC_G5
Function block and function number	00007
Source code published/not published	Not published
Function block and function version	1.01

Compatible Models

Item	Name	Model numbers	Version
Device	G5-series Servo Drive	R88D-KN□-ECT	Version 2.10 or later
	G5-series Servo Motor (with absolute encoder)	R88M-K□□□□□T(-□) R88M-K□□□□□S(-□) R88M-K□□□□□C(-□)	-

Variables

Name	Meaning	I/O	Description	Valid range	Unit	Default
Execute	Execute	Input	TRUE: Executes the instruction. FALSE: Does not execute the instruction.	TRUE or FALSE	---	FALSE
NodeAdr	Slave Node Address	Input	Node address of the Servo Drive to initialize	1 to 512*1		1
TimeOut	Timeout time	Input	0: 2.0 s 1 to 65535: 0.1 to 6553.5 s	0 to 65535	0.1 s	0 (2.0 s)
Done	Done	Output	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE or FALSE	---	---
Busy	Executing	Output	TRUE: Execution processing is in progress. FALSE: Execution processing is not in progress.	TRUE or FALSE		---
Error	Error	Output	TRUE: Error end FALSE: Normal end, execution in progress, or execution condition not met	TRUE or FALSE		---
ErrorID	Error Code	Output	This is the error ID for an error end. The value is WORD#16#0 for a normal end.	*2		---
ErrorIDEx	Expansion Error Code	Output	Abort code for SDO access specified by CoE*3.*4	*5		---

*1. The valid range is 1 to 192 for NJ-series CPU Units.

*2. For *ErrorID*, refer to the EC_CoESDOWrite instruction in *Error Codes That You Can Check with ErrorID* in the *i*nstructions reference manual.

*3. CoE stands for CAN application protocol over EtherCAT.

*4. It is output when the value of *ErrorID* is DWORD#16#1804.

*5. Refer to *SDO Abort Codes* in the instructions reference manual for a list of SDO abort codes.

	Boolean	Bit strings				Integers								Real numbers		Times, durations, dates, and text strings				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Execute	OK																			
NodeAdr							OK													
TimeOut							OK													
Done	OK																			
Busy	OK																			
Error	OK																			
ErrorID			OK																	
ErrorIDEx				OK																

Function

This function block initializes the absolute encoder (clears the multi-rotation counter for the absolute encoder) for the specified Servo Drive. When the Servo Drive changes to a Fault state (Error No. 27) after the initialization is completed, cycle the control power supply to the Servo Drive.

To execute this function block, it is required that the object of servo parameter 3015 hex (Operation Switch when Using Absolute Encoder) is 0 (Use as absolute encoder) or 2 (Use as absolute encoder but ignore multi-rotation counter overflow).

After sending the initialization command to the Servo Drive, the command part waits for the response for the time specified with *TimeOut* (Timeout). The value of *ErrorIDEx* is 0 for a normal end.

Precautions for Correct Use

- After completion of the function block, the G5-series Servo Drive changes to a Fault state. Cycle the control power supply to the Servo Drive.
- Execute the function block while the Servo is OFF. If the Servo Drive is assigned to the Motion Control Function Module, make sure that the Axis Disabled (*_MC_AX[***].Status.Disabled*) in the Axis Variable is TRUE before you execute the function block.
- Do not use Linear Motor/AC Servo Drives (R88D-KN□-ECT-L). Make sure that the Servo Drive is a compatible model before executing this function block.
- Do not attempt to turn OFF the power supply to the Controller and EtherCAT slaves or stop the EtherCAT communications until the processing for this function block ends normally or ends in an error.

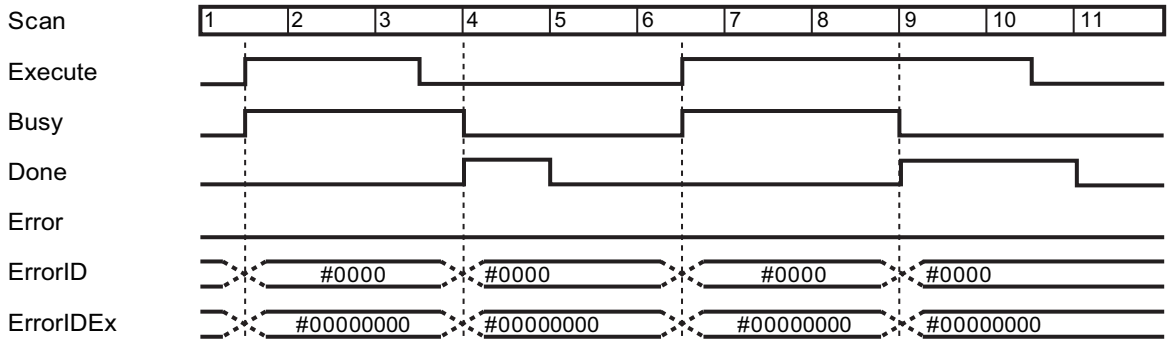
Timing Charts

The following figures show the timing charts for the function block.

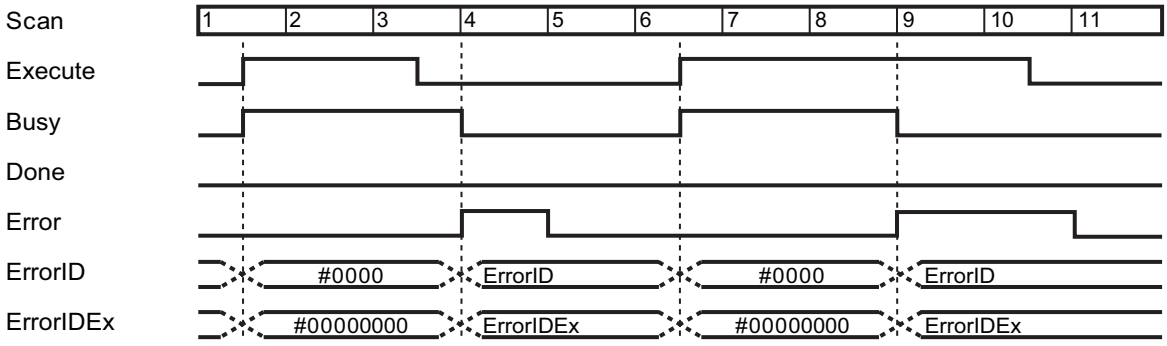
- *Busy* (Executing) changes to TRUE when *Execute* changes to TRUE.
- *Done* changes to TRUE when the initialize process is ended.
- If an error occurs when execution of the function block is in progress, *Error* changes to TRUE and *Busy* (Executing) changes to FALSE.

  You can find out the cause of the error by accessing the values output to *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code).
- If *Execute* changes to FALSE before execution of the function block is ended, *Done* and *Error* are TRUE only for one task period after execution of the function block is ended.
- If *Execute* remains TRUE even after execution of the function block is ended, the output values of *Done* and *Error* are held.

● Timing Chart for Normal End



● Timing Chart for Error End



Additional Information

- This function block initializes absolute encoders by writing 6a64 6165 hex to the object of servo parameter 4102 hex (Absolute Encoder Setup) with SDO mailbox communications.

Precautions for Correct Use

- Execution of this function block will be continued until processing is ended even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is ended. Use this to confirm normal ending of processing.
- This function block executes one EC_CoESDOWrite instruction or one EC_CoESDORead instruction within the part. Refer to the *EC_CoESDOWrite instruction* in the instructions reference manual for the maximum number of instructions that can be executed at the same time.
- An error will occur in the following cases. *Error* will change to TRUE.
  a) The EtherCAT master is not in a state that allows message communications.
  b) The slave specified with *NodeAdr* does not exist.
  c) The slave specified with *NodeAdr* is not in a state that allows communications.
  d) The slave is not in a state that allows writing of parameters.
  e) The number of EtherCAT communications instructions such as EC_CoESDOWrite exceeded the maximum number of instructions that can be executed at the same time.

Related System-defined Variables

Name	Meaning	Data type	Description
_EC_MBXSlavTbl	Message Communications Enabled Slave Table	Array[1..512] OF BOOL	This table indicates the slaves that can perform message communications. Slaves are given in the table in the order of slave node addresses. TRUE: Message communications are possible. FALSE: Message communications are not possible.

Troubleshooting

Error code	Expansion error code	Status	Description	Correction
16#0000	16#00000000	Normal End	---	---
16#3C12	16#00000001	VendorID Not Matched	The device is not supported by this function block (A non-OMRON device is specified).	Check that the specified device is a G5-series Servo Drive.
16#3C12	16#00000002	Unsupported Unit Version	The function block was executed for an unsupported G5-series Servo Drive.	Check the unit version of the G5-series Servo Drive.
16#3C12	16#00000003	Initialization Failure	The encoder initialization was not completed within one minute.	• Execute the function block again. The encoder was located near the position of one rotation at the start of the initialization and fell in a state of a multi-rotation due to vibration etc. This may be assumed as an initialization failure when the Servo Drive checks the completion of the initialization. • Replace the G5-series Servo Drive.
16#3C12	16#00000004	Unsupported Device	The device is not supported by this function block.	Check that the specified device is a G5-series Servo Drive.
16#0400	16#00000000	Input Value Out of Range	The value of *NodeAdr* is out of range.	Set the value of *NodeAdr* correctly.
16#1800	16#00000000	EtherCAT Communications Error	The EtherCAT network is not in a usable status.	Check the operation status of the EtherCAT network by checking the status of the EtherCAT master. Use this information to correct the cause of the problem.
16#1801	16#00000000	EtherCAT Slave Does Not Respond	The target slave does not exist.	Specify an existing node address.
			The target slave is not in an operating condition.	Check the status of the target EtherCAT slave. Make sure that the target slave is in a usable status.
16#1802	16#00000000	EtherCAT Timeout	Communications with the target slave timed out.	Check the operating status of the target slave and correct the cause of the problem.
16#1803	16#00000000	Reception Buffer Overflow	The receive data from an EtherCAT slave overflowed the receive buffer.	Set the size of the reception buffer to a value larger than the size of the receive data from the slave.
16#1804	SDO Abort Code[*1]	SDO Abort Error	The cause of this error depends on the specifications of the EtherCAT slave.	Refer to the manual for the EtherCAT slave and correct the problem.
16#1808	16#00000000	Communications Resource Overflow	More than 32 EtherCAT communications instructions were executed at the same time.	Correct the user program so that no more than 32 EtherCAT communications instructions are executed at the same time.

*1. An SDO abort code is shown here. Refer to *SDO Abort Codes* in the instructions reference manual for a list of SDO abort codes.

Sample Programming

Precautions for Correct Use

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

This sample programming initializes the absolute encoder for a Servo Drive.

Conditions

- The target Servo Drive is Node 1.
- The target Servo Drive is assigned to axis 1 in the Motion Control Function Module.
- The timeout time is 2 s (default).

Processing

1. Make sure that the target slave is in a state that allows communications, that axis 1 is disabled, and that the same instance is not being executed.
2. Execute INITENCODER_G5_instance.

LD

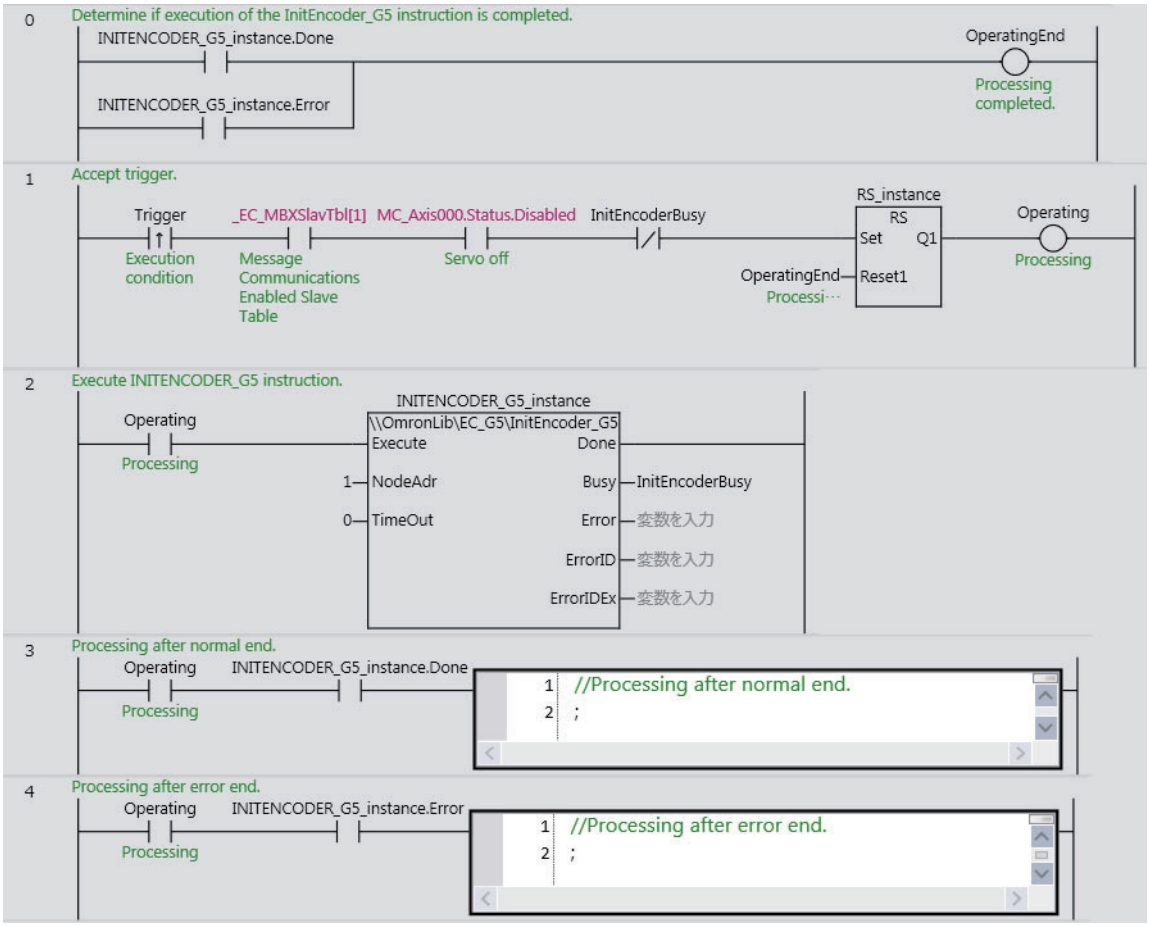
External Variables

Variable	Data type	Constant	Comment
_EC_MBXSlavTbl	ARRAY[1..N *1] OF BOOL	✓	Message Communications Enabled Slave Table
MC_Axis000	_sAXIS_REF	✓	Servo off

*1. N = 192 for NJ501, NJ301 and NJ101. N = 512 for NX701.

Internal Variables

Name	Data type	Default	Comment
INITENCODER_G5_instance	OmronLib\EC_G5\InitEncoder_G5		
OperatingEnd	BOOL		Processing completed
Trigger	BOOL	FALSE	Execution condition
Operating	BOOL		Processing
RS_instance	RS		
InitEncoderBusy	BOOL		



ST

● External Variables

Variable	Data type	Constant	Comment
_EC_MBXSlavTbll	ARRAY[1..N *1] OF BOOL	✓	Message Communications Enabled Slave Table
MC_Axis000	_sAXIS_REF	✓	Servo off

*1. N = 192 for NJ501, NJ301 and NJ101. N = 512 for NX701.

● Internal Variables

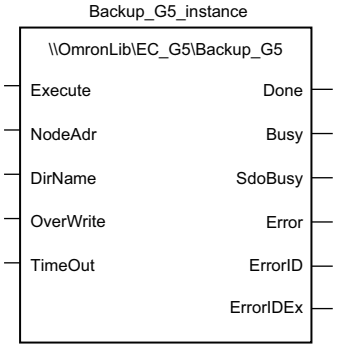
Name	Data type	Default	Comment
INITENCODER_G5_instance	OmronLib\EC_G5\InitEncoder_G5		
Trigger	BOOL	FALSE	Execution condition
LastTrigger	BOOL	FALSE	Value of Trigger from previous task
OperatingStart	BOOL	FALSE	Processing started.
Operating	BOOL	FALSE	Processing
InitEncoderBusy	BOOL	FALSE	

```
// Detect when Trigger changes to TRUE.
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_EC_MBXSlavTbl[1]=TRUE) AND
(MC_Axis000.Status.Disabled=TRUE) AND(InitEncoderBusy=FALSE) ) THEN
   OperatingStart:=TRUE;
   Operating :=TRUE;
END_IF;
LastTrigger:=Trigger;

// Initialize InitEncoder_G5 instruction.
IF (OperatingStart=TRUE) THEN
   INITENCODER_G5_instance(
      Execute :=FALSE);
   OperatingStart:=FALSE;
END_IF;
// Execute Backup_G5 instruction.
IF (Operating=TRUE) THEN
   INITENCODER_G5_instance(
      Execute :=TRUE,
      NodeAdr:=1,
      TimeOut:=0,
      Busy=>InitEncoderBusy);
   IF (INITENCODER_G5_instance.Done=TRUE) THEN
      // Processing after normal end.
      Operating:=FALSE;
   END_IF;
   IF (INITENCODER_G5_instance.Error=TRUE) THEN
      // Processing after error end.
      Operating:=FALSE;
   END_IF;
END_IF;
```

Backup_G5

The Backup_G5 function block backs up the parameters for a G5-series Servo Drive and stores them as a backup file on the SD Memory Card inserted in the Controller.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
Backup_G5	G5-series Servo Drive Parameter Backup	FB		Backup_G5_instance(Execute, NodeAdr, DirName, OverWrite, TimeOut, Done, Busy, SdoBusy, Error, ErrorID, ErrorIDEx);

Function Block and Function Information

Item	Description
Library file name	OmronLib_EC_G5_V1_1.slr
Namespace	OmronLib\EC_G5
Function block and function number	00008
Source code published/not published	Not published
Function block and function version	1.01

Compatible Models

Item	Name	Model numbers	Version
Device	100-VAC Input Drive	R88D-KNA5L-ECT	Version 2.10 or later
		R88D-KN01L-ECT	
		R88D-KN02L-ECT	
		R88D-KN04L-ECT	
	200-VAC Input Drive	R88D-KN01H-ECT	
		R88D-KN02H-ECT	
		R88D-KN04H-ECT	
		R88D-KN08H-ECT	
		R88D-KN10H-ECT	
		R88D-KN15H-ECT	
		R88D-KN20H-ECT	
		R88D-KN30H-ECT	
		R88D-KN50H-ECT	
		R88D-KN75H-ECT	
		R88D-KN150H-ECT	
	400-VAC Input Drive	R88D-KN06F-ECT	
		R88D-KN10F-ECT	
		R88D-KN15F-ECT	
		R88D-KN20F-ECT	
		R88D-KN30F-ECT	
		R88D-KN50F-ECT	
		R88D-KN75F-ECT	
		R88D-KN100F-ECT	
	SD Memory Card	HMC-SD□□□	

Variables

Name	Meaning	I/O	Description	Valid range	Unit	Default
Execute	Execute	Input	TRUE: Executes the instruction. FALSE: Does not execute the instruction.	TRUE or FALSE	---	FALSE
NodeAdr	Slave Node Address	Input	Specifies the node address.	1 to 512*1		1
DirName	Directory Name	Input	Specifies the directory on the SD Memory Card. The backup file(s) stored in this directory will be restored.	40 bytes max.		" *2
OverWrite	Overwrite Enable	Input	TRUE: Enables overwrite. FALSE: Prohibits overwrite.	TRUE or FALSE		FALSE
TimeOut	Timeout Time	Input	0: 2.0 s 1 to 65535: 0.1 to 6553.5 s	0 to 65535	0.1 s	0 (2.0 s)
Done	Done	Output	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE or FALSE	---	---
Busy	Executing	Output	TRUE: Execution processing is in progress. FALSE: Execution processing is not in progress.	TRUE or FALSE		---
SdoBusy	Processing Sdo Communications	Output	This flag is TRUE when an SDO communications command is sent to a G5-series Servo Drive.*3	TRUE or FALSE		---
Error	Error	Output	TRUE: Error end FALSE: Normal end, execution in progress, or execution condition not met	TRUE or FALSE		---
ErrorID	Error Code	Output	This is the error ID for an error end. The value is WORD#16#0 for a normal end.	*4		---
ErrorIDEx	Expansion Error Code	Output	This is the error ID for an Expansion Unit Hardware Error. The value is DWORD#16#0 for a normal end.	*4		---

*1. The valid range is 1 to 192 for NJ-series CPU Units.

*2. If *DirName* is omitted, this specifies the root directory on the SD Memory Card.

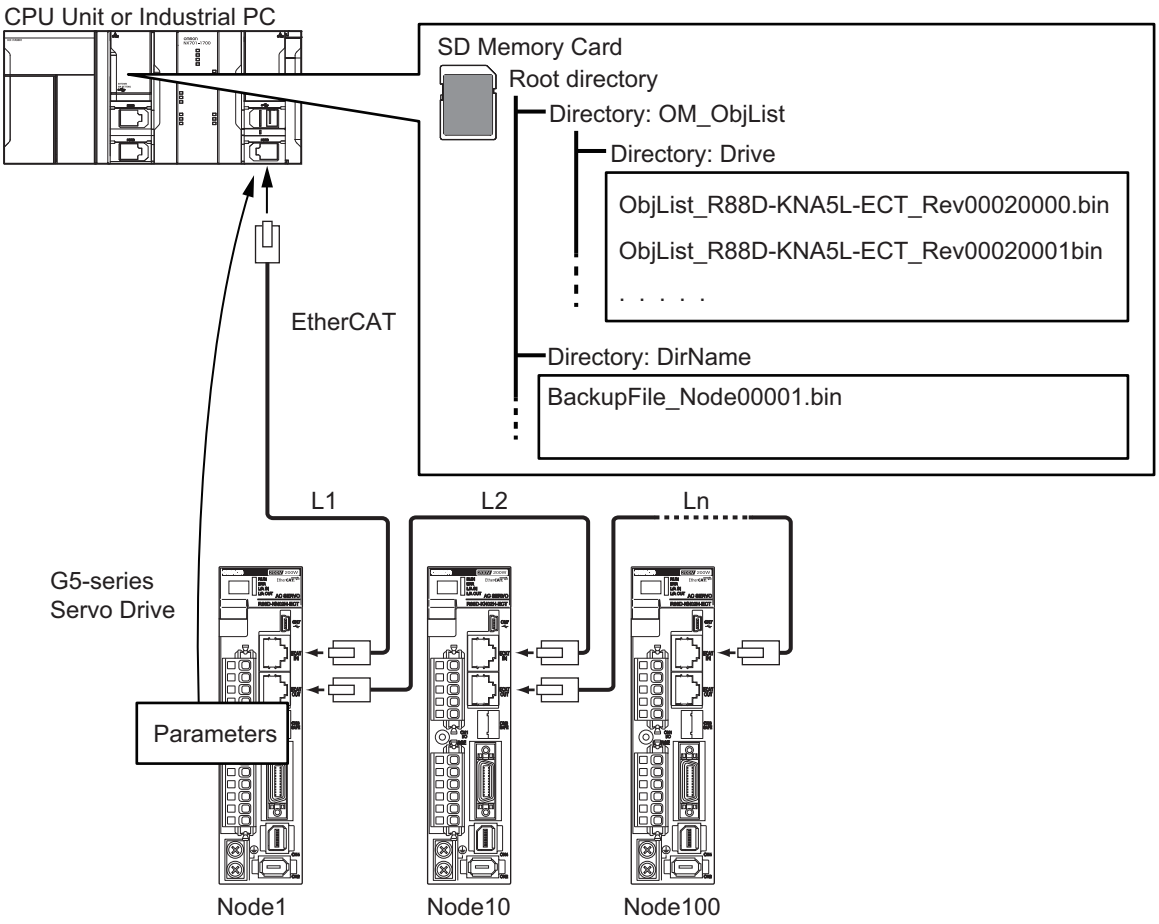
*3. You can use it to identify that communications processing is in progress.

*4. For details, refer to *Troubleshooting* on page 48.

	Boolean	Bit strings				Integers								Real numbers		Times, durations, dates, and text strings				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Execute	OK																			
NodeAdr							OK													
DirName																				OK
OverWrite	OK																			
TimeOut							OK													
Done	OK																			
Busy	OK																			
SdoBusy	OK																			
Error	OK																			
ErrorID			OK																	
ErrorIDEx				OK																

Function

- The Backup_G5 function block reads parameters for the G5-series Servo Drive at the node address specified with *NodeAdr* (Node Address) when *Execute* changes to TRUE and creates a backup file in the directory specified with *DirName* (Directory Name) on the SD Memory Card.
- To use this function block, you need to create in advance a directory named OM_ObjList\Drive in the root directory on the SD Memory Card and store object lists provided by OMRON in it. Be sure to store all the object lists. Then, access the object lists to back up the parameters with this function block.
- A backup file is binary data that has the name “BackupFile_Node□□□□□.Bin”. Here, □□□□□ is a slave node address in decimal (e.g., 00001 for Node 1).
- If the directory specified with *DirName* (Directory Name) does not exist, the function block creates a new directory with the specified directory name. If *DirName* (Directory Name) is omitted, the function block assumes that the root directory on the SD Memory Card is specified.



Procedure

1. Create a directory named OM_ObjList\Drive in the root directory on the SD Memory Card and store all the object lists provided by OMRON in it.
2. Execute the function block.

   For *NodeAdr* (Slave Node Address), specify the node address of the backup target slave.

   If necessary, set *DirName* (Directory Name) and *TimeOut* (Timeout Time).
3. A backup file is created in the directory specified with *DirName* (Directory Name) when execution of the function block is ended.

Precautions for Correct Use

- Do not change the names and the contents of any object list.
- Execute the function block while the Servo is OFF. If the Servo Drive is assigned to the Motion Control Function Module, make sure that the Axis Disabled (*_MC_AX[***].Status.Disabled*) in the Axis Variable is TRUE before you execute the function block.
- Do not attempt to turn OFF the power supply to the Controller and EtherCAT slaves or stop the EtherCAT communications until the processing for this function block ends normally or ends in an error.

Additional Information

If you turn OFF the power supply to the G5-series Servo Drive when replacing a G5-series Servo Drive or when applying the restored parameters, an EtherCAT slave communications error will occur.

The EtherCAT slave communications error does not occur, if you perform EtherCAT slave disconnection before turning OFF the power supply to the G5-series Servo Drive.

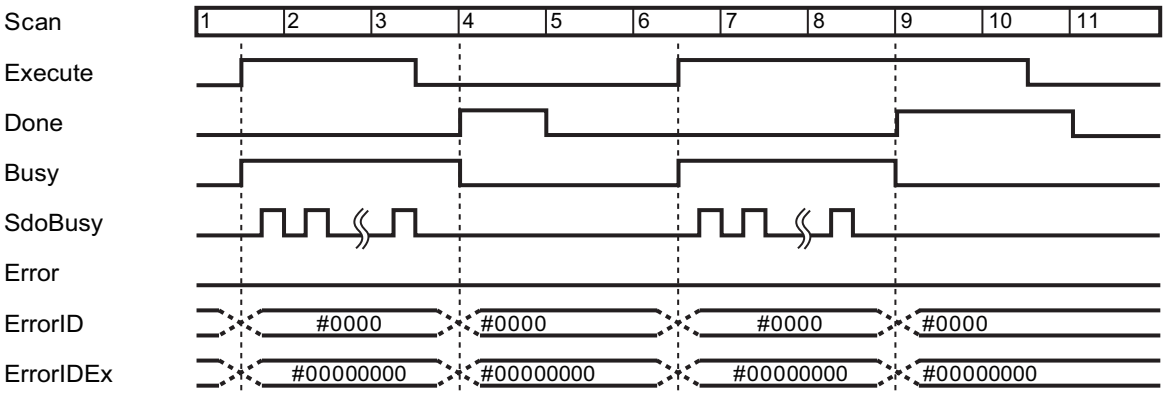
Timing Charts

The following figures show the timing charts for the function block.

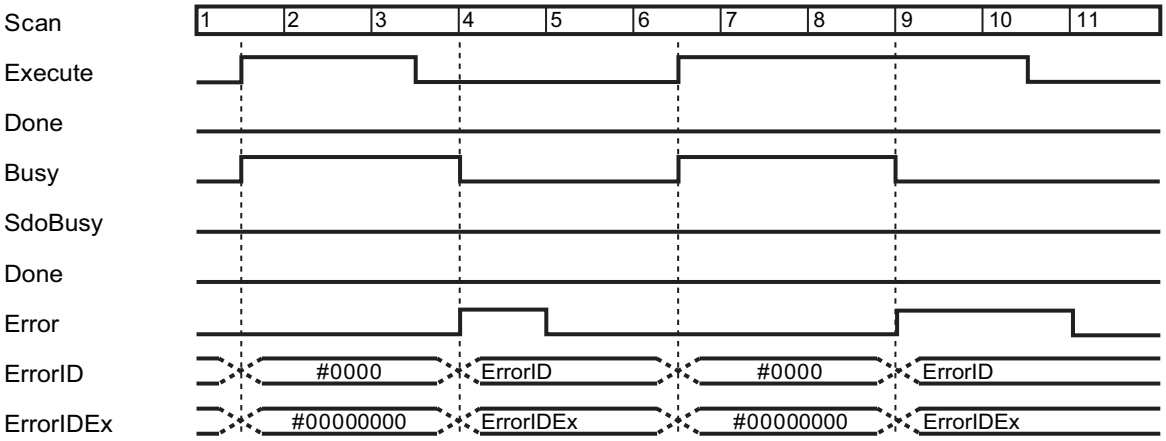
- *Busy* (Executing) changes to TRUE when *Execute* changes to TRUE.
- When backup file creation is ended, *Done* changes to TRUE.
- If an error occurs when execution of the function block is in progress, *Error* changes to TRUE and *Busy* (Executing) changes to FALSE.

  You can find out the cause of the error by accessing the values output to *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code).
- If *Execute* changes to FALSE before execution of the function block is ended, *Done* and *Error* are TRUE only for one task period after execution of the function block is ended.
- If *Execute* remains TRUE even after execution of the function block is ended, the output values of *Done* and *Error* are held.

● Timing Chart for Normal End



● Timing Chart for Error End



Additional Information

- An object list is binary data that stores, by the revision number of EtherCAT slaves, SDO parameter information (Index, Subindex, and Complete Access) for the backup target parameters.
- For information on the characters that can be used in *DirName*, refer to *Specifications of Supported SD Memory Cards, Folders, and Files* in the software user's manual.
- In backup or restore processing, the value of *SdoBusy* is TRUE when data objects are being sent. The value of *SdoBusy* is FALSE when data objects are not being sent due to the internal processing of the function block.

Precautions for Correct Use

- Execution of this function block will be continued until processing is ended even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is ended. Use this to confirm normal ending of processing.
- This function block executes one EC_CoESDORead instruction within the part. Refer to the *EC_CoESDORead instruction* in the instructions reference manual for the maximum number of instructions that can be executed at the same time.
- This function block executes one FileWriteVar instruction within the part. Refer to the *FileWriteVar instruction* in the instructions reference manual for the maximum number of instructions that can be executed at the same time.
- An error will occur in the following cases. *Error* will change to TRUE.
  a) No object list exists.
  b) The EtherCAT master is not in a state that allows message communications.
  c) The slave specified with *NodeAdr* does not exist.
  d) The slave specified with *NodeAdr* is not in a state that allows communications.
  e) The slave is not in a state that allows reading of parameters.
  f) The number of EtherCAT communications instructions such as EC_CoESDORead exceeded the maximum number of instructions that can be executed at the same time.
  g) SD Memory Card access failed (no SD Memory Card inserted, write protection enabled, etc.).
  h) The number of SD Memory Card related instructions such as FileWriteVar exceeded the maximum number of instructions that can be executed at the same time.

Related System-defined Variables

Name	Meaning	Data type	Description
_Card1Ready	SD Memory Card Ready Flag	BOOL	TRUE when the SD Memory Card is recognized. It is FALSE when the SD Memory Card is not recognized. TRUE: Can be used. FALSE: Cannot be used.
_EC_MBXSlavTbl	Message Communications Enabled Slave Table	Array[1..512] OF BOOL	This table indicates the slaves that can perform message communications. Slaves are given in the table in the order of slave node addresses. TRUE: Message communications are possible. FALSE: Message communications are not possible.

Troubleshooting

Error code	Expansion error code	Status	Description	Correction
16#0000	16#00000000	Normal End	---	---
16#3C00	16#00000001	VendorID Not Matched	The VendorID in object lists does not match the VendorID of the backup target node.	Use object lists that match the VendorID of the backup target node.
16#3C00	16#00000002	ProductCode Not Matched	The ProductCode in object lists does not match the ProductCode of the backup target node.	Use object lists that match the ProductCode of the backup target node.
16#3C00	16#00000003	Unsupported Device	The device is not supported by this function block.	Check that the specified device is a G5-series Servo Drive.
16#0400	16#00000000	Input Value Out of Range	The value of *NodeAdr* is out of range.	Set the value of *NodeAdr* correctly.
			The directory name specified with *DirName* contains one or more characters that cannot be used.	Set *DirName* correctly.
			The directory name specified with *DirName* is too long.	Check the length of the directory name specified with *DirName* and set it to within the valid range.
16#1400	16#00000000	SD Memory Card Access Failure	An SD Memory Card is either not inserted or is not inserted properly.	Insert an SD Memory Card correctly.
			The SD Memory Card is broken.	Replace the SD Memory Card with one that operates normally.
			The SD Memory Card slot is broken.	If this error persists even after making the above two corrections, replace the CPU Unit or the Industrial PC.
16#1401	16#00000000	SD Memory Card Write-protected	An attempt was made to write to a write-protected SD Memory Card.	Remove write protection from the SD Memory Card. Slide the small switch on the side of the SD Memory Card from the LOCK position to the writable position.
16#1402	16#00000000	SD Memory Card Insufficient Capacity	The SD Memory Card ran out of free space.	Replace the SD Memory Card for one with sufficient available capacity.
16#1403	16#00000000	File Does Not Exist	Object lists are not stored.	Check the SD Memory Card and store correct object lists on it.
16#1405	16#00000000	File Already in Use	An instruction attempted to read or write a file already being accessed by another instruction.	Correct the user program so that this function block is executed only when the *Busy* output variable for all other instructions for the same file is FALSE.
16#1409	16#00000000	That File Name Already Exists	The backup file exists on the SD Memory Card when *OverWrite* is FALSE.	Change the directory in which to store the backup file. Or, delete the existing backup file in advance.

Error code	Expansion error code	Status	Description	Correction
16#140A	16#00000000	Write Access Denied	The file or directory specified for the function block to write is write-protected.	Remove write protection from the file or directory specified for the function block. Or, change the file name of the file to write.
16#140B	16#00000000	Too Many Files Open	The maximum number of open files was exceeded when opening a file for the function block.	Correct the user program to decrease the number of open files.
16#140E	16#00000000	SD Memory Card Access Failed	The SD Memory Card is broken.	Replace the SD Memory Card.
			The SD Memory Card slot is broken.	If this error occurs even after making the above correction, replace the CPU Unit or the Industrial PC.
16#1800	16#00000000	EtherCAT Communications Error	The EtherCAT network is not in a usable status.	Check the operation status of the EtherCAT network by checking the status of the EtherCAT master. Use this information to correct the cause of the problem.
16#1801	16#00000000	EtherCAT Slave Does Not Respond	The target slave does not exist.	Specify an existing node address.
			The target slave is not in an operating condition.	Check the status of the target EtherCAT slave. Make sure that the target slave is in a usable status.
16#1802	16#00000000	EtherCAT Timeout	Communications with the target slave timed out.	Check the operating status of the target slave and correct the cause of the problem.
16#1803	16#00000000	Reception Buffer Overflow	The receive data from an EtherCAT slave overflowed the receive buffer.	Set the size of the reception buffer to a value larger than the size of the receive data from the slave.
16#1804	SDO Abort Code*1	SDO Abort Error	The cause of this error depends on the specifications of the EtherCAT slave.	Refer to the manual for the EtherCAT slave and correct the problem.
16#1808	16#00000000	Communications Resource Overflow	More than 32 EtherCAT communications instructions were executed at the same time.	Correct the user program so that no more than 32 EtherCAT communications instructions are executed at the same time.

*1. Refer to *SDO Abort Codes* in the instructions reference manual for a list of SDO abort codes.

Sample Programming

Precautions for Correct Use

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

This sample programming backs up and then restores parameters for a Servo Drive.

Conditions

- The target G5-series Servo Drive is Node 1.
- The target G5-series Servo Drive is assigned to axis 0 in the Motion Control Function Module.
- The backup file is created in the TestDir directory.
- The backup file is write-protected (default).
- The timeout time is 2 s (default).

Backup Processing

1. Make sure that the target slave is in a state that allows communications, that axis 0 is disabled, that the same instance is not being executed, and that parameters are not being restored to the same node.
2. Execute BACKUP_G5_instance.

Restore Processing

1. Make sure that the Servo is OFF for axis 0 and disconnect the target slave from the network.
2. Turn OFF the power supply to the target slave and replace the target slave.
3. Connect the target slave to the network again.
4. Make sure that the target slave is in a state that allows communications, that axis 0 is disabled, that the same instance is not being executed, and that parameters are not being backed up from the same node.
5. Execute RESTORE_G5_instance.
6. After completion of the restore processing, disconnect the target slave from the network, cycle the control power supply to the target slave, and connect the target slave to the network again.

LD

● External Variables

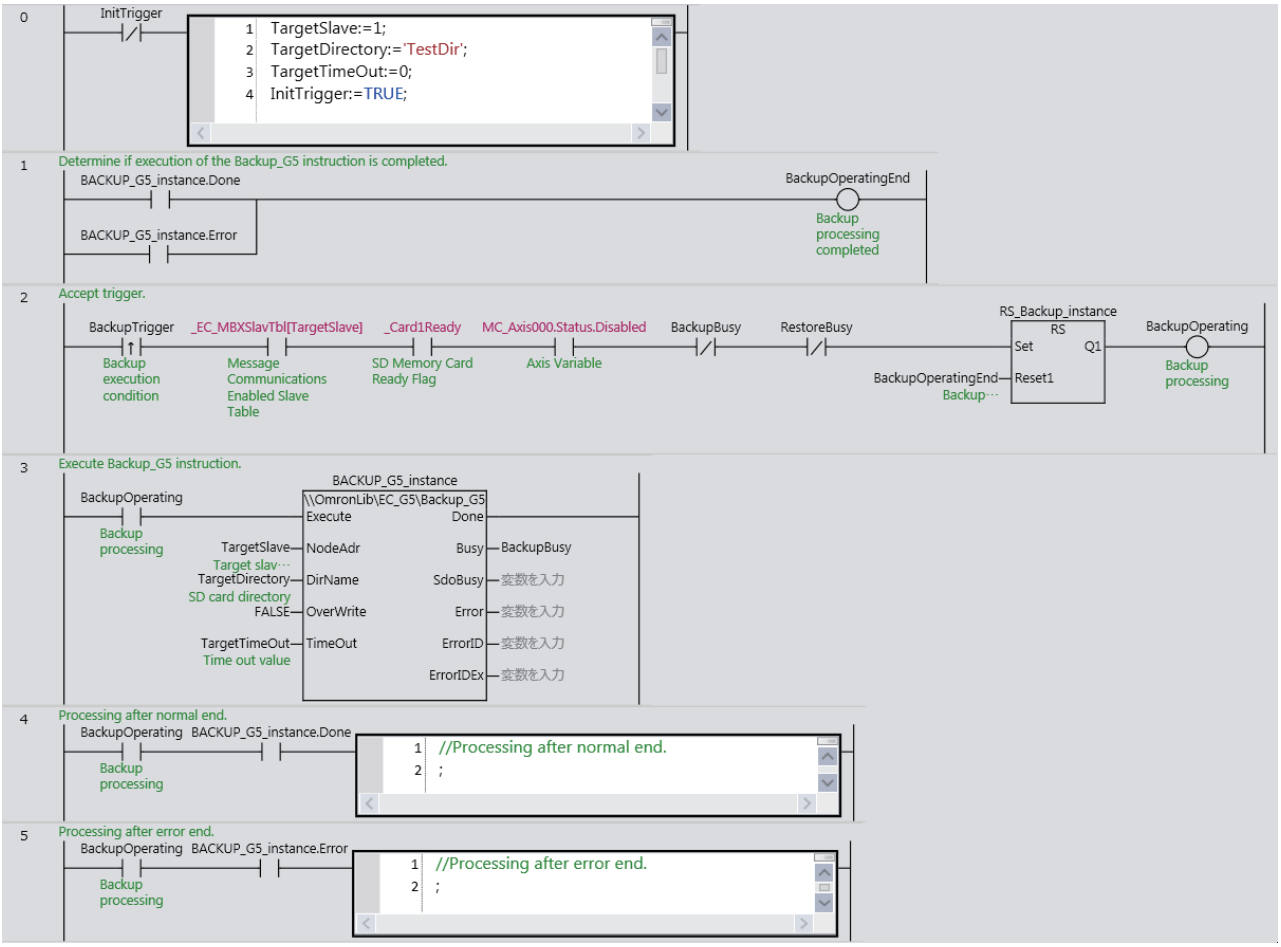
Name	Data type	Constant	Comment
_EC_MBXSlavTbl	ARRAY[1..N*1] OF BOOL	✓	Message Communications Enabled Slave Table
_Card1Ready	BOOL	✓	SD Memory Card Ready Flag
MC_Axis000	_sAXIS_REF	✓	Servo off

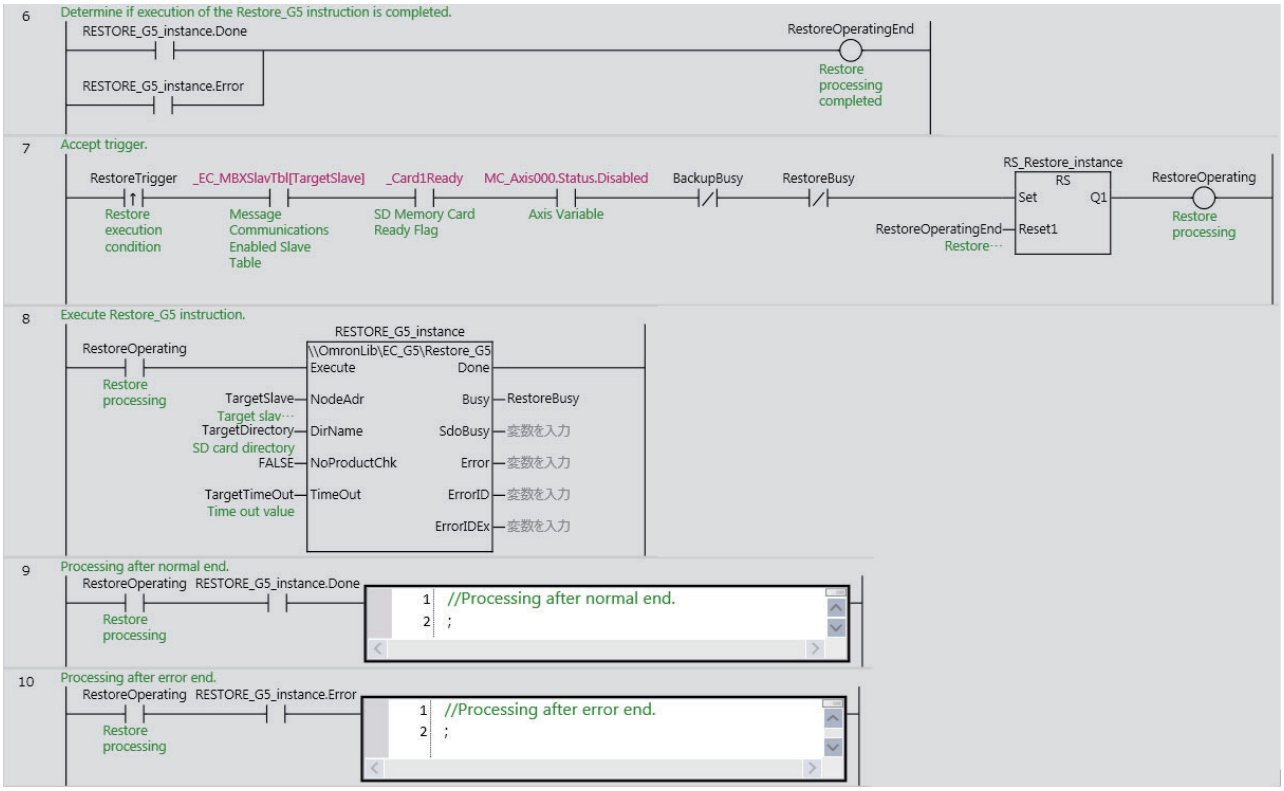
*1. N = 192 for NJ501, NJ301 and NJ101. N = 512 for NX701.

● Internal Variables

Name	Data type	Default	Comment
BACKUP_G5_instance	OmronLib\EC_G5\Backup_G5		
RESTORE_G5_instance	OmronLib\EC_G5\Restore_G5		
InitTrigger	BOOL	FALSE	
BackupTrigger	BOOL	FALSE	Backup execution condition
RestoreTrigger	BOOL	FALSE	Restore execution condition
TargetSlave	UINT	1	Target slave node Address
TargetDirectory	STRING[41]	''	SD card directory
TargetTimeOut	UINT	0	Time out value

Name	Data type	Default	Comment
BackupBusy	BOOL		
RestoreBusy	BOOL		
BackupOperating	BOOL		Backup processing
RestoreOperating	BOOL		Restore processing
BackupOperatingEnd	BOOL		Backup processing completed
RestoreOperatingEnd	BOOL		Restore processing completed
RS_Backup_instance	RS		
RS_Restore_instance	RS		




6
Determine if execution of the Restore_G5 instruction is completed.
RESTORE_G5_instance.Done
RESTORE_G5_instance.Error
RestoreOperatingEnd
Restore processing completed
7
Accept trigger.
RestoreTrigger
Restore execution condition
_EC_MBXSlavTbl[TargetSlave]
Message Communications Enabled Slave Table
_Card1Ready
SD Memory Card Ready Flag
MC_Axis000.Status.Disabled
Axis Variable
BackupBusy
RestoreBusy
RS_Restore_instance
RS
Set
Q1
Reset1
RestoreOperatingEnd
Restore···
RestoreOperating
Restore processing
8
Execute Restore_G5 instruction.
RestoreOperating
Restore processing
RESTORE_G5_instance
\\OmronLib\EC_G5\Restore_G5
Execute
Done
TargetSlave
Target slav···
NodeAdr
Busy
RestoreBusy
TargetDirectory
SD card directory
DirName
SdoBusy
変数を入力
FALSE
NoProductChk
Error
変数を入力
TargetTimeOut
Time out value
TimeOut
ErrorID
変数を入力
ErrorIDEx
変数を入力
9
Processing after normal end.
RestoreOperating
RESTORE_G5_instance.Done
Restore processing
1 //Processing after normal end.
2 ;
10
Processing after error end.
RestoreOperating
RESTORE_G5_instance.Error
Restore processing
1 //Processing after error end.
2 ;

ST

● External Variables

Variable	Data type	Constant	Comment
_EC_MBXSlavTbl	ARRAY[1..N *1] OF BOOL	✓	Message Communications Enabled Slave Table
_Card1Ready	BOOL	✓	SD Memory Card Ready Flag
MC_Axis000	_sAXIS_REF	✓	Servo off

*1. N = 192 for NJ501, NJ301 and NJ101. N = 512 for NX701.

● Internal Variables

Name	Data type	Default	Comment
BACKUP_G5_instance	OmronLib\EC_G5\Backup_G5		
RESTORE_G5_instance	OmronLib\EC_G5\Restore_G5		
InitTrigger	BOOL	FALSE	
BackupTrigger	BOOL	FALSE	Backup execution condition
RestoreTrigger	BOOL	FALSE	Restore execution condition
TargetSlave	UINT	1	Target slave node Address
TargetDirectory	STRING[41]	''	SD card directory
TargetTimeOut	UINT	0	Time out value
BackupBusy	BOOL	FALSE	
RestoreBusy	BOOL	FALSE	
BackupLastTrigger	BOOL	FALSE	Value of BackupTrigger from previous task
RestoreLastTrigger	BOOL	FALSE	Value of RestoreTrigger from previous task
BackupOperating	BOOL	FALSE	Backup processing
RestoreOperating	BOOL	FALSE	Restore processing
BackupOperatingStart	BOOL	FALSE	Backup processing started
RestoreOperatingStart	BOOL	FALSE	Restore processing started

```
// Backup and Resotre parameters setting.
IF InitTrigger=FALSE THEN
   TargetSlave:=1;
   TargetDirectory:='TestDir';
   TargetTimeOut:=0;
   InitTrigger:=TRUE;
END_IF;

// Detect when BackupTrigger changes to TRUE.
IF ( (BackupTrigger=TRUE) AND (BackupLastTrigger=FALSE) AND (_EC_MBXSlavTbl
                [TargetSlave]=TRUE) AND (MC_Axis000.Status.Disabled=TRUE) AND
                 (_Card1Ready=TRUE) AND (BackupBusy=FALSE) AND
                (RestoreBusy=FALSE) ) THEN
   BackupOperatingStart:=TRUE;
   BackupOperating :=TRUE;
END_IF;
BackupLastTrigger:=BackupTrigger;

// Initialize Backup_G5 instruction.
IF (BackupOperatingStart=TRUE) THEN
   BACKUP_G5_instance(
      Execute :=FALSE);
   BackupOperatingStart:=FALSE;
```

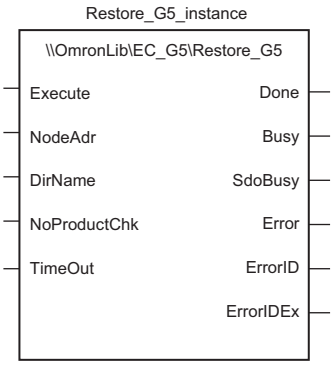
```
END_IF;
// Execute Backup_G5 instruction.
IF (BackupOperating=TRUE) THEN
   BACKUP_G5_instance(
      Execute :=TRUE,
      NodeAdr:=TargetSlave,
      DirName:=TargetDirectory,
      OverWrite:=FALSE,
      TimeOut:=TargetTimeOut,
      Busy=>BackupBusy);
   IF (BACKUP_G5_instance.Done=TRUE) THEN
      // Processing after normal end.
      BackupOperating:=FALSE;
   END_IF;
   IF (BACKUP_G5_instance.Error=TRUE) THEN
      // Processing after error end.
      BackupOperating:=FALSE;
   END_IF;
END_IF;

// Detect when RestoreTrigger changes to TRUE.
IF ( (RestoreTrigger=TRUE) AND (RestoreLastTrigger=FALSE) AND
               (_EC_MBXSlavTbl[TargetSlave]=TRUE) AND
               (MC_Axis000.Status.Disabled=TRUE) AND (_Card1Ready=TRUE) AND
               (BackupBusy=FALSE) AND (RestoreBusy=FALSE) ) THEN
   RestoreOperatingStart:=TRUE;
   RestoreOperating :=TRUE;
END_IF;
RestoreLastTrigger:=RestoreTrigger;

// Initialize Restore_G5 instruction.
IF (RestoreOperatingStart=TRUE) THEN
   RESTORE_G5_instance(
      Execute :=FALSE);
   RestoreOperatingStart:=FALSE;
END_IF;
// Execute Restore_G5 instruction.
IF (RestoreOperating=TRUE) THEN
   RESTORE_G5_instance(
      Execute :=TRUE,
      NodeAdr:=TargetSlave,
      DirName:=TargetDirectory,
      NoProductChk:=FALSE,
      TimeOut:=TargetTimeOut,
      Busy=>RestoreBusy);
   IF (RESTORE_G5_instance.Done=TRUE) THEN
      // Processing after normal end.
      RestoreOperating:=FALSE;
   END_IF;
   IF (RESTORE_G5_instance.Error=TRUE) THEN
      // Processing after error end.
      RestoreOperating:=FALSE;
   END_IF;
END_IF;
```

Restore_G5

The Restore_G5 function block restores the backup file(s) created by the Backup_G5 function block on the SD Memory Card inserted in the Controller to a G5-series Servo Drive.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
Restore_G5	G5-series Servo Drive Parameter Restore	FB		Restore_G5_instance(Execute, NodeAdr, DirName, NoProductChk, TimeOut, Done, Busy, SdoBusy, Error, ErrorID, ErrorIDEx);

Function Block and Function Information

Item	Description
Library file name	OmronLib_EC_G5_V1_1.slr
Namespace	OmronLib\EC_G5
Function block and function number	00009
Source code published/not published	Not published
Function block and function version	1.01

Compatible Models

Item	Name	Model numbers	Version
Device	100-VAC Input Drive	R88D-KNA5L-ECT	Version 2.10 or later
		R88D-KN01L-ECT	
		R88D-KN02L-ECT	
		R88D-KN04L-ECT	
	200-VAC Input Drive	R88D-KN01H-ECT	
		R88D-KN02H-ECT	
		R88D-KN04H-ECT	
		R88D-KN08H-ECT	
		R88D-KN10H-ECT	
		R88D-KN15H-ECT	
		R88D-KN20H-ECT	
		R88D-KN30H-ECT	
		R88D-KN50H-ECT	
		R88D-KN75H-ECT	
		R88D-KN150H-ECT	
	400-VAC Input Drive	R88D-KN06F-ECT	
		R88D-KN10F-ECT	
		R88D-KN15F-ECT	
		R88D-KN20F-ECT	
		R88D-KN30F-ECT	
		R88D-KN50F-ECT	
		R88D-KN75F-ECT	
		R88D-KN100F-ECT	
	SD Memory Card	HMC-SD□□□	

Variables

Name	Meaning	I/O	Description	Valid range	Unit	Default
Execute	Execute	Input	TRUE: Executes the instruction. FALSE: Does not execute the instruction.	TRUE or FALSE	---	FALSE
NodeAdr	Slave Node Address	Input	Specifies the node address.	1 to 512*1		1
DirName	Directory Name	Input	Specifies the directory on the SD Memory Card. The backup file(s) stored in this directory will be restored.	40 bytes max.		" *2
NoProductChk	Product Code Check	Input	Specifies whether or not to execute restore when the product code in the backup data differs from the product code of the restore destination. TRUE: Executes restore. FALSE: Prohibits restore.	TRUE or FALSE		FALSE
TimeOut	Timeout Time	Input	0: 2.0 s 1 to 65535: 0.1 to 6553.5 s	0 to 65535	0.1 s	0 (2.0 s)
Done	Done	Output	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE or FALSE	---	---
Busy	Executing	Output	TRUE: Execution processing is in progress. FALSE: Execution processing is not in progress.	TRUE or FALSE		---
SdoBusy	Processing Sdo Communications	Output	This flag is TRUE when an SDO communications command is sent to a G5-series Servo Drive.*3	TRUE or FALSE		---
Error	Error	Output	TRUE: Error end FALSE: Normal end, execution in progress, or execution condition not met	TRUE or FALSE		---
ErrorID	Error Code	Output	This is the error ID for an error end. The value is WORD#16#0 for a normal end.	*4		---
ErrorIDEx	Expansion Error Code	Output	This is the error ID for an Expansion Unit Hardware Error. The value is DWORD#16#0 for a normal end.	*4		---

*1. The valid range is 1 to 192 for NJ-series CPU Units.

*2. If *DirName* is omitted, this specifies the root directory on the SD Memory Card.

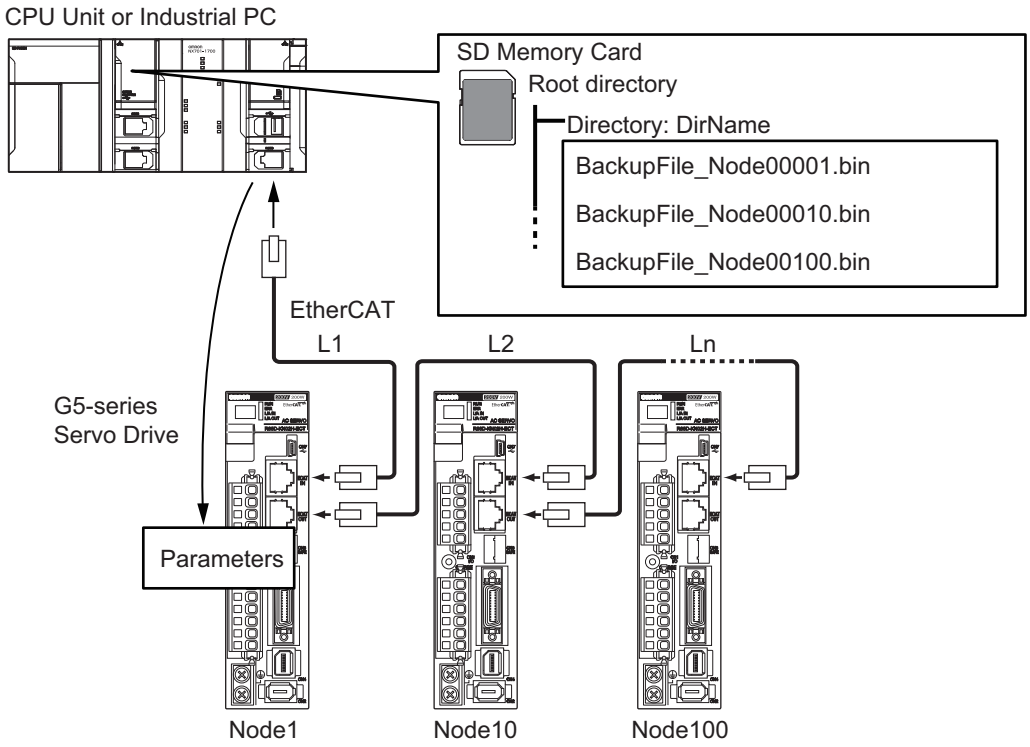
*3. You can use it to identify that communications processing is in progress.

*4. For details, refer to *Troubleshooting* on page 62.

	Boolean	Bit strings				Integers								Real numbers		Times, durations, dates, and text strings				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Execute	OK																			
NodeAdr							OK													
DirName																				OK
NoProductChk	OK																			
TimeOut							OK													
Done	OK																			
Busy	OK																			
SdoBusy	OK																			
Error	OK																			
ErrorID			OK																	
ErrorIDEx				OK																

Function

- When *Execute* changes to TRUE, the Restore_G5 function block restores the backup file(s) in the directory specified with *DirName* (Directory Name) on the SD Memory Card to the G5-series Servo Drive specified with *NodeAdr* (Node Address).
- Before you restore a file, use the Backup_G5 function block to create one or more backup files.
- The backup file stored at the node address specified with *NodeAdr* (Node Address) is the restore target. For example, if *NodeAdr* is UINT#1, the file “BackupFile_Node00001.Bin” is the restore target.
- If *DirName* (Directory Name) is omitted, the function block assumes that the root directory on the SD Memory Card is specified.
- This function block will perform restore if the revision of the backup source device is either the same as or older than the revision of the restore destination device. If the revision of the backup source device is newer than the revision of the restore destination device, an error will occur.
- To restore backup data to a node at a different node address from that of the backup source device, change the node address portion of the backup file name to the node address of the restore destination device.
- To restore backup data when the model of the backup source device differs from the model of the restore destination device, set *NoProductChk* to TRUE. This enables you to execute restore even if you change the model of your Servo Drive.



Procedure

1 Execute the function block.

For *NodeAdr* (Slave Node Address), specify the node address of the restore target slave.

If necessary, set *DirName* (Directory Name) and *TimeOut* (Timeout Time).

2 The backup file in the directory specified with *DirName* (Directory Name) is restored to the G5-series Servo Drive when execution of the function block is ended.

Precautions for Correct Use

- Execute the function block while the Servo is OFF. If the Servo Drive is assigned to the Motion Control Function Module, make sure that the Axis Disabled (*_MC_AX[***].Status.Disabled*) in the Axis Variable is TRUE before you execute the function block.
- After completion of Restore_G5 function block, cycle the control power supply to the G5-series Servo Drive to apply the restored parameters.
- Do not attempt to turn OFF the power supply to the Controller and EtherCAT slaves or stop the EtherCAT communications until the processing for this function block ends normally or ends in an error.

Additional Information

If you turn OFF the power supply to the G5-series Servo Drive when replacing a G5-series Servo Drive or when applying the restored parameters, an EtherCAT slave communications error will occur.

The EtherCAT slave communications error does not occur, if you perform EtherCAT slave disconnection before turning OFF the power supply to the G5-series Servo Drive.

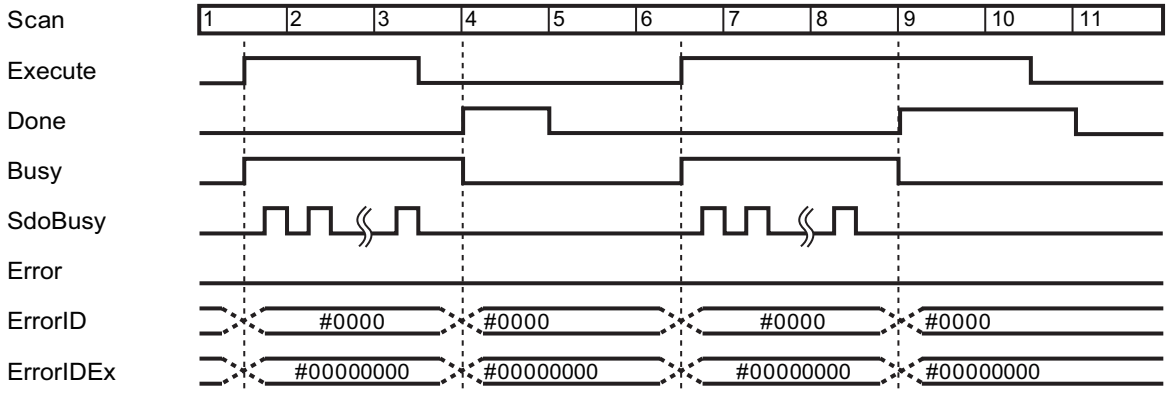
Timing Charts

The following figures show the timing charts for the function block.

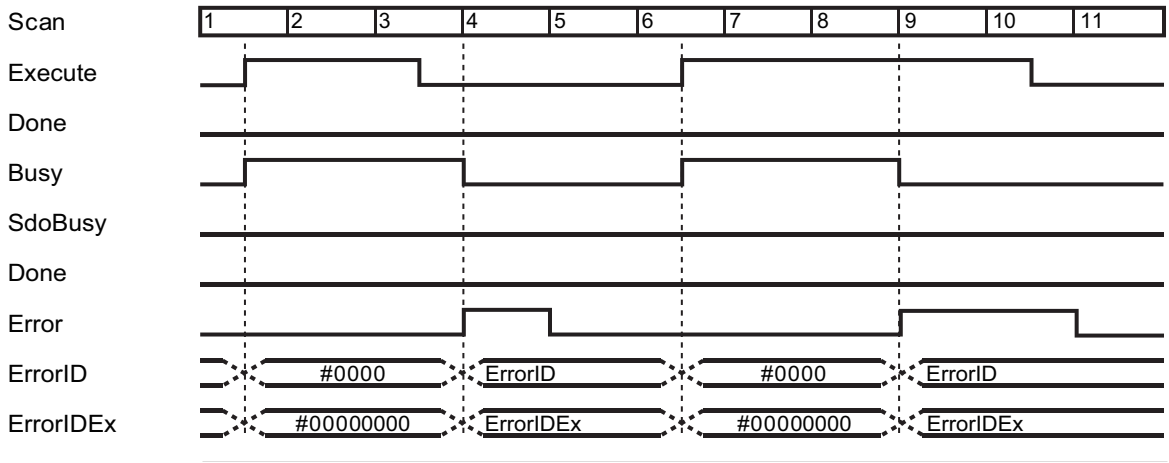
- *Busy* (Executing) changes to TRUE when *Execute* changes to TRUE.
- *Done* changes to TRUE when restore is ended.
- If an error occurs when execution of the function block is in progress, *Error* changes to TRUE and *Busy* (Executing) changes to FALSE.

  You can find out the cause of the error by accessing the values output to *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code).
- If *Execute* changes to FALSE before execution of the function block is ended, *Done* and *Error* are TRUE only for one task period after execution of the function block is ended.
- If *Execute* remains TRUE even after execution of the function block is ended, the output values of *Done* and *Error* are held.

● Timing Chart for Normal End



● Timing Chart for Error End



Additional Information

- For information on the characters that can be used in *DirName*, refer to *Specifications of Supported SD Memory Cards, Folders, and Files* in the software user's manual.
- In backup or restore processing, the value of *SdoBusy* is TRUE when data objects are being sent. The value of *SdoBusy* is FALSE when data objects are not being sent due to the internal processing of the function block.

Precautions for Correct Use

- Execution of this function block will be continued until processing is ended even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is ended. Use this to confirm normal ending of processing.
- This function block executes one EC_CoESDOWrite instruction or one EC_CoESDORead within the part. Refer to the *EC_CoESDOWrite instruction* in the instructions reference manual for the maximum number of instructions that can be executed at the same time.
- This function block executes one FileReadVar instruction within the part. Refer to the *FileReadVar instruction* in the instructions reference manual for the maximum number of instructions that can be executed at the same time.
- An error will occur in the following cases. *Error* will change to TRUE.
  a) The EtherCAT master is not in a state that allows message communications.
  b) The slave specified with *NodeAdr* does not exist.
  c) The slave specified with *NodeAdr* is not in a state that allows communications.
  d) The model specified in the backup file differs from the model of the restore destination device and *NoProductChk* is FALSE.
  e) The slave is not in a state that allows writing of parameters.
  f) The number of EtherCAT communications instructions such as EC_CoESDOWrite exceeded the maximum number of instructions that can be executed at the same time.
  g) SD Memory Card access failed (no SD Memory Card inserted, a file is being accessed, etc.).
  h) The number of SD Memory Card related instructions such as FileReadVar exceeded the maximum number of instructions that can be executed at the same time.

Precautions for Correct Use

Setting the *NoProductChk* input variable to TRUE allows you to execute restore even when the product code in the backup data differs from the product code of the restore destination. Make sure that the restore destination is a G5-series Servo Drive before you attempt to restore the parameters.

Related System-defined Variables

Name	Meaning	Data type	Description
_Card1Ready	SD Memory Card Ready Flag	BOOL	TRUE when the SD Memory Card is recognized. It is FALSE when the SD Memory Card is not recognized. TRUE: Can be used. FALSE: Cannot be used.
_EC_MBXSlavTbl	Message Communications Enabled Slave Table	Array[1..512] OF BOOL	This table indicates the slaves that can perform message communications. Slaves are given in the table in the order of slave node addresses. TRUE: Message communications are possible. FALSE: Message communications are not possible.

Troubleshooting

Error code	Expansion error code	Status	Description	Correction
16#0000	16#00000000	Normal End	---	---
16#3C01	16#00000001	Unit Not Matched	The model of the restore destination does not match the model specified in the backup file.	Set the model of the restore destination so that it matches the model specified in the backup file.
16#3C01	16#00000002	Backup Data Disabled	The backup file is not created correctly.	Create again the backup file correctly.
16#3C01	16#00000003	Revision Not Matched	The revision number of the restore destination is older than the revision number in the backup file.	Specify the Servo Drive with a revision number that is either the same as or newer than the revision number in the backup file as the restore destination.
16#3C01	16#00000004	Unsupported Device	The device is not supported by this function block.	Check that the specified device is a G5-series Servo Drive.
16#0400	16#00000000	Input Value Out of Range	The value of *NodeAdr* is out of range.	Set the value of *NodeAdr* correctly.
			The directory name specified with *DirName* contains one or more characters that cannot be used.	Set *DirName* correctly.
			The directory name specified with *DirName* is too long.	Check the length of the directory name specified with *DirName* so that it is within the valid range.

Error code	Expansion error code	Status	Description	Correction
16#1400	16#00000000	SD Memory Card Access Failure	An SD Memory Card is either not inserted or is not inserted properly.	Insert an SD Memory Card correctly.
			The SD Memory Card is broken.	Replace the SD Memory Card with one that operates normally.
			The SD Memory Card slot is broken.	If this error persists even after making the above two corrections, replace the CPU Unit or the Industrial PC.
16#1403	16#00000000	File Does Not Exist	There is no backup file to restore in the directory specified with *DirName*.	Specify the directory in which a backup file exists specified in this function block.
16#1405	16#00000000	File Already in Use	An instruction attempted to read or write a file already being accessed by another instruction.	Correct the user program so that this function block is executed only when the *Busy* output variable for all other instructions for the same file is FALSE.
16#140B	16#00000000	Too Many Files Open	The maximum number of open files was exceeded when opening a file for the function block.	Correct the user program to decrease the number of open files.
16#140E	16#00000000	SD Memory Card Access Failed	The SD Memory Card is broken.	Replace the SD Memory Card.
			The SD Memory Card slot is broken.	If this error occurs even after making the above correction, replace the CPU Unit or the Industrial PC.
16#1800	16#00000000	EtherCAT Communications Error	The EtherCAT network is not in a usable status.	Check the operation status of the EtherCAT network by checking the status of the EtherCAT master. Use this information to correct the cause of the problem.
16#1801	16#00000000	EtherCAT Slave Does Not Respond	The target slave does not exist.	Specify an existing node address.
			The target slave is not in an operating condition.	Check the status of the target EtherCAT slave. Make sure that the target slave is in a usable status.
16#1802	16#00000000	EtherCAT Timeout	Communications with the target slave timed out.	Check the operating status of the target slave and correct the cause of the problem.
16#1803	16#00000000	Reception Buffer Overflow	The receive data from an EtherCAT slave overflowed the receive buffer.	Set the size of the reception buffer to a value larger than the size of the receive data from the slave.
16#1804	SDO Abort Code*1	SDO Abort Error	The cause of this error depends on the specifications of the EtherCAT slave.	Refer to the manual for the EtherCAT slave and correct the problem.
16#1808	16#00000000	Communications Resource Overflow	More than 32 EtherCAT communications instructions were executed at the same time.	Correct the user program so that no more than 32 EtherCAT communications instructions are executed at the same time.

*1. Refer to *SDO Abort Codes* in the instructions reference manual for a list of SDO abort codes.

Sample Programming

Refer to the sample programming for the following instructions: *Backup_G5* on page 40.

Additional Information

The following gives an object list for the backup and restore targets.

Refer to the *G5-series AC Servomotors/Servo Drives with Built-in EtherCAT Communications User's Manual* (Cat. No. I576) for details on each object.

Index	Subindex	Access	Data attribute*1
2200 hex	0	RW	C
2201 hex	0	RW	C
3000 hex	0	RW	C
3001 hex	0	RW	R
3002 hex	0	RW	B
3003 hex	0	RW	B
3004 hex	0	RW	B
3013 hex	0	RW	B
3015 hex	0	RW	C
3016 hex	0	RW	C
3017 hex	0	RW	C
3100 hex	0	RW	B
3101 hex	0	RW	B
3102 hex	0	RW	B
3103 hex	0	RW	B
3104 hex	0	RW	B
3105 hex	0	RW	B
3106 hex	0	RW	B
3107 hex	0	RW	B
3108 hex	0	RW	B
3109 hex	0	RW	B
3110 hex	0	RW	B
3111 hex	0	RW	B
3112 hex	0	RW	B
3113 hex	0	RW	B
3114 hex	0	RW	B
3115 hex	0	RW	B
3116 hex	0	RW	B
3117 hex	0	RW	B
3118 hex	0	RW	B
3119 hex	0	RW	B
3120 hex	0	RW	B
3121 hex	0	RW	B
3122 hex	0	RW	B
3123 hex	0	RW	B
3124 hex	0	RW	B
3125 hex	0	RW	B
3126 hex	0	RW	B
3127 hex	0	RW	B
3200 hex	0	RW	B
3201 hex	0	RW	B
3202 hex	0	RW	B

Index	Subindex	Access	Data attribute*1
3203 hex	0	RW	B
3204 hex	0	RW	B
3205 hex	0	RW	B
3206 hex	0	RW	B
3207 hex	0	RW	B
3208 hex	0	RW	B
3209 hex	0	RW	B
3210 hex	0	RW	B
3211 hex	0	RW	B
3212 hex	0	RW	B
3213 hex	0	RW	B
3214 hex	0	RW	B
3215 hex	0	RW	B
3216 hex	0	RW	B
3217 hex	0	RW	B
3218 hex	0	RW	B
3219 hex	0	RW	B
3220 hex	0	RW	B
3221 hex	0	RW	B
3222 hex	0	RW	B
3312 hex	0	RW	B
3313 hex	0	RW	B
3314 hex	0	RW	B
3317 hex	0	RW	B
3321 hex	0	RW	B
3323 hex	0	RW	R
3324 hex	0	RW	R
3325 hex	0	RW	R
3326 hex	0	RW	R
3327 hex	0	RW	R
3328 hex	0	RW	C
3329 hex	0	RW	C
3400 hex	0	RW	C
3401 hex	0	RW	C
3402 hex	0	RW	C
3403 hex	0	RW	C
3404 hex	0	RW	C
3405 hex	0	RW	C
3406 hex	0	RW	C
3407 hex	0	RW	C
3410 hex	0	RW	C
3411 hex	0	RW	C
3416 hex	0	RW	A
3417 hex	0	RW	A
3418 hex	0	RW	A
3419 hex	0	RW	A
3421 hex	0	RW	A
3432 hex	0	RW	A
3433 hex	0	RW	A
3434 hex	0	RW	A
3435 hex	0	RW	A
3436 hex	0	RW	A
3437 hex	0	RW	B

Index	Subindex	Access	Data attribute*1
3438 hex	0	RW	B
3439 hex	0	RW	B
3440 hex	0	RW	A
3441 hex	0	RW	A
3442 hex	0	RW	A
3504 hex	0	RW	C
3505 hex	0	RW	C
3508 hex	0	RW	B
3509 hex	0	RW	C
3511 hex	0	RW	B
3512 hex	0	RW	A
3513 hex	0	RW	A
3514 hex	0	RW	A
3515 hex	0	RW	C
3520 hex	0	RW	C
3521 hex	0	RW	B
3522 hex	0	RW	B
3525 hex	0	RW	B
3526 hex	0	RW	B
3602 hex	0	RW	B
3605 hex	0	RW	B
3606 hex	0	RW	B
3607 hex	0	RW	B
3608 hex	0	RW	B
3609 hex	0	RW	B
3610 hex	0	RW	B
3611 hex	0	RW	B
3614 hex	0	RW	B
3615 hex	0	RW	A
3618 hex	0	RW	R
3623 hex	0	RW	B
3624 hex	0	RW	B
3631 hex	0	RW	B
3632 hex	0	RW	B
3634 hex	0	RW	B
3635 hex	0	RW	B
3637 hex	0	RW	B
3638 hex	0	RW	C
3700 hex	0	RW	A
3701 hex	0	RW	R
3703 hex	0	RW	A
3704 hex	0	RW	C
3705 hex	0	RW	B
3706 hex	0	RW	B
3758 hex	0	RW	B
3759 hex	0	RW	R
3781 hex	0	RW	C
3800 hex	0	RW	C
3801 hex	0	RW	A
3803 hex	0	RW	A
3818 hex	0	RW	B
605B hex	0	RW	B
605C hex	0	RW	B

Index	Subindex	Access	Data attribute*1
605D hex	0	RW	B
605E hex	0	RW	B
6065 hex	0	RW	A
6067 hex	0	RW	A
607C hex	0	RW	C
607D hex	1	RW	A
607D hex	2	RW	A
6083 hex	0	RW	B
6084 hex	0	RW	B
6091 hex	1	RW	C
6091 hex	2	RW	C
6098 hex	0	RW	B
6099 hex	1	RW	B
6099 hex	2	RW	B

*1. This is the timing when changes to writable objects are enabled.

A: Always enabled

B: Prohibited to change during motor rotation or commands.

If it is changed during motor rotation or commands, the update timing will be unknown.

C: Updated after the control power is reset, or after a Config command is executed via EtherCAT communications.

D: Changeable only when the EtherCAT communications state is Pre-Operational (Pre-Op).

R: Updated when the control power supply is reset.

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 71.

*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 71.

*2. It is provided in English and Japanese.

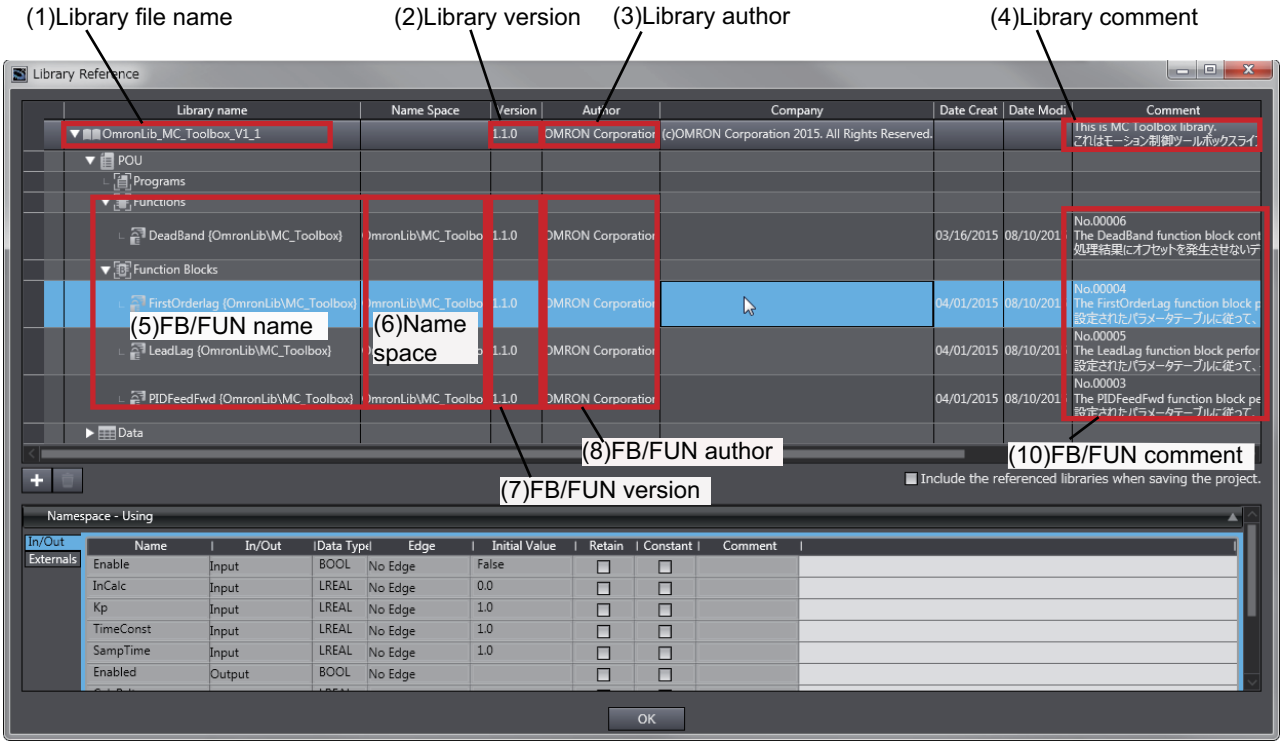
Referring to Attributes of Libraries, Function Blocks and Functions

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

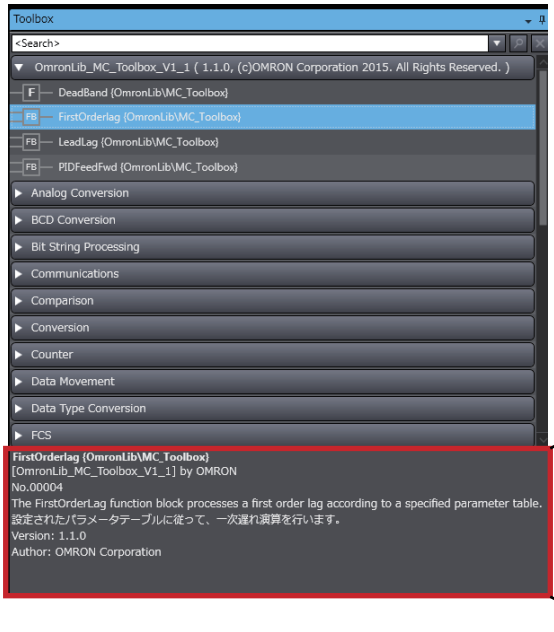
When you refer to the libraries, the library information is displayed at the locations shown below.



(b) Toolbox Pane

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

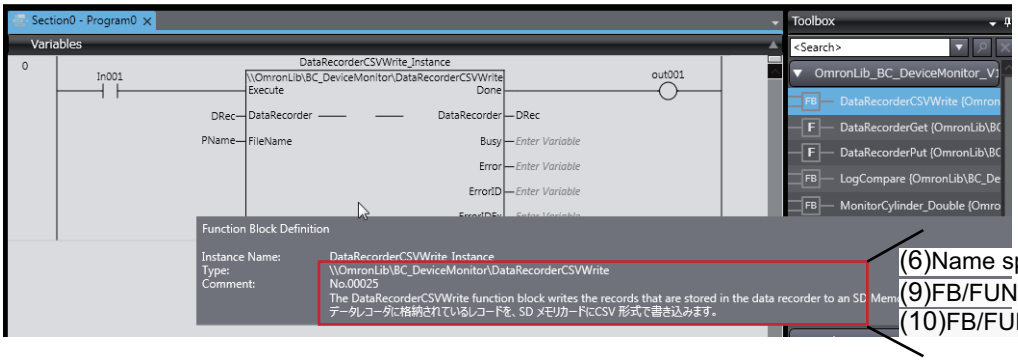
The text “by OMRON” which is shown on the right of the library name (1) indicates that this library was provided by OMRON.



(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.

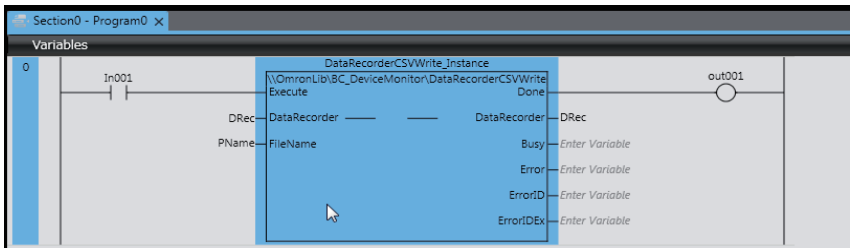
The following are the examples of items that you may need to customize.

- 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.

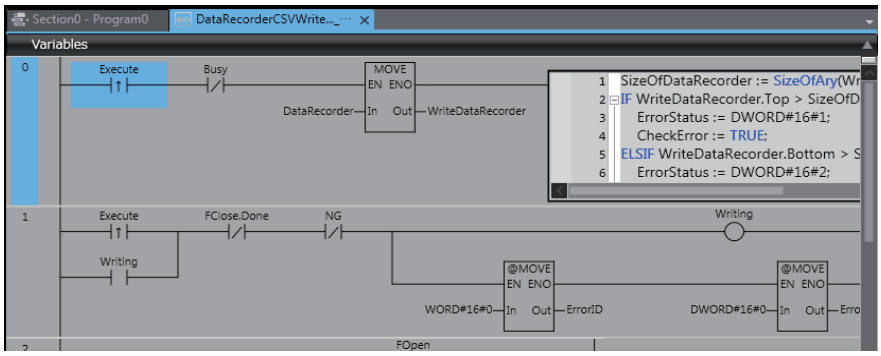
Use the following procedure to refer to the source codes of function blocks and functions.

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. W548-E1-05 0119