

**OMRON**

# Sysmac Library

---

## User's Manual for EtherCAT<sup>®</sup> 1S Series Library SYSMAC-XR011

## NOTE

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

No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

## Trademarks

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

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

## Copyrights

Microsoft product screen shots reprinted with permission from Microsoft Corporation.

# Introduction

---

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

This manual contains information that is necessary to use the function blocks in the EtherCAT 1S 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 1S Series Library is used to initialize the absolute encoder, back up and restore the parameters for an OMRON 1S-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 *SYS-MAC-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

---

<b>Introduction .....</b>	<b>1</b>
Features of the Library.....	1
Intended Audience.....	1
Applicable Products.....	1
<b>Manual Structure .....</b>	<b>2</b>
Special Information.....	2
<b>CONTENTS.....</b>	<b>4</b>
<b>Terms and Conditions Agreement.....</b>	<b>6</b>
Warranty, Limitations of Liability.....	6
Application Considerations.....	7
Disclaimers.....	7
<b>Safety Precautions .....</b>	<b>8</b>
Definition of Precautionary Information.....	8
Symbols.....	8
Cautions.....	9
<b>Precautions for Correct Use.....</b>	<b>10</b>
<b>Related Manuals .....</b>	<b>11</b>
<b>Revision History .....</b>	<b>14</b>
<b>Procedure to Use Sysmac Libraries .....</b>	<b>15</b>
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
<b>Common Specifications of Function Blocks.....</b>	<b>23</b>
Common Variables.....	24
Precautions.....	30
<b>Individual Specifications of Function Blocks.....</b>	<b>31</b>
InitEncoder_1S.....	32
Backup_1S.....	41
Restore_1S.....	55
MotorSetup_1S.....	64
Restart_1S.....	73
EasyTuning_1S.....	78
<b>Appendix .....</b>	<b>91</b>
Referring to Library Information.....	92
Referring to Function Block and Function Source Codes.....	95
Objects to Back up and Restore.....	96



# 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.

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

## Symbols

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

**Cautions** **Caution**

---

Read all related manuals carefully before you use this library.



---

Emergency stop circuits, interlock circuits, limit circuits, and similar safety measures must be provided in external control circuits.



---

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



---

The Sysmac Library and manuals are assumed to be used by personnel that is given in Intended Audience in this manual. Otherwise, do not use them.



---

The sample programming shows only the portion of a program that uses the function or function block from the library.



---

When using sample programming in actual devices, also program safety circuits, device interlocks, I/O with other devices, and other control procedures.



# 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 1S-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.
- After completion of Restore\_1S function block, cycle the control power supply to the 1S-series Servo Drive to apply the restored parameters.
- When the model which was backed up differs from the model to be restored, confirm that the restore can be performed with the object list which was backed up, and then change *NoproductChk* of this function block to TRUE.
- Confirm that the models to be backed up are applicable models and versions before you perform the backup.

## Using Sample Programming

---

- 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 <i>NJ/NY-series G code Instructions Reference Manual</i> (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 <i>NJ/NY-series NC Integrated Controller User's Manual</i> (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.
AC Servomotors/Servo Drivers 1S-series with Built-in EtherCAT® Communications User's Manual	I586	R88M-1□ R88D-1SN□-ECT	Learning how to use the Servomotors/Servo Drives with built-in EtherCAT Communications.	Describes the hardware, setup methods and functions of the Servomotors/Servo Drives with built-in EtherCAT Communications.
	I621	R88M-1A□ R88D-1SAN□-ECT		

# Revision History

---

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

**Cat. No. W571-E1-06**

↑  
Revision code

Revision code	Date	Revised content
01	July 2016	Original production
02	November 2016	Changed the manual name.
03	March 2017	Compatible with R88D-1SN□□□-ECT Version 1.1
04	January 2019	<ul style="list-style-type: none"> <li>• Added compatible models.</li> <li>• Compatible with R88D-1SN□□□-ECT Version 1.2</li> </ul>
05	April 2019	<ul style="list-style-type: none"> <li>• Added compatible models.</li> <li>• Compatible with 1S Series other than R88D-1SN□□□-ECT</li> </ul>
06	July 2019	Added following function blocks: <ul style="list-style-type: none"> <li>• MotorSetup_1S</li> <li>• Restart_1S</li> <li>• EasyTuning_1S</li> </ul>



# 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

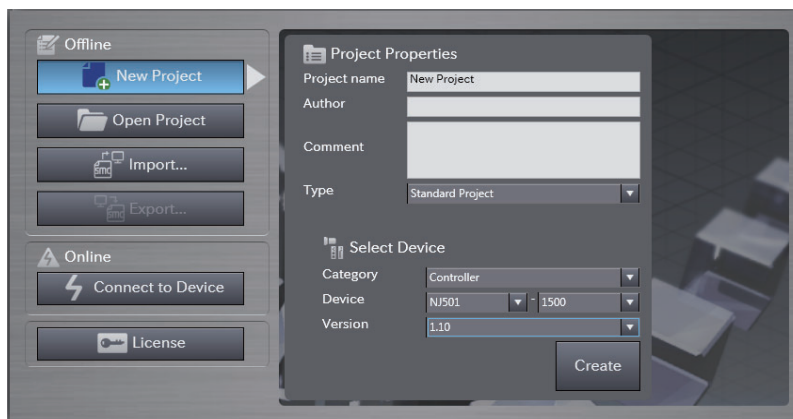


## Version Information

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.

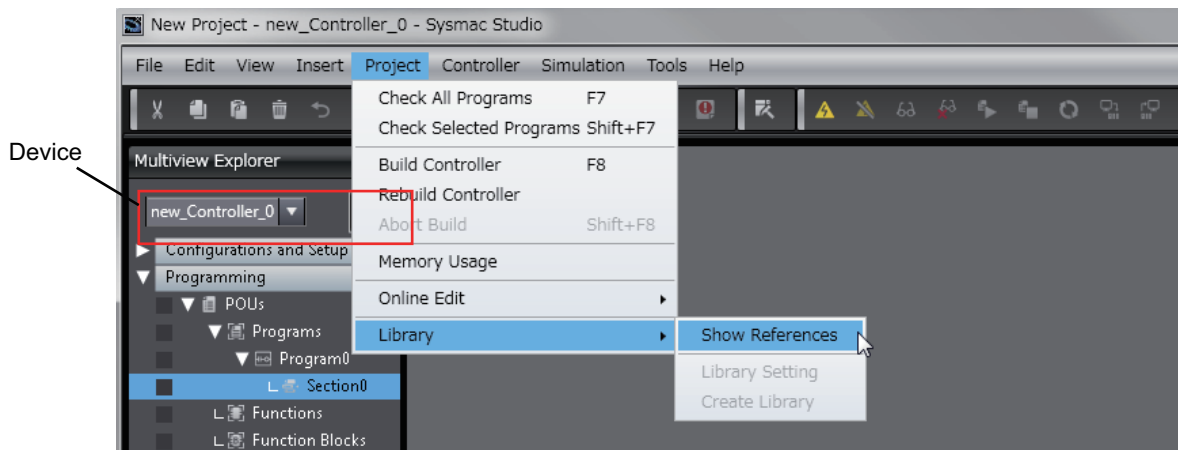


## Precautions for Correct Use


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.

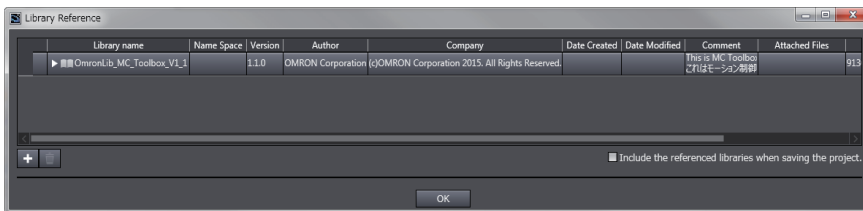
**2** Select **Project – Library – Show References**.



**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.



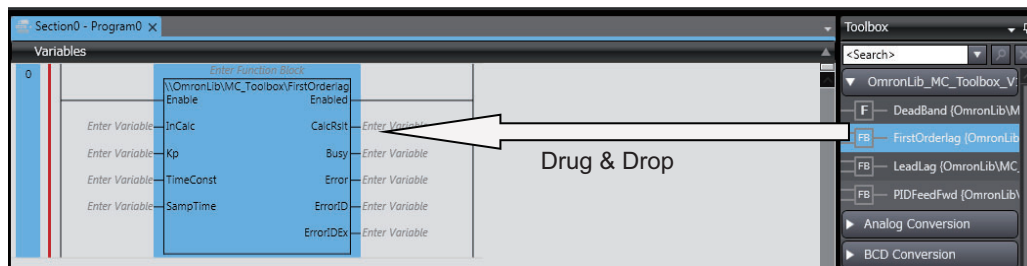
The Sysmac Library file is read into the project.

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

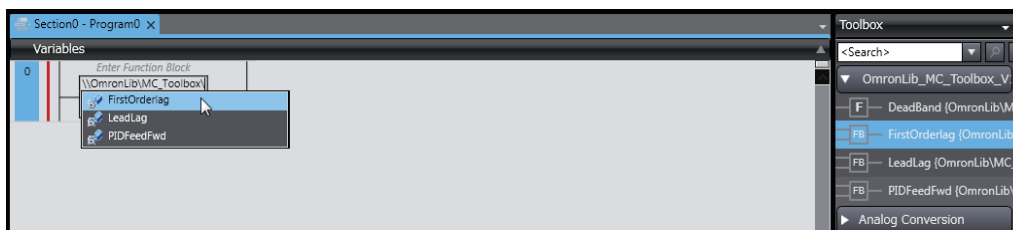
For the procedure for adding and setting libraries in the above screen, refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)*.

**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).



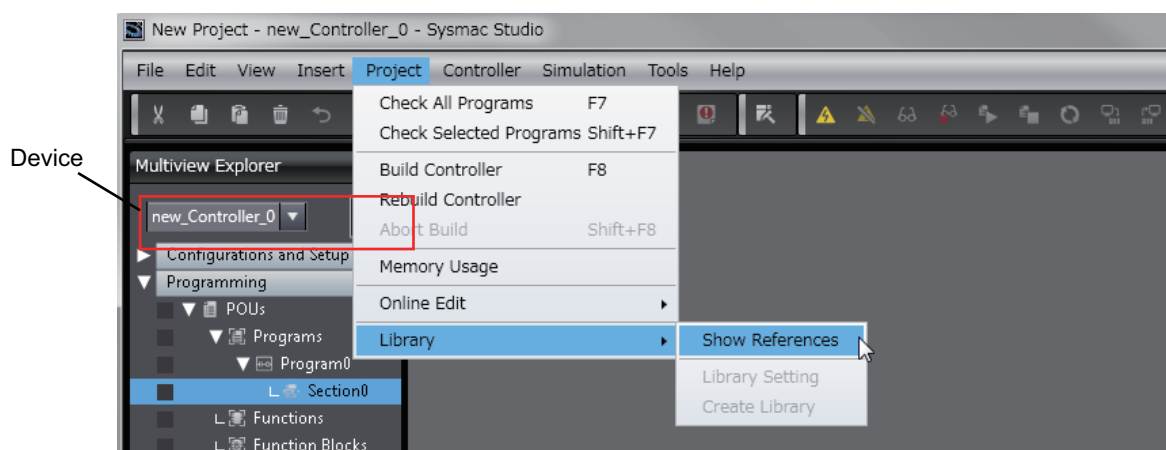
## Precautions for Correct Use

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

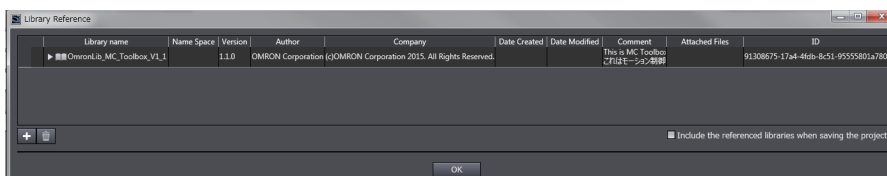
- 1 Start the Sysmac Studio and open a project in which any old-version Sysmac Library is included.
- 2 Select **Project – Library – Show References**.



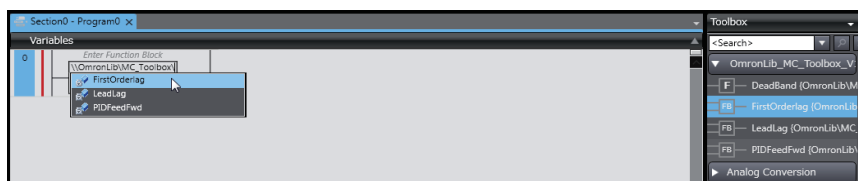
## Precautions for Correct Use

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

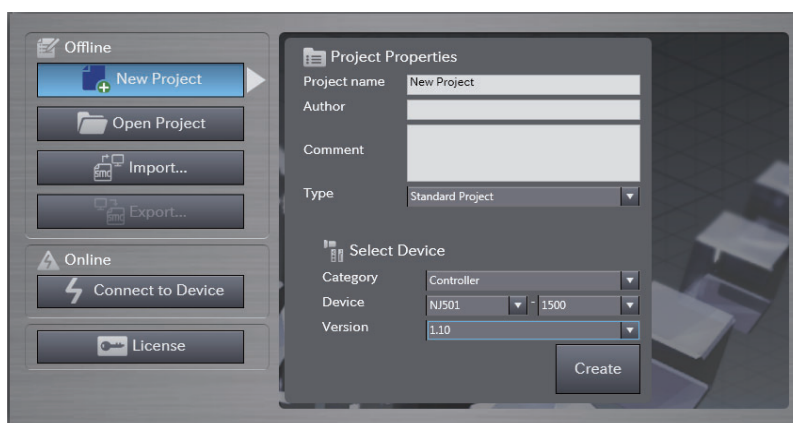
You can use Sysmac Libraries uploaded from a CPU Unit or an Industrial PC to your computer if they are not installed.

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.



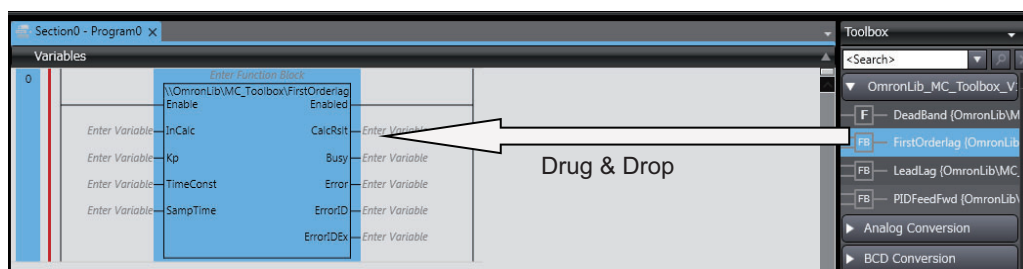
- 2 Connect the computer to the CPU Unit or the Industrial PC and place it online.

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

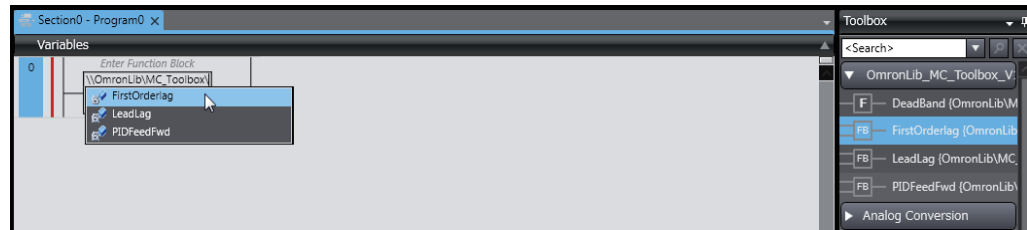
Now, when you select the Ladder Editor or ST Editor, the function blocks and functions included in the Sysmac Library used in the uploaded POUs appear in the Toolbox.

- 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 Studio 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 them 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

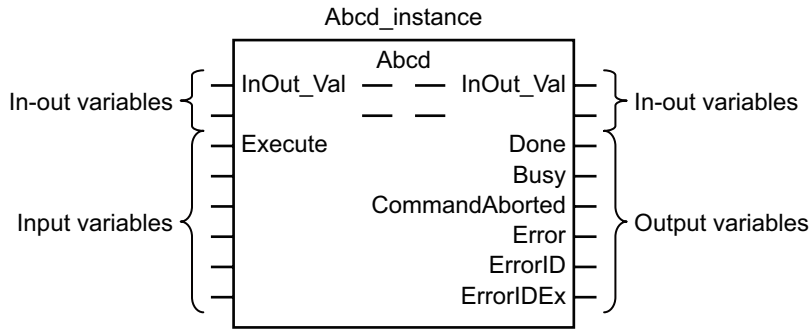
Common input variables and output variables used in functions and function blocks are as follows.

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 continuously 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.

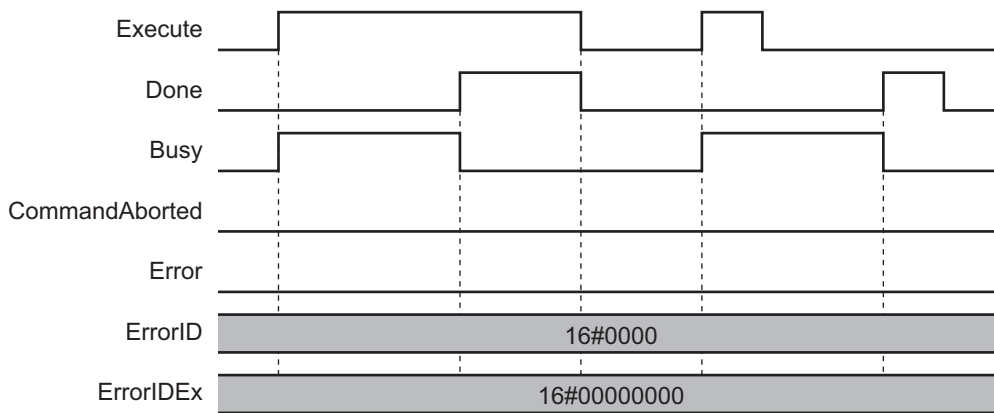


- If *Execute* is TRUE and *Done*, *CommandAborted*, or *Error* changes to TRUE, *Done*, *CommandAborted*, and *Error* changes to FALSE when *Execute* is changed to FALSE.
- If *Execute* is FALSE and *Done*, *CommandAborted*, or *Error* changes to TRUE, *Done*, *CommandAborted*, and *Error* changes to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

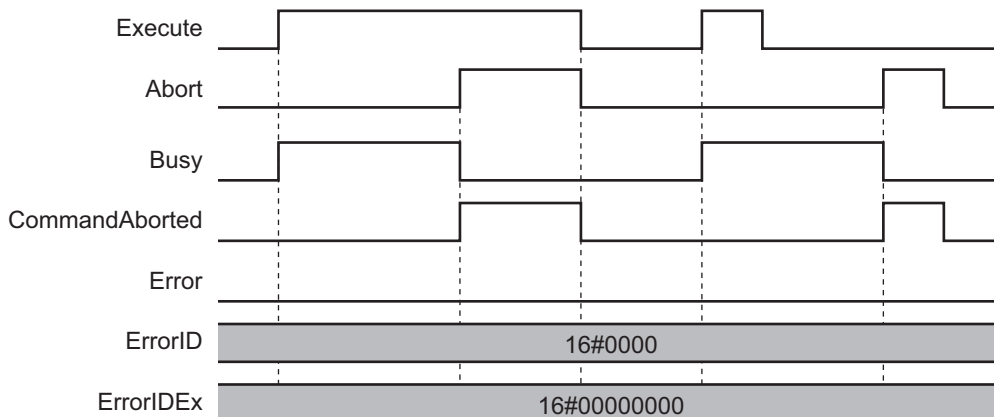
## Timing Charts

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

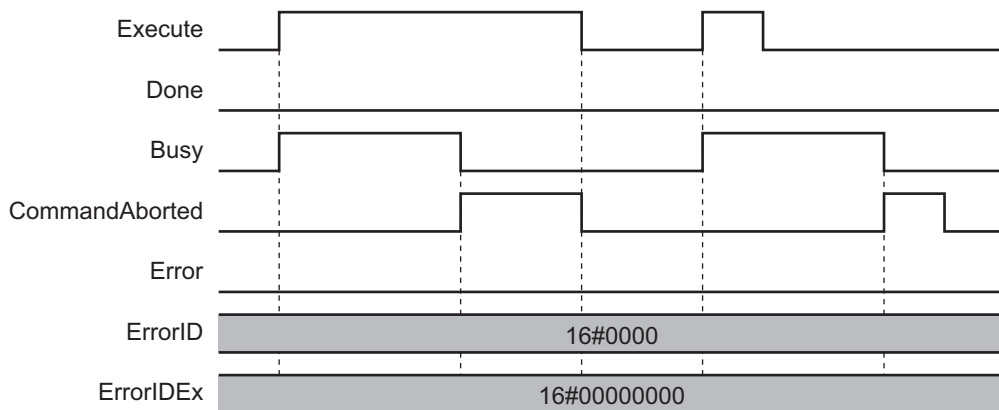
### ● Normal End



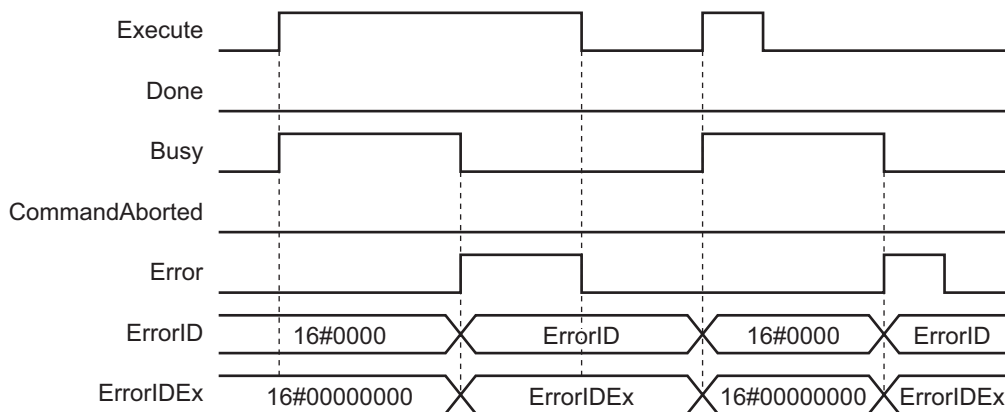
### ● Canceled Execution



● **Aborted Execution**

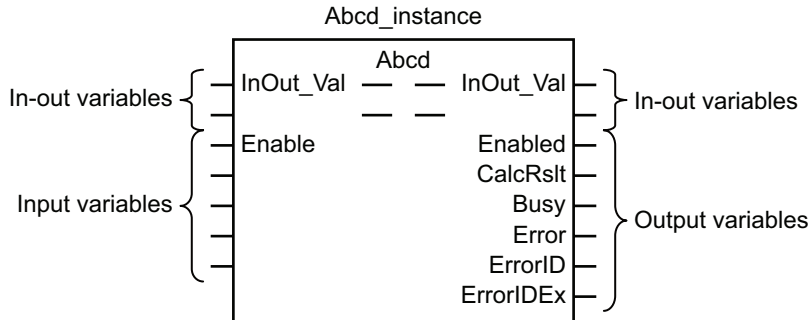


● **Errors**



## 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.

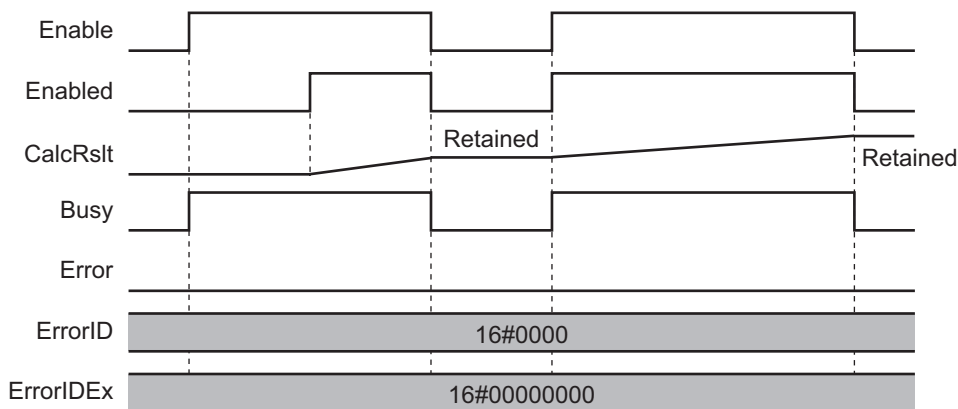


- If an error occurs, the relevant error code and expansion error code are set in *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code). The error codes are retained even after *Error* changes to FALSE, but *ErrorID* is set to 16#0000 and *ErrorIDEx* is set to 16#0000 0000 when *Enable* changes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

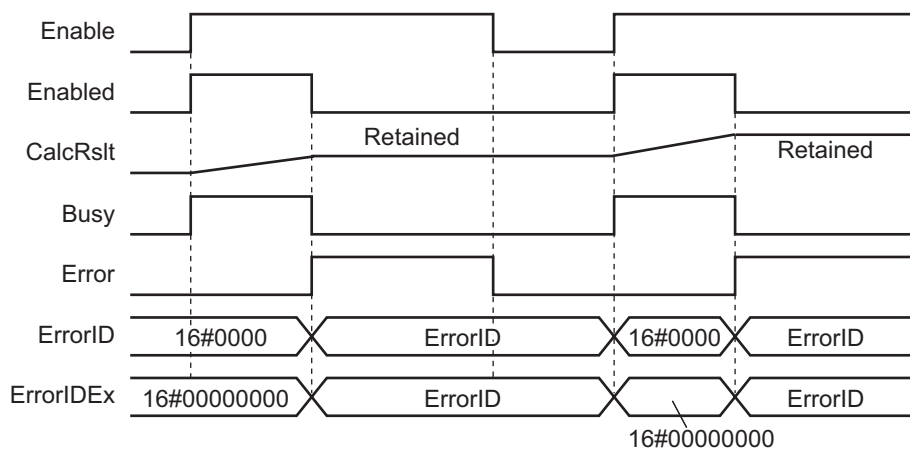
## Timing Charts

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

### ● Normal End



● Errors



# 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_1S	1S-series Servo Drive Absolute Encoder Initialize	P.32
Backup_1S	1S-series Servo Drive Parameter Backup	P.41
Restore_1S	1S-series Servo Drive Parameter Restore	P.55
MotorSetup_1S	1S-series Servo Drive Motor Replacement Detected	P.64
Restart_1S	1S-series Servo Drive Function Block Restarts	P.73
EasyTuning_1S	1S-series Servo Drive Easy tuning	P.78

# InitEncoder\_1S

The InitEncoder\_1S function block initializes the absolute encoder (clears the multi-rotation counter for the absolute encoder) for a 1S-series Servo Drive.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
InitEncoder_1S	1S-series Servo Drive Absolute Encoder Initialize	FB		InitEncoder_1S_instance( Execute, NodeAdr, TimeOut, Done, Busy, Error, ErrorID, ErrorIDEx);

## Library Information

Item	Description
Library file name	OmronLib_EC_1S_V2_0.slr
Namespace	OmronLib\EC_1S
Function block and function number	00042
Source code published/not published	Not published
Function block and function version	1.00

## Compatible Models

Item	Name	Model numbers	Version
Device	1S-series Servo Drive	R88D-1SN□□□-ECT	Version 1.00 or later
		R88D-1SAN□□□-ECT	

## 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* <sup>1</sup>	---	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	This is the error ID for an Expansion Unit Hardware Error. The value is DWORD#16#0 for a normal end.	*2	---	---

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

\*2. For details, refer to *Troubleshooting* on page 37

Name	Bo ole an	Bit strings					Integers							Real num- bers		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.

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.

## Timing Charts

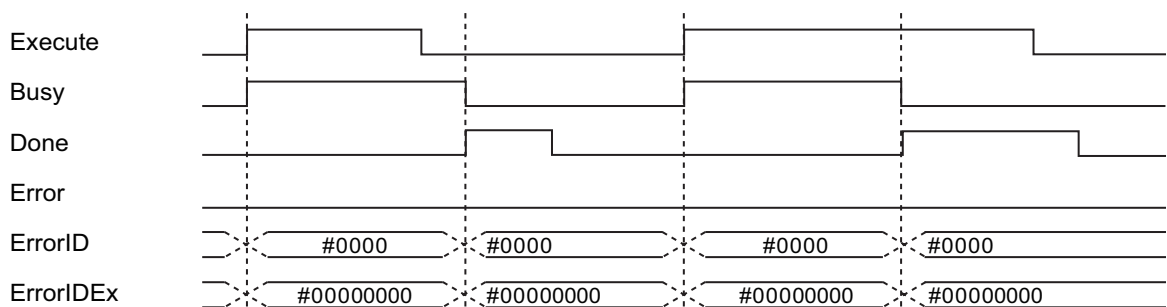
The following figures show the timing charts for the function block.

- *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.

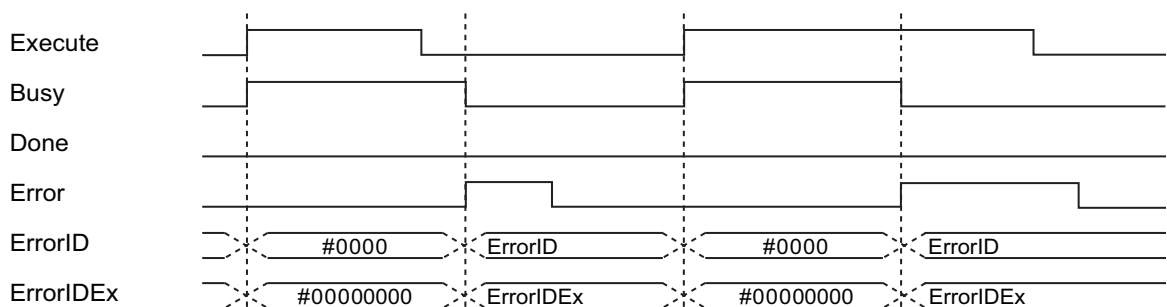
You can find out the cause of the error by accessing the values output to *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code).

- 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



## Related System-defined Variables

Name	Meaning	Data type	Description
_EC_MBXSlavTbl	Message Communications Enabled Slave Table	Array[1..512* <sup>1</sup> ] 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.

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

## Additional Information

This function block initializes absolute encoders by writing 6a64 6165 hex to the object of servo parameter Index:4510 hex Subindex:F1 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.
- After completion of the function block, the 1S-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.
- 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.
- Sample programming shows only the portion of a program that uses the function or function block from the library.
- When using sample programming in 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.
- 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.

## Troubleshooting

Error code	Expansion error code	Status	Description	Correction
16#0000	16#00000000	Normal End	---	---
16#0400	16#00000000	Input Value Out of Range	The value of <i>NodeAdr</i> is out of range.	Set the value of <i>NodeAdr</i> 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 <sup>*1</sup>	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.
16#3C2C	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 1S-series Servo Drive.
	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 1S-series Servo Drive.
	16#00000004	Unsupported Device	The device is not supported by this function block.	Check that the specified device is a 1S-series Servo Drive.

\*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

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 0 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 0 is disabled, and that the same instance is not being executed.
- 2** Execute INITENCODER\_1S\_instance.

### LD

#### ● Internal Variables

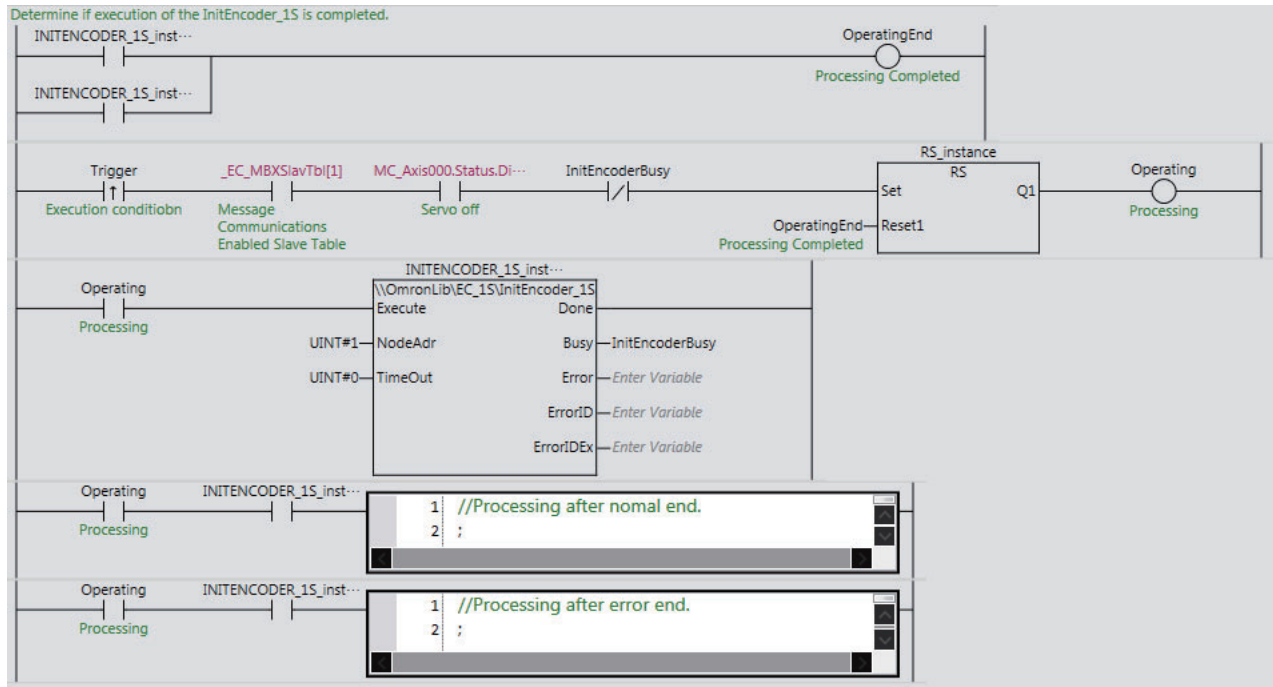
Name	Data type	Default	Comment
INITENCODER_1S_instance	OmronLib\EC_1S\InitEncoder_1S		
OperatingEnd	BOOL		Processing completed
Trigger	BOOL	FALSE	Execution condition
Operating	BOOL		Processing
RS_instance	RS		
InitEncoderBusy	BOOL		

#### ● 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.





## ST

### ● Internal Variables

Name	Data type	Default	Comment
INITENCODER_1S_instance	OmronLib\EC_1S\InitEncoder_1S		
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	

### ● 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.

```
// 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_1S instruction.
IF (OperatingStart=TRUE) THEN
  INITENCODER_1S_instance(
    Execute :=FALSE);
  OperatingStart:=FALSE;
END_IF;
// Execute Backup_1S instruction.
```

```
IF (Operating=TRUE) THEN
  INITENCODER_1S_instance(
    Execute :=TRUE,
    NodeAdr:=UINT#1,
    TimeOut:=UINT#0,
    Busy=>InitEncoderBusy);
  IF (INITENCODER_1S_instance.Done=TRUE) THEN
    // Processing after normal end.
    Operating:=FALSE;
  END_IF;
  IF (INITENCODER_1S_instance.Error=TRUE) THEN
    // Processing after error end.
    Operating:=FALSE;
  END_IF;
END_IF;
```

# Backup\_1S

The Backup\_1S function block backs up the parameters for a 1S-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_1S	1S-series Servo Drive Parameter Backup	FB		Backup_1S_instance( Execute, NodeAdr, DirName, OverWrite, TimeOut, Done, Busy, SdoBusy, Error, ErrorID, ErrorIDEx);

## Library Information

Item	Description
Library file name	OmronLib_EC_1S_V2_0.slr
Namespace	OmronLib\EC_1S
Function block and function number	00043
Source code published/not published	Not published
Function block and function version	1.00

## Compatible Models

Item	Name	Model numbers	Version
Device	1S-series Servo Drive	R88D-1SN□□□-ECT	Version 1.00 or later
		R88D-1SAN□□□-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 <sup>*1</sup>	---	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 1S-series Servo Drive. <sup>*3</sup>	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.

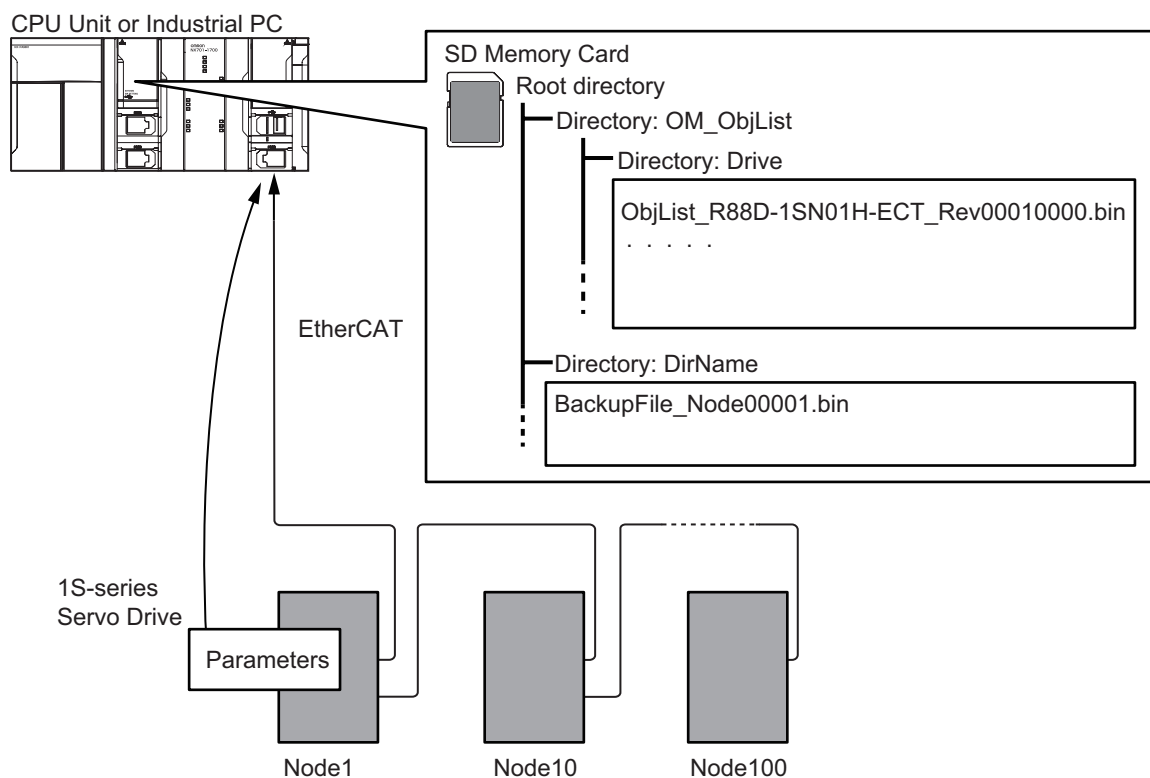
\*3. You can use it to identify that communications processing is in progress.

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

Name	Bo ole an	Bit strings					Integers							Real num- bers		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\_1S function block reads parameters for the 1S-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.
- Object lists that are provided by OMRON are installed in 'C:\OMRON\Data\Attachement' when installing libraries. Uncompress a ZIP file for an applicable series and store the files in the SD Memory Card.
- 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

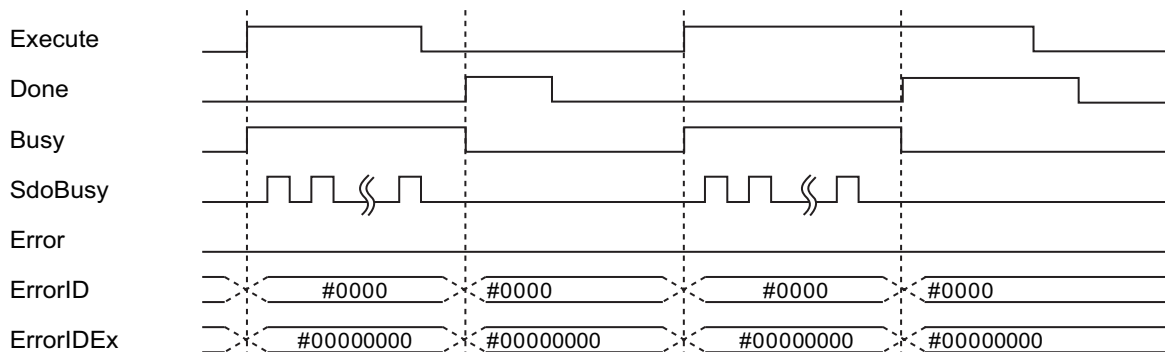
- 1** Create a directory named OM\_ObjList\Drive in the root directory on the SD Memory Card and store all the object lists provided by OMRON in it.
- 2** Execute the function block.  
For *NodeAdr* (Slave Node Address), specify the node address of the backup target slave.  
If necessary, set *DirName* (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.

## Timing Charts

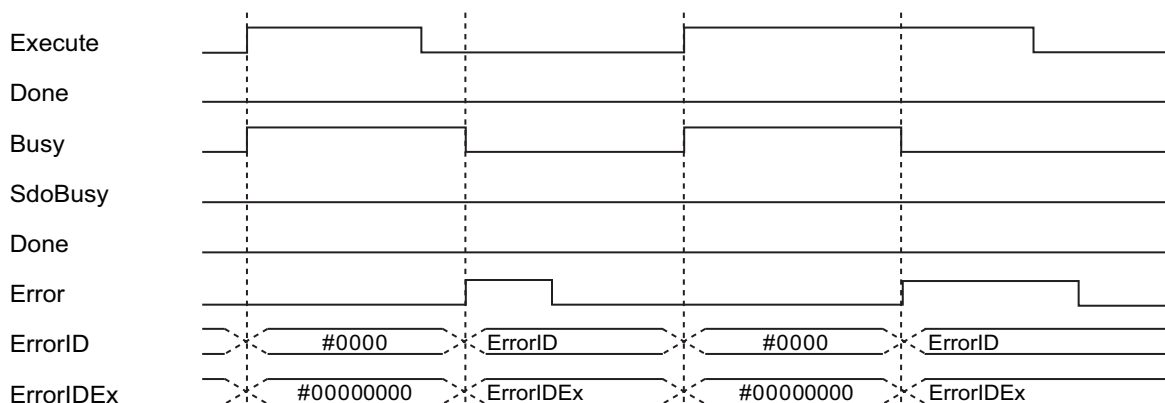
The following figures show the timing charts for the function block.

- *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



## 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* <sup>1</sup> ] 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.

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

## 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.
- If you turn OFF the power supply to the 1S-series Servo Drive when replacing a 1S-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 1S-series Servo Drive.
- Refer to *Objects to Back up and Restore* on page 96 for details on the objects to back up and restore.



## 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.
- Do not change the names and the contents of any object list.
- 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.
- Sample programming shows only the portion of a program that uses the function or function block from the library.
- When using sample programming in 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.
- 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.

## Troubleshooting

Error code	Expansion error code	Status	Description	Correction
16#0000	16#00000000	Normal End	---	---
16#0400	16#00000000	Input Value Out of Range	The value of <i>NodeAdr</i> is out of range.	Set the value of <i>NodeAdr</i> correctly.
			The directory name specified with <i>DirName</i> contains one or more characters that cannot be used.	Set <i>DirName</i> correctly.
			The directory name specified with <i>DirName</i> is too long.	Check the length of the directory name specified with <i>DirName</i> 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 <i>Busy</i> 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 <i>OverWrite</i> is FALSE.	Change the directory in which to store the backup file. Or, delete the existing backup file in advance.
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.

Error code	Expansion error code	Status	Description	Correction
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 <sup>*1</sup>	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.
16#3C2D	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#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#00000003	Unsupported Device	The device is not supported by this function block.	Check that the specified device is a 1S-series Servo Drive.

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

## Sample Programming

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

### Conditions

- The target 1S-series Servo Drive is Node 1.
- The target 1S-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\_1S\_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\_1S\_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

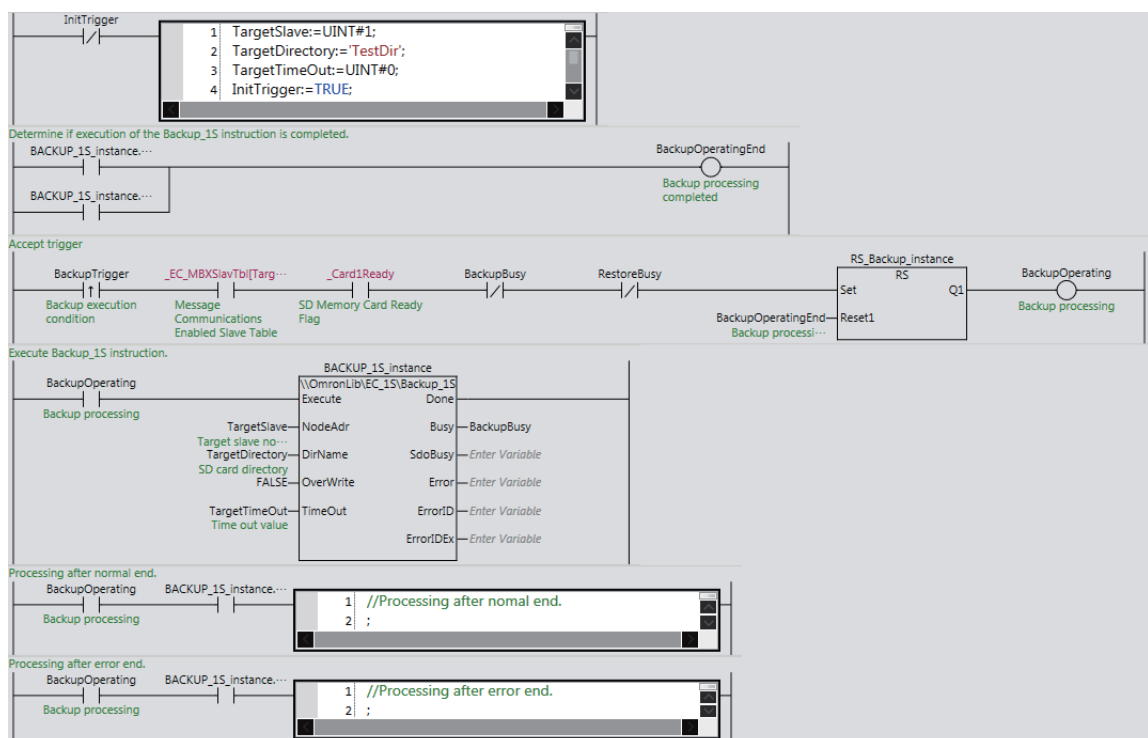
## ● Internal Variables

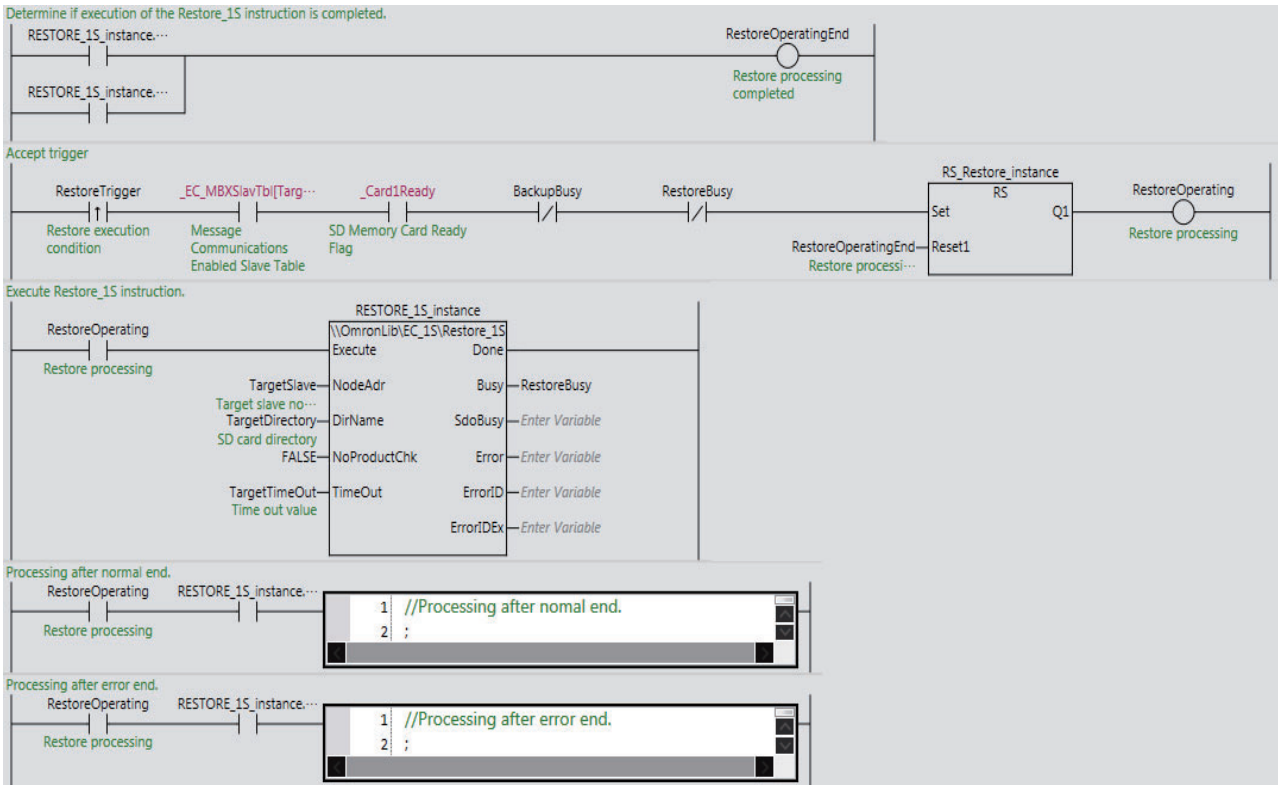
Name	Data type	Default	Comment
BACKUP_1S_instance	OmronLib\EC_1S\Backup_1S		
RESTORE_1S_instance	OmronLib\EC_1S\Restore_1S		
InitTrigger	BOOL	FALSE	
BackupTrigger	BOOL	FALSE	Backup execution condition
RestoreTrigger	BOOL	FALSE	Restore execution condition
TargetSlave	UINT	1	Target slave node Address
TargetDirectory	STRING[41]	"	SD card directory
TargetTimeOut	UINT	0	Time out value
BackupBusy	BOOL		
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		

## ● 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

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





## ST

### ● Internal Variables

Name	Data type	Default	Comment
BACKUP_1S_instance	OmronLib\EC_1S\Backup_1S		
RESTORE_1S_instance	OmronLib\EC_1S\Restore_1S		
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

### ● 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

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

```
// Backup and Resotre parameters setting.
IF InitTrigger=FALSE THEN
  TargetSlave:=UINT#1;
  TargetDirectory:='TestDir';
  TargetTimeOut:=UINT#0;
  InitTrigger:=TRUE;
END_IF;
// Detect when BackupTrigger changes to TRUE.
IF ( (BackupTrigger=TRUE) AND (BackupLastTrigger=FALSE) AND (_EC_MBXSlavTbl[Target-
Slave]=TRUE) AND
      (_Card1Ready=TRUE) AND (BackupBusy=FALSE) AND
      (RestoreBusy=FALSE) ) THEN
  BackupOperatingStart:=TRUE;
  BackupOperating :=TRUE;
END_IF;
BackupLastTrigger:=BackupTrigger;
// Initialize Backup_1S instruction.
IF (BackupOperatingStart=TRUE) THEN
  BACKUP_1S_instance(
    Execute :=FALSE);
  BackupOperatingStart:=FALSE;
END_IF;
// Execute Backup_1S instruction.
IF (BackupOperating=TRUE) THEN
  BACKUP_1S_instance(
```

```

Execute :=TRUE,
NodeAdr:=TargetSlave,
DirName:=TargetDirectory,
OverWrite:=FALSE,
TimeOut:=TargetTimeOut,
Busy=>BackupBusy);
IF (BACKUP_1S_instance.Done=TRUE) THEN
  // Processing after normal end.
  BackupOperating:=FALSE;
END_IF;
IF (BACKUP_1S_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
      (_Card1Ready=TRUE) AND (BackupBusy=FALSE) AND
      (RestoreBusy=FALSE) ) THEN
  RestoreOperatingStart:=TRUE;
  RestoreOperating :=TRUE;
END_IF;
RestoreLastTrigger:=RestoreTrigger;
// Initialize Restore_1S instruction.
IF (RestoreOperatingStart=TRUE) THEN
  RESTORE_1S_instance(
  Execute :=FALSE);
  RestoreOperatingStart:=FALSE;
END_IF;
// Execute Restore_1S instruction.
IF (RestoreOperating=TRUE) THEN
  RESTORE_1S_instance(
  Execute :=TRUE,
  NodeAdr:=TargetSlave,
  DirName:=TargetDirectory,
  NoProductChk:=FALSE,
  TimeOut:=TargetTimeOut,
  Busy=>RestoreBusy);
  IF (RESTORE_1S_instance.Done=TRUE) THEN
    // Processing after normal end.
    RestoreOperating:=FALSE;
  END_IF;
  IF (RESTORE_1S_instance.Error=TRUE) THEN
    // Processing after error end.
    RestoreOperating:=FALSE;
  END_IF;
END_IF;
END_IF;

```



# Restore\_1S

The Restore\_1S function block restores the backup file(s) created by the Backup\_1S function block on the SD Memory Card inserted in the Controller to a 1S-series Servo Drive.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
Restore_1S	1S-series Servo Drive Parameter Restore	FB		<pre>Restore_1S_instance(   Execute,   NodeAdr,   DirName,   NoProductChk,   TimeOut,   Done,   Busy,   SdoBusy,   Error,   ErrorID,   ErrorIDEx);</pre>

## Library Information

Item	Description
Library file name	OmronLib_EC_1S_V2_0.slr
Namespace	OmronLib\EC_1S
Function block and function number	00044
Source code published/not published	Not published
Function block and function version	1.00

## Compatible Models

Item	Name	Model numbers	Version
Device	1S-series Servo Drive	R88D-1SN□□□-ECT	Version 1.00 or later
		R88D-1SAN□□□-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 <sup>*1</sup>	---	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 1S-series Servo Drive. <sup>*3</sup>	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.

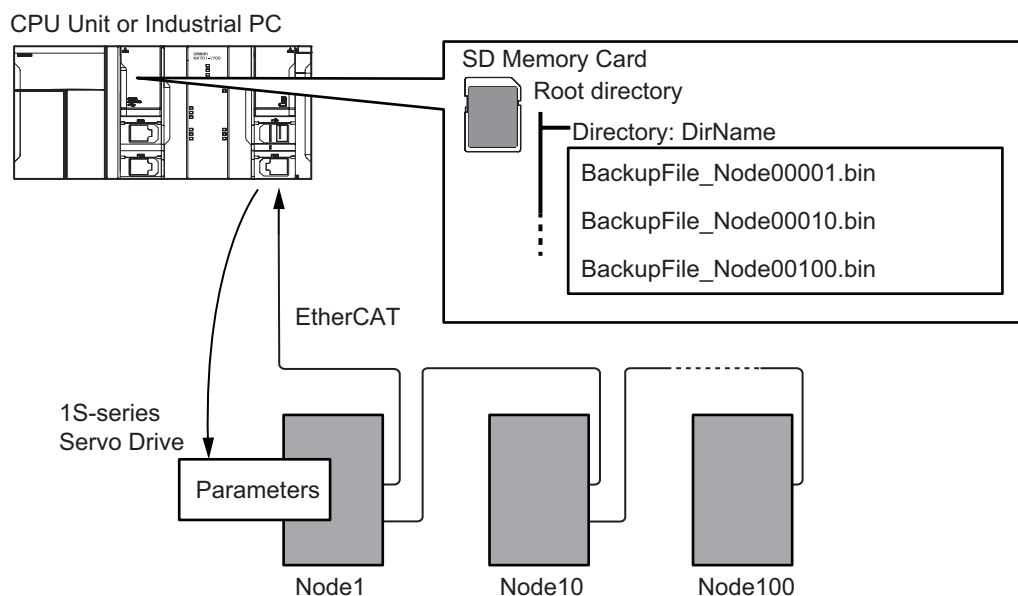
\*3. You can use it to identify that communications processing is in progress.

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

Name	Bo ole an	Bit strings				Integers							Real num- bers		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\_1S function block restores the backup file(s) in the directory specified with *DirName* (Directory Name) on the SD Memory Card to the 1S-series Servo Drive specified with *NodeAdr* (Node Address).
- Before you restore a file, use the Backup\_1S 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. If the parameters in the backup file do not exist in the restore destination device, an SDO Abort error will occur.



### ● 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 1S-series Servo Drive when execution of the function block is ended.

## Timing Charts

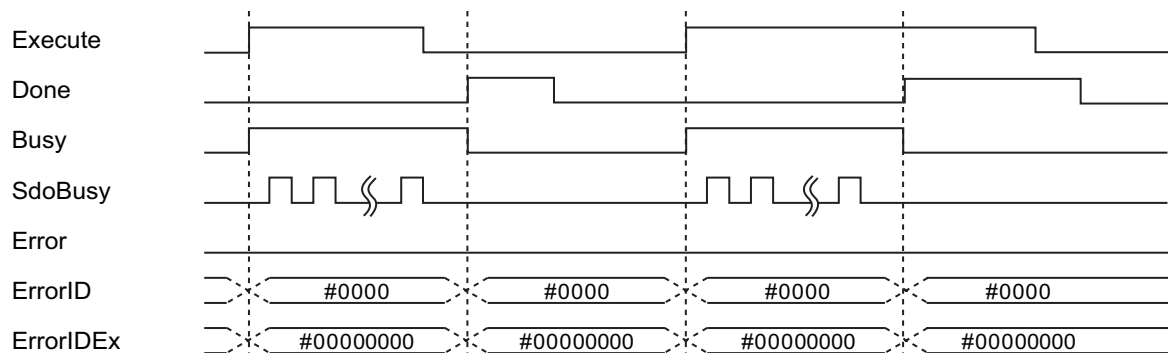
The following figures show the timing charts for the function block.

- *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.

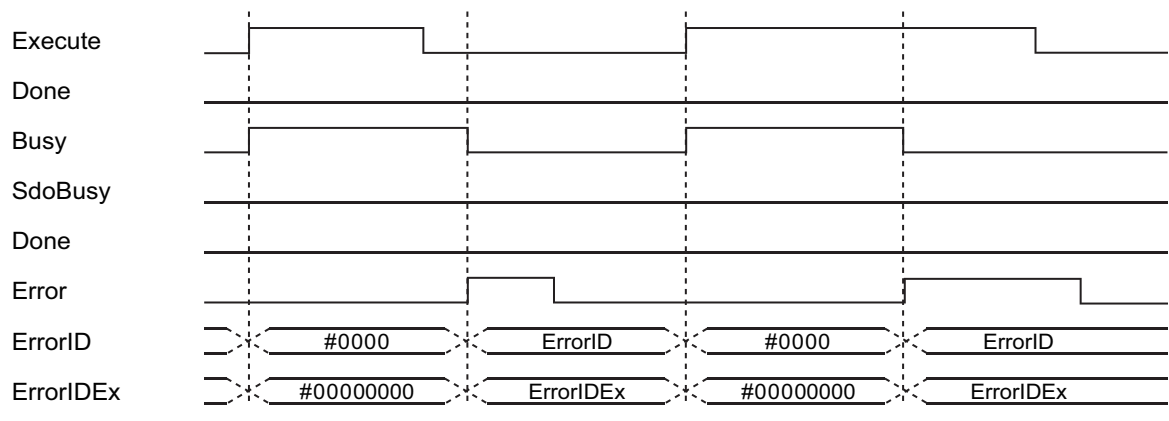
You can find out the cause of the error by accessing the values output to *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code).

- 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



## 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* <sup>1</sup> ] 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.

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

## 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.
- If you turn OFF the power supply to the 1S-series Servo Drive when replacing a 1S-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 1S-series Servo Drive.
- Refer to *Objects to Back up and Restore* on page 96 for details on the objects to back up and restore.

## 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.
- 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\_1S function block, cycle the control power supply to the 1S-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.
- 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 1S-series Servo Drive before you attempt to restore the parameters.
- 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.

## Troubleshooting

Error code	Expansion error code	Status	Description	Correction
16#0000	16#00000000	Normal End	---	---
16#0400	16#00000000	Input Value Out of Range	The value of <i>NodeAdr</i> is out of range.	Set the value of <i>NodeAdr</i> correctly.
			The directory name specified with <i>DirName</i> contains one or more characters that cannot be used.	Set <i>DirName</i> correctly.
			The directory name specified with <i>DirName</i> is too long.	Check the length of the directory name specified with <i>DirName</i> so that it is 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#1403	16#00000000	File Does Not Exist	There is no backup file to restore in the directory specified with <i>DirName</i> .	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 <i>Busy</i> 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.



Error code	Expansion error code	Status	Description	Correction
16#1802	16#00000000	EtherCAT Time-out	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 <sup>*1</sup>	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.
16#3C2E	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#00000002	Backup Data Disabled	The backup file is not created correctly.	Create again the backup file correctly.
	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#00000004	Unsupported Device	The device is not supported by this function block.	Check that the specified device is a 1S-series Servo Drive.

\*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\_1S* on page 41.

# MotorSetup\_1S

The MotorSetup\_1S function block resets the Motor Replacement Detected (Error No. 95.05) for a 1S-series Servo Drive.

Function block name	Name	FB/FUN	Graphic expression	ST expression
MotorSetup_1S	Reset the Motor Replacement Detected	FB		MotorSetup_1S_instance( Execute, NodeAdr, TimeOut, Done, Busy, Error, ErrorID, ErrorIDEx);

## Library Information

Item	Description
Library file name	OmronLib_EC_1S_V2_0.slr
Namespace	OmronLib\EC_1S
Function block and function number	00173
Source code published/not published	Not published
Function block and function version	1.00

## Compatible Models

Item	Name	Model numbers	Version
Device	1S-series Servo Drive	R88D-1SN□□□-ECT	Version 1.00 or later
		R88D-1SAN□□□-ECT	

## 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 for resetting.	1 to 512* <sup>1</sup>		1
TimeOut	Timeout Time	Input	0: 2.0s 1 to 65535: 0.1 to 6553.5s	0 to 65535	0.1s	0 (2.0s)
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	This is the error ID for an Expansion Unit Hardware Error. The value is DWORD#16#0 for a normal end.	*2		

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

\*2. For details, refer to *Troubleshooting* on page 68

Name	Bo ole an	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 resets the Motor Replacement Detected for the specified Servo Drive. Use this function block when the Motor Replacement Detected (Error Display No. 95.05) occurs because of replacing a Servo Drive or a Servomotor.

You cannot reset The Motor Replacement Detected only this function block. When this function block changes done, you need execute the Unit Restart or cycle the control power supply of the Servo Drive.

After sending the reset command to the Servo Drive, this function block waits for the response only for the time specified with *TimeOut* (Timeout). It waits that the reset process ends after receiving the response from the Servo Drive. This function block becomes Normal End when the reset process ends, Done changes to TRUE.

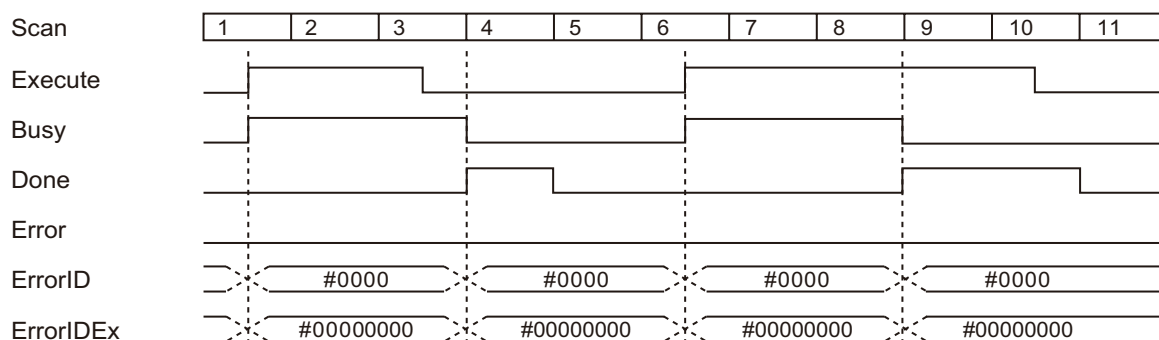
This function block becomes Error End when the reset process does not end within one minute, Error changes to TRUE.

## Timing Charts

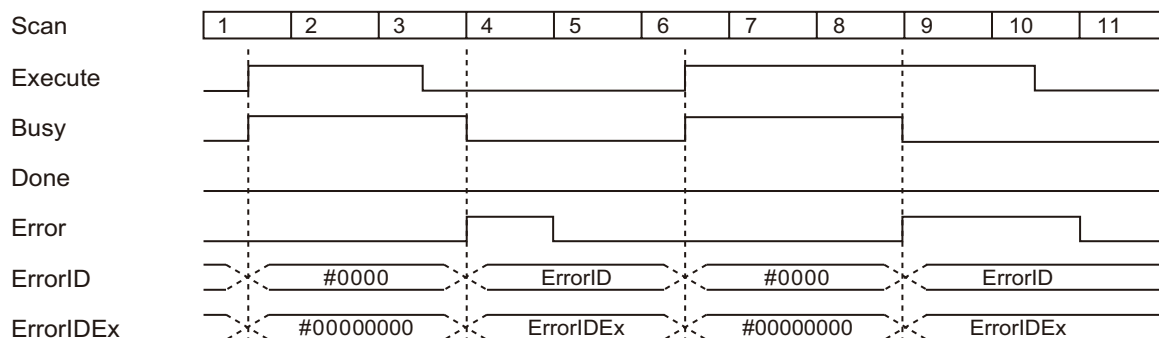
The following figures show the timing charts for the function block.

- Busy (Executing) changes to TRUE when Execute changes to TRUE.
- Done changes to TRUE when the reset 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 resets the Motor Replacement Detected by writing 7465 736d hex to the object of servo parameter Index: 4410 hex/Subindex:F1 hex (Motor 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.
- Execute the function block while the Servo is OFF.
- 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.
- Sample programming shows only the portion of a program that uses the function or function block from the library.
- When using sample programming in 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.
- 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 and EC\_CoESDORead exceeded the maximum number of instructions that can be executed at the same time.

## Troubleshooting

Error code	Expansion error code	Status	Description	Correction
16#0000	16#00000000	Normal End	---	---
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 <sup>*1</sup>	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.
16#3CEE	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 1S-series Servo Drive.
16#3CEE	16#00000003	Reset Failure	The Motor Replacement Detected resetting was not completed within one minute.	Execute the function block again.
16#3CEE	16#00000004	Unsupported Device	The device is not supported by this function block.	Check that the specified device is a 1S-series Servo Drive.

\*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

This sample programming resets the Motor Replacement Detected for a Servo Drive.

### Conditions

- The target Servo Drive is Node 1.
- The target Servo Drive is assigned to axis 0 in the Motion Control Function Module.
- The timeout time is 2 s (default).
- The Motor Replacement Detected (Error No. 95.05) has been occurring by replacement of Servomotor or Servo Drive.

### Processing

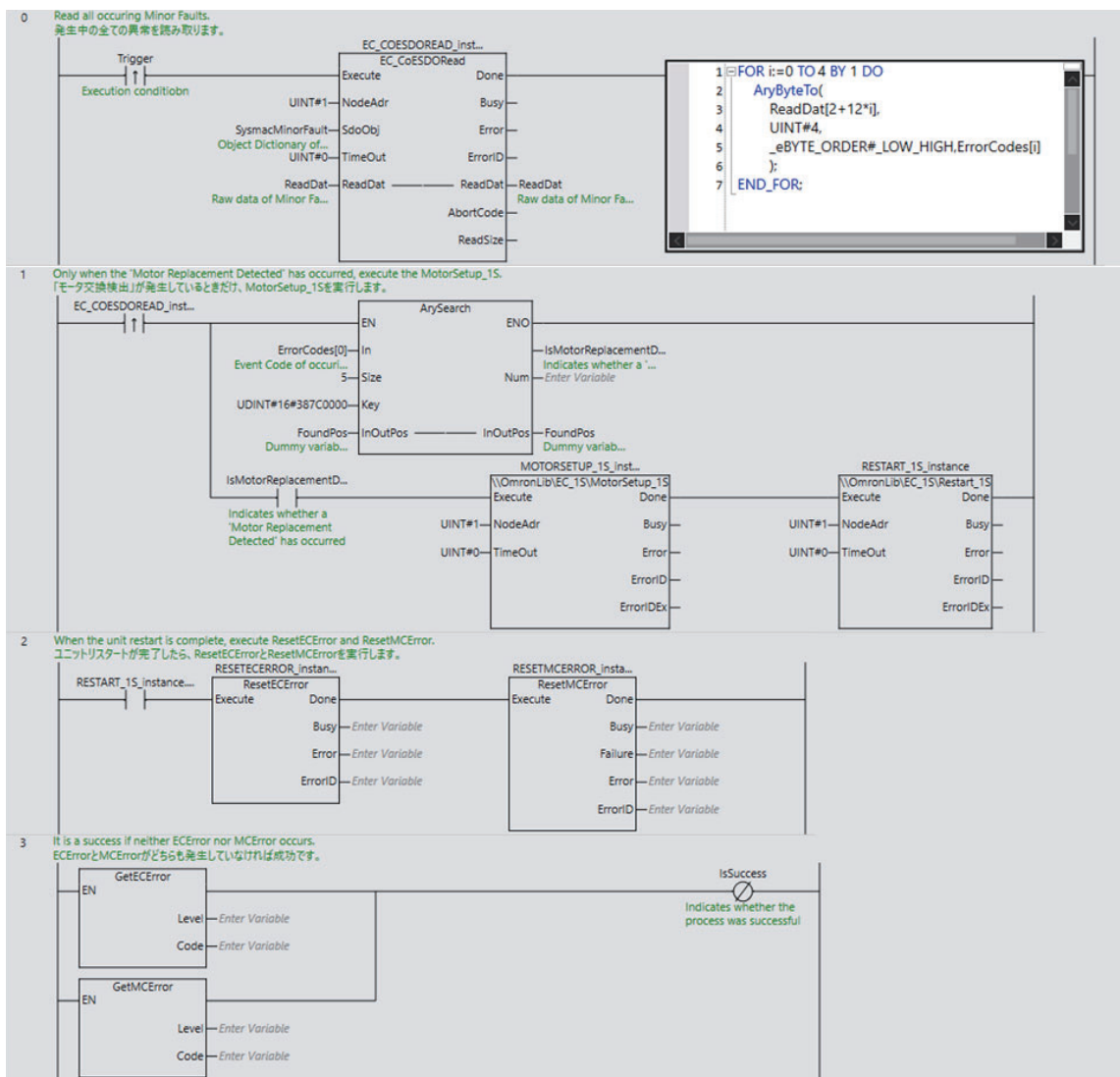
- 1** Make sure that the Motor Replacement Detected (Error No. 95.05) occurs, the target slave is in a state that allows communications, that axis 0 is disabled.
- 2** For reset the Motor Replacement Detected, execute MOTORSETUP\_1S\_instance.
- 3** For restart Servo Drive, execute RESTART\_1S\_instance.
- 4** For reset the EtherCAT Slave Communications Error, execute ResetECError and ResetMCError.

### LD

#### ● Internal Variables

Name	Data type	Default	Comment
Trigger	BOOL		Execution condition
IsMotorReplacement Detected	BOOL		Indicates whether a 'Motor Replacement Detected' has occurred
IsSuccess	BOOL		Indicates whether the process was successful
SysmacMinorFault	_sSDO_ACCESS	(Index := 16#2004, Subindex := 0, IsCompleteAccess := True)	Object Dictionary of Sysmac Minor Fault for Complete Access
ReadDat	ARRAY[0..61] OF BYTE		Raw data of Minor Fault(Need convert)
i	DINT		Index for 'FOR' instruction
ErrorCodes	ARRAY[0..4] OF UDINT		Event Code of occurring Minor Fault
FoundPos	UINT		Dummy variable(Required to use ArySearch)

Name	Data type	Default	Comment
EC_COESDORREAD_insta nce	EC_CoESDORRead		
MOTORSETUP_1S_ instance	OmronLib\EC_1S\MotorSetup_1S		
RESTART_1S_instance	OmronLib\EC_1S\Restart_1S		
RESETECERROR_ instance	ResetECError		
RESETECERROR_ instance	ResetMCError		





# ST

## ● Internal Variables

Name	Data type	Default	Comment
Trigger	BOOL		Execution condition
IsMotorReplacement Detected	BOOL		Indicates whether a 'Motor Replacement Detected' has occurred
IsSuccess	BOOL		Indicates whether the process was successful
SysmacMinorFault	_sSDO_ACCESS	(Index := 16#2004, Subindex := 0, IsCompleteAccess := True)	Object Dictionary of Sysmac Minor Fault for Complete Access
ReadDat	ARRAY[0..61] OF BYTE		Raw data of Minor Fault(Need convert)
i	DINT		Index for 'FOR' instruction
ErrorCodes	ARRAY[0..4] OF UDINT		Event Code of occurring Minor Fault
FoundPos	UINT		Dummy variable(Required to use ArySearch)
R_TRIG_instance	R_TRIG		
EC_COESDOREAD_instance	EC_CoESDORead		
MOTORSETUP_1S_instance	OmronLib\EC_1S\MotorSetup_1S		
RESTART_1S_instance	OmronLib\EC_1S\Restart_1S		
RESETECERROR_instance	ResetECError		
RESETMCERROR_instance	ResetMCErr		

```

R_TRIG_instance (Clk:=EC_COESDOREAD_instance.Done);
//Read all occurring Minor Faults.
EC_COESDOREAD_instance(
  Execute:=Trigger,
  NodeAdr:=UINT#1,
  SdoObj:=SysmacMinorFault,
  ReadDat:=ReadDat);
IF R_TRIG_instance.Q THEN
  FOR i:=0 TO 4 BY 1 DO
    AryByteTo(ReadDat [2+12*i],UINT#4,_eBYTE_ORDER#_LOW_HIGH,ErrorCodes[i]);
  END_FOR;
  IsMotorReplacementDetected := ArySearch(
    In:=ErrorCodes[0],
    Size:=UINT#5,
    Key:=UDINT#16#387C0000,
    InOutPos:=FoundPos);

```

```
END_IF;
//Only when the 'Motor Replacement Detected' has occurred, execute the MotorSetup_1S.
MOTORSETUP_1S_instance(
    Execute:=R_TRIG_instance.Q AND IsMotorReplacementDetected,
    NodeAdr:=UINT#1);
RESTART_1S_instance(
    Execute:=MOTORSETUP_1S_instance.Done,
    NodeAdr:=UINT#1);
//When the unit restart is complete, execute ResetMCErrror and ResetECErrror.
RESETECErrror_instance(Execute:=RESTART_1S_instance.Done);
RESETMCErrror_instance(Execute:=RESETECErrror_instance.Done);
//It is a success if neither ECErrror nor MCErrror occurs.
IsSuccess:= NOT GetECErrror() AND NOT GetMCErrror();
```

# Restart\_1S

The Restart\_1S function block restarts a 1S-series Servo Drive.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
Restart_1S	1S-series Servo Drive Restart	FB		Restart_1S_instance( Execute, NodeAdr, TimeOut, Done, Busy, Error, ErrorID, ErrorIDEx);

## Library Information

Item	Description
Library file name	OmronLib_EC_1S_V2_0.slr
Namespace	OmronLib\EC_1S
Function block and function number	00174
Source code published/not published	Not published
Function block and function version	1.00

## Compatible Models

Item	Name	Model numbers	Version
Device	1S-series Servo Drive	R88D-1SN□□□-ECT	Version 1.00 or later
		R88D-1SAN□□□-ECT	

## 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 restart	1 to 512* <sup>1</sup>		1
TimeOut	Timeout time	Input	0 : 2.0s 1 to 65535: 0.1 to 6553.5s	0 to 65535	0.1s	0 (2.0s)
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	This is the error ID for an Expansion Unit Hardware Error. The value is DWORD#16#0 for a normal end.	*2		---

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

\*2. For details, refer to *Troubleshooting* on page 77

Name	Bo ole an	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 executes unit restart for the specified Servo Drive.

Use this function block when you need the Unit Restart such as cases after the absolute encoder initialization or the Motor Replacement Detected resetting.

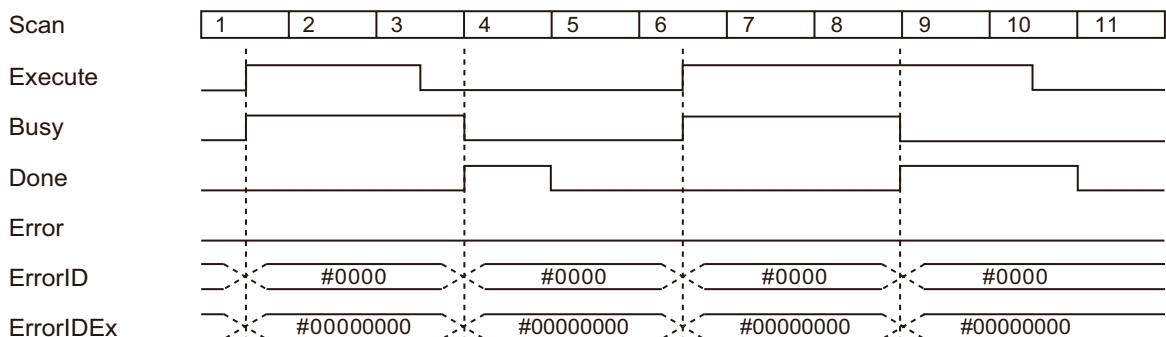
After sending the unit restart command to the Servo Drive, this function block waits for the response only for the time specified with *TimeOut* (Timeout). After the function block receives the response from the Servo Drive, it waits that the Servo Drive re-start is done. This function block becomes Normal End when the Unit Restart completes, Done changes to TRUE.

## Timing Charts

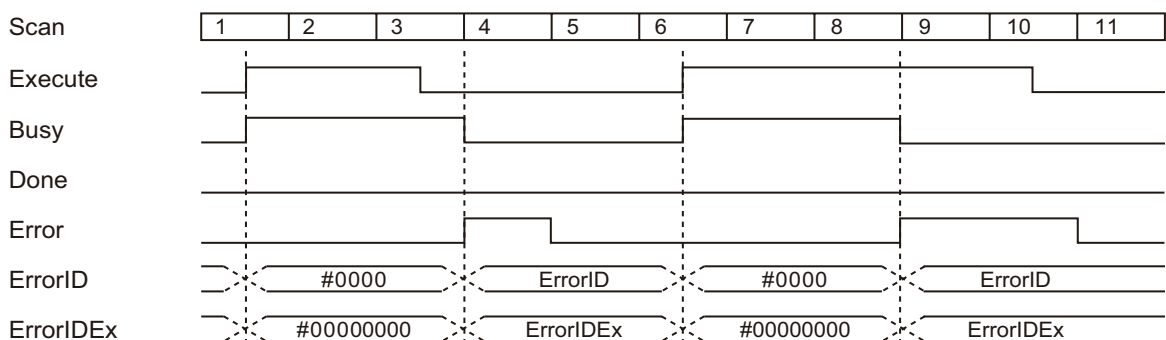
The following figures show the timing charts for the function block.

- *Busy* (Executing) changes to TRUE when *Execute* changes to TRUE.
- *Done* changes to TRUE when the Unit Restart completes.
- 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 restarts Servo Drive by writing 7465 7365 7261 hex to the object of servo parameter Index:2400 hex Subindex:01 hex (Unit Restart) 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.
- Execute the function block while the Servo is OFF.
- 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.
- If the specified Servo Drive is assigned to an Axis Variable, the EtherCAT Slave Communication Error Occurs after this function block execution. To reset it, execute ResetMCError or MC\_Reset.
- After the initialization process completes, the Servo Drive is not in a state that allows message communications. To start message communications, execute the ResetECError.
- 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.

## Troubleshooting

Error code	Expansion error code	Status	Description	Correction
16#0000	16#00000000	Normal End	---	---
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.
16#3CEF	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 1S-series Servo Drive.
16#3CEF	16#00000004	Unsupported Device	The device is not supported by this function block.	Check that the specified device is a 1S-series Servo Drive.

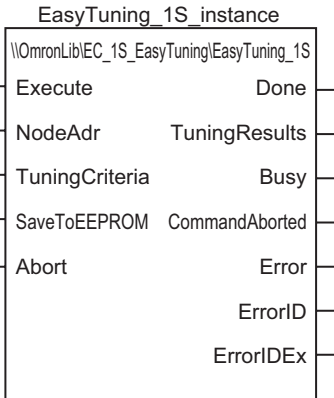
\*1. An SDO abort code is shown here. Refer to SDO Abort Codes in the instructions reference manual for a list of SDO abort codes.

## Sample Programming

Refer to Sample Programming of *MotorSetup\_1S* on page 64

# EasyTuning\_1S

The EasyTuning\_1S function block tunes an axis by executing the easy tuning for a 1S-series Servo Drive.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
EasyTuning_1S	1S-series Servo Drive Easy Tuning	FB		EasyTuning_1S_instance( Execute, NodeAdr, TuningCriteria, SaveToEEPROM, Abort, Done, TuningResults, Busy, CommandAborted, Error, CommandAborted, Error, ErrorID, ErrorIDEx);

## Library Information

Item	Description
Library file name	OmronLib_EC_1S_EasyTuning_V1_0.slr
Namespace	OmronLib\EC_1S_EasyTuning
Function block and function number	00236
Source code published/not published	Not published
Function block and function version	1.00

## Compatible Models

Item	Name	Model numbers	Version
Device	1S-series Servo Drive	R88D-1SN□□□-ECT	Version 1.00 or later
		R88D-1SAN□□□-ECT	



## Variables

Name	Meaning	I/O	Description	Valid range	Unit	Default
Execute	Execute	Input	TRUE: Executes the instruction. FALSE: Does not execute the instruction.	Depends on data type.	---	FALSE
NodeAdr	Slave Node Address	Input	Node address of the Servo Drive to tune gain parameters	1 to 512* <sup>1</sup>	---	1
Tuning Criteria	Tuning criteria	Input	Tuning methods and criteria inputs	---	---	---
SaveToEEPROM	Save to EEPROM	Input	TRUE: Saves results to EEPROM. FALSE: Does not save results to EEPROM.	Depends on data type.	---	FALSE
Abort	Abort	Input	TRUE: Aborts the execution. FALSE: Does not abort the execution.	Depends on data type.	---	FALSE
Done	Done	Output	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	Depends on data type.	---	---
Tuning Results	Tuning Results	Output	Displays tuning results.	---		---
Busy	Executing	Output	TRUE: Execution processing is in progress. FALSE: Execution processing is not in progress.	Depends on data type.		---
Command Aborted	Command Aborted	Output	TRUE: Execution processing is aborted. FALSE: Execution processing is not aborted.	Depends on data type.		---
Error	Error	Output	TRUE: Error end FALSE: Normal end, execution in progress, or execution condition not met	Depends on data type.		---
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	This is the error ID for an Expansion Unit Hardware Error. The value is DWORD#16#0 for a normal end.	*2		---

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

\*2. For details, refer to *Troubleshooting* on page 87

## Structures

The data type of the TuningCriteria input variable to this function block is OmronLib\EC\_1S\_EasyTuning\TUNING\_CRITERIA. The table below gives the specifications.

Name	Meaning	Description	Data Type	Valid range	Unit	Default
TuningCriteria	Tuning criteria		OmronLib\EC_1S_EasyTuning\TUNING_CRITERIA	---	---	
EstimateInertiaRatio	Update Estimated Inertia Ratio	TRUE: The amount of load inertia is automatically estimated and the setting of Inertia Ratio (3001 hex - 01 hex) is updated based on the result. FALSE: The Drive's present setting of Inertia Ratio (3001 hex - 01 hex) is maintained.	BOOL	Depends on data type	---	TRUE
CompensateFriction	Update Compensate Friction	TRUE: The load friction is automatically estimated and the torque compensation settings (Viscous Friction Coefficient (3310 hex - 01 hex), Positive Dynamic Friction Compensation (3310 hex - 03 hex), and Negative Dynamic Friction Compensation (3310 hex - 04 hex)) are updated based on the result. FALSE: The torque compensation values (Viscous Friction Coefficient (3310 hex - 01 hex), Positive Dynamic Friction Compensation (3310 hex - 03 hex), and Negative Dynamic Friction Compensation (3310 hex - 04 hex)) are set to 0 to disable the friction compensation.	BOOL	Depends on data type.	---	TRUE
CompensateUnbalancedLoad	Update Compensate Unbalanced Load	TRUE: The amount of unbalanced load is automatically estimated and the setting of the torque compensation value (Unbalanced Load Compensation (3310 hex - 02 hex)) are updated based on the result. FALSE: The torque compensation value (Unbalanced Load Compensation (3310 hex - 02 hex)) is set to 0 to disable the friction compensation.	BOOL	Depends on data type.	---	TRUE
AdjustNotchFilter	Adjust Notch Filter	TRUE: The resonance frequency of the load is automatically estimated and the settings of Notch Filters (3321 hex to 3324 hex) are updated based on the result. After the tuning, Adaptive Notch Filter (3320 hex - 01 hex) is set to Disabled. FALSE: The Drive's present settings of Adaptive Notch Filter (3320 hex - 01 hex) and Notch Filters (3321 hex to 3324 hex) are maintained.	BOOL	Depends on data type.	---	TRUE
ObtainBestStabilizationTime	Obtain Best Stabilization Time	TRUE: Although vibration occurs while auto tuning is being performed, the function block automatically sets the stabilization time as short as possible within the range that does not cause micro vibration in the Drive. FALSE: Set the stabilization time that is applied as the tuning completion condition.	BOOL	Depends on data type.	---	FALSE

Name	Meaning	Description	Data Type	Valid range	Unit	Default
StabilizationTime	Stabilization Time Criterion	Stabilization Time Criterion if ObtainBestStabilizationTime is FALSE. This parameter is not used if ObtainBestStabilizationTime is TRUE.	UINT	Depends on data type.	ms	100
VibrationLevel	Vibration Detection Level Criterion	Adjust the gain so that the torque vibration does not exceed this setting value. Set the percentage to the rated torque of the motor.	REAL	0 to 500	%	1
PositionWindow	Position Window	Set the in-position width used for measuring the stabilization time. This setting value is applied to the Positioning Completion Notification - Positioning Window (3B51 hex - 01 hex).	UDINT	1 to 2147483647		
FocusOnOvershoot	Focus on Overshoot	TRUE: Focus on overshoot suppression Priority is given to suppression of overshoot in the tuning. FALSE: Focus on positioning Priority is given to reducing the stabilization time. Depending on the device and tuning conditions, overshoot may occur.	BOOL	Depends on data type.	---	FALSE
LinearMovement Interpolation	Linear Movement Interpolation	TRUE: Linear 1st order interpolation is applied. FALSE: Smooth* <sup>1</sup> 2nd order interpolation is applied.	BOOL	Depends on data type.	---	TRUE

\*1. Available in R88D-1SAN□□□-ECT or R88D-1SN□□□-ECT Version 1.2 and later.

The data type of the TuningResults input variable to this function block is OmronLib\EC\_1S\_EasyTuning\TUNING\_RESULTS. The table below gives the specifications.

Name	Meaning	Description	Data Type	Valid range	Unit	Default
TuningResults			OmronLib\EC_1S_EasyTuning\TUNING_RESULTS	---	---	---
TuningStep	Tuning Step	Number of current Tuning step.	USINT	*1		0
AllTrials	All Trials	All measurement results.	ARRAY[0..32] OF OmronLib\EC_1S_EasyTuning\STRIALS	---		---
Stabilization Time	Stabilization Time	Gives the stabilization time.	UDINT	Depends on data type.		0
Overshoot	Overshoot	Gives the overshoot.	REAL	Depends on data type.		0
Speed Proportional Gain1	Speed Proportional Gain 1	Gives the 1st velocity proportional gain.	REAL	Depends on data type.	Hz	0
Position Proportional Gain1	Position Proportional Gain 1	Gives the 1st position proportional gain.	REAL	Depends on data type.	Hz	0
Estimated InertiaRatio	Estimated Inertia Ratio	Gives the Inertia Ratio Estimate Value.	UDINT	Depends on data type.	%	0
Estimated Unbalanced Load	Estimated Unbalanced Load	Gives the Unbalanced Load Estimate Value.	REAL	Depends on data type.	%	0
Estimated Dynamic Friction	Estimated Viscous Friction	Gives the Dynamic Friction Compensation Estimate Value.	REAL	Depends on data type.	%	0
Estimated ViscousFriction	Estimated Viscous Friction	Gives the Viscous Friction Estimate Value.	REAL	Depends on data type.	%	0
CommandTime	Command Time	Gives the command time.	UDINT	Depends on data type.	ms	0
TactTime	Tact Time	Gives the tact time.	UDINT	Depends on data type.	ms	0
Command Speed	Command Speed	Gives the command Velocity.	UDINT	Depends on data type.	r/min	0
MotorSpeed	Motor Speed	Gives the motor Velocity.	UDINT	Depends on data type.	r/min	0
NotchFilter Frequency3	Notch Filter Frequency 3	Gives the notch frequency of the 3rd resonance suppression notch filter.	REAL	50 to 5000	Hz	0
NotchFilter Frequency4	Notch Filter Frequency 4	Gives the notch frequency of the 4th resonance suppression notch filter.	REAL	50 to 5000	Hz	0
FinalTrial	Final Trial	Gives the adopted trial number. Reference only while Done is TRUE.	USINT	Depends on data type.	---	0

\*1. For details, refer to *TuningStep* on page 84.

Name	Bo le an	Bit strings				Integers							Real num- bers		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													
TuningCriteria	Structure sTUNING_CRITERIA																			
SaveToEEPROM	OK																			
Abort	OK																			
Done	OK																			
TuningResults	Structure sTUNING_RESULTS																			
Busy	OK																			
Command Aborted	OK																			
Error	OK																			
ErrorID			OK																	
ErrorIDEx				OK																

## Function

- This function block perform the easy tuning for the 1S-series Servo Drive at the node address specified with NodeAdr (Node Address). You can obtain a result equivalent to that of the easy tuning performed in Setting and Tuning Wizard on Sysmac Studio. Use this function block if you want to perform the easy tuning without Sysmac Studio. However, this function block does not support a damping control. If you need the damping control, use Sysmac Studio.
- To perform a tuning with this function block, you need to operate an axis by motion control instructions. Execute motion control instructions intermittently until Done changes to TRUE. The motion control instruction must fulfill the following conditions. Otherwise, errors may occur or you obtain invalid tuning results.
  - a) Instruction type: Positioning
  - b) Time from a motion control start to a positioning completion: Within 10 seconds
  - c) Time from a positioning completion to the next motion control start: 100ms or more
- You can abort tuning process by changing the value of Abort to TRUE. If this function block is re-executed after the aborting, the tuning process performed all over again. If you need re-execute this function block, make sure Abort is FALSE before change the value of Execute to TRUE.
- Specify parameters for tuning, using TuningCriteria.
- You can select if the tuning results are saved in a Servo Drive with SaveToEEPROM (Save to EEPROM).
  - a) If an input value is TRUE, the results are saved to the EEPROM of the Servo Drive at the timing when the value of Done changes to TRUE. If the value of CommandAborted or Error is TRUE, the results are not saved.
  - b) If an input value is FALSE, the tuning results are not saved to the EEPROM of the Servo Drive. If you cut off the power supply or execute the Unit Restart to the Servo Drive without saved, the results are lost. If you need to save the tuning results manually, execute Store Parameters by writing 6576 6173 hex to Index:1010 hex/Subindex:01 hex (Store Parameters) with EC\_CoES-DOWrite.
- TuningResults gives the tuning progress status and the final result.
  - a) TuningStep gives the tuning progress status. When you attempt to know the meaning of the TuningStep, refer to TuningStep that will be mentioned later.
  - b) AllTrials gives the latest 33 tuning results. When the function block performs the 34th tuning, the oldest result is removed. Member variable of AllTrials is the same parameter as that of Auto Tune Monitor given in an easy tuning of Sysmac Studio. When you attempt to know the meaning of the parameter, refer to Easy Tuning (Single Drive) of SYSMAC-SE2□□□ Sysmac Studio Drive Functions Operation Manual (I589-E1).
  - c) FinalTrial gives the element number in AllTrials that is adopted tuning result.

## TuningStep

TuningStep of output variable show the following meanings.

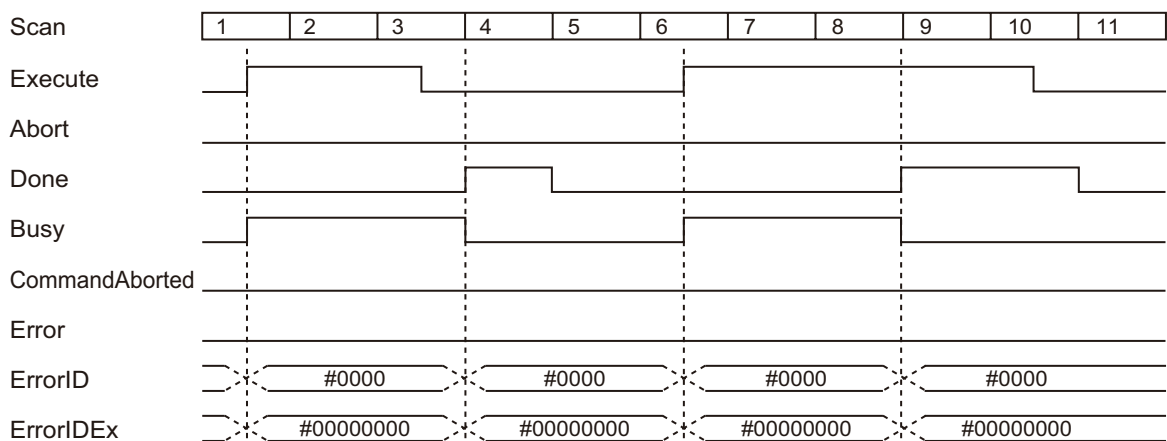
Step	Description
0	Tuning not started yet
1	Waiting the positioning command
10	Estimating Load Characteristics
20	Tuning gain and filter settings
30	Boosting feed-forward gains
40	Completed successfully
45	Canceled execution by user
50	Stopped execution due to error

## Timing Charts

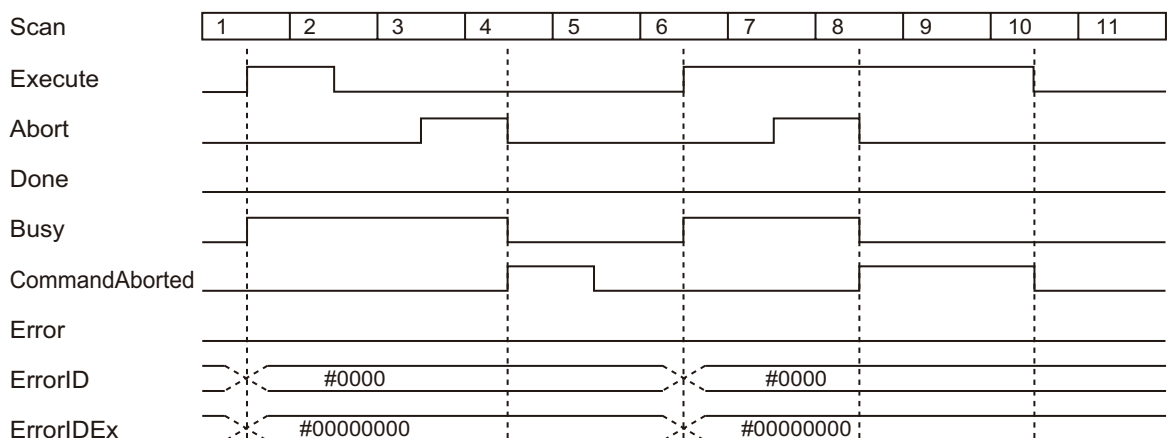
The following figures shows the timing charts for the function block.

- Busy (Executing) changes to TRUE when Execute changes to TRUE.
- Done changes to TRUE when the tuning process is succeeded.
- If the value of Abort is changed from FALSE to TRUE, the tuning process is stopped and CommandAborted 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.

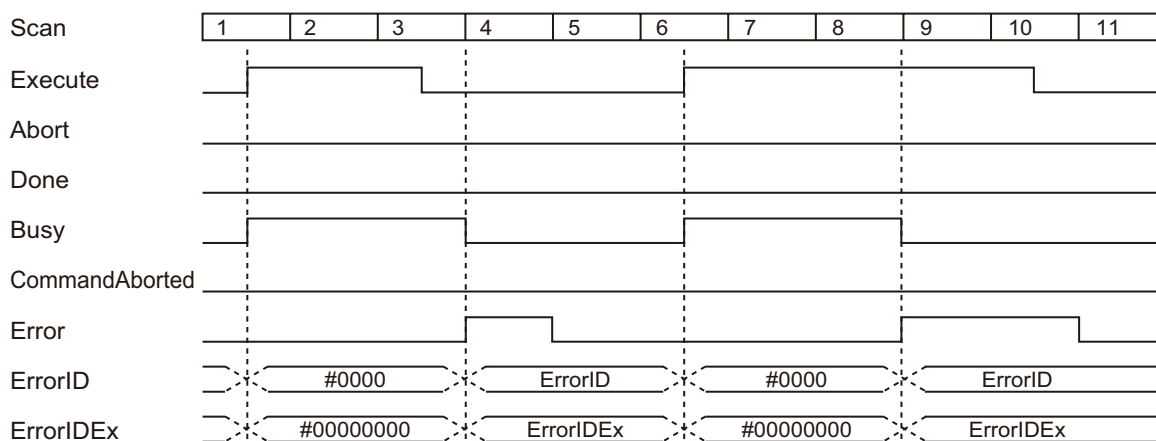
### ● Timing Chart for Normal End



### ● Timing Chart for Canceled Execution



## ● Timing Chart for Error End



## Precautions for Correct Use

- Execution of this function block will be continued until processing is ended even if the value of Execute changes to FALSE or the execution time exceeds the task period. The value of Done changes to TRUE when processing is ended. Use this to confirm normal ending of processing.
- This function block executes four EtherCAT communications instructions at maximum within the part. EtherCAT communications instructions have concurrent executions restriction. Create a user program not to violate this restriction. 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.
- Do not use or open the screens of Sysmac Studio drive functions when you use this function block.
- 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.
- Sample programming shows only the portion of a program that uses the function or function block from the library.
- When using sample programming in 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.
- 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.



## Troubleshooting

Error code	Expansion error code	Status	Description	Correction
16#0000	16#00000000	Normal End	---	---
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 Time out	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 <sup>*1</sup>	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.
16#3D1F	16#00000001	VibrationLevel value Out of Range	The value of VibrationLevel is out of range.	Set the value of VibrationLevel correctly.
16#3D1F	16#00000002	PositionWindow value Out of Range	The value of PositionWindow is out of range.	Set the value of PositionWindow correctly.
16#3D1F	16#00000003	2nd Order Interpolation is not Supported	The specified Servo Drive does not support the 2nd order interpolation.	When setting FALSE: Smooth to LinearMovementInterpolation, use R88D-1SAN□□□-ECT or R88D-1SN□□□-ECT Version 1.2 and later.
16#3D1F	16#00000005	Load Characteristic Estimation Failure	Failed in estimating load characteristics.	Check that the error is not applied to Conditions that interfere with the Load Characteristic Estimation function, referring to instruction manual for the target Servo Drive.
16#3D1F	16#00000006	Drive Error Detected	The specified servo drive detected warning or error.	Remove the cause of the warning and errors.
16#3D1F	16#00000007	Vibration Detected	Vibration was detected.	Increase the StabilizationTime.
16#3D1F	16#00000008	Torque Saturation Detected	Torque saturation was detected.	Reduce motor acceleration / deceleration.

Error code	Expansion error code	Status	Description	Correction
16#3D1F	16#00000009	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 1S-series Servo Drive.
16#3D1F	16#0000000A	Unsupported Device	The device is not supported by this function block.	Check that the specified device is a 1S-series Servo Drive.
16#3D1F	16#0000000B	Criteria Not Found	The criteria cannot be found.	Relieve the TuningCriteria.

\*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

This sample programming executes Easy Tuning for a Servo Drive.

### Conditions

- The target Servo Drive is Node 1.
- The target Servo Drive is assigned to axis 0 in the Motion Control Function Module.

### Processing

- 1** Make sure that the Sysmac Studio drive functions are not used and not displayed its screens.
- 2** Make sure that the target slave is in a state that allows communications.
- 3** Execute EASYTUNING\_1S\_instance to start Easy Tuning.
- 4** Check that Busy of EASYTUNING\_1S\_instance changes to TRUE and execute MC\_POWER\_instance and MC\_MOVE\_RELATIVE\_instance.
- 5** Reverse movement starts in three seconds after Done of MC\_MOVE\_RELATIVE\_instance changes to TRUE.
- 6** Check that Done of EASYTUNING\_1S\_instance changes to TRUE.

### LD

#### ● Internal Variables

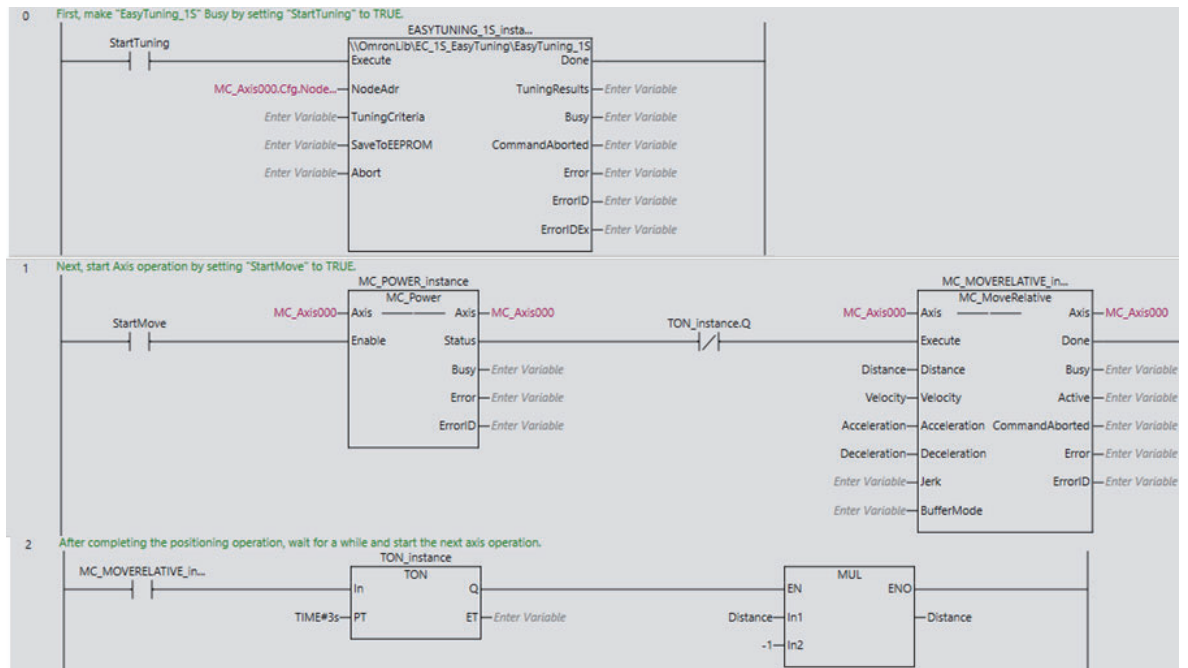
Name	Data type	Default	Comment
StartMove	BOOL		
EASYTUNING_1S_instance	OmronLib\EC_1S_EasyTuning\EasyTuning_1S		
StartTuning	BOOL		
MC_POWER_instance	MC_Power		
MC_MOVERELATIVE_instance	MC_MoveRelative		
Distance	LREAL	30	
Velocity	LREAL	100	
Acceleration	LREAL	1000	
Deceleration	LREAL	1000	
TON_instance	TON		

## ● External Variables

Name	Data type	Constant	Comment
MC_Axis000	_sAXIS_REF	✓	

## ● Global Variables

Name	Data type	Default	AT	Constant	Comment
MC_Axis000	_sAXIS_REF		MC://_MC_AX[0]	✓	



## ST

## ● Internal Variables

Name	Data type	Default	Comment
StartMove	BOOL		
EASYTUNING_1S_instance	OmronLib\EC_1S_EasyTuning\EasyTuning_1S		
StartTuning	BOOL		
MC_POWER_instance	MC_Power		
MC_MOVERELATIVE_instance	MC_MoveRelative		
Distance	LREAL	30	
Velocity	LREAL	100	
Acceleration	LREAL	1000	
Deceleration	LREAL	1000	
TON_instance	TON		

## ● External Variables

Name	Data type	Constant	Comment
MC_Axis000	_sAXIS_REF	✓	

## ● Global Variables

Name	Data type	Default	AT	Constant	Comment
MC_Axis000	_sAXIS_REF		MC://_MC_AX[0]	✓	

```
//First, make "EasyTuning_1S" Busy by setting "StartTuning" to TRUE.
EASYTUNING_1S_instance(
  Execute:=StartTuning,
  NodeAdr:=MC_Axis000.Cfg.NodeAddress
);
//Next, start Axis operation by setting "StartMove" to TRUE.
MC_POWER_instance(
  Axis:=MC_Axis000,
  Enable:=StartMove
);
MC_MOVERELATIVE_instance(
  Axis:=MC_Axis000,
  Execute:=MC_POWER_instance.Status AND (NOT TON_instance.Q),
  Distance:=Distance,
  Velocity:=Velocity,
  Acceleration:=Acceleration,
  Deceleration:=Deceleration
);
//After completing the positioning operation, wait for a while and start the next
axis operation.
TON_instance(
  In:=MC_MOVERELATIVE_instance.Done,
  PT:=TIME#3s
);
IF TON_instance.Q THEN
  Distance := Distance * -1;
END_IF;
```

# 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 93.

\*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 93.

\*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.

(1)Library file name      (2)Library version      (3)Library author      (4)Library comment

Library name	Name Space	Version	Author	Company	Date Creat	Date Modi	Comment
OmronLib_MC_Toolbox_V1_1		1.1.0	OMRON Corporation	(c)OMRON Corporation 2015. All Rights Reserved.			This is MC Toolbox library. これはモーション制御ツールボックスライ
POU							
Programs							
Functions							
DeadBand (OmronLib_MC_Toolbox)	OmronLib\MC_Toolbo	1.1.0	OMRON Corporation		03/16/2015	08/10/201	No.00006 The DeadBand function block cont 処理結果にオフセットが発生させないデ
FirstOrderlag (OmronLib_MC_Toolbox)	OmronLib\MC_Toolbo	1.1.0	OMRON Corporation		04/01/2015	08/10/201	No.00004 The FirstOrderLag function block p 設定されたパラメータテーブルに従って、
LeadLag (OmronLib_MC_Toolbox)	OmronLib\MC_Toolbo	1.1.0	OMRON Corporation		04/01/2015	08/10/201	No.00005 The LeadLag function block perfor 設定されたパラメータテーブルに従って、
PIDFeedFwd (OmronLib_MC_Toolbox)	OmronLib\MC_Toolbo	1.1.0	OMRON Corporation		04/01/2015	08/10/201	No.00003 The PIDFeedFwd function block pe 設定されたパラメータテーブルに従って、

(5)FB/FUN name      (6)Name space      (7)FB/FUN version      (8)FB/FUN author      (10)FB/FUN comment

Namespace - Using

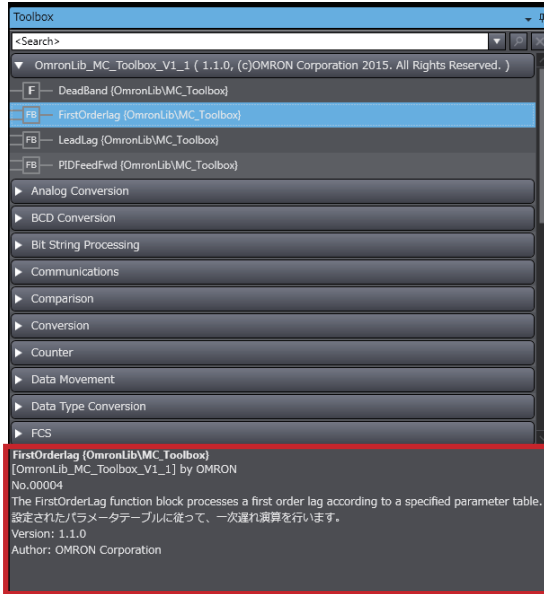
In/Out	Name	In/Out	Data Type	Edge	Initial Value	Retain	Constant	Comment
Externals	Enable	Input	BOOL	No Edge	False	<input type="checkbox"/>	<input type="checkbox"/>	
	InCalc	Input	LREAL	No Edge	0.0	<input type="checkbox"/>	<input type="checkbox"/>	
	Kp	Input	LREAL	No Edge	1.0	<input type="checkbox"/>	<input type="checkbox"/>	
	TimeConst	Input	LREAL	No Edge	1.0	<input type="checkbox"/>	<input type="checkbox"/>	
	SampTime	Input	LREAL	No Edge	1.0	<input type="checkbox"/>	<input type="checkbox"/>	
	Enabled	Output	BOOL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>	

OK

(b) Toolbox Pane

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

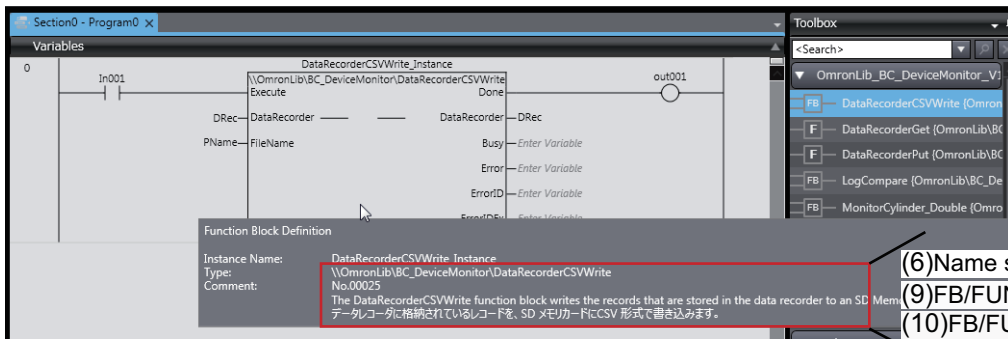
The text “by OMRON” which is shown on the right of the library name (1) indicates that this library was provided by OMRON.



- (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.

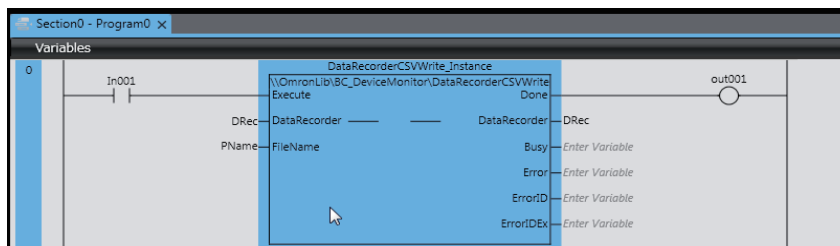
The following are the examples of items that you may need to customize.

- 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.

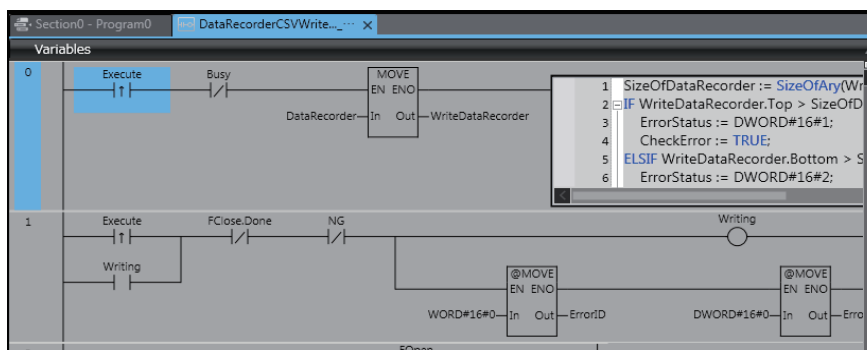
Use the following procedure to refer to the source codes of function blocks and functions.

- 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.



# Objects to Back up and Restore

The following table gives the objects to back up and restore. Refer to the *AC Servomotors/Servo Drives 1S-series with Built-in EtherCAT Communications User's Manual* (Cat. No. I586) or the *AC Servomotors/Servo Drives 1S-series with Built-in EtherCAT Communications and Safety Functionality User's Manual* (Cat. No. I621) for details on objects.

Index(Hex)	Subindex(Hex)	Name
2200	00	Communications Error Setting
2201	00	Sync Not Received Timeout Setting
3000	-	Basic Functions
	01	Motor Rotation Direction Selection
	03	Control Method Selection
3001	-	Machine
	01	Inertia Ratio
	02	Backlash Compensation Selection
	03	Backlash Compensation Amount
	04	Backlash Compensation Time Constant
3011	-	Position Command Filter
	01	FIR Filter Enable
	02	FIR Filter Moving Average Time
	03	IIR Filter Enable
	04	IIR Filter Cutoff Frequency
3012	-	Damping Control
	01	Damping Filter 1 Selection
	02	Damping Filter 2 Selection
3013	-	Damping Filter 1
	01	1st Frequency
	02	1st Damping Time Coefficient
	03	2nd Frequency
	04	2nd Damping Time Coefficient
	05	3rd Frequency
	06	3rd Damping Time Coefficient
	07	4th Frequency
	08	4th Damping Time Coefficient
3014	-	Damping Filter 2
	01	1st Frequency
	02	1st Damping Time Coefficient
	03	2nd Frequency
	04	2nd Damping Time Coefficient
	05	3rd Frequency
	06	3rd Damping Time Coefficient
	07	4th Frequency
	08	4th Damping Time Coefficient
3021	-	Velocity Command Filter
	01	Acceleration Time
	02	Deceleration Time
	03	IIR Filter Enable
	04	Filter Cutoff Frequency
3031	-	Velocity Limit in Torque Control
	01	Velocity Limit Value

Index(Hex)	Subindex(Hex)	Name
3041	-	Command Dividing Function
	01	Operation Selection in csv
	10	Interpolation method selection in csp <sup>*2</sup>
3112	-	ODF Velocity Feed-forward
	01	Gain
	02	LPF Enable
	03	LPF Cutoff Frequency
3113	-	ODF Torque Feed-forward
	01	Gain
	02	LPF Enable
	03	LPF Cutoff Frequency
3120	-	TDF Position Control
	01	Command Following Gain
	10	Command Following Gain selection <sup>*1</sup>
	11	Command Following Gain 2 <sup>*1</sup>
3121	-	TDF Velocity Control
	01	Command Following Gain
	10	Command Following Gain selection <sup>*1</sup>
	11	Command Following Gain 2 <sup>*1</sup>
3212	-	Gain Switching in Position Control
	01	Mode Selection
	02	Delay Time
	03	Speed
	04	Time
3213	-	1st Position Control Gain
	01	Proportional Gain
3214	-	2nd Position Control Gain
	01	Proportional Gain
3222	-	Gain Switching in Velocity Control
	01	Mode Selection
3223	-	1st Velocity Control Gain
	01	Proportional Gain
	02	Integral Gain
3224	-	2nd Velocity Control Gain
	01	Proportional Gain
	02	Integral Gain
3232	-	Filter Switching in Torque Control
	01	Mode Selection
3233	-	1st Torque Command Filter
	01	Enable
	02	Cutoff Frequency
3234	-	2nd Torque Command Filter
	01	Enable
	02	Cutoff Frequency
3310	-	Torque Compensation
	01	Viscous Friction Coefficient
	02	Unbalanced Load Compensation
	03	Positive Dynamic Friction Compensation
	04	Negative Dynamic Friction Compensation

\*1. R88D-1SN□□□-ECT: Version 1.1 and later

\*2. R88D-1SN□□□-ECT: Version 1.2 and later

Index(Hex)	Subindex(Hex)	Name
3320	-	Adaptive Notch Filter
	01	Adaptive Notch Selection
	03	Resonance Detection Threshold
3321	-	1st Notch Filter
	01	Enable
	02	Frequency
	03	Q-value
	04	Depth
3322	-	2nd Notch Filter
	01	Enable
	02	Frequency
	03	Q-value
	04	Depth
3323	-	3rd Notch Filter
	01	Enable
	02	Frequency
	03	Q-value
	04	Depth
3324	-	4th Notch Filter
	01	Enable
	02	Frequency
	03	Q-value
	04	Depth
3330	-	Torque Limit
	01	Switching Selection
	02	Max Torque
	03	Positive Torque Limit Value
	04	Negative Torque Limit Value
	05	Positive Torque Limit Value 2
	06	Negative Torque Limit Value 2
3A00	-	Homing
	01	Zero Position Range
3B10	-	Drive Prohibition
	01	Enable
	02	Stop Selection
3B11	-	Software Position Limit
	01	Enable Selection
	02	Stop Selection
3B21	-	Deceleration Stop
	01	Torque
3B40	-	Zone Notification 1
	01	Lower Limit
	02	Upper Limit
3B41	-	Zone Notification 2
	01	Lower Limit
	02	Upper Limit
3B52	-	Positioning Completion Notification 2
	01	Position Window
	02	Notification Condition

Index(Hex)	Subindex(Hex)	Name
3B60	-	Speed Detection Function
	01	Velocity Attainment Detection Level
	02	Zero Speed Detection Level
	03	Velocity Conformity Detection Range
	04	Excessive Speed Detection Level
	05	Excessive Velocity Deviation Detection Level
3B70	-	Vibration Detection
	01	Detection Level
3B71	-	Runaway detection* <sup>1</sup>
	01	Enable* <sup>1</sup>
3B80	-	Load Characteristic Estimation
	01	Inertia Ratio Update Selection
	02	Viscous Friction Compensation Update Selection
	03	Unbalanced Load Compensation Update Selection
	04	Dynamic Friction Compensation Update Selection
	05	Viscous Friction Tuning Coefficient
	06	Estimation Sensitivity Selection
4020	-	Warning Customization
	01	Warning Mask 1 Selection
	03	Warning Mask 3 Selection
	04	Warning Hold Selection
	05	Warning Level Change 1 Selection
	07	Warning Level Change 3 Selection
4021	-	Warning Output 1 Setting
	01	Selection 1
	03	Selection 3
4022	-	Warning Output 2 Setting
	01	Selection 1
	03	Selection 3
4030	-	Information Customization
	01	Information Level Change Selection
4110	-	Monitor Data via PDO
	01	Target Object 1
	02	Target Object 2
	03	Target Object 3
	04	Target Object 4
4140	-	Lifetime Information
	01	Safety Relay Lifetime Warning Detection Threshold* <sup>3</sup>
4150	-	Overload
	01	Warning Notification Level
4210	-	Display
	01	LED Display Selection
4310	-	Regeneration
	01	External Regeneration Resistor Selection
	02	External Regeneration Resistance
	03	External Regeneration Allowable Power
	04	External Regeneration Overload Ratio
4320	-	Main Circuit Power Supply
	01	Momentary Hold Time
	02	Phase Loss Detection Enable
	03	Capacitor Discharge Enable

\*1. R88D-1SN□□□-ECT: Version 1.1 and later

\*3. Only R88D-1SAN□□□-ECT

Index(Hex)	Subindex(Hex)	Name
4510	-	Encoder
	01	Operation Selection when Using Absolute Encoder
	02	Absolute Encoder Counter Overflow Warning Level
4610	-	Brake Interlock Output
	01	Enable
	02	Timeout at Servo OFF
	03	Threshold Speed at Servo OFF
	04	Hardware Delay Time
4611	-	External Dynamic Brake Interlock* <sup>4</sup>
	01	Enable Control Output* <sup>4</sup>
	02	Enable Monitor Input* <sup>4</sup>
	04	Hardware Delay Time* <sup>4</sup>
4620	-	Encoder Dividing Pulse Output
	01	Enable
	02	Dividing Numerator
	03	Dividing Denominator
	04	Output Reverse Selection
4630	-	Positive Drive Prohibition Input
	01	Port Selection
	02	Logic Selection
4631	-	Negative Drive Prohibition Input
	01	Port Selection
	02	Logic Selection
4632	-	External Latch Input 1
	01	Port Selection
	02	Logic Selection
4633	-	External Latch Input 2
	01	Port Selection
	02	Logic Selection
4634	-	Home Proximity Input
	01	Port Selection
	02	Logic Selection
4635	-	Positive Torque Limit Input
	01	Port Selection
	02	Logic Selection
4636	-	Negative Torque Limit Input
	01	Port Selection
	02	Logic Selection
4637	-	Error Stop Input
	01	Port Selection
	02	Logic Selection
4638	-	Monitor Input 1
	01	Port Selection
	02	Logic Selection
4639	-	Monitor Input 2
	01	Port Selection
	02	Logic Selection
463A	-	Monitor Input 3
	01	Port Selection
	02	Logic Selection

\*4. Only R88D-1SN55□-ECT/ -1SN75□-ECT/ -1SN150□-ECT

Index(Hex)	Subindex(Hex)	Name
463B	-	Monitor Input 4
	01	Port Selection
	02	Logic Selection
463C	-	Monitor Input 5
	01	Port Selection
	02	Logic Selection
463D	-	Monitor Input 6
	01	Port Selection
	02	Logic Selection
463E	-	Monitor Input 7
	01	Port Selection
	02	Logic Selection
463F	-	Monitor Input 8
	01	Port Selection
	02	Logic Selection
4641	-	External Dynamic Brake Relay Monitor Input <sup>*4</sup>
	01	Port Selection <sup>*4</sup>
	02	Logic Selection <sup>*4</sup>
4650	-	Error Output
	01	Port Selection
	02	Logic Selection
4651	-	Servo Ready Output
	01	Port Selection
	02	Logic Selection
4652	-	Positioning Completion Output 1
	01	Port Selection
	02	Logic Selection
4653	-	Positioning Completion Output 2
	01	Port Selection
	02	Logic Selection
4654	-	Velocity Attainment Detection Output
	01	Port Selection
	02	Logic Selection
4655	-	Torque Limit Output
	01	Port Selection
	02	Logic Selection
4656	-	Zero Speed Detection Output
	01	Port Selection
	02	Logic Selection
4657	-	Velocity Conformity Output
	01	Port Selection
	02	Logic Selection
4658	-	Warning Output 1
	01	Port Selection
	02	Logic Selection
4659	-	Warning Output 2
	01	Port Selection
	02	Logic Selection
465A	-	Velocity Limiting Output
	01	Port Selection
	02	Logic Selection

\*4. Only R88D-1SN55□-ECT/ -1SN75□-ECT/ -1SN150□-ECT

Index(Hex)	Subindex(Hex)	Name
465B	-	Error Clear Attribute Output
	01	Port Selection
	02	Logic Selection
465C	-	Remote Output 1
	01	Port Selection
	02	Logic Selection
465D	-	Remote Output 2
	01	Port Selection
	02	Logic Selection
465E	-	Remote Output 3
	01	Port Selection
	02	Logic Selection
465F	-	Zone Notification Output 1
	01	Port Selection
	02	Logic Selection
4660	-	Zone Notification Output 2
	01	Port Selection
	02	Logic Selection
4661	-	Position Command Status Output
	01	Port Selection
	02	Logic Selection
4662	-	Distribution Completed Output
	01	Port Selection
	02	Logic Selection
4663	-	External brake interlock output <sup>*2</sup>
	01	Port Selection
	02	Logic Selection
4664	-	External Dynamic Brake Interlock Output <sup>*4</sup>
	01	Port Selection <sup>*4</sup>
	02	Logic Selection <sup>*4</sup>
605B	00	Shutdown option code
605C	00	Disable operation option code
605D	00	Halt option code
605E	00	Fault reaction option code
6065	00	Following error window
6067	00	Position window
607C	00	Home offset
607D	-	Software position limit
	01	Min position limit
	02	Max position limit
6091	-	Gear ratio
	01	Motor revolutions
	02	Shaft revolutions
6098	00	Homing method
6099	-	Homing speeds
	01	Speed during search for switch
	02	Speed during search for zero
609A	00	Homing acceleration
60C2	-	Interpolation time period
	01	Interpolation time period value
	02	Interpolation time index

\*2. R88D-1SN□□□-ECT: Version 1.2 and later

\*4. Only R88D-1SN55□-ECT/ -1SN75□-ECT/ -1SN150□-ECT



Index(Hex)	Subindex(Hex)	Name
60D0	-	Touch probe source
	01	Touch probe 1 source
	02	Touch probe 2 source
60DA	00	Function Settings
60FE	-	Digital outputs
	02	Bit mask





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

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

**Regional Headquarters**

**OMRON EUROPE B.V.**

Wegalaan 67-69, 2132 JD Hoofddorp  
The Netherlands  
Tel: (31)2356-81-300/Fax: (31)2356-81-388

**OMRON ELECTRONICS LLC**

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

**OMRON ASIA PACIFIC PTE. LTD.**

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

**OMRON (CHINA) CO., LTD.**

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

**Authorized Distributor:**

© OMRON Corporation 2016-2019 All Rights Reserved.  
In the interest of product improvement,  
specifications are subject to change without notice.

**Cat. No. W571-E1-06**

0719