

## Machine Automation Controller NJ/NX-series

---

### Sysmac Library User's Manual for Adept Robot Control Library

NX701-□□□□

NJ501-□□□□

NJ301-□□□□


NJ101-□□□□

## 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 USA 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.

This manual provides information required to use the function blocks in the Adept Robot Control Library. ("Function block" is sometimes abbreviated as "FB".) Please read this manual and make sure you understand the functionality and performance of the NJ/NX-series CPU Unit before you attempt to use it in a control system.

This manual contains the specifications of the Function Block. It does not include restrictions on use of the Controller, Units, or components, or restrictions due to combinations. Make sure to read the user's manual for each product before use.

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

## Features of the Library

The Adept Robot Control Library is used to directly control Adept Robots from NJ/NX-series Controller.

You can use this library to control any types of robots like parallel, SCARA and articulated from PLC with common instructions and a common programming method.

## 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.
- Personnel with knowledge of control logic.

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

## Applicable Products

This manual covers the following products.

NX-series CPU Units

- NX701-□□□□

NJ-series CPU Units

- NJ501-□□□□
- NJ301-□□□□
- NJ101-□□□□

Part of the specifications and restrictions for the CPU Units are given in other manuals. Refer to *Related Manuals* on page 10.

# 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 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>3</b>
<b>Terms and Conditions Agreement .....</b>	<b>5</b>
Warranty, Limitations of Liability .....	5
Application Considerations .....	6
Disclaimers .....	6
<b>Safety Precautions.....</b>	<b>7</b>
Warning .....	8
Cautions .....	8
<b>Precautions for Correct Use .....</b>	<b>9</b>
<b>Related Manuals.....</b>	<b>10</b>
<b>Terminology.....</b>	<b>12</b>
<b>Revision History.....</b>	<b>13</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 .....	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>
Common data types .....	32
ARB_SetToolTransform.....	36
ARB_ResetToolTransform.....	42
ARB_DefineLocation .....	48
ARB_DefinePallet.....	55
ARB_ResetRobotError .....	64
ARB_RobotControl .....	70
ARB_TeachPosition.....	82
ARB_InputOutputSignals .....	90
ARB_TeachPendantControl .....	99
ARB_ReadLatch.....	107
ARB_MoveCommand.....	114
ARB_PickAndPlaceCommand .....	122
ARB_Jog .....	129
ARB_AlignToolCommand.....	137
Execution Conditions Table .....	143
Multi-execution of Function Blocks.....	145
Re-execution of Function Blocks.....	146
Errors from Controller.....	147
<b>Appendix .....</b>	<b>149</b>
Referring to Library Information.....	150
Referring to Function Block and Function Source Codes .....	153



# 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 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.

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.

## Warning

### Warning

If you wish to control robots using the libraries in this guide, make sure to install the robot and perform the test run according to the *Robot Safety Guide* (Cat. No. I590).



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



## Cautions

### Caution

Read all related manuals carefully before you use this library. Especially read the *Robot Safety Guide* (Cat. No. I590) prior to above.



Implement safety measures given in the manuals for the robots used before you use this library.



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



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



You must confirm that the user program and parameter values are appropriate to the specifications and operation methods of the devices.



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



Understand the contents of sample programming before you use the sample programming and create the user program.



# 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.
- You cannot change the source code of the functions or function blocks that are provided in the Sysmac Library.

## 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.

## Operation

---

- Specify the input parameter values within the valid range.
- In the function or function block with an Enabled output variable, if the value of Enabled is FALSE, do not use the processing result of the function or function block as a command value to the control target.

# 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 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	<p>An introduction to the entire NX-series system is provided along with the following information on the CPU Unit.</p> <ul style="list-style-type: none"> <li>• Features and system configuration</li> <li>• Introduction</li> <li>• Part names and functions</li> <li>• General specifications</li> <li>• Installation and wiring</li> <li>• Maintenance and inspection</li> </ul> <p>Use this manual together with the <i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501).</p>
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.	<p>An introduction to the entire NJ-series system is provided along with the following information on the CPU Unit.</p> <ul style="list-style-type: none"> <li>• Features and system configuration</li> <li>• Introduction</li> <li>• Part names and functions</li> <li>• General specifications</li> <li>• Installation and wiring</li> <li>• Maintenance and inspection</li> </ul> <p>Use this manual together with the <i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501).</p>
NJ/NX-series CPU Unit Software User's Manual	W501	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning how to program and set up an NJ/NX-series CPU Unit. Mainly software information is provided.	<p>The following information is provided on a Controller built with an NJ/NX-series CPU Unit.</p> <ul style="list-style-type: none"> <li>• CPU Unit operation</li> <li>• CPU Unit features</li> <li>• Initial settings</li> <li>• Programming based on IEC 61131-3 language specifications</li> </ul> <p>Use this manual together with the <i>NX-series CPU Unit Hardware User's Manual</i> (Cat. No. W535) or <i>NJ-series CPU Unit Hardware User's Manual</i> (Cat. No. W500).</p>
NJ/NX-series Instructions Reference Manual	W502	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning detailed specifications on the basic instructions of an NJ/NX-series CPU Unit.	<p>The instructions in the instruction set (IEC 61131-3 specifications) are described. When programming, use this manual together with the <i>NX-series CPU Unit Hardware User's Manual</i> (Cat. No. W535) or <i>NJ-series CPU Unit Hardware User's Manual</i> (Cat. No. W500) and with the <i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501).</p>
NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual	W506	NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Using the built-in EtherNet/IP port on an NJ-series CPU Unit.	<p>Information on the built-in EtherNet/IP port is provided. Information is provided on the basic setup, tag data links, and other features. Use this manual together with the <i>NJ-series CPU Unit Hardware User's Manual</i> (Cat. No. W500) and <i>NJ-series CPU Unit Software User's Manual</i> (Cat. No. W501).</p>
Robot Safety Guide	I590	-	Using any of the Adept robots.	Safety information on the robots are provided.
NJ-series EtherNet/IP Connection Guide OMRON Corporation Adept Robot of ePLC	P649	-	Using any of the Adept robots.	Information on wiring and operation settings for the robot controllers, and settings for an NJ-series CPU Unit.

Manual name	Cat. No.	Model numbers	Application	Description
NJ-series Startup Guide OMRON Corporation Adept Robot Control Library	P103	-	Using Adept robot control libraries.	Information on how to start up function blocks for controlling an OMRON robot controller from an NJ-series CPU Unit, when the robot controller and the NJ-series CPU Unit are used in combination.
Cobra 350 Robot User's Guide	I591	Cobra350	Using any of the robots shown on the left.	Describes the functionality of the robots.
eCobra 600, 800, and 800 Inverted Robots User's Guide	I593	eCobra 600/800/800 Inverted		
Hornet 565 Robot User's Guide	I596	Hornet 565		
Quattro 650H/650HS/800H/ 800HS User's Guide	I597	Quattro 650H/650HS/80 0H/800HS		
Viper 650/850 Robot with eMB-60R User's Guide	I599	Viper 650/850 eMB-60R		
T20 User's Guide	I601	T20	Using an Adept T20 pen- dant.	Describes the functionality of the pendant.
SmartController EX User's Guide	I602	SmartController EX	Using SmartController EX.	Information on installation and operation for the SmartController EX and sDIO products (sold sepa- rately).

# Terminology

---

Term	Cat. No.
Joint coordinates	Coordinate system whose elements are positions of each joint of the robot. To see which joint corresponds to which element actually, please refer the user's manual of the robot which you will use.
Location	A functionality which defines a position in Cartesian coordinates or in Joint coordinates into the robot, and uses it.
Location number	A number which specifies a location to be defined in the robot or to be used.
Yaw	Yaw is a rotation about the local reference frame Z axis.
Pitch	Pitch is defined as a rotation about the Y axis, after yaw has been applied.
Roll	Roll is defined as a rotation about the Z axis after yaw and pitch have been applied.

# Revision History

---

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

<b>Cat. No.</b>	<b>W575-E1-01</b>
-----------------	-------------------

↑  
Revision code

Revision code	Date	Revised content
01	April 2016	Original production





# 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

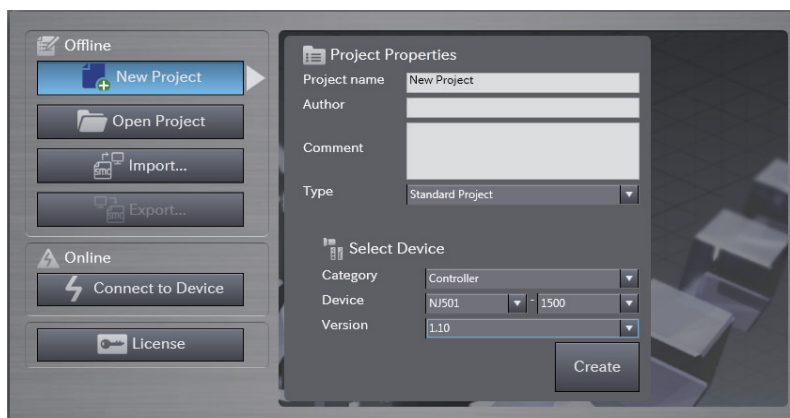


## Version Information

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.

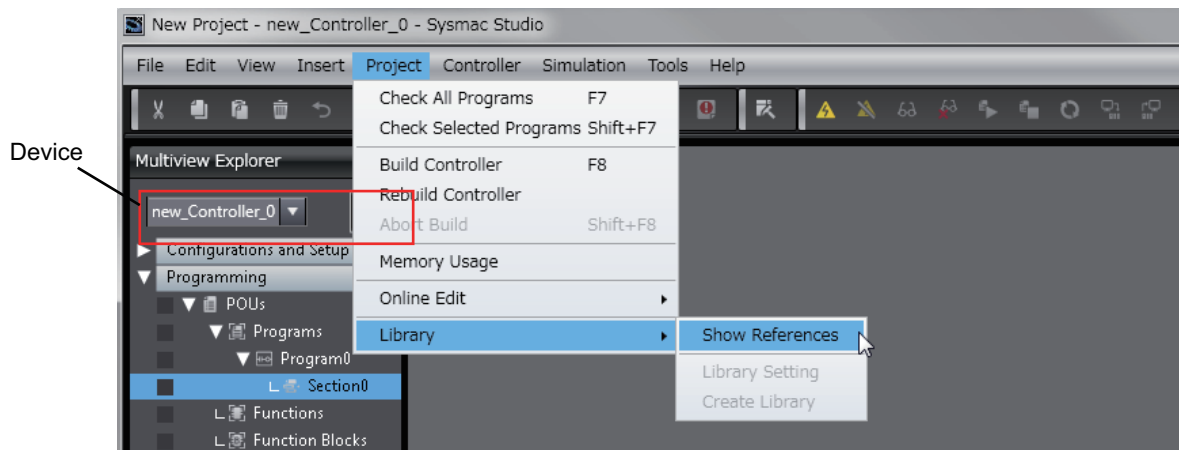


## Precautions for Correct Use


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.

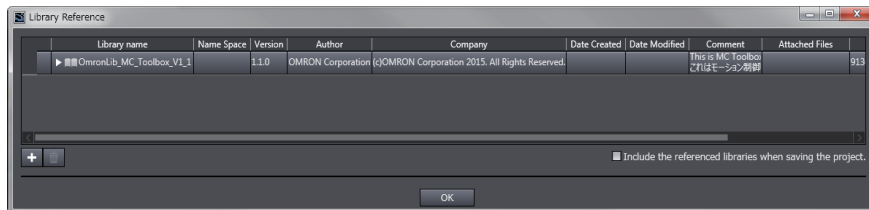
## 2 Select **Project – Library – Show References**.



### 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. If you do not select an NJ/NX-series CPU Unit as the device, Library References does not appear in the above menu. When the device selected currently is an NJ/NX-series CPU Unit, the device icon  is displayed in the Multiview Explorer.

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



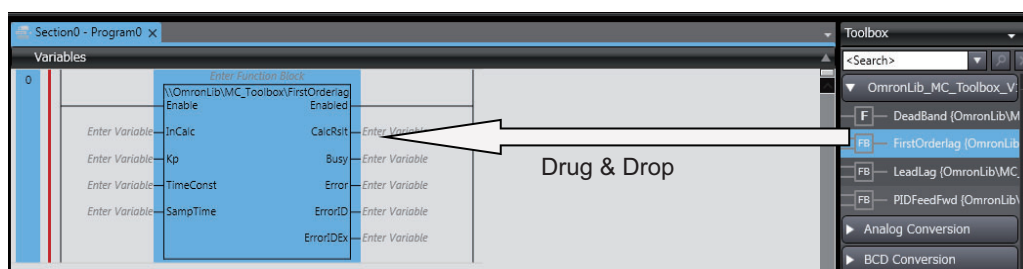
The Sysmac Library file is read into the project.

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

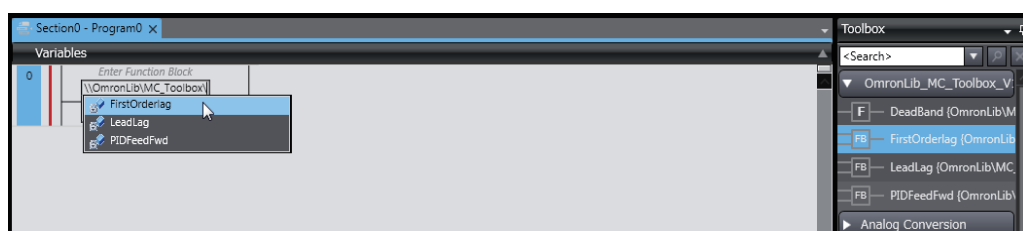
For the procedure for adding and setting libraries in the above screen, refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504).

## 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).



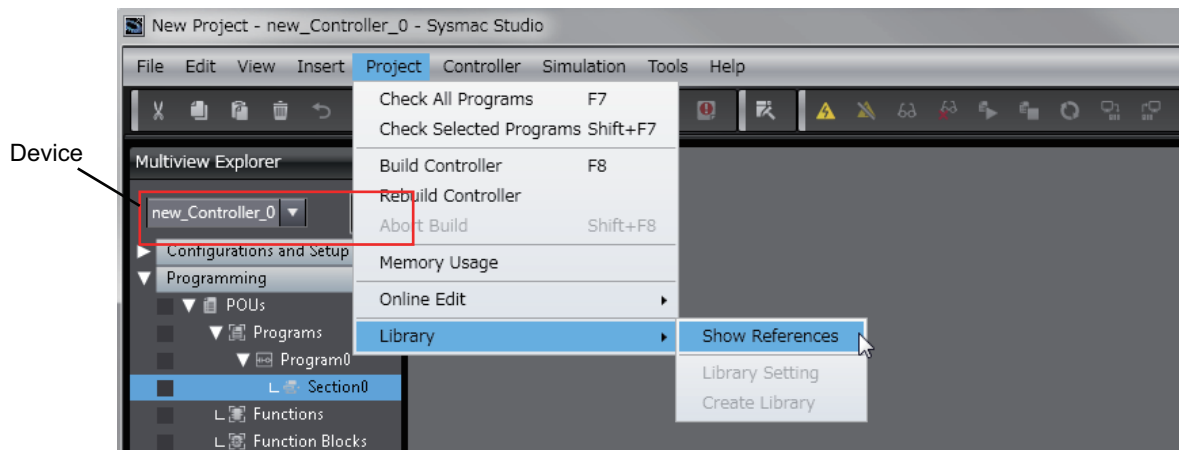
### Precautions for Correct Use

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

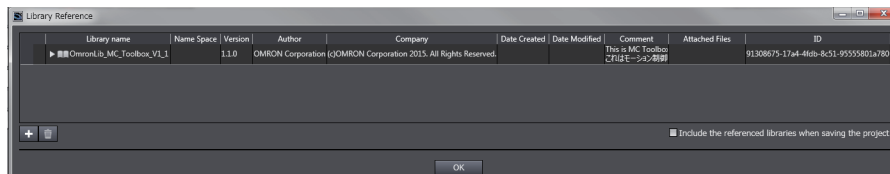
- 1 Start the Sysmac Studio and open a project in which any old-version Sysmac Library is included.
- 2 Select **Project – Library – Show References**.



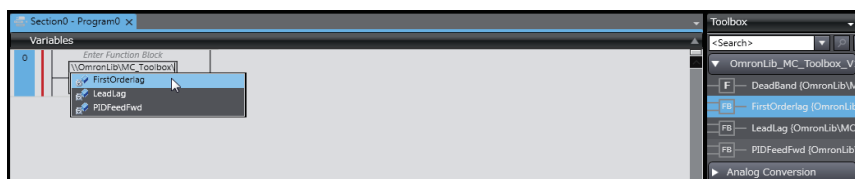
### Precautions for Correct Use

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. Otherwise, Library References does not appear in the above menu. When the device selected currently is an NJ/NX-series CPU Unit, 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.



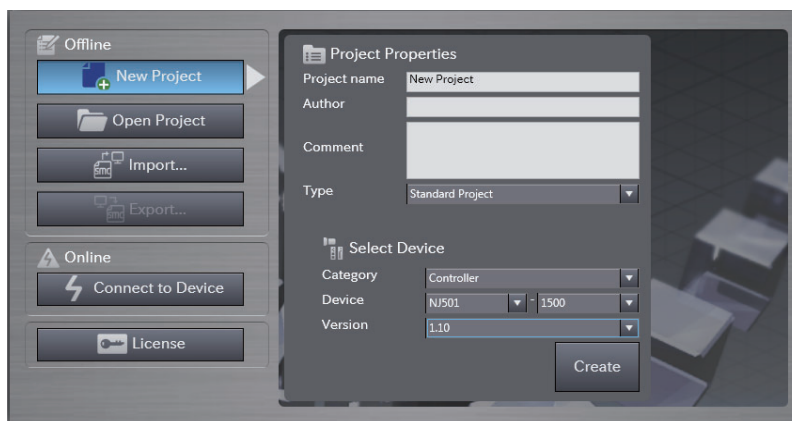
# Procedure to Use Sysmac Libraries Uploaded from a CPU Unit

You can use Sysmac Libraries uploaded from a CPU Unit to your computer if they are not installed. The procedure to use uploaded Sysmac Libraries from a CPU Unit 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.

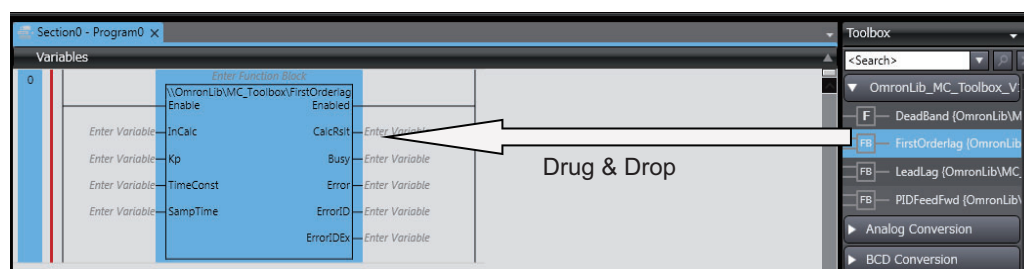


- 2 Connect the computer to the CPU Unit and place it online.
- 3 Upload POU in which any Sysmac Library is used to the computer.

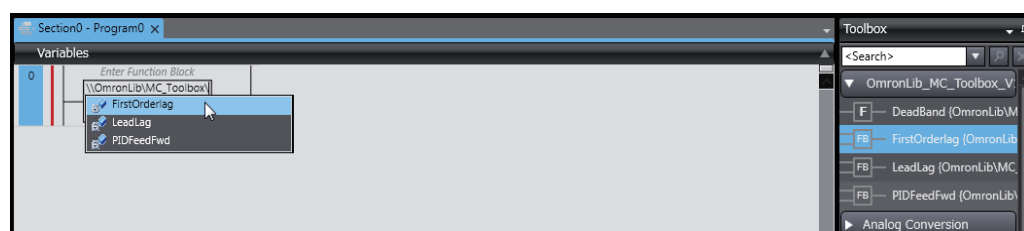
Now, when you select the Ladder Editor or ST Editor, the function blocks and functions included in the Sysmac Library used in the uploaded POUs appear in the Toolbox.

#### 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 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.

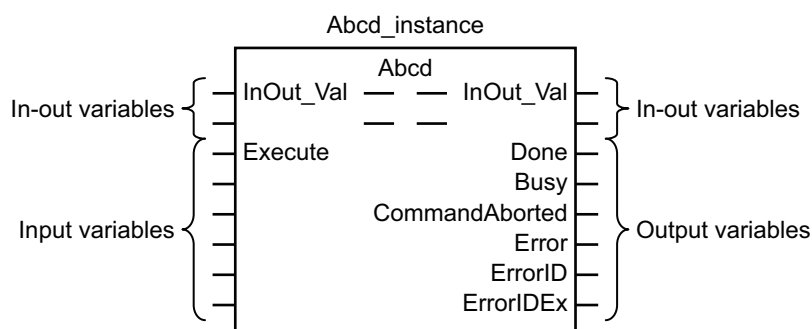
## Definition of Input Variables and Output Variables

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.
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.
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.

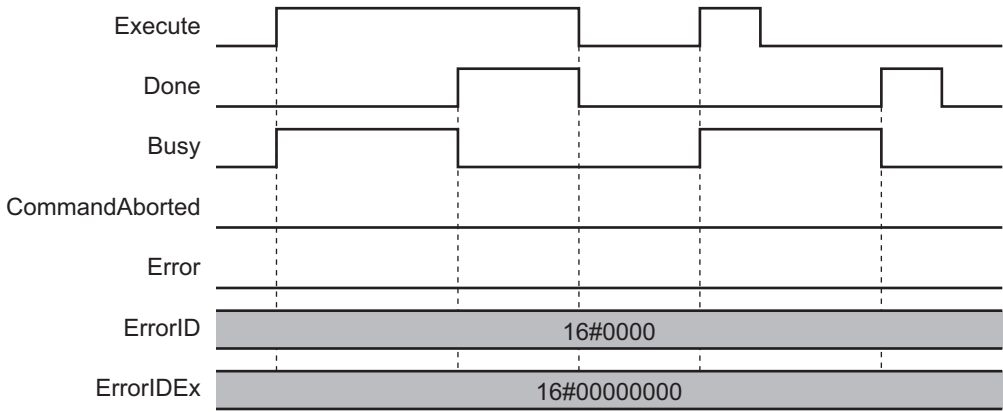


- If *Execute* is TRUE and *Done*, *CommandAborted*, or *Error* changes to TRUE, *Done*, *CommandAborted*, and *Error* changes to FALSE when *Execute* is changed to FALSE.
- If *Execute* is FALSE and *Done*, *CommandAborted*, or *Error* changes to TRUE, *Done*, *CommandAborted*, and *Error* changes to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

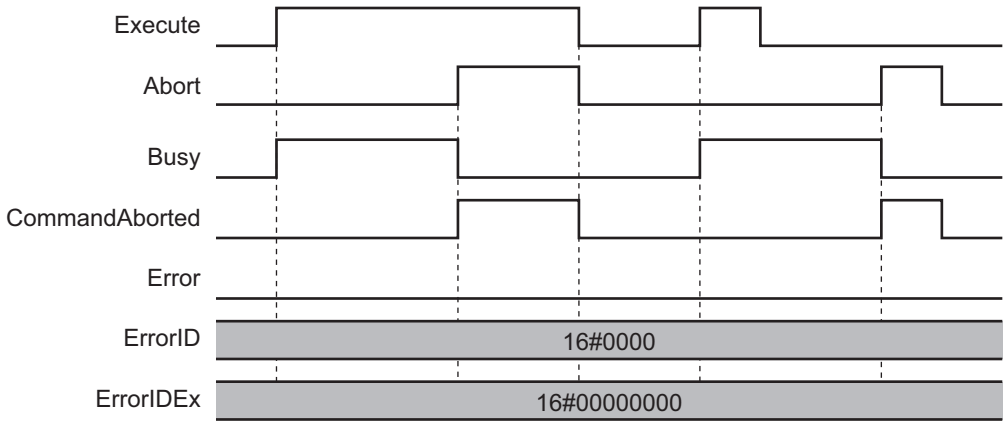
## Timing Charts

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

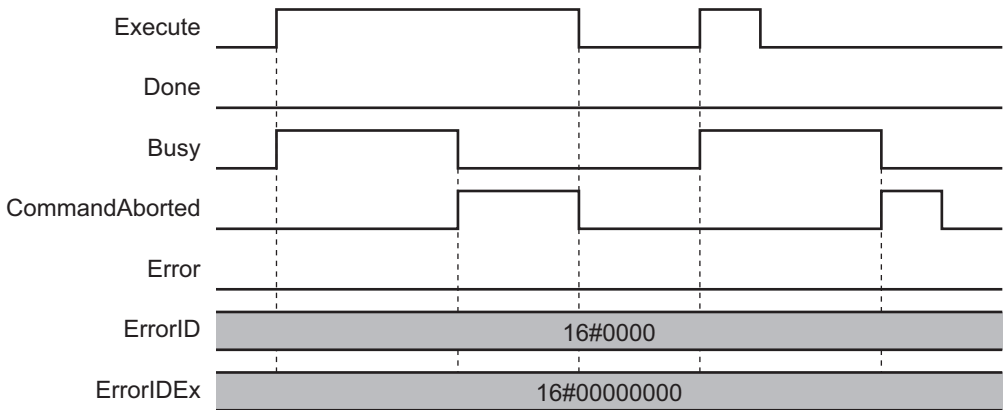
● **Normal End**



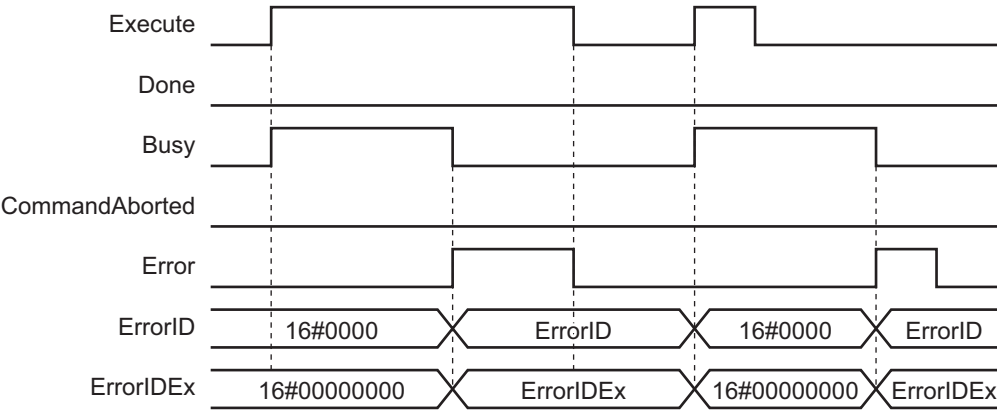
● **Canceled Execution**



● **Aborted Execution**

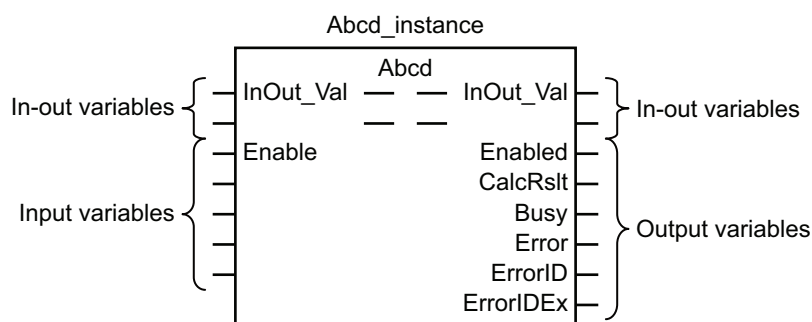


● Errors



## 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.

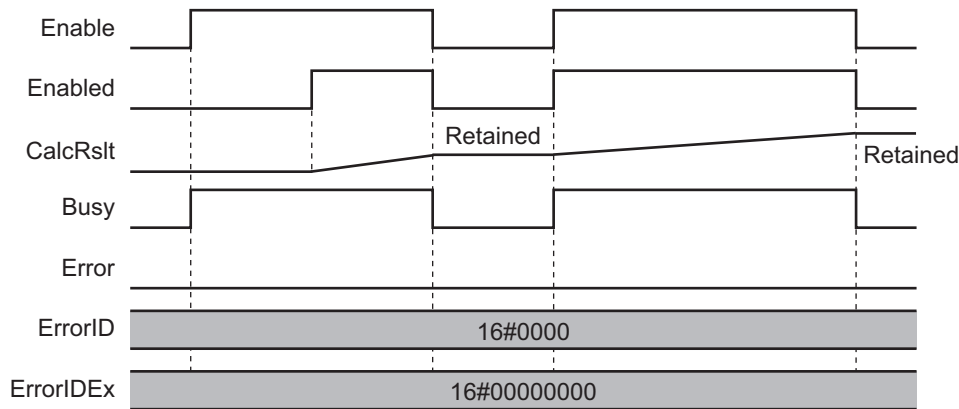


- If an error occurs, the relevant error code and expansion error code are set in *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code). The error codes are retained even after *Error* changes to FALSE, but *ErrorID* is set to 16#0000 and *ErrorIDEx* is set to 16#0000 0000 when *Enable* changes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

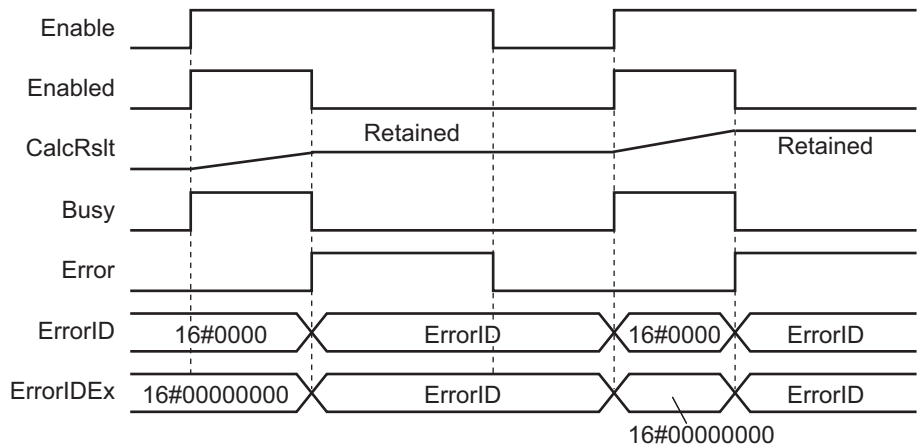
## Timing Charts

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 *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501).

## 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 *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507).



# Individual Specifications of Function Blocks

Instruction	Name	Page
ARB_SetToolTransform	Set Tool Transform	P. 36
ARB_ResetToolTransform	Reset Tool Transform	P. 42
ARB_DefineLocation	Define Location	P. 48
ARB_DefinePallet	Define Pallet	P. 55
ARB_ResetRobotError	Reset Error	P. 64
ARB_RobotControl	Robot Control	P. 70
ARB_TeachPosition	Teach Position	P. 82
ARB_InputOutputSignals	Input Output Signals	P. 90
ARB_TeachPendantControl	Teach Pendant Control	P. 99
ARB_ReadLatch	Read Latch	P. 107
ARB_MoveCommand	MoveCommand	P. 114
ARB_PickAndPlaceCommand	PickAndPlaceCommand	P. 122
ARB_Jog	Jog	P. 129
ARB_AlignToolCommand	AlignToolCommand	P. 137

# Common data types

This section describes the instructions (Function blocks) and data types that are in the library.

## sARB\_ROBOT\_DATA\_REF

Member Variable	Meaning	Data type	Valid range	Function
ConnectionStatus	Connection Status	BOOL	TRUE or FALSE	Specify the network connection status for the corresponding EIP slave number.
CommandData	Command Data	ARRAY [0..213] OF BYTE	---	Specify the memory used to share command data with the Robot Controller.
StatusData	Status Data	ARRAY [0..283] OF BYTE	---	Specify the memory used to share status data with the Robot Controller.

**sARB\_MOVE\_POSITION\_REF**

Member Variable	Meaning	Data type	Valid range	Default	Function
JointCoord	Joint coordinates	BOOL	TRUE or FALSE	FALSE	Specifies how the robot controller will process the positions.  If TRUE position is in Joint coordinates, otherwise is in Cartesian coordinates.
LocationNumber	Location number	INT	0 and [2...999]	0	Identifies a position stored in the robot controller.  When 0, it means that Location Number is not used and Position or PalletNumber have to be used instead.
PalletNumber	Pallet number	INT	0 to 100	0	Identifies the pallet definition stored in the robot controller to be used.  When 0, it means that Pallet-Number is not used and Position or Location Number has to be used instead.
PalletIndex	Pallet index	INT	Positive number, Negative number, or 0	0	Identifies the destination location within the pallet.
Position	Target position	ARRAY [0..5] OF REAL	Positive number, Negative number, or 0	[0,0,0,0,0,0]	Target position: Position[0]: X axis coordinate or Joint_1 Position[1]: Y axis coordinate or Joint_2 Position[2]: Z axis coordinate or Joint_3 Position[3]: Yaw angle coordinate or Joint_4 Position[4]: Pitch angle coordinate or Joint_5 Position[5]: Roll angle coordinate or Joint_6

Note Position, location number and pallet data cannot be used at the same time, the order of preference is like the one displayed in the structure.

**sARB\_MOVE\_CONFIG\_REF**

Member Variable	Meaning	Data type	Valid range	Default	Function
Relative	Relative Move	BOOL	TRUE or FALSE	FALSE	Determines if the movement is relative to the robot's current position or to absolute coordinates. When TRUE the movement is relative otherwise Absolute is set.
CoarseNulling	Coarse nulling tolerances	BOOL	TRUE or FALSE	FALSE	Determines the nulling tolerances that will apply at the end of the motion.
SingleTurn	Single turn	BOOL	TRUE or FALSE	FALSE	Allows full rotations of the robot wrist joints, enabling this bit limits rotations to the range +/- 180 degrees.
AbsoluteApproach	Absolute Approach	BOOL	TRUE or FALSE	FALSE	Determines the effect of the defined approach height value. When it is TRUE the robot moves to the specified location using the height value as absolute Z, otherwise the approach to the location is using the Z offset from the defined location.
ApproachHeight	Approach height value	REAL	Positive number, Negative number, or 0	0	Value that defines the approach height.
Righty	Righty arm configuration	BOOL	TRUE or FALSE	FALSE	When TRUE righty configuration is applied otherwise is lefty. Note that this is only effective for Cartesian coordinates.
Below	Below arm configuration	BOOL	TRUE or FALSE	FALSE	When TRUE below configuration is applied otherwise is above. Note that this is only effective for Cartesian coordinates.
Flip	Flip arm configuration	BOOL	TRUE or FALSE	FALSE	When TRUE flip configuration is applied otherwise is no flip. Note that this is only effective for Cartesian coordinates.

**sARB\_MOTION\_PARAMS\_REF**

Member Variable	Meaning	Data type	Valid range	Function
Speed	Target speed	INT	1 to 120	Set target speed (percentage of robot maximum speed configured in the controller)
Acceleration	Target acceleration	INT	0 to 1000	Set target acceleration (percentage of robot maximum acceleration configured in the controller)
Deceleration	Target deceleration	INT	0 to 1000	Set target deceleration (percentage of robot maximum deceleration configured in the controller)
AccelerationProfile	Acceleration Type	INT	0, 1, 2, 3, 4	Specify the acceleration and deceleration profile. 0 = Trapezoidal movement profile 1,2,3,4 = S-shaped moving profile
SpeedLimit	Speed limit	INT	0 to 100	Maximum joint speed (percentage of joint maximum speed configured in the controller)

**sARB\_JOG\_PARAMS\_REF**

Member variable	Meaning	Data type	Valid range	Default	Function
Speed	Target speed	INT	0 to 100	0	Set target speed (percentage of robot maximum speed configured in the controller)
SpeedLimit	Speed limit	INT	0 to 100	0	Maximum joint speed (percentage of joint maximum speed configured in the controller)

**sARB\_ALIGN\_CONFIG\_REF**

Member variable	Meaning	Data type	Valid range	Function
CoarseNulling	Coarse nulling tolerances	BOOL	TRUE or FALSE	Determines the nulling tolerances that will apply at the end of the motion.
SingleTurn	Single turn	BOOL	TRUE or FALSE	Allows full rotations of the robot wrist joints, enabling this bit limits rotations to the range +/- 180 degrees.

# ARB\_SetToolTransform

The ARB\_SetToolTransform instruction defines and sets a robot tool by setting a tool system transformation to the robot.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_SetTool Transform	Set Tool Transform	FB		<pre>ARB_SetToolTransform_instance(   RobotData:=parameter,   Execute:=parameter,   ToolData:=parameter,   Done =&gt;parameter,   Busy =&gt;parameter,   Error =&gt;parameter,   ErrorID=&gt;parameter,   ErrorIDEx=&gt;parameter );</pre>

## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.
ToolData	Tool definition	sARB_TOOLDATA_REF	---	---	Tool definition.

#### ● sARB\_TOOLDATA\_REF

Name	Meaning	Data type	Valid range	Default	Description
LocationNumber	Location number	INT	0 and [2...999]	0	Identifies a location stored in the robot controller used to define the robot tool.  When 0, it means that LocationNumber is not used and ToolTransform has to be used instead.
ToolTransform	Tool Coordinate System Transformation definition	ARRAY [0..5] OF REAL	Positive number, Negative number, or 0	[0.0,...,0.0]	X, Y, Z, Yaw, Pitch, Roll defined respect to the robot tool flange.

## Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the execution is completed. Tool is set.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error code	WORD	*1	Specify error number.
ErrorIDEx	Extended error ID	DWORD	*1	Specify extended error number.

\*1. Refer to *Troubleshooting* on page 41 for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	<ul style="list-style-type: none"> <li>When execution is completed.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one control period when <i>Execute</i> is FALSE.</li> </ul>
Busy	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to TRUE.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>
Error	<ul style="list-style-type: none"> <li>When there is an error in the execution conditions or input parameters for the instruction.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to FALSE.</li> </ul>

## In-Out Variables

Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOT_DATA_REF	---	Specify data needed for the robot controller.

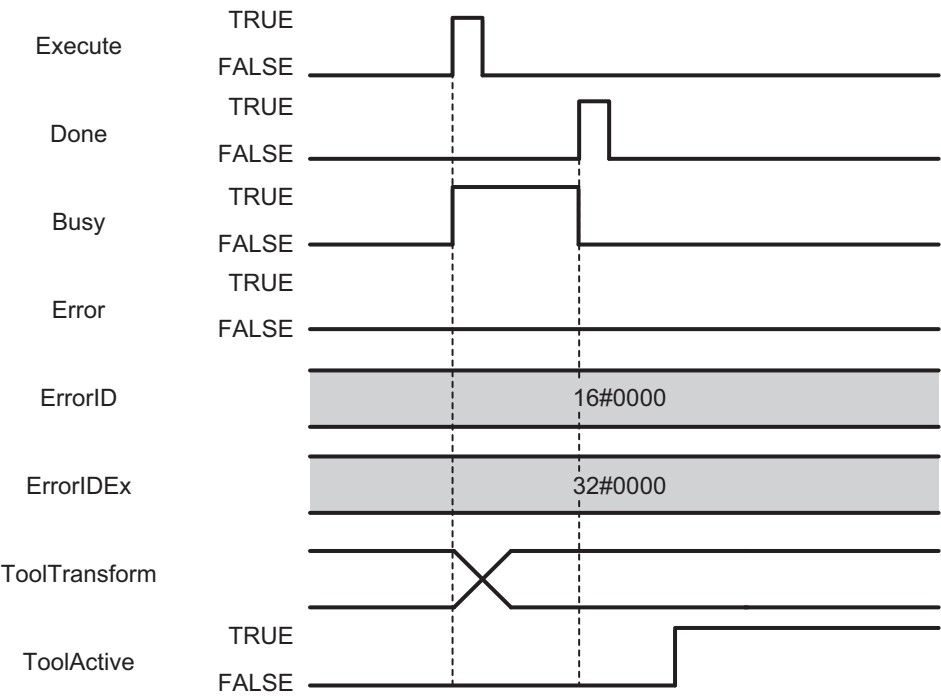
## Function

- This instruction is used to set a tool coordinate system for the robot.
- Only one tool can be defined and activated for each robot.
- When the tool coordinate system is successfully registered, it is used by movement commands.
- The setting tool will be retained till it is reset.



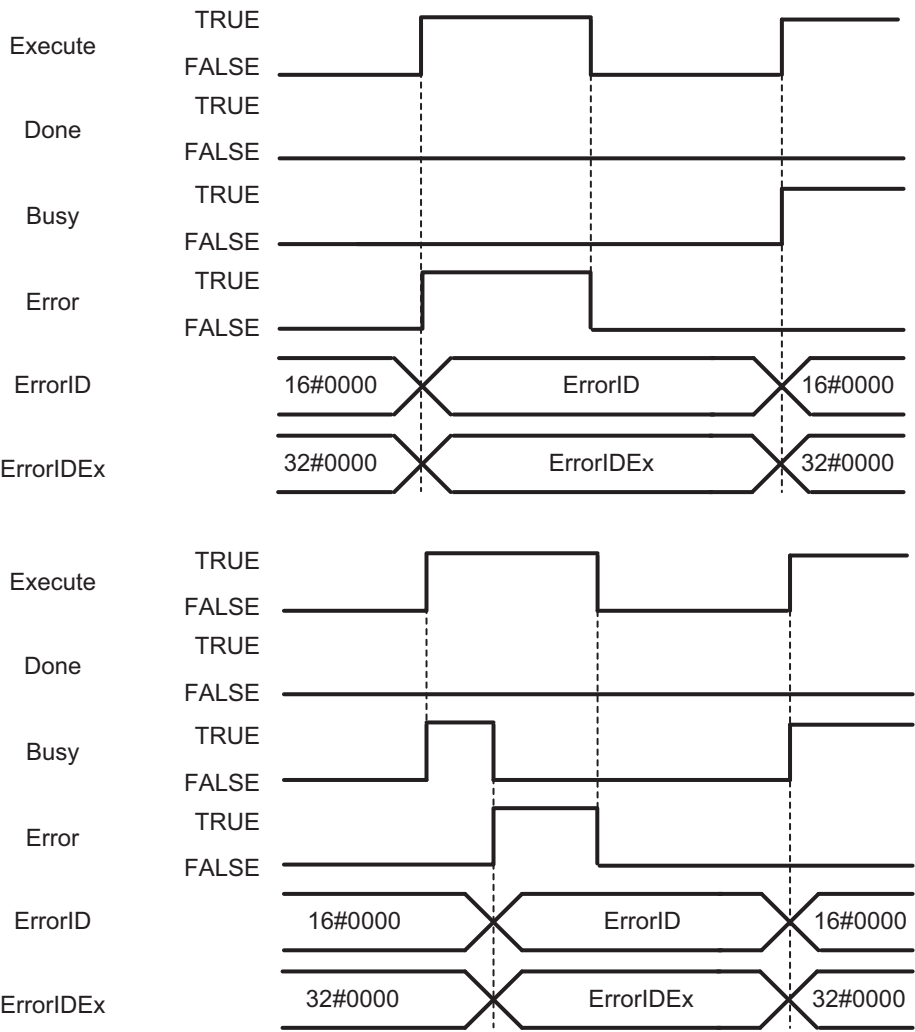
Timing charts

● Normal execution



● Error execution

If an error occurs during instruction execution, *Error* will change to TRUE. You can find out the cause of the error by referring to the value output by *ErrorID* and *ErrorIDEx*.



## Troubleshooting

Error code	Extended error ID	Status	Meaning	Correction
16#0000	16#00000000	Normal End	---	---
16#3C32	16#00000001	Robot controller error	Error occurs in the robot controller.	Check the RobotError variable output in ARB_RobotControl and perform corrections for the error.
	16#00000002	Communication error	Ethernet/IP communication error.	Execute the instruction after the successful EtherNet/IP communication recovery.
	16#00000003	Invalid operation	The instruction could not be executed in this controller state. <i>See Execution Conditions Table on page 143.</i>	Execute the instruction after the function block is ready for execution.
	16#00000004	Re-execution disabled	Instruction cannot be re-executed	Wait for the Done variable output changes to TRUE and then execute the function block again.
	16#00000005	Multi-execution disabled	Multiple instructions that cannot be executed simultaneously were executed for the same target.	Wait for the Done variable output in the previously executed instructions changes to TRUE and then execute the function block.
	16#00000006	Location Number Out of Range	The value specified for the parameter LocationNumber in the input variable ToolData is out of range.	Set the LocationNumber within range.

# ARB\_ResetToolTransform

The ARB\_ResetToolTransform instruction resets the active robot tool which is set to the robot.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_ResetToolTransform	Reset Tool Transform	FB	<div>ARB_ResetToolTransform_instance</div> <div><div><div>\\OmronLib\EIP_Adept</div><div>\\ARB_ResetToolTransform</div></div><div><div>RobotData</div><div>Execute</div></div><div><div>RobotData</div><div>Done</div><div>Busy</div><div>Error</div><div>ErrorID</div><div>ErrorIDEx</div></div></div>	ARB_ResetToolTransform_instance( RobotData:=parameter, Execute:=parameter, Done =>parameter, Busy =>parameter, Error =>parameter, ErrorID=>parameter, ErrorIDEx=>parameter );

ARB\_ResetToolTransform\_instance(  
RobotData:=parameter,  
Execute:=parameter,  
Done =>parameter,  
Busy =>parameter,  
Error =>parameter,  
ErrorID=>parameter,  
ErrorIDEx=>parameter  
);

## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.

### Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the execution is completed. Tool is reset
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error ID	WORD	*1	Specify error number.
ErrorIDEx	Extended error ID	DWORD	*1	Specify extended error number.

\*1. Refer to *Troubleshooting* on page 47 for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	<ul style="list-style-type: none"> <li>When execution is completed.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one control period when <i>Execute</i> is FALSE.</li> </ul>
Busy	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to TRUE.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>
Error	<ul style="list-style-type: none"> <li>When there is an error in the execution conditions or input parameters for the instruction.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to FALSE.</li> </ul>

### In-Out Variables

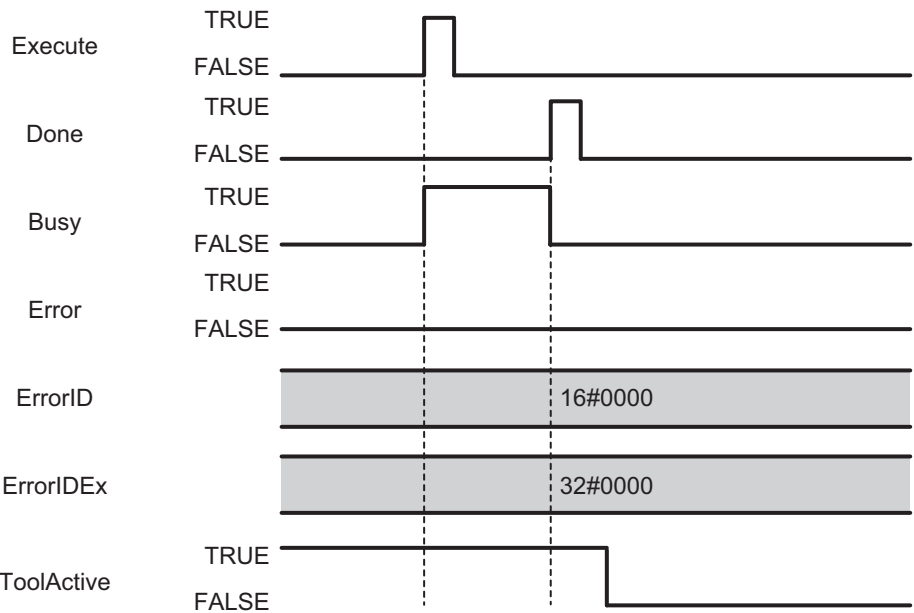
Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOTDATA_REF	---	Specify data needed for the robot controller.

## Function

This instruction is used to reset the active tool coordinate system for the robot.  
If there is no active tool, this instruction has no effect.

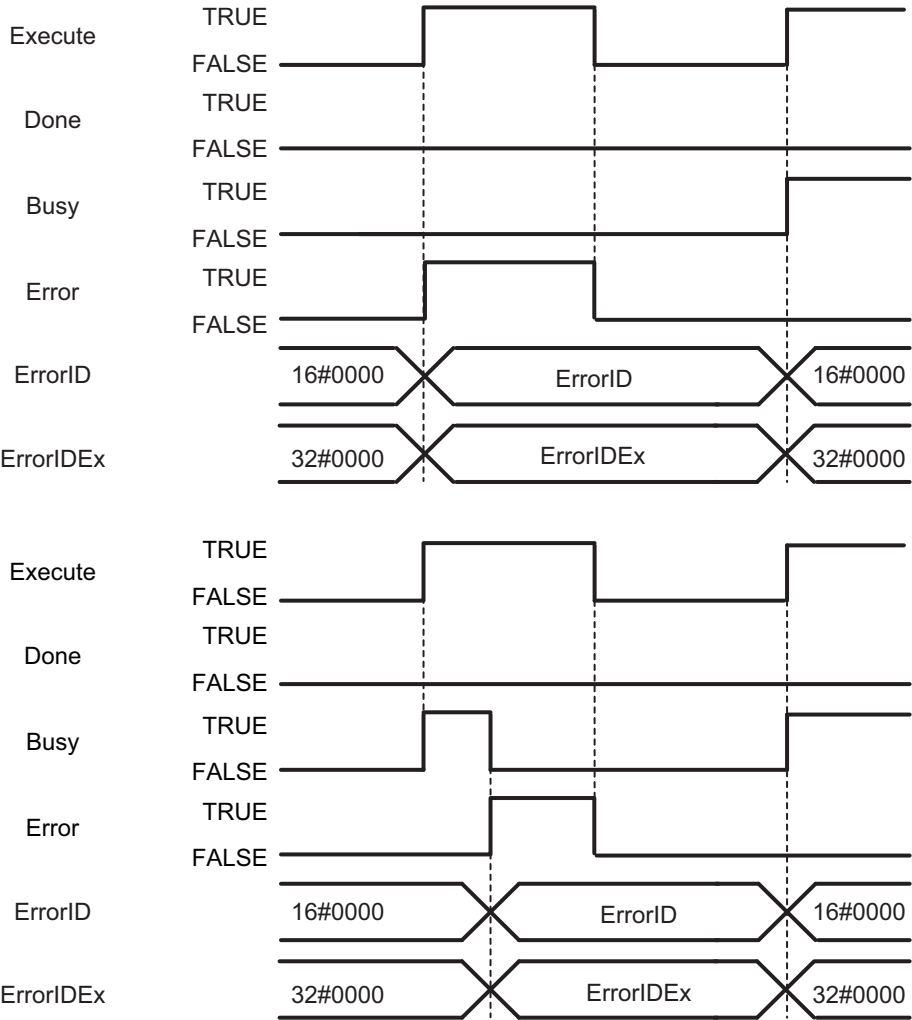
Timing charts

● Normal execution



● Error execution

If an error occurs during instruction execution, *Error* will change to TRUE. You can find out the cause of the error by referring to the value output by *ErrorID* and *ErrorIDEx*.





## Troubleshooting

Error code	Extended error ID	Status	Meaning	Correction
16#0000	16#00000000	The service ended normally.	---	---
16#3C33	16#00000001	Robot controller error	Error occurs in the robot controller.	Check the RobotError variable output in ARB_RobotControl and perform corrections for the error.
	16#00000002	Communication error	Ethernet/IP communication error.	Execute the instruction after the successful EtherNet/IP communication recovery.
	16#00000003	Invalid operation	The instruction could not be executed in this controller state. <i>See Execution Conditions Table on page 143.</i>	Execute the instruction after the function block is ready for execution.
	16#00000004	Re-execution disabled	Instruction cannot be re-executed	Wait for the Done variable output changes to TRUE and then execute the function block again.
	16#00000005	Multi-execution disabled	Multiple instructions that cannot be executed simultaneously were executed for the same target.	Wait for the Done variable output in the previously executed instructions changes to TRUE and then execute the function block.

# ARB\_DefineLocation

The ARB\_DefinePosition instruction defines a position in the robot controller.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_DefineLocation	Define Location	FB	<div>ARB_DefineLocation_instance</div> <div><div>\\OmronLib\EIP_Adept</div><div>\\ARB_DefineLocation</div><div><div>RobotData</div><div>Execute</div><div>PositionData</div></div><div><div>RobotData</div><div>Done</div><div>Busy</div><div>Error</div><div>ErrorID</div><div>ErrorIDEx</div></div></div>	ARB_DefineLocation_instance( RobotData:=parameter, Execute:=parameter, PositionData:=parameter, Done =>parameter, Busy =>parameter, Error =>parameter, ErrorID=>parameter, ErrorIDEx=>parameter );

ARB\_DefineLocation\_instance(  
RobotData:=parameter,  
Execute:=parameter,  
PositionData:=parameter,  
Done =>parameter,  
Busy =>parameter,  
Error =>parameter,  
ErrorID=>parameter,  
ErrorIDEx=>parameter  
);

## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.
PositionData	Position definition	sARB_POSITIONDATA_REF	---	---	Position definition.

#### ● sARB\_POSITIONDATA\_REF

Member Variable	Meaning	Data type	Valid range	Description
LocationNumber	Location number	INT	2 to 999	Specify the number of a location where the position is stored.
Position	Position	ARRAY [0..5] OF REAL	Positive number, Negative number, or 0	Specify the position: Position[0]: X axis coordinate or Joint_1 Position[1]: Y axis coordinate or Joint_2 Position[2]: Z axis coordinate or Joint_3 Position[3]: Yaw angle coordinate or Joint_4 Position[4]: Pitch angle coordinate or Joint_5 Position[5]: Roll angle coordinate or Joint_6

## Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the execution is completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error ID	WORD	*1	Specify error number.
ErrorIDEx	Extended error ID	DWORD	*1	Specify extended error number.

\*1. Refer to *Troubleshooting* on page 54 for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	<ul style="list-style-type: none"> <li>TRUE when the execution is completed.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one control period when <i>Execute</i> is FALSE.</li> </ul>
Busy	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to TRUE.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>
Error	<ul style="list-style-type: none"> <li>When there is an error in the execution conditions or input parameters for the instruction.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to FALSE.</li> </ul>

## In-Out Variables

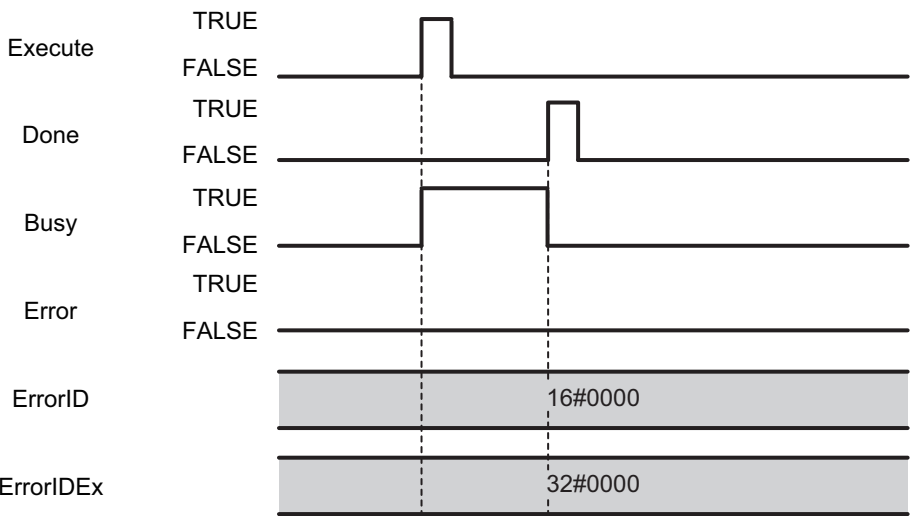
Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOT_DATA_REF	---	Specify data needed for the robot controller.

## Function

- This instruction is used to download a position into the robot controller.
- It's possible to define a position in both, Cartesian and Joint space.
- Positions must be downloaded to Locations each time the Adept controller is restarted.

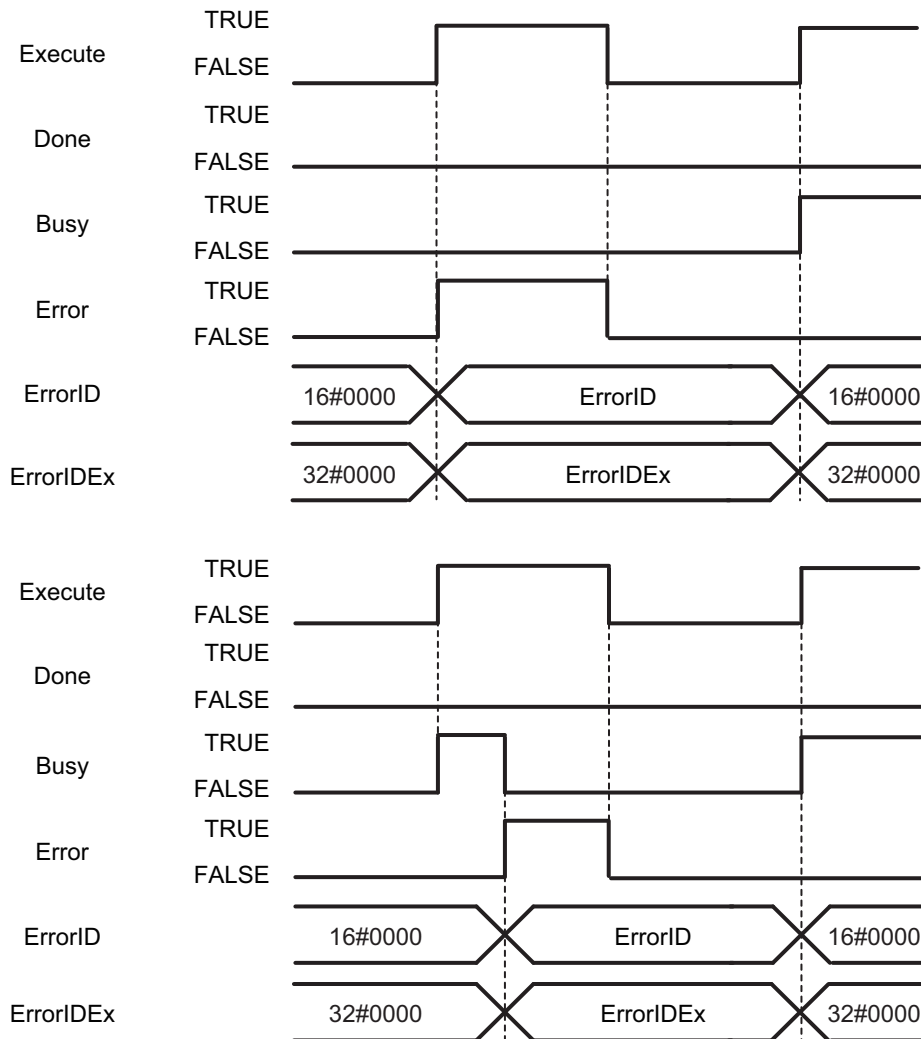
Timing charts

● Normal execution



## ● Error execution

If an error occurs during instruction execution, *Error* will change to TRUE. You can find out the cause of the error by referring to the value output by *ErrorID* and *ErrorIDEx*.



## Troubleshooting

Error code	Extended error ID	Status	Meaning	Correction
16#0000	16#00000000	The service ended normally.	---	---
16#3C34	16#00000001	Robot controller error	Error occurs in the robot controller.	Check the RobotError variable output in ARB_RobotControl and perform corrections for the error.
	16#00000002	Communication error	Ethernet/IP communication error	Execute the instruction after the successful EtherNet/IP communication recovery.
	16#00000003	Invalid operation	The instruction could not be executed in this controller state. <i>See Execution Conditions Table on page 143.</i>	Execute the instruction after the function block is ready for execution.
	16#00000004	Re-execution disabled	Instruction cannot be re-executed	Wait for the Done variable output changes to TRUE and then execute the function block again.
	16#00000005	Multi-execution disabled	Multiple instructions that cannot be executed simultaneously were executed for the same target.	Wait for the Done variable output in the previously executed instructions changes to TRUE and then execute the function block.
	16#00000006	Location Number Out of Range	The value specified for the parameter LocationNumber in the input variable Position-Data is out of range.	Set the LocationNumber within range.



# ARB\_DefinePallet

The ARB\_DefinePallet instruction defines all pallet information in the robot controller.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_DefinePallet	Define Pallet	FB	<div>ARB_DefinePallet_instance</div> <div><div><div>\\OmronLib\EIP_Adept</div><div>ARB_DefinePallet</div></div><div><div>RobotData</div><div>Execute</div><div>PalletData</div></div><div><div>RobotData</div><div>Done</div><div>Busy</div><div>Error</div><div>ErrorID</div><div>ErrorIDEx</div></div></div>	ARB_DefinePallet_instance( RobotData:=parameter, Execute:=parameter, PalletData:=parameter, Done =>parameter, Busy =>parameter, Error =>parameter, ErrorID=>parameter, ErrorIDEx=>parameter );

ARB\_DefinePallet\_instance(  
RobotData:=parameter,  
Execute:=parameter,  
PalletData:=parameter,  
Done =>parameter,  
Busy =>parameter,  
Error =>parameter,  
ErrorID=>parameter,  
ErrorIDEx=>parameter  
);

## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.
PalletData	Pallet definition	sARB_PAL LETDATA_ REF	---	---	Pallet definition data.

#### ● sARB\_PALLETDATA\_REF

Member Variable	Meaning	Data type	Valid range	Description
PalletNumber	Pallet number	INT	1...100	Specify the number of the pallet.
StartingLocation- Number	Starting location number	INT	2...999	Specify the number of a location where the position is stored.
FirstRowLocation- Number	First row location number	INT	2...999	Specify the number of a location where the position is stored.
LastRowLocation- Number	Last row position number	INT	2...999	Specify the number of a location where the position is stored.
NumberPosition- FirstRow	Number of posi- tion in the first row	INT	Positive num- ber, Negative number, or 0	Number of location in the first row
NumberRows	Number of rows	INT	Positive num- ber, Negative number, or 0	Number of rows
PalletConfig	Configuration	INT	[0, 1, 2, 3, 4]	Pallet configuration type
TravDirection	Traverse direction	INT	[0, 1]	Traverse direction

## Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the execution is completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error ID	WORD	*1	Specify error number.
ErrorIDEx	Extended error ID	DWORD	*1	Specify extended error number.

\*1. Refer to *Troubleshooting* on page 63 for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	<ul style="list-style-type: none"> <li>When execution is completed.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one control period when <i>Execute</i> is FALSE.</li> </ul>
Busy	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to TRUE.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>
Error	<ul style="list-style-type: none"> <li>When there is an error in the execution conditions or input parameters for the instruction.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to FALSE.</li> </ul>

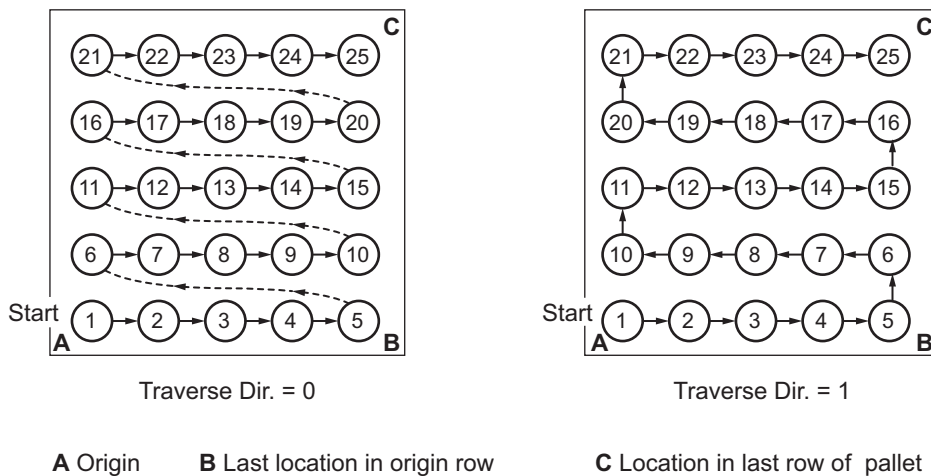
## In-Out Variables

Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOT_DATA_REF	---	Specify data needed for the robot controller.

## Function

- This instruction is used to define all information related to the pallet.
- It uses three locations that have to be already defined: StartingLocation (A), FirstRowLocation (B) and LastRowLocation (C).
- The origin is the first position in the pallet which corresponds to 1 (not 0).
- The spacing between rows is constant for all rows.
- The even-numbered rows, if offset, are always offset by half the distance between positions in a row.
- When defining the pallet, if location C is on the same line (row) as A and B, an error will be reported.
- In the case of a single-row pallet, location C must be the same as location B.

Image below represents an overview of the pallet layout definition elements:

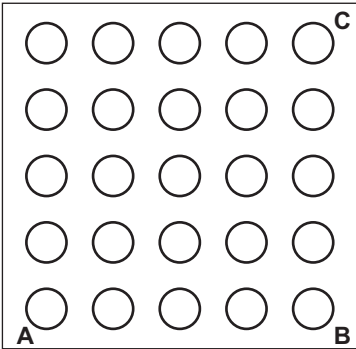


● Possible pallet configuration codes:

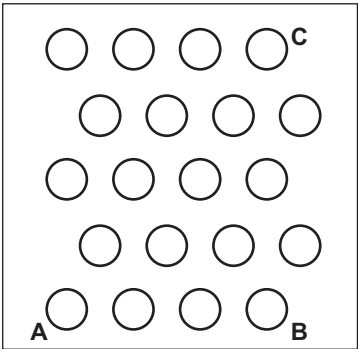
A Origin

B Last location in origin row

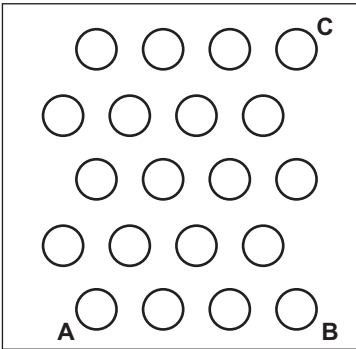
C Location in last row of pallet



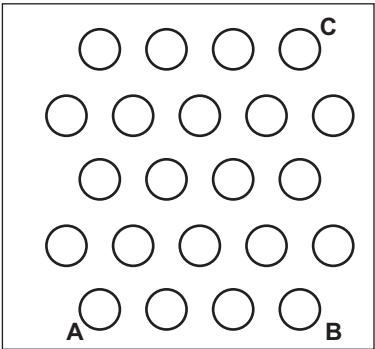
Config. Code = 0



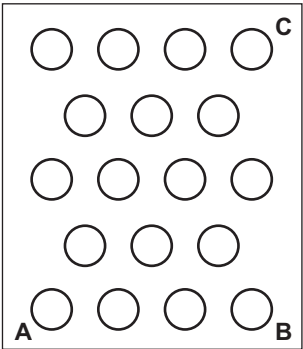
Config. Code = 1



Config. Code = 2



Config. Code = 3



Config. Code = 4

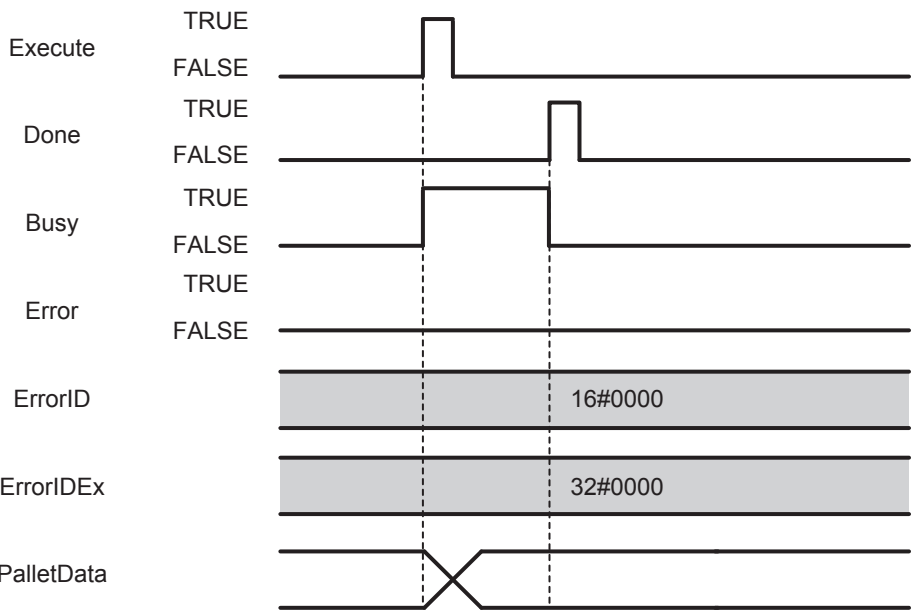
## Precautions for Correct Use

A pallet is defined by executing ARB\_DefinePallet, but it can be re-defined from the both, the function block and from changing the pallet definition data in the robot controller (re-defining one of the three location numbers A-B-C used by the pallet).

The second way might be unexpected behaviour for user and an explanation will be added to the User Manual.

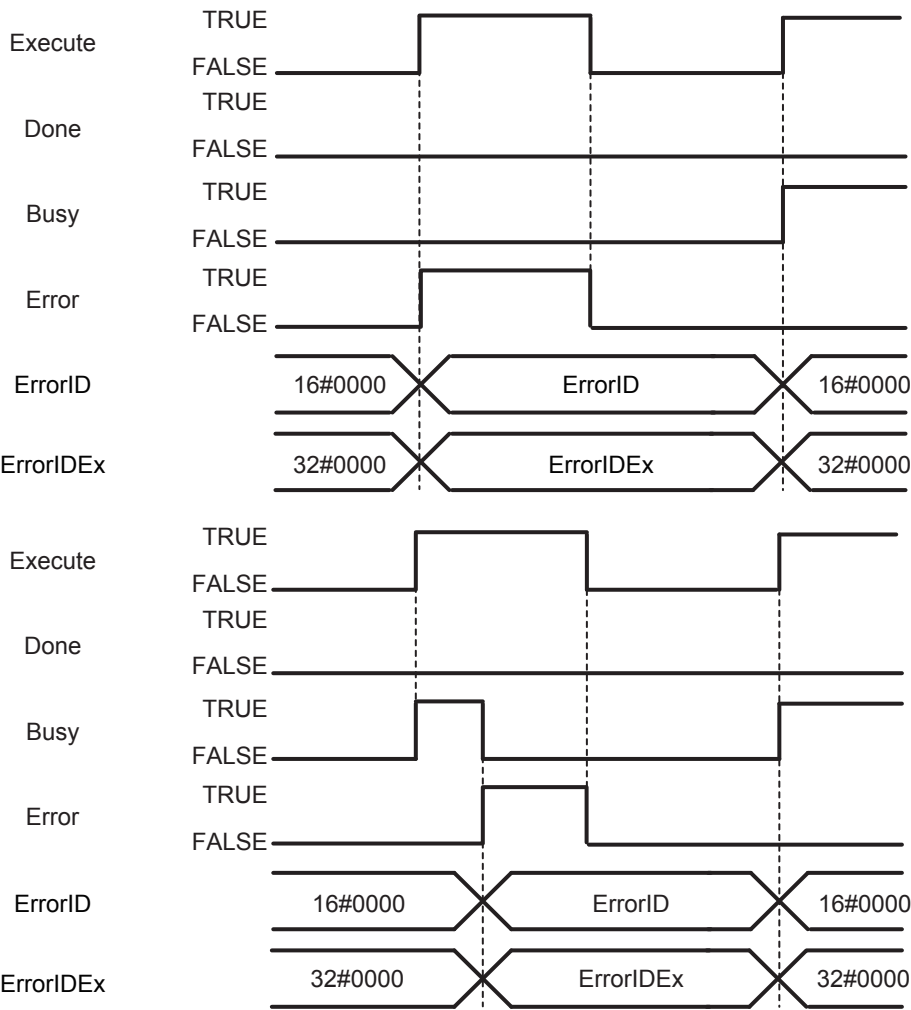
Timing charts

● Normal execution



● Error execution

If an error occurs during instruction execution, *Error* will change to TRUE. You can find out the cause of the error by referring to the value output by *ErrorID* and *ErrorIDEx*.





## Troubleshooting

Error code	Extended error ID	Status	Meaning	Correction
16#0000	16#00000000	The service ended normally.	---	---
16#3C35	16#00000001	Robot controller error	Error occurs in the robot controller.	Check the RobotError variable output in ARB_RobotControl and perform corrections for the error.
	16#00000002	Communication error	Ethernet/IP communication error	Execute the instruction after the successful EtherNet/IP communication recovery.
	16#00000003	Invalid operation	The instruction could not be executed in this controller state. <i>See Execution Conditions Table on page 143.</i>	Execute the instruction after the function block is ready for execution.
	16#00000004	Re-execution disabled	Instruction cannot be re-executed	Wait for the Done variable output changes to TRUE and then execute the function block again.
	16#00000005	Multi-execution disabled	Multiple instructions that cannot be executed simultaneously were executed for the same target.	Wait for the Done variable output in the previously executed instructions changes to TRUE and then execute the function block.
	16#00000006	Location Number Out of Range	The value specified any of the parameters in the input variable PalletData is out of range.	Set the parameter value within range.
	16#00000007	Pallet Number Out of Range	The value specified for the parameter PalletNumber in the input variable PalletData is out of range.	Set the PalletNumber within range.
	16#00000008	Pallet Configuration Selection Out of Range	The value specified for the parameter PalletConfig in the input variable PalletData is out of range.	Set the PalletConfig within range.
	16#00000009	Traverse Direction Out of Range	The value specified for the parameter TravDirection in the input variable PalletData is out of range.	Set the TravDirection within range.

# ARB\_ResetRobotError

The ARB\_ResetRobotError instruction resets any existing error in the ePLC software.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_ResetRobotError	Reset Error	FB	<div>ARB_ResetRobotError_instance</div> <div><div><div>\\OmronLib\EIP_Adept</div><div>ARB_ResetRobotError</div></div><div><div>RobotData</div><div>Execute</div></div><div><div>RobotData</div><div>Done</div><div>Busy</div><div>Error</div><div>ErrorID</div><div>ErrorIDEx</div></div></div>	ARB_ResetRobotError_instance( RobotData:=parameter, Execute:=parameter, Done =>parameter, Busy =>parameter, Error =>parameter, ErrorID:=parameter, ErrorIDEx:=parameter );

ARB\_ResetRobotError\_instance(  
RobotData:=parameter,  
Execute:=parameter,  
Done =>parameter,  
Busy =>parameter,  
Error =>parameter,  
ErrorID=>parameter,  
ErrorIDEx=>parameter  
);

## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.

### Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the execution is completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error ID	WORD	*1	Specify error number.
ErrorIDEx	Extended error ID	DWORD	*1	Specify extended error number.

\*1. Refer to *Troubleshooting* on page 69 for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	<ul style="list-style-type: none"> <li>TRUE when the execution is completed.</li> </ul>	<ul style="list-style-type: none"> <li>When Execute is TRUE and changes to FALSE.</li> <li>After one control period when Execute is FALSE.</li> </ul>
Busy	<ul style="list-style-type: none"> <li>When Execute changes to TRUE.</li> </ul>	<ul style="list-style-type: none"> <li>When Done changes to TRUE.</li> <li>When Error changes to TRUE</li> </ul>
Error	<ul style="list-style-type: none"> <li>When there is an error in the execution conditions or input parameters for the instruction.</li> </ul>	<ul style="list-style-type: none"> <li>When Execute changes to FALSE.</li> </ul>

### In-out variables

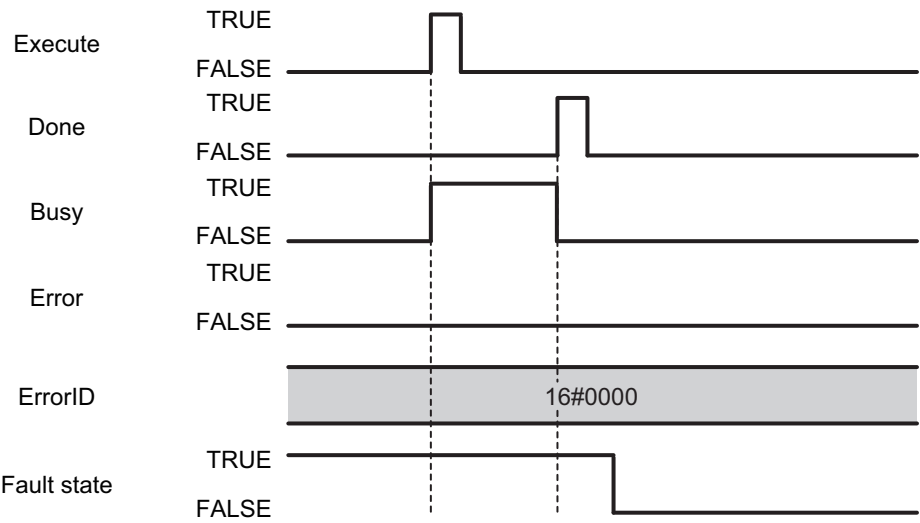
Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOT_DATA_REF	---	Specify data needed for the robot controller.

## Function

- Resets the ePLC connect software if it is in a fault state. The current error message and error number tags are cleared by the reset command.
- When an error occurs, the robot goes to fault state. The reset command will release the robot from the fault state.
- In order to complete the reset correctly there cannot be any other command in execution.
- If reset is executed when the robot controller is not in fault state, then the reset operation is not send to the controller.

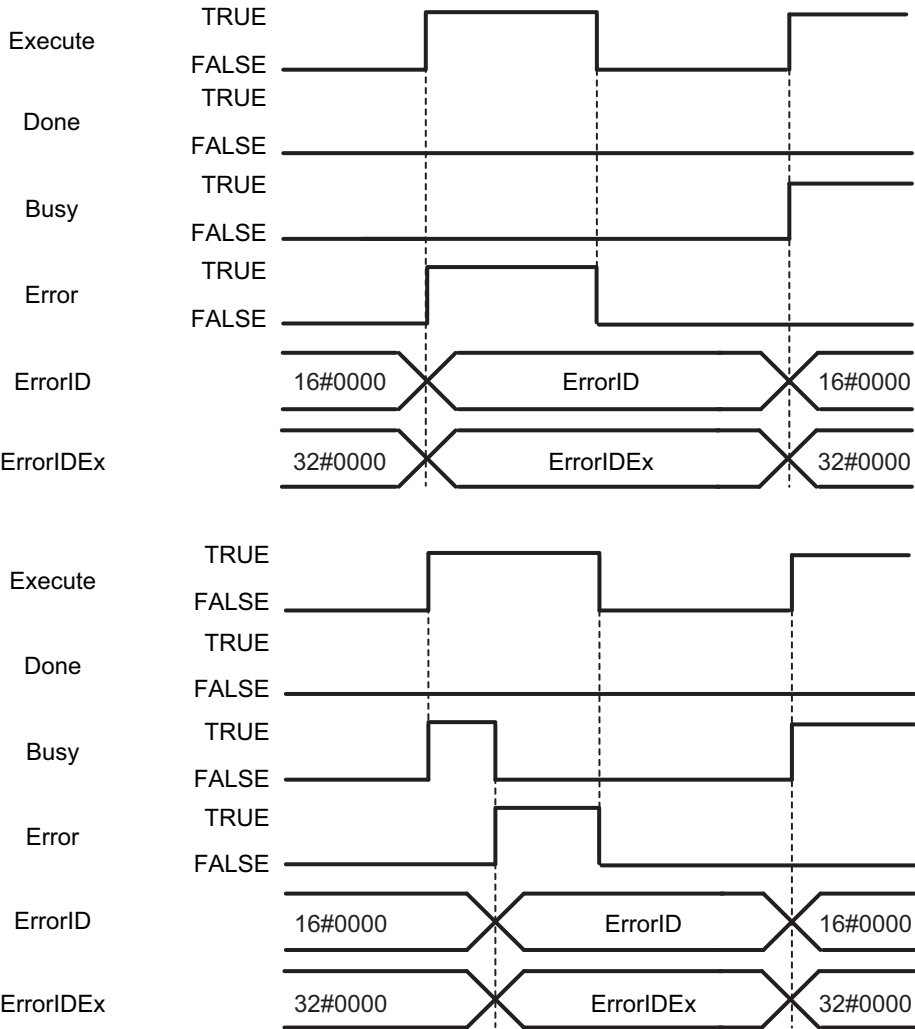
Timing charts

● Normal execution



● Error execution

If an error occurs during instruction execution, *Error* will change to TRUE. You can find out the cause of the error by referring to the value output by *ErrorID* and *ErrorIDEx*.



## Troubleshooting

Error code	Extended error ID	Status	Meaning	Correction
16#0000	16#00000000	The service ended normally.	---	---
16#3C36	16#00000001	Robot controller error	Error occurs in the robot controller.	Check the RobotError variable output in ARB_RobotControl and perform corrections for the error.
	16#00000002	Communication error	Ethernet/IP communication error	Execute the instruction after the successful EtherNet/IP communication recovery.
	16#00000004	Re-execution disabled	Instruction cannot be re-executed	Wait for the Done variable output changes to TRUE and then execute the function block again.

# ARB\_RobotControl

The ARB\_RobotControl instruction controls the main robot settings and operations and monitors the robot states, positions, configuration and errors.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_RobotControl	Stop Control	FB	<div>ARB_RobotControl_instance</div> <div><div>\\OmronLib\EIP_Adept</div><div>ARB_RobotControl</div></div> <div><div>RobotData</div><div>Enable</div><div>Power</div><div>Calibrate</div><div>CancelMotion</div><div>StopOnInput</div></div> <div><div>RobotData</div><div>Enabled</div><div>Busy</div><div>Connected</div><div>RobotState</div><div>RobotMotion</div><div>RobotPosition</div><div>RobotConfig</div><div>RobotError</div></div>	ARB_RobotControl_instance( RobotData:=parameter, Enable:=parameter, Power:=parameter, Calibrate:=parameter, CancelMotion:=parameter, StopOnInput:=parameter, Enabled =>parameter, Busy =>parameter, Connected =>parameter, RobotState=>parameter, RobotMotion=>parameter, RobotPosition=>parameter, RobotConfig=>parameter, RobotError=>parameter );

ARB\_RobotControl\_instance(  
RobotData:=parameter,  
Enable:=parameter,  
Power:=parameter,  
Calibrate:=parameter,  
CancelMotion:=parameter,  
StopOnInput:=parameter,  
Enabled =>parameter,  
Busy =>parameter,  
Connected =>parameter,  
RobotState=>parameter,  
RobotMotion=>parameter,  
RobotPosition=>parameter,  
RobotConfig=>parameter,  
RobotError=>parameter  
);



## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Enable	Enable	BOOL	TRUE or FALSE	FALSE	The instruction is executed while Enable is to TRUE.
Power	Enable Power	BOOL	TRUE or FALSE	FALSE	Enable Power of the robot when input is TRUE
Calibrate	Calibrate Robot	BOOL	TRUE or FALSE	FALSE	Robot calibration starts when the input is set to TRUE
CancelMotion	Cancel Robot Movement	BOOL	TRUE or FALSE	FALSE	The brake instruction is executed when the input is set to TRUE
StopOnInput	Stop on latch input	BOOL	TRUE or FALSE	FALSE	The option to stop on input is enabled while the input is TRUE

### Output Variables

Name	Meaning	Data type	Valid range	Description
Enabled	Done	BOOL	TRUE or FALSE	TRUE while the instruction is being executed without error.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Connected	Connected	BOOL	TRUE or FALSE	TRUE when the connection with the controller is working.
RobotState	Robot State	sARB_ROBOTSTATE_REF	---	Robot State
RobotMotion	Robot Motion	sARB_ROBOT_MOTION_REF	---	Robot motion data.
RobotPosition	Robot Position	sARB_ROBOTPOS_REF	---	Robot Actual Position in Cartesian and Joint Coordinate Systems
RobotConfig	Robot Configuration	sARB_ROBOTCONFIG_REF	---	Robot configuration status
RobotError	Robot Error	sARB_ROBOTERROR_REF	---	Robot controller error state and message

### ● sARB\_ROBOTSTATE\_REF

Name	Meaning	Data type	Valid range	Default	Description
SysInitState	System Initialized State	BOOL	TRUE or FALSE	FALSE	Status of the initialization of the system. TRUE: System is initialized FALSE: System is not initialized
PowerState	Robot Power state	BOOL	TRUE or FALSE	FALSE	Status of the robot power. TRUE: Robot power is turned ON FALSE: Robot power is turned OFF
CalibratedState	Robot Calibrated state	BOOL	TRUE or FALSE	FALSE	Status of the robot calibration. TRUE: Robot is calibrated FALSE: Robot is not calibrated
ErrorState	Robot Error State	BOOL	TRUE or FALSE	FALSE	Status of the system TRUE: System is in error FALSE: System is not in error
EStopState	Emergency Stop State	BOOL	TRUE or FALSE	FALSE	Status of Emergency Stop input TRUE: E-stop circuit is open FALSE: E-stop circuit is closed

### ● sARB\_ROBOTMOTION\_REF

Name	Meaning	Data type	Valid range	Default	Description
InMotionState	In Motion State	BOOL	TRUE or FALSE	FALSE	Status of the robot motion: TRUE: Robot is moving FALSE: Robot is not moving
InPositionState	In Position State	BOOL	TRUE or FALSE	FALSE	0=Not in position; 1=In position
MotionCompleted	Percentage of motion completed	INT	0 to 100	0	Returns the percentage of the current motion that has completed. The value 100 indicates that no motion is in progress. For continuous-path motions, the value of Motion Completed increases during each motion until the transition to the next motion, and then the value suddenly changes to close to 0 to reflect the start of the next motion. That is, Motion Completed does not reach 100 before it is reset to reflect the next motion.
MotionCounter	Motion Counter	INT	Positive number, Negative number, or 0	0	The Current Motion Counter is incremented each time a motion begins. It can be used to determine that the previous commanded motion has started.

### ● sARB\_ROBOT\_POS\_REF

Name	Meaning	Data type	Valid range	Default	Description
CartesianPosition	Cartesian Position	ARRAY [0..5] OF REAL	Positive number, Negative number, or 0	[0.0 ... 0.0]	Actual Robot Position in Cartesian Coordinates
JointPosition	Joint Position	ARRAY [0..5] OF REAL	Positive number, Negative number, or 0	[0.0 ... 0.0]	Actual Robot Position in Joint Coordinates

### ● sARB\_ROBOTCONFIG\_REF

Name	Meaning	Data type	Valid range	Default	Description
ToolActive	Tool is set	BOOL	TRUE or FALSE	FALSE	State of the tool activation of the robot: TRUE: Tool is active FALSE: Tool is not active (null tool)
RightyConfig	Righty configuration	BOOL	TRUE or FALSE	FALSE	State of the Righty/Lefty configuration of the robot. TRUE: Robot has RIGHTY configuration FALSE: Robot has LEFTY configuration
BelowConfig	Below Configuration	BOOL	TRUE or FALSE	FALSE	State of the Below/Above configuration of the robot. TRUE: Robot has BELOW configuration FALSE: Robot has ABOVE configuration
FlipConfig	Flip Configuration	BOOL	TRUE or FALSE	FALSE	State of the flip/NoFlip configuration of the robot. TRUE: Robot has FLIP configuration FALSE: Robot has NOFLIP configuration

### ● sARB\_ROBOTERROR\_REF

Name	Meaning	Data type	Valid range	Default	Description
ErrorNumber	Error Number	DINT	Positive number, Negative number, or 0	0	Contains the current ePLC Connect software error number. Refer to <i>Errors from Controller</i> on page 147 for details.
ErrorMessage	Error Message	STRING[82]	---	---	Contains the current ePLC Connect software message string

## ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Enabled	<ul style="list-style-type: none"><li>When <i>Enable</i> changes to TRUE.</li></ul>	<ul style="list-style-type: none"><li>When <i>Enable</i> changes to FALSE.</li></ul>
Busy	<ul style="list-style-type: none"><li>When <i>Enable</i> changes to TRUE.</li></ul>	<ul style="list-style-type: none"><li>When <i>Enable</i> changes to FALSE.</li><li>When <i>Error</i> changes to TRUE.</li></ul>
Error	<ul style="list-style-type: none"><li>When there is an error in the execution conditions or input parameters for the instruction.</li></ul>	<ul style="list-style-type: none"><li>When <i>Enable</i> is FALSE</li></ul>

## In-Out Variables

Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOT_DATA_REF	---	Specify data needed for the robot controller.

## Function

The instruction enables the use of the main robot control settings and operations and the monitoring of robot states, positions, configuration and errors.

In order to execute any operation from the function block or to monitor updated outputs the Enable input must be set and the function block must be in Enabled state.

## Operations and Settings of the robot

### ● Power

Power function allows switching the power at the robot on and off to enable the movement of the robot.

The input bit should remain true to keep the power on state.

In order to execute any robot movement or definition (location, pallet, etc.) the robot must be in power on state, otherwise an error occurs.

### ● Calibrate

Initialize the robot positioning system with the robot's current position. When the operation is triggered, the controller executes the main calibration program which causes the robot-specific calibration routines to be loaded and robot calibrated.

The calibration status is output through the field Calibrated in the RobotState structure.

Calibrate operation may operate differently for each type of robot.

If the robot is already calibrated (RobotState.Calibrated=TRUE) no calibration occurs when rising up the calibration input.

In order to execute any robot movement or definition the robot must be calibrated, otherwise an error occurs.

### ● Cancel Motion

Cancels the current motion and decelerates the robot to a stop. Motion can be cancelled if it has been already commanded and not while movement command is being sent. It only affects motion of ARB\_MoveComamnd, ARB\_AlignToolCommand and ARB\_PickAndPlaceCommand.

### ● Stop on input

Immediate stop when digital input is latched. A "stop-on-digital-signal" functionality is available. Motion can be stopped if it has been already commanded and not while movement command is being sent. It only affects motion of ARB\_MoveComamnd, ARB\_AlignToolCommand and ARB\_PickAndPlaceCommand. With this feature, any Adept robot system can be programmed to stop rapidly on a digital-input latch event. Requires the Enhanced Trajectory Generator license and several configuration changes in eV+.

## Status and Monitor

### ● Robot State

Provides information about the state of the robot and the ePLC Connect software. See detailed description of the states in the structure sARB\_ROBOTSTATE\_REF.

### ● Robot Position

Provides current position of the robot in Cartesian coordinates and Joint coordinates.

Positions are a 6-elements array of REAL:

Name	Data type	Meaning	Valid range	Default	Units
CartesianPosition [0]	REAL	Coordinate X Position	Positive number, Negative number, or 0	0.0	mm
CartesianPosition [1]	REAL	Coordinate Y Position	Positive number, Negative number, or 0	0.0	mm
CartesianPosition [2]	REAL	Coordinate Z Position	Positive number, Negative number, or 0	0.0	mm
CartesianPosition [3]	REAL	Rotation around coordinate X	Positive number, Negative number, or 0	0.0	°
CartesianPosition [4]	REAL	Rotation around coordinate Y	Positive number, Negative number, or 0	0.0	°
CartesianPosition [5]	REAL	Rotation around coordinate Z	Positive number, Negative number, or 0	0.0	°

Name	Data type	Meaning	Valid range	Default	Units
JointPosition [0]	REAL	Joint-1 Position	Positive number, Negative number, or 0	0.0	°
JointPosition [1]	REAL	Joint-2 Position	Positive number, Negative number, or 0	0.0	°
JointPosition [2]	REAL	Joint-3 Position	Positive number, Negative number, or 0	0.0	°
JointPosition [3]	REAL	Joint-4 Position	Positive number, Negative number, or 0	0.0	°
JointPosition [4]	REAL	Joint-5 Position	Positive number, Negative number, or 0	0.0	°
JointPosition [5]	REAL	Joint-6 Position	Positive number, Negative number, or 0	0.0	°

### ● Robot Configuration

Provides information about the current configuration of the robot. See detailed description of the states in the structure sARB\_ROBOTCONFIG\_REF.

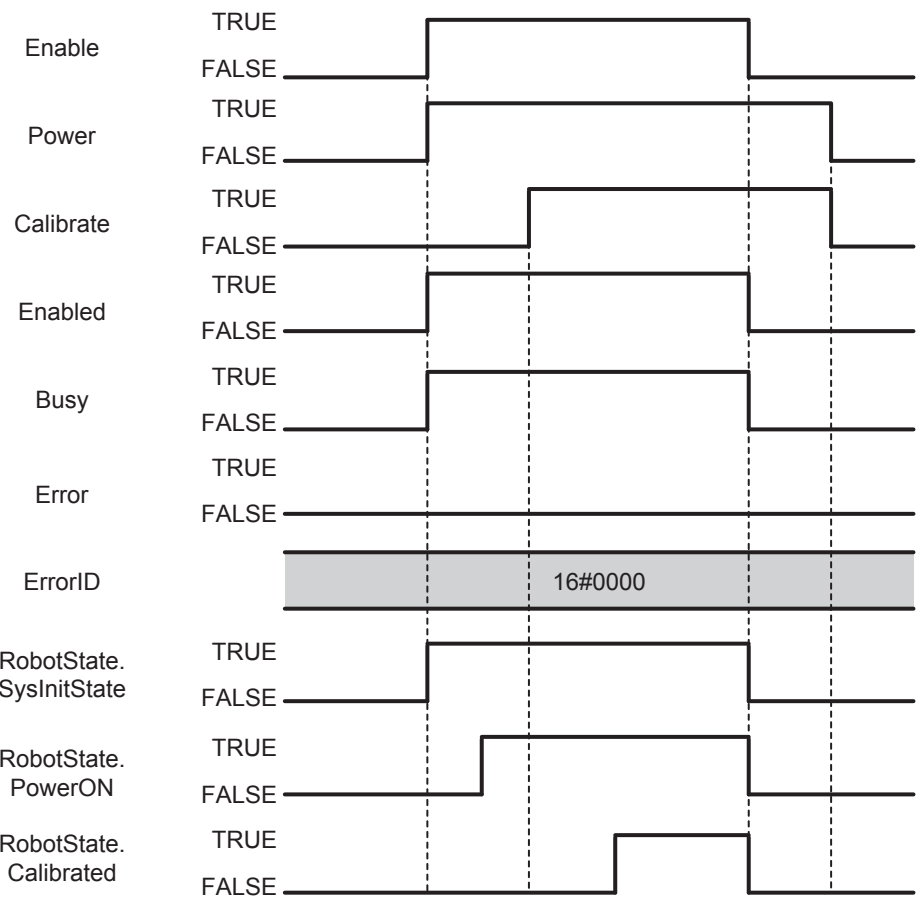
### ● Robot Error

Provides information about the errors of the robot and the ePLC Connect software. See detailed description of the states in the structure sARB\_ROBOTERROR\_REF.

See detailed description of the error codes and messages in the *Troubleshooting*.

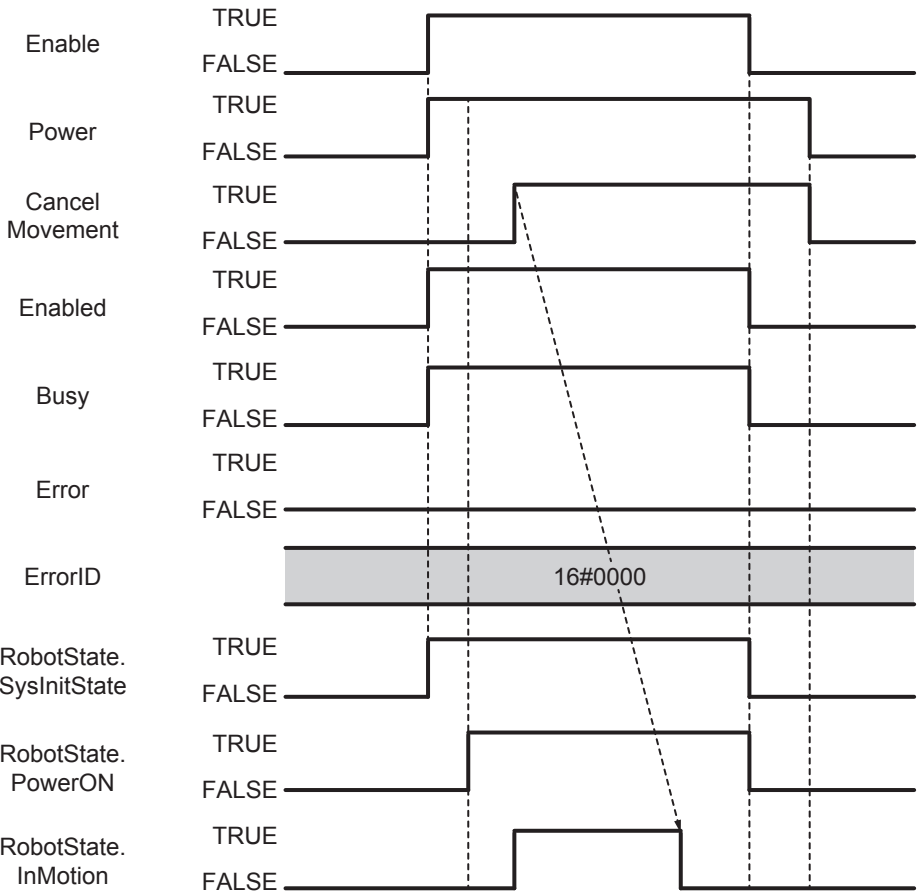
Timing charts

● Power and Calibration

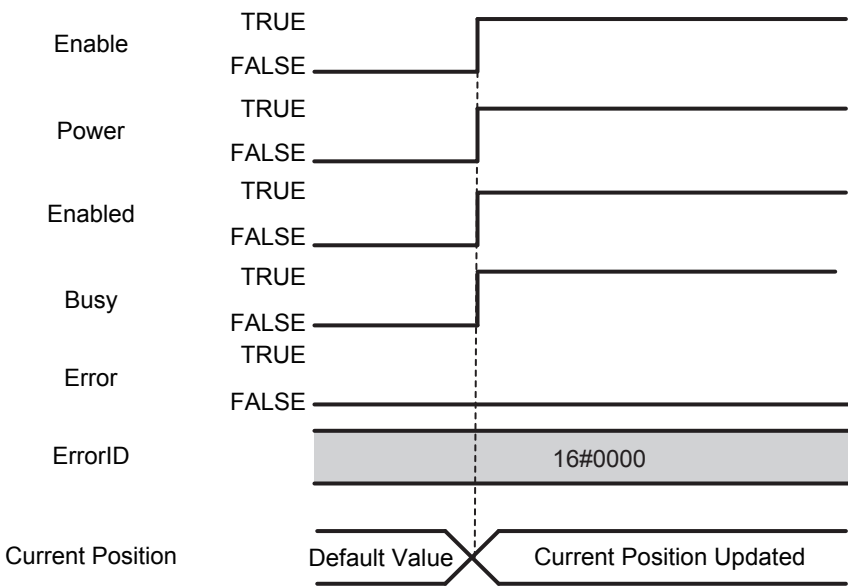




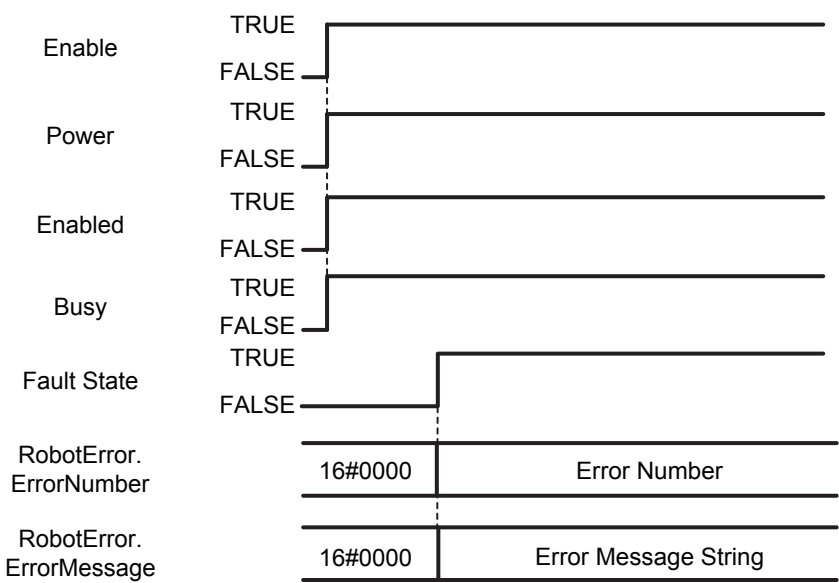
● **Cancel Movement**



● Current Position



● Error State and Messages



## Errors

There is no error state on this function block because as long as it is enabled it provides control of the robot to the user and it is monitoring status and errors of the robot and the ePLC Connect software.

In case of connection problem, instruction can be enabled, but 'Connected' output will be FALSE.

# ARB\_TeachPosition

The ARB\_TeachPosition instruction teaches the current robot position and configuration.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_TeachPosition	Teach Position	FB	<div>ARB_TeachPosition_instance</div> <div><div><div>\\OmronLib\EIP_Adept</div><div>\\ARB_TeachPosition</div></div><div><div>RobotData</div><div>Execute</div><div>LocationNumber</div><div>JointCoord</div></div><div><div>RobotData</div><div>Done</div><div>Busy</div><div>Error</div><div>ErrorID</div><div>ErrorIDEx</div><div>MotionRecord</div></div></div>	ARB_TeachPosition_instance( RobotData:=parameter, Execute:=parameter, LocationNumber:= parameter, JointCoord:= parameter, Done =>parameter, Busy =>parameter, Error =>parameter, ErrorID=>parameter, ErrorIDEx=>parameter, MotionRecord=>parameter );

ARB\_TeachPosition\_instance(  
RobotData:=parameter,  
Execute:=parameter,  
LocationNumber:= parameter,  
JointCoord:= parameter,  
Done =>parameter,  
Busy =>parameter,  
Error =>parameter,  
ErrorID=>parameter,  
ErrorIDEx=>parameter,  
MotionRecord=>parameter  
);

## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.
LocationNumber	Location number	INT	2 to 999	0	Number of a location in the robot controller where the location is stored
JointCoord	Joint Coordinate	BOOL	TRUE or FALSE	0	Position taught in joint or Cartesian coordinates TRUE: joint coordinates FALSE: Cartesian coordinates

### Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the Teaching is completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs.
ErrorIDEx	Extended error ID	DWORD	*1	Specify extended error number.
MotionRecord	Motion Recorded Data	sARB_MOTIONRECORD_REF	---	Contains the information of the taught data

\*1. Refer to *Troubleshooting* on page 89 for details.

## ● sARB\_MOTIONRECORD\_REF

Name	Meaning	Data type	Valid range	Default	Description
Position	Position taught	ARRAY [0..5] OF REAL	Positive number, Negative number, or 0	[0.0 ... 0.0]	Position taught in joint coordinates or Cartesian coordinates depending on the JointCoord input. Position[0]: X axis coordinate or Joint_1 Position[1]: Y axis coordinate or Joint_2 Position[2]: Z axis coordinate or Joint_3 Position[3]: Yaw angle coordinate or Joint_4 Position[4]: Pitch angle coordinate or Joint_5 Position[5]: Roll angle coordinate or Joint_6
JointCoord	Joint Coord- inates	BOOL	TRUE or FALSE	FALSE	Position taught in Joint or Cartesian coordinates. TRUE: joint coordinates FALSE: Cartesian coordinates
RightyConfig	Righty con- figuration	BOOL	TRUE or FALSE	FALSE	Righty/Lefty configuration of the robot when teaching is done. TRUE: Robot has RIGHTY configu- ration FALSE: Robot has LEFTY configu- ration
BelowConfig	Below Con- figuration	BOOL	TRUE or FALSE	FALSE	Below/Aboveconfiguration of the robot when teaching is done. TRUE: Robot has BELOW configu- ration FALSE: Robot has ABOVE configu- ration
FlipConfig	Flip Config- uration	BOOL	TRUE or FALSE	FALSE	Flip/NoFlip configuration of the robot when teaching is done. TRUE: Robot has FLIP configuration FALSE: Robot has NOFLIP configu- ration

## ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	<ul style="list-style-type: none"> <li>When an execution is completed.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one control period when <i>Execute</i> is FALSE.</li> </ul>
Busy	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to TRUE.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>
Error	<ul style="list-style-type: none"> <li>When there is an error in the execution conditions or input parameters for the instruction.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> input is FALSE</li> </ul>

## In-Out Variables

Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOT_DATA_REF	---	Specify data needed for the robot controller.

## Function

This instruction is used to teach current position into the robot controller. It means that current position is defined with the input location number.

It is possible to teach position in Cartesian or Joint space depending on the Joint Coordinate Input.

When the teaching is done the current robot position and configuration is also recorded in the output structure called Motion Record.

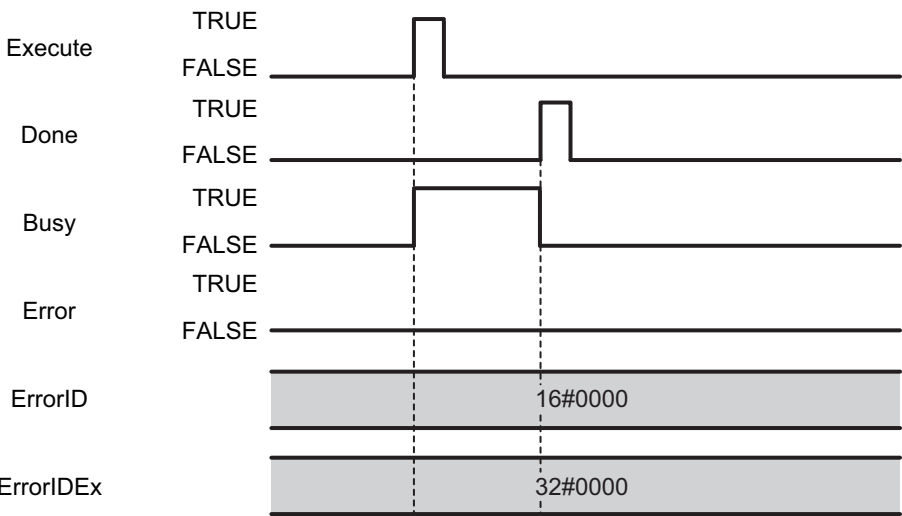
Taught positions must be downloaded each time the Adept controller is restarted.

MotionRecord output is set to its default value when instruction is executed and it is updated with the recorded values when the instruction is completed.



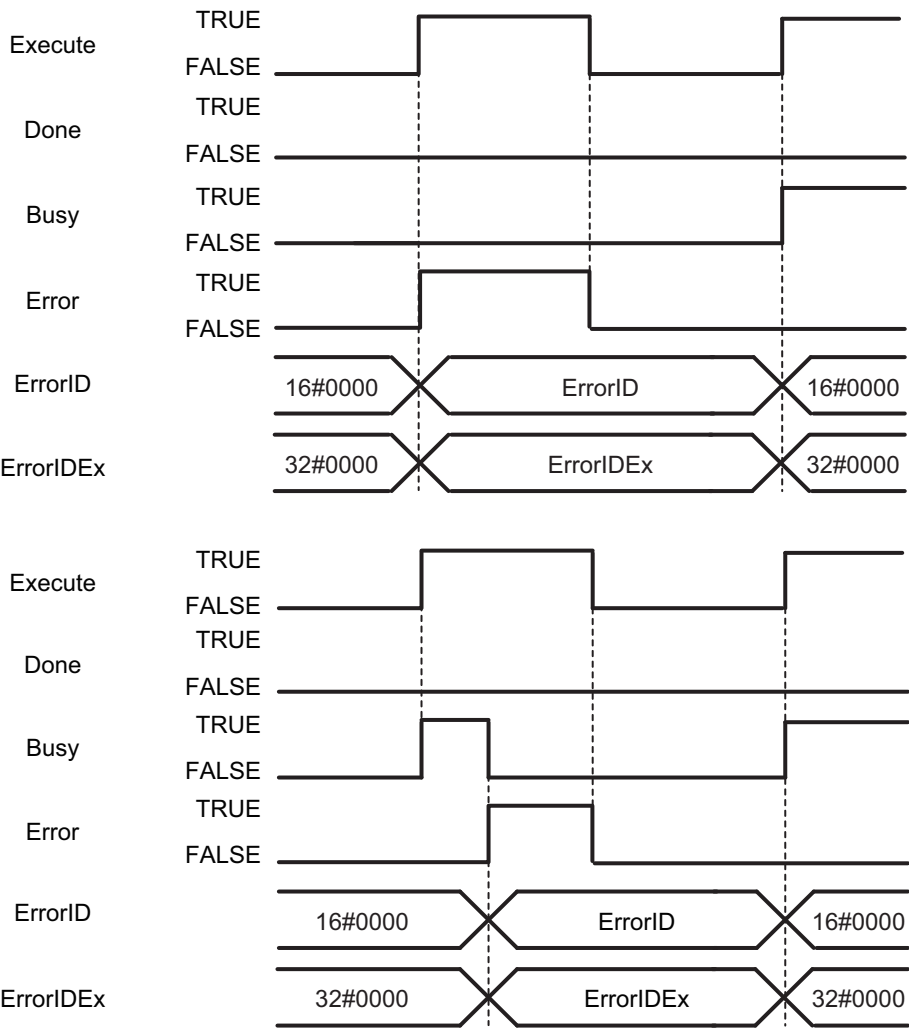
Timing charts

● Normal execution



● Error execution

If an error occurs during instruction execution, *Error* will change to TRUE. You can find out the cause of the error by referring to the value output by *ErrorID* and *ErrorIDEx*.



## Troubleshooting

Error code	Extended error ID	Status	Meaning	Correction
16#0000	16#00000000	The service ended normally.	---	---
16#3C38	16#00000001	Robot controller error	Error occurs in the robot controller.	Check the RobotError variable output in ARB_RobotControl and perform corrections for the error.
	16#00000002	Communication error	Ethernet/IP communication error	Execute the instruction after the successful EtherNet/IP communication recovery.
	16#00000003	Invalid operation	The instruction could not be executed in this controller state. <i>See Execution Conditions Table on page 143.</i>	Execute the instruction after the function block is ready for execution.
	16#00000004	Re-execution disabled	Instruction cannot be re-executed	Wait for the Done variable output changes to TRUE and then execute the function block again.
	16#00000005	Multi-execution disabled	Multiple instructions that cannot be executed simultaneously were executed for the same target.	Wait for the Done variable output in the previously executed instructions changes to TRUE and then execute the function block.
	16#00000006	Location Number Out of Range	The value specified the input variable LocationNumber is out of range.	Set the LocationNumber within range.

# ARB\_InputOutputSignals

The scope of the ARB\_InputOutputSingals instruction is to communicate with the robot controller through its digital input and outputs.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_InputOutputSignals	Input Output Signals	FB	<div>ARB_InputOutputSignals_instance</div> <div><div><div>\\OmronLib\EIP_Adept</div><div>VARB_InputOutputSignals</div></div><div><div>RobotData</div><div>-----</div><div>RobotData</div></div><div><div>Enable</div><div></div><div>Enabled</div></div><div><div>ControlOutputs</div><div></div><div>Busy</div></div><div><div></div><div></div><div>Error</div></div><div><div></div><div></div><div>ErrorID</div></div><div><div></div><div></div><div>ErrorIDEx</div></div><div><div></div><div></div><div>ReadInputs</div></div></div>	ARB_InputOutputSignals_instance ( RobotData:=parameter, Enable:=parameter, ControlOutputs:=parameter, Enabled =>parameter, Busy =>parameter, Error =>parameter, ErrorID=>parameter, ErrorIDEx=>parameter, ReadInputs=>parameter )

ARB\_InputOutputSignals\_instance  
(  
  RobotData:=parameter,  
  Enable:=parameter,  
  ControlOutputs:=parameter,  
  Enabled =>parameter,  
  Busy =>parameter,  
  Error =>parameter,  
  ErrorID=>parameter,  
  ErrorIDEx=>parameter,  
  ReadInputs=>parameter  
);

## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Enable	Enable	BOOL	TRUE or FALSE	FALSE	The instruction is activated when Enable is TRUE.  In this moment it reads/writes the inputs/outputs from/to the controller.
ControlOutputs	Signals written to the outputs of the robot controller.	sARB_OUTPUTSIGNALS_REF	---	All FALSE	A structure that represents all the digital outputs.

#### ● sARB\_OUTPUTSIGNALS\_REF

Member Variable	Meaning	Data type	Valid range	Description
Output1	Output Signal 1	BOOL	TRUE or FALSE	Digital Output Signal 1
Output2	Output Signal 2	BOOL	TRUE or FALSE	Digital Output Signal 2
Output3	Output Signal 3	BOOL	TRUE or FALSE	Digital Output Signal 3
Output4	Output Signal 4	BOOL	TRUE or FALSE	Digital Output Signal 4
Output5	Output Signal 5	BOOL	TRUE or FALSE	Digital Output Signal 5
Output6	Output Signal 6	BOOL	TRUE or FALSE	Digital Output Signal 6
Output7	Output Signal 7	BOOL	TRUE or FALSE	Digital Output Signal 7
Output8	Output Signal 8	BOOL	TRUE or FALSE	Digital Output Signal 8
Output3001	Output Signal 3001	BOOL	TRUE or FALSE	Digital Output Signal 3001
Output3002	Output Signal 3002	BOOL	TRUE or FALSE	Digital Output Signal 3002
Output3003	Output Signal 3003	BOOL	TRUE or FALSE	Digital Output Signal 3003
Output3004	Output Signal 3004	BOOL	TRUE or FALSE	Digital Output Signal 3004
OutReserved1	Reserved bit 1	BOOL	TRUE or FALSE	---
OutReserved2	Reserved bit 2	BOOL	TRUE or FALSE	---
OutReserved3	Reserved bit 3	BOOL	TRUE or FALSE	---
OutReserved4	Reserved bit 4	BOOL	TRUE or FALSE	---

## Output Variables

Name	Meaning	Data type	Valid range	Description
Enabled	Done	BOOL	TRUE or FALSE	TRUE when the instruction is started.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error ID	WORD	*1	Contains the error code when an error occurs.
ErrorIDEx	Extended error ID	DWORD	*1	Specify extended error number.
ReadInputs	Signals read from the inputs of the robot controller.	sARB_INPUTSIGNALS_REF	---	A structure that represent all hardware and software inputs.

\*1. Refer to *Troubleshooting* on page 98 for details.

### ● sARB\_INPUTSIGNALS\_REF

Member Variable	Meaning	Data type	Valid range	Description
Input1001	Input Signal 1001	BOOL	TRUE or FALSE	Digital Input Signal 1001
Input1002	Input Signal 1002	BOOL	TRUE or FALSE	Digital Input Signal 1002
Input1003	Input Signal 1003	BOOL	TRUE or FALSE	Digital Input Signal 1003
Input1004	Input Signal 1004	BOOL	TRUE or FALSE	Digital Input Signal 1004
Input1005	Input Signal 1005	BOOL	TRUE or FALSE	Digital Input Signal 1005
Input1006	Input Signal 1006	BOOL	TRUE or FALSE	Digital Input Signal 1006
Input1007	Input Signal 1007	BOOL	TRUE or FALSE	Digital Input Signal 1007
Input1008	Input Signal 1008	BOOL	TRUE or FALSE	Digital Input Signal 1008
Input1009	Input Signal 1009	BOOL	TRUE or FALSE	Digital Input Signal 1009
Input1010	Input Signal 1010	BOOL	TRUE or FALSE	Digital Input Signal 1010
Input1011	Input Signal 1011	BOOL	TRUE or FALSE	Digital Input Signal 1011
Input1012	Input Signal 1012	BOOL	TRUE or FALSE	Digital Input Signal 1012
Input2001	Input Signal 2001	BOOL	TRUE or FALSE	Digital Input Signal 2001
Input2002	Input Signal 2002	BOOL	TRUE or FALSE	Digital Input Signal 2002
Input2003	Input Signal 2003	BOOL	TRUE or FALSE	Digital Input Signal 2003
Input2004	Input Signal 2004	BOOL	TRUE or FALSE	Digital Input Signal 2004

## ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Enabled	<ul style="list-style-type: none"> <li>When <i>Enable</i> changes to TRUE.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Enable</i> changes to FALSE.</li> </ul>
Busy	<ul style="list-style-type: none"> <li>When <i>Enable</i> changes to TRUE.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Enable</i> changes to FALSE.</li> <li>When <i>Error</i> changes to TRUE.</li> </ul>
Error	<ul style="list-style-type: none"> <li>When there is an error in the execution conditions or input parameters for the instruction.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Enable</i> is FALSE</li> </ul>

## In-out variables

---

Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOT_DATA_REF	---	Specify data needed for the robot controller.



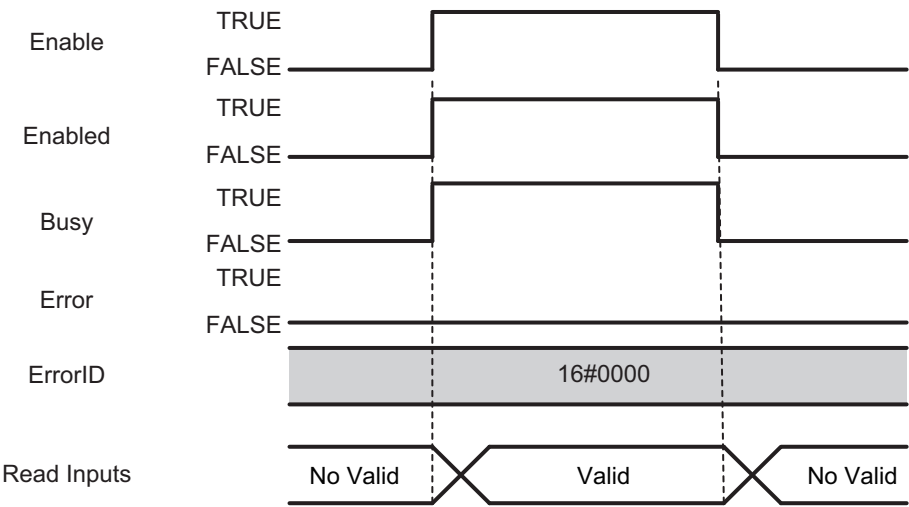
## Function

This instruction writes the value of its *ControlOutputs* variable to the robot controller outputs and it reads the robot controller inputs in its *ReadInputs* variable.

Note When the function block is disabled the signals outputs in the robot controller keep the latest state.

Timing charts

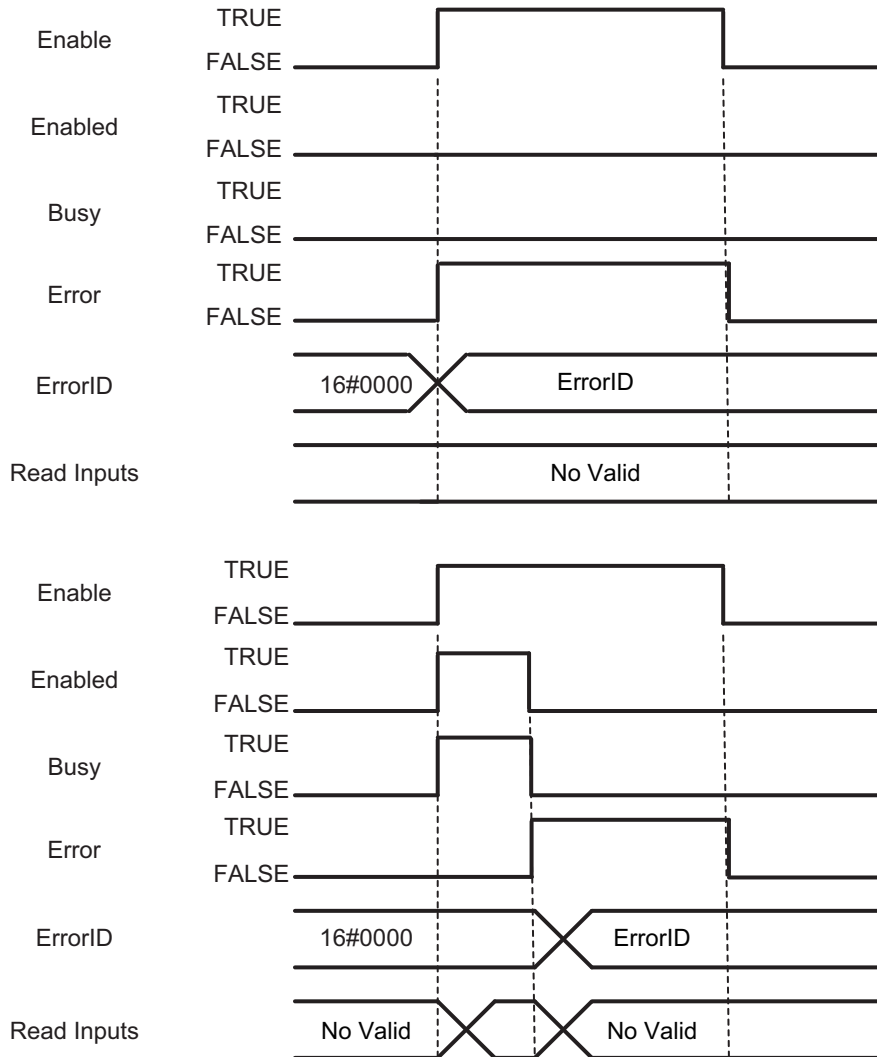
● Normal execution



## ● Error execution

If an error occurs during instruction execution, *Error* will change to TRUE. You can find out the cause of the error by referring to the value output by *ErrorID*.

This function block is only affected by the connection error. All other errors will not stop the execution of this FB.

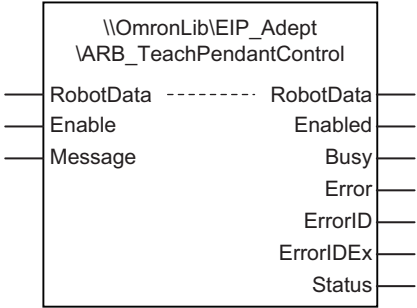


## Troubleshooting

Error code	Extended error ID	Status	Meaning	Correction
16#0000	16#00000000	The service ended normally.	---	---
16#3C39	16#00000001	Robot controller error	Error occurs in the robot controller.	Check the RobotError variable output in ARB_RobotControl and perform corrections for the error.
	16#00000002	Communication error	Ethernet/IP communication error	Execute the instruction after the successful EtherNet/IP communication recovery.

# ARB\_TeachPendantControl

The ARB\_TeachPendantControl instruction is used to send and receive information from the manual control pendant attached to the robot controller.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_TeachPendantControl	Teach Pendant Control	FB		<pre>ARB_TeachPendantControl_instance(   RobotData:=parameter,   Enable:=parameter,   Message:=parameter,   Enabled=&gt;parameter,   Busy=&gt;parameter,   Error=&gt;parameter,   ErrorID=&gt;parameter,   ErrorIDEx=&gt;parameter,   Status=&gt;parameter );</pre>

## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Enable	Enable	BOOL	TRUE or FALSE	FALSE	The instruction is executed while <i>Enable</i> is TRUE.  In this moment it reads/writes the inputs/outputs from/to the controller.
Message	Message	STRING[82]	---		Text to be written in the screen of the teach pendant.

### Output Variables

Name	Meaning	Data type	Valid range	Description
Enabled	Done	BOOL	TRUE or FALSE	TRUE when the instruction is started and teach pendant is connected.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs.
ErrorIDEx	Extended error ID	DWORD	*1	Specify extended error number.
Status		sARB_TEACHPENDANT_STATUS_REF	---	

\*1. Refer to *Troubleshooting* on page 106 for details.

## ● sARB\_TEACHPENDANT\_STATUS\_REF

Member Variable	Meaning	Data type	Valid range	Description
Speed	Speed Status	INT	Positive number, Negative number, or 0	Status of Teach Pendant Speed potentiometer
ButtonF1	Button F1 Status	BOOL	TRUE or FALSE	Status of Teach Pendant Button F1
ButtonF2	Button F2 Status	BOOL	TRUE or FALSE	Status of Teach Pendant Button F2
ButtonF3	Button F3 Status	BOOL	TRUE or FALSE	Status of Teach Pendant Button F3
ButtonF4	Button F4 Status	BOOL	TRUE or FALSE	Status of Teach Pendant Button F4
ButtonMenu	Button Menu	BOOL	TRUE or FALSE	Status of Teach Pendant Button Menu
ButtonOk	Button Ok	BOOL	TRUE or FALSE	Status of Teach Pendant Button Ok
ButtonCancel	Button Cancel	BOOL	TRUE or FALSE	Status of Teach Pendant Button Cancel
ButtonUp	Button Up	BOOL	TRUE or FALSE	Status of Teach Pendant Button Up
ButtonDown	Button Down	BOOL	TRUE or FALSE	Status of Teach Pendant Button Down
ButtonLeft	Button Left	BOOL	TRUE or FALSE	Status of Teach Pendant Button Left
ButtonRight	Button Right	BOOL	TRUE or FALSE	Status of Teach Pendant Button Right
ButtonSelectRobot	Button Select Robot	BOOL	TRUE or FALSE	Status of Teach Pendant Button Select Robot
ButtonJogMode	Button JogMode	BOOL	TRUE or FALSE	Status of Teach Pendant Button JogMode
ButtonSlow	Button Slow	BOOL	TRUE or FALSE	Status of Teach Pendant Button Slow
ButtonSpeedMinus	Button Speed Minus	BOOL	TRUE or FALSE	Status of Teach Pendant Button Speed Minus
ButtonSpeedPlus	Button Speed Plus	BOOL	TRUE or FALSE	Status of Teach Pendant Button Speed Plus
ButtonPosXOrJ1	Button Positive J1 Or X	BOOL	TRUE or FALSE	Status of Teach Pendant Button Positive J1 Or X
ButtonNegXOrJ1	Button Negative J1 Or X	BOOL	TRUE or FALSE	Status of Teach Pendant Button Negative J1 Or X
ButtonPosYOrJ2	Button Positive J2 Or Y	BOOL	TRUE or FALSE	Status of Teach Pendant Button Positive J2 Or Y
ButtonNegYOrJ2	Button Negative J2 Or Y	BOOL	TRUE or FALSE	Status of Teach Pendant Button Negative J2 Or Y
ButtonPosZOrJ3	Button Positive J3 Or Z	BOOL	TRUE or FALSE	Status of Teach Pendant Button Positive J3 Or Z
ButtonNegZOrJ3	Button Negative J3 Or Z	BOOL	TRUE or FALSE	Status of Teach Pendant Button Negative J3 Or Z
ButtonPosRxOrJ4	Button Positive J4 Or Rx	BOOL	TRUE or FALSE	Status of Teach Pendant Button Positive J4 Or Rx
ButtonNegRxOrJ4	Button Negative J4 Or Rx	BOOL	TRUE or FALSE	Status of Teach Pendant Button Negative J4 Or Rx
ButtonPosRyOrJ5	Button Positive J5 Or Ry	BOOL	TRUE or FALSE	Status of Teach Pendant Button Positive J5 Or Ry

Member Variable	Meaning	Data type	Valid range	Description
ButtonNegRyOrJ5	Button Negative J5 Or Ry	BOOL	TRUE or FALSE	Status of Teach Pendant Button Negative J5 Or Ry
ButtonPosRzOrJ6	Button Positive J6 Or Rz	BOOL	TRUE or FALSE	Status of Teach Pendant Button Positive J6 Or Rz
ButtonNegRzOrJ6	Button Negative J6 Or Rz	BOOL	TRUE or FALSE	Status of Teach Pendant Button Negative J6 Or Rz

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Enabled	<ul style="list-style-type: none"> <li>When <i>Enable</i> changes to TRUE and the manual control pendant is attached to the robot controller.</li> </ul>	<ul style="list-style-type: none"> <li>When instruction is active and <i>Enable</i> changes to FALSE.</li> <li>When an mcp connection error occurs</li> </ul>
Busy	<ul style="list-style-type: none"> <li>When <i>Enable</i> changes to TRUE.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Enable</i> changes to FALSE.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>
Error	<ul style="list-style-type: none"> <li>When there is an error in the execution conditions or input parameters for the instruction.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Enable</i> is FALSE</li> </ul>

## In-out variables

---

Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOT_DATA_REF	---	Specify data needed for the robot controller.



## Function

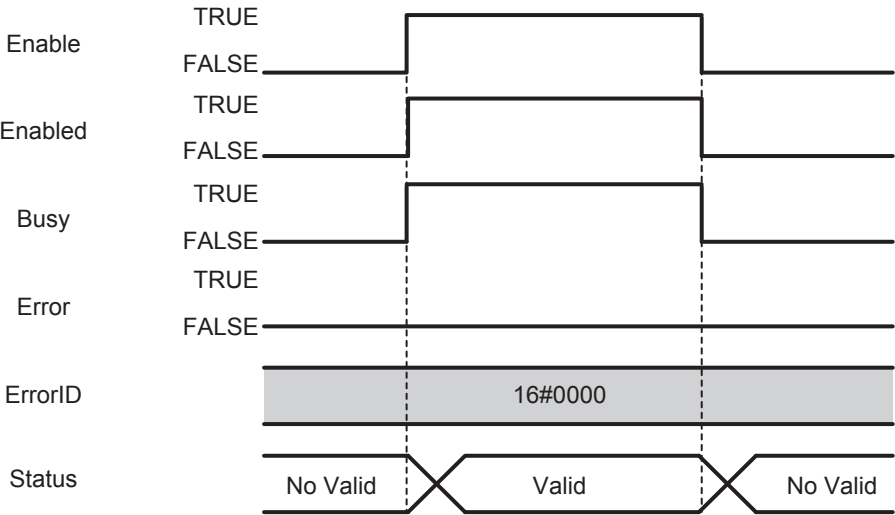
- This instruction is used to switch on/off leads in the manual control pendant, send a text message and receive if any button is pressed. Therefore, it only works when there is a manual control pendant attached to the robot controller.
- It applies to the Adept T20 pendant which is used with the Adept SmartController EX. It does not apply to the T1 or T2 pendants (SmartController CX).

Note When the function block is disabled the leds in the teach pendant keep the latest state.

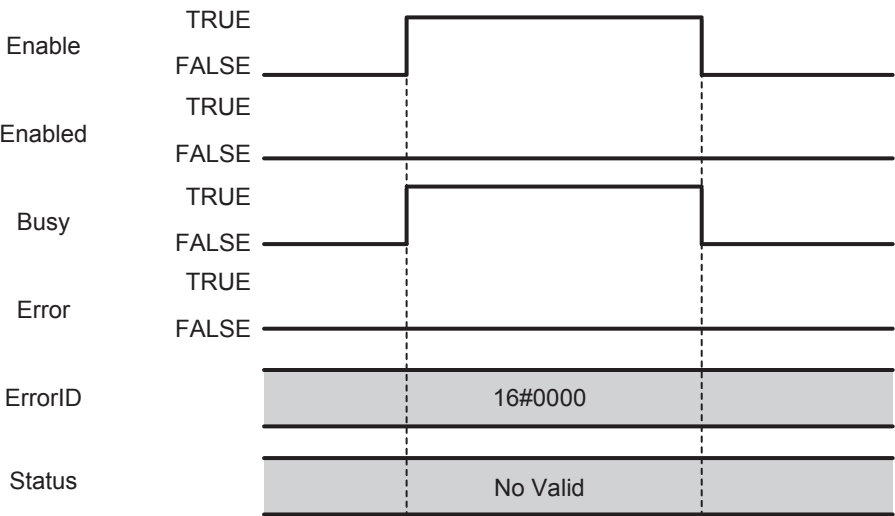
Timing charts

● Normal execution

- The manual control pendant (MCP) is attached to the robot controller.



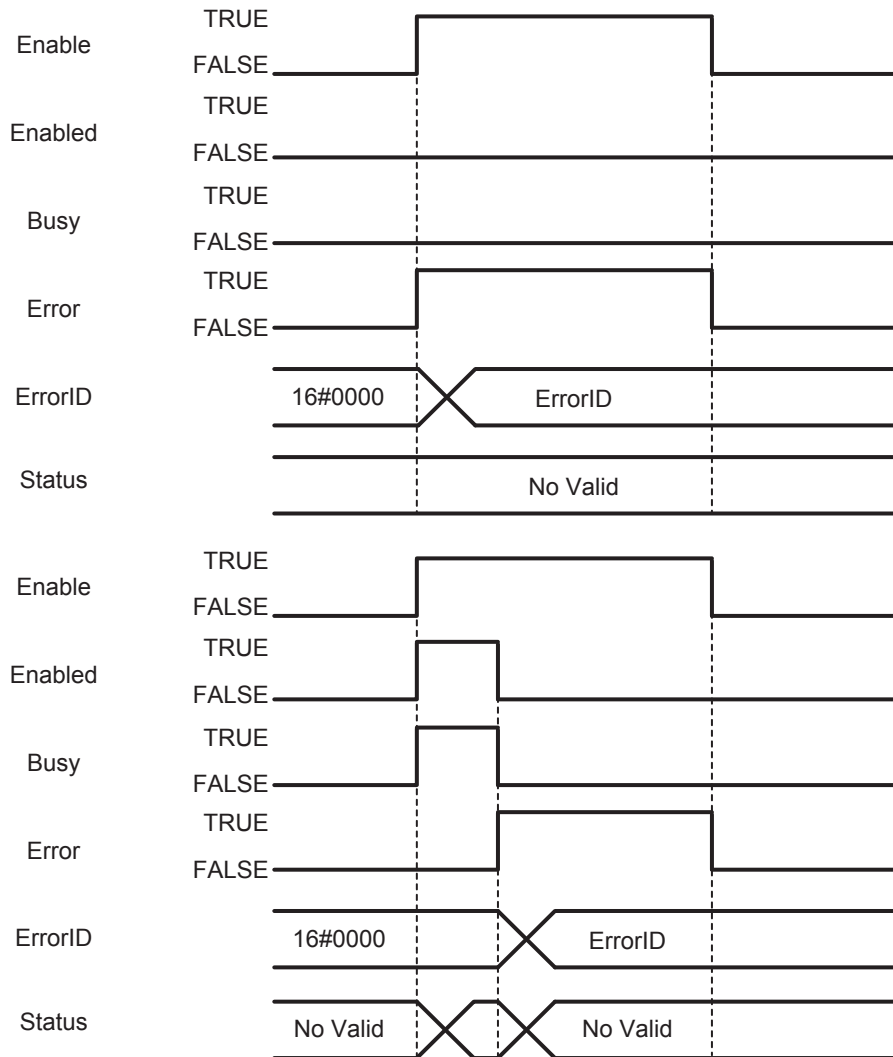
- The manual control pendant (MCP) is not attached to the robot controller.



## ● Error execution

If an error occurs during instruction execution, Error will change to TRUE. You can find out the cause of the error by referring to the value output by ErrorID.

This function block is only affected by the connection error. All other errors will not stop the execution of this FB.



## Troubleshooting

Error code	Extended error ID	Status	Meaning	Correction
16#0000	16#00000000	The service ended normally.	---	---
16#3C3A	16#00000001	Robot controller error	Error occurs in the robot controller.	Check the RobotError variable output in ARB_RobotControl and perform corrections for the error.
	16#00000002	Communication error	Ethernet/IP communication error	Execute the instruction after the successful EtherNet/IP communication recovery.

# ARB\_ReadLatch

The ARB\_ReadLatch instruction is used to get robot the Cartesian position of the robot when an external trigger is raised.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_ReadL atch	Read Latch	FB	<div>ARB_ReadLatch_instance</div> <div><div><div>\\OmronLib\EIP_Adept</div><div>ARB_ReadLatch</div></div><div><div>RobotData</div><div>-----</div><div>RobotData</div></div><div><div>Enable</div><div></div><div>Enabled</div></div><div><div>ReadLatch</div><div></div><div>Busy</div></div><div><div></div><div></div><div>Error</div></div><div><div></div><div></div><div>ErrorID</div></div><div><div></div><div></div><div>ErrorIDEx</div></div><div><div></div><div></div><div>IsLatched</div></div><div><div></div><div></div><div>Position</div></div></div>	ARB_ReadLatch_instance( RobotData:=parameter, Enable:=parameter, ReadLatch:=parameter, Enabled =>parameter, Busy =>parameter, Error =>parameter, ErrorID=>parameter, ErrorIDEx=>parameter, IsLatched=>parameter, Position=>parameter );

ARB\_ReadLatch\_instance(  
RobotData:=parameter,  
Enable:=parameter,  
ReadLatch:=parameter,  
Enabled =>parameter,  
Busy =>parameter,  
Error =>parameter,  
ErrorID=>parameter,  
ErrorIDEx=>parameter,  
IsLatched=>parameter,  
Position=>parameter  
);

## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Enable	Enable	BOOL	TRUE or FALSE	FALSE	The instruction is executed while Enable is to TRUE.
ReadLatch	Read latch command	BOOL	TRUE or FALSE	FALSE	Activate the read latch command

### Output Variables

Name	Meaning	Data type	Valid range	Description
Enabled	Done	BOOL	TRUE or FALSE	TRUE when the instruction is started.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is active.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs.
ErrorIDEx	Extended error ID	DWORD	*1	Specify extended error number.
IsLatched	Latched	BOOL	TRUE or FALSE	TRUE in case that latch signal was raised in the robot controller, otherwise it is FALSE. otherwise it is FALSE.
Position	Latched Position	ARRAY [0..5] OF REAL	---	Position of the robot when the latched signal is rised: Position1[0]: X axis coordinate Position1[1]: Y axis coordinate Position1[2]: Z axis coordinate Position1[3]: Yaw angle coordinate Position1[4]: Pitch angle coordinate Position1[5]: Roll angle coordinate

\*1. Refer to *Troubleshooting* on page 113 for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Enabled	• When <i>Enable</i> changes to TRUE.	• When <i>Enable</i> changes to FALSE.
Busy	• When <i>Enable</i> changes to TRUE.	• When <i>Enable</i> changes to FALSE. • When <i>Error</i> changes to TRUE.
Error	• When there is an error in the execution conditions or input parameters for the instruction.	• When <i>Enable</i> is FALSE

## In-out variables

Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOT_DATA_REF	---	Specify data needed for the robot controller.

## Function

- This function returns the Cartesian position of the robot when an external latch signal is raised; this signal is configured in the robot controller.
- When the input ReadLatch is TRUE and an external latch signal is raised the IsLatched output is set to TRUE and the Position output could be read. IsLatched stays TRUE while input ReadLatch is TRUE. Meanwhile all latch signals are detected and the output LatchPosition is updated with the robot position related to the last latch signal.
- When enabling the FB (Enable set to TRUE), the previous latches stored in the Robot controller pile will be cleared automatically. When the Enabled output is TRUE the latch clearing will be completed and then latch inputs using the FB is possible. Clearing the latches can take several RPI. Once the clearing process has been completed, the first latch received (IsLatched=TRUE) is the last latch signal received in the Robot controller.
- Latch Position output is only updated when the IsLatched signal is received (value equal to TRUE)

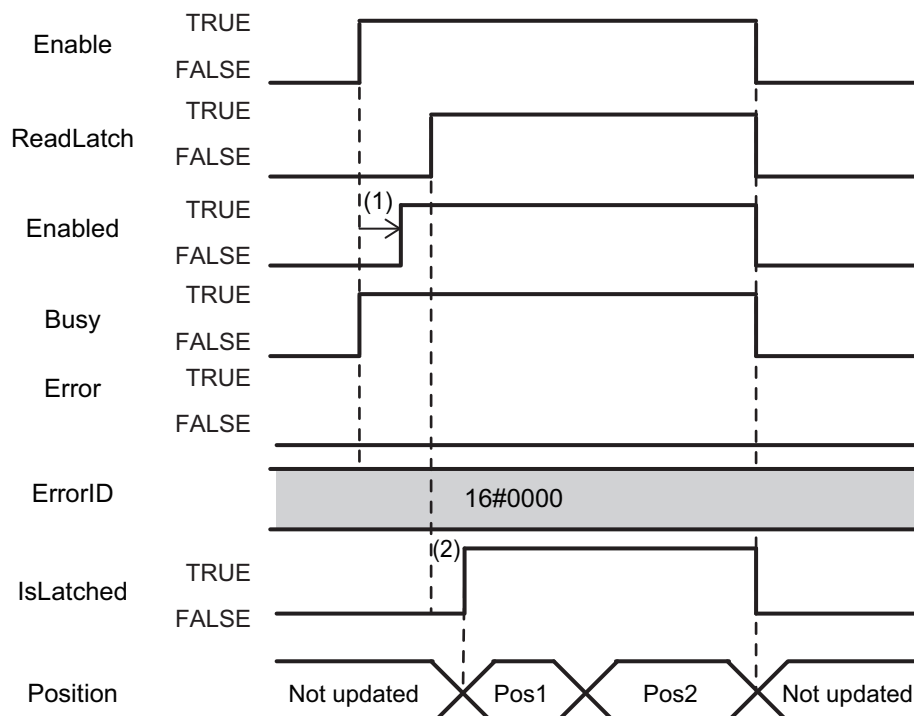
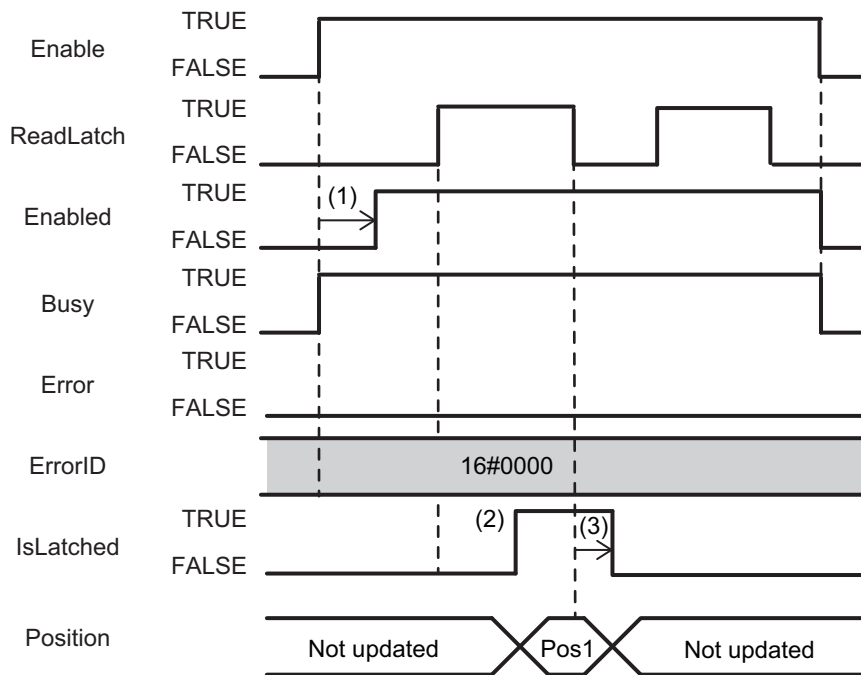
## Additional Information

Due to the clearing process, user can use only one instance of the FB with the same robot. If 2 or more instances are used, the FBs might hang in the cleaning process forever.



## Timing charts

### ● Normal execution



(1): Time to clear the previous latches

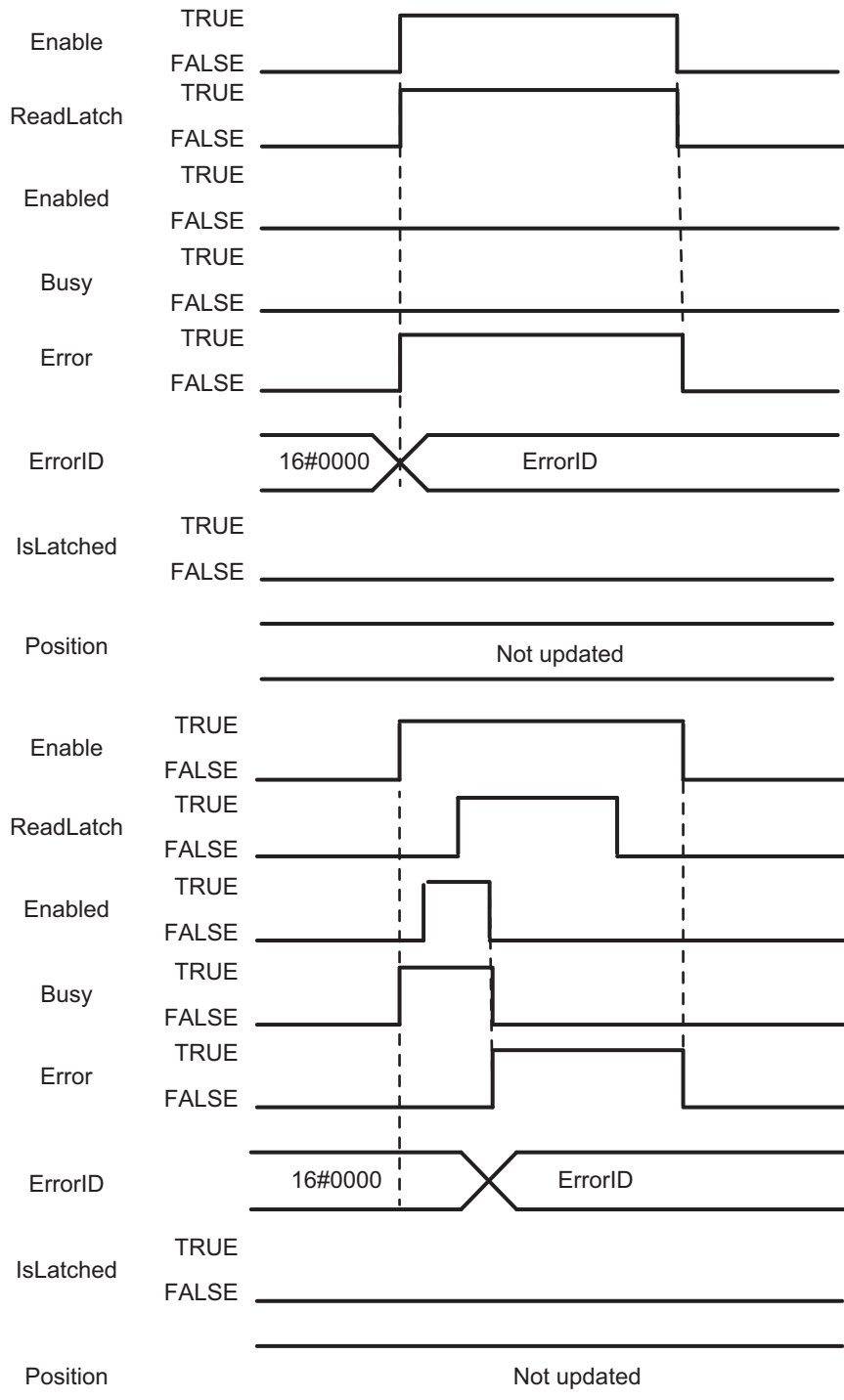
(2): Latch signal is received

(3): 1 RPI between ReadLatch input is reset and IsLatched output is reset

## ● Error execution

If an error occurs during instruction execution, *Error* will change to TRUE. You can find out the cause of the error by referring to the value output by *ErrorID*.

This function block is only affected by the connection error. All other errors will not stop the execution of this FB.



## Troubleshooting

Error code	Extended error ID	Status	Meaning	Correction
16#0000	16#00000000	The service ended normally.	---	---
16#3C3B	16#00000001	Robot controller error	Error occurs in the robot controller.	Check the RobotError variable output in ARB_RobotControl and perform corrections for the error.
	16#00000002	Communication error	Ethernet/IP communication error	Execute the instruction after the successful EtherNet/IP communication recovery.
	16#00000003	Invalid operation	The instruction could not be executed in this controller state. <i>See Execution Conditions Table on page 143.</i>	Execute the instruction after the function block is ready for execution.

# ARB\_MoveCommand

The ARB\_MoveCommand instruction moves the robot to a target position using a linear movement or a point to point operation depending on the user selection.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_Move Command	MoveCommand	FB	<div>ARB_MoveCommand_instance</div> <div><div><div>\\OmronLib\EIP_Adept</div><div>\\ARB_MoveCommand</div></div><div><div>RobotData</div><div>Execute</div><div>Position</div><div>StraightLineMove</div><div>Blending</div><div>MotionParams</div><div>MoveConfig</div></div><div><div>RobotData</div><div>Done</div><div>Busy</div><div>Error</div><div>ErrorId</div><div>ErrorIdEx</div></div></div>	ARB_MoveCommand_instance( RobotData:=parameter, Execute:=parameter, Position:=parameter, StraightLineMove:=parameter, Blending:=parameter, MotionParams:=parameter, MoveConfig:=parameter, Done =>parameter, Busy =>parameter, Error =>parameter, ErrorId=>parameter, ErrorIdEx=>parameter );

ARB\_MoveCommand\_instance(  
RobotData:=parameter,  
Execute:=parameter,  
Position:=parameter,  
StraightLineMove:=parameter,  
Blending:=parameter,  
MotionParams:=parameter,  
MoveConfig:=parameter,  
Done =>parameter,  
Busy =>parameter,  
Error =>parameter,  
ErrorId=>parameter,  
ErrorIdEx=>parameter  
);

## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.
Position	Position	sARB_MOVE_POSITION_REF	---*1		Sets the parameters used to define the robot target position for the move command.
StraightLineMove	Straight Line movement	BOOL	TRUE or FALSE	FALSE	Defines if the movement will be executed as a linear trajectory in Cartesian space, otherwise as a PTP trajectory.  TRUE: Linear Movement FALSE: PTP Movement
Blending	Blending	BOOL	TRUE or FALSE	FALSE	Enables blending between the executed movement and the next movement executed.  TRUE: Blending enabled (Continuous Path) FALSE: No blending enabled
MotionParams	Motion parameters	sARB_MOTION_PARAMS_REF	---*1	---	Set the motion parameters.
MoveConfig	Motion Configuration	sARB_MOVE_CONFIG_REF	---*1	---	Sets the motion configuration settings.

\*1. See structure definition

## Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged by the robot controller.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is being executed.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error ID	WORD	*1	Contains the error code when an error occurs.
ErrorIDEx	Extended error ID	DWORD	*1	Specify extended error number.

\*1. Refer to *Troubleshooting* on page 120 for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	<ul style="list-style-type: none"> <li>When instruction is acknowledge by the robot controller.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to FALSE.</li> <li>After one control period when <i>Execute</i> is FALSE.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>
Busy	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to TRUE.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>
Error	<ul style="list-style-type: none"> <li>When there is an error in the execution conditions or input parameters for the instruction.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to FALSE.</li> </ul>

## In-Out Variables

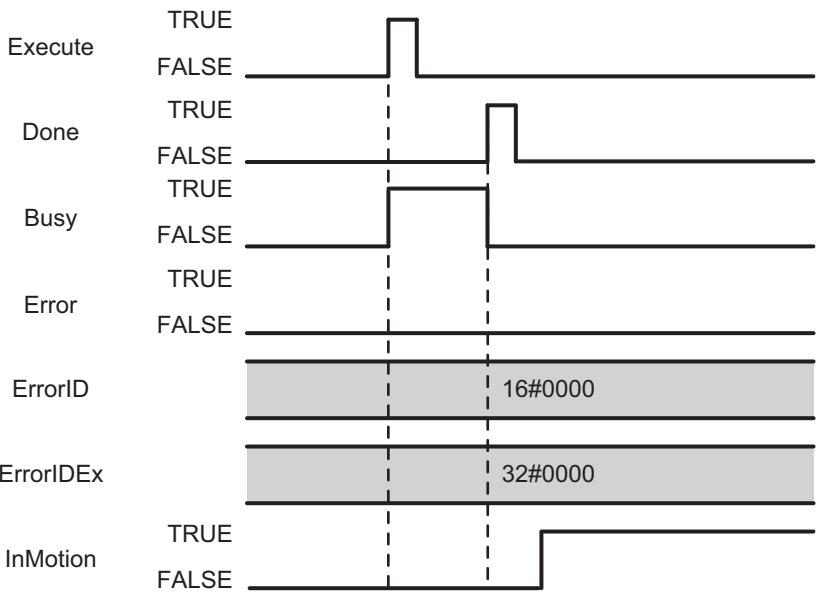
Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOT_DATA_REF	---	Specify data needed for the robot controller.

## Function

- This instruction moves the robot from the current position to the target position specified in the position input variable.
- If the StraightLineMov input variable is TRUE then the movement will be done using a linear movement, otherwise it will execute a point to point operation.
- Changing the arm configuration (Righty, Flip and Below configurations) is only effective in the robot when working in Cartesian coordinates and the linear movement has not been selected (PTP movement).
- The configuration in the Relative input variable can be overridden depending on the selection of the approach height (absolute or relative).
- Target position is provided by the Position input:
  - In case LocationNumber is higher than zero, this pre-defined position will be used.
  - In case LocationNumber is zero and PalletNumber is different than zero, the movement will be relative to the pallet. PalletNumber, PalletIndex and Position array will be used.
  - In the case that LocationNumber and PalletNumber are both equal to zero. Position array will be used as target position.
- Once the command has been completed (Done output is TRUE) another command can be executed. If the Blending input of the first one is TRUE, the second command will be blended before the first one achieves its target position. If not, the first command will achieve its target position and later the second command will start.
- Cancel Motion input can stop the movement once it has been commanded.

Timing charts

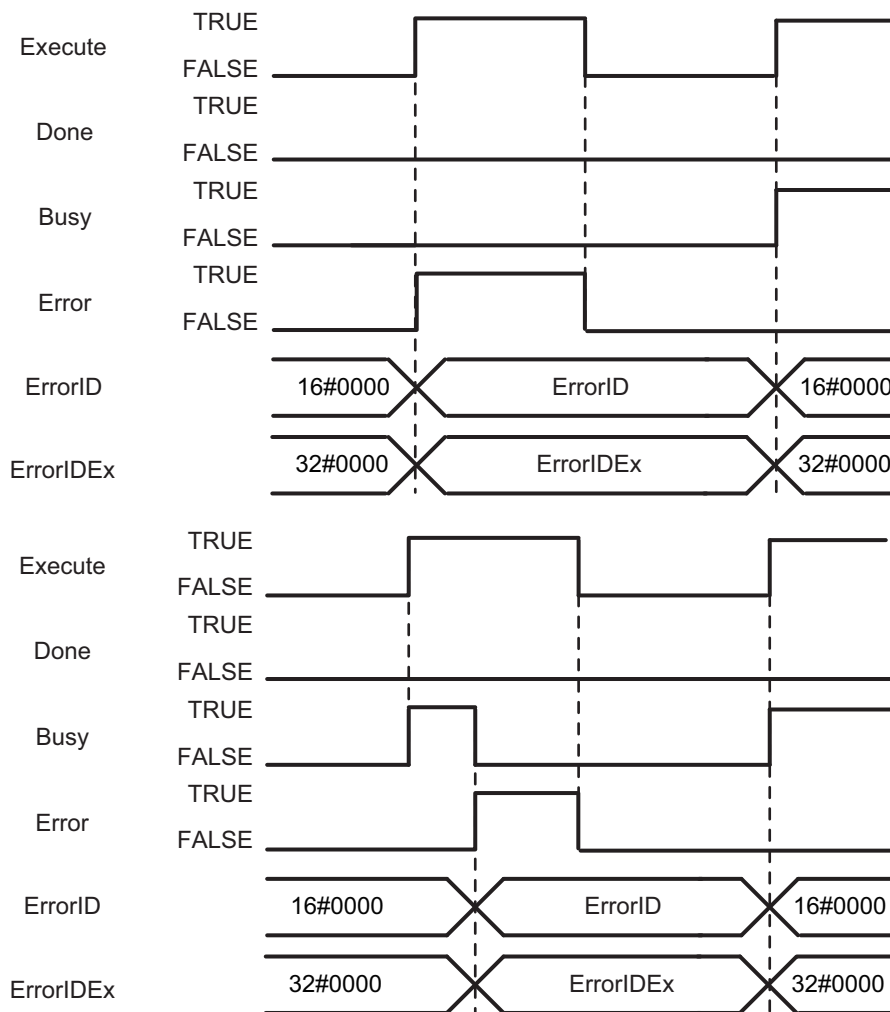
● Normal execution





## ● Error execution

If an error occurs during instruction execution, *Error* will change to TRUE. You can find out the cause of the error by referring to the value output by *ErrorID* and *ErrorIDEx*.



## Troubleshooting

Error code	Extended error ID	Status	Meaning	Correction
16#0000	16#00000000	The service ended normally.	---	---
16#3C3C	16#00000001	Robot controller error	Error occurs in the robot controller.	Check the RobotError variable output in ARB_RobotControl and perform corrections for the error.
	16#00000002	Communication error	Ethernet/IP communication error	Execute the instruction after the successful EtherNet/IP communication recovery.
	16#00000003	Invalid operation	The instruction could not be executed in this controller state. <i>See Execution Conditions Table on page 143.</i>	Execute the instruction after the function block is ready for execution.
	16#00000004	Re-execution disabled	Instruction cannot be re-executed	Wait for the Done variable output changes to TRUE and then execute the function block again.
	16#00000005	Multi-execution disabled	Multiple instructions that cannot be executed simultaneously were executed for the same target.	Wait for the Done variable output in the previously executed instructions changes to TRUE and then execute the function block.
	16#00000006	Location Number Out of Range	The value specified for the parameter LocationNumber in the input variable Position is out of range.	Set the LocationNumber within range.
	16#00000007	Pallet Number Out of Range	The value specified for the parameter PalletNumber in the input variable Position is out of range.	Set the PalletNumber within range.
	16#0000000B	Speed Out of Range	The value specified for the parameter Speed in the input variable MotionParams is out of range.	Set the Speed within range.
	16#0000000C	Acceleration Out of Range	The value specified for the parameter Acceleration in the input variable MotionParams is out of range.	Set the Acceleration within range.
	16#0000000D	Deceleration Out of Range	The value specified for the parameter Deceleration in the input variable MotionParams is out of range.	Set the Deceleration within range.
	16#0000000E	Acceleration Type Selection Out of Range	The value specified for the parameter AccelerationType in the input variable MotionParams is out of range.	Set the AccelerationType within range.

Error code	Extended error ID	Status	Meaning	Correction
16#3C3C	16#0000000F	Speed Limit Out of Range	The value specified for the parameter SpeedLimit in the input variable MotionParams is out of range.	Set the SpeedLimit within range.
	16#00000010	Pallet Index Out of Range	The value specified for the parameter PalletIndex in the input variable Position is out of range.	Set the PalletIndex within range.
	16#00000012	Illegal Motion Stop request	Cancel Motion or Stop on Input executed during execution of MoveCommand. Current Movement not started yet, so it cannot be stopped.	Wait for completion of the Current Movement and then execute the instruction again.
	16#00000013	Emergency Stop state	Emergency Stop input has been set during the execution of the instruction	Reset the Emergency Stop input.

# ARB\_PickAndPlaceCommand

The ARB\_PickAndPlaceCommand instruction performs a movement composed by depart, move and approach to a target position.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_PickAndPlaceCommand	PickAndPlace-Command	FB		<pre>ARB_PickAndPlaceCommand_instance(   RobotData:=parameter,   Execute:=parameter,   Position:=parameter,   Blending:=parameter,   MotionParams:=parameter,   MoveConfig:=parameter,   Done =&gt;parameter,   Busy =&gt;parameter,   Error =&gt;parameter,   ErrorId=&gt;parameter,   ErrorIdEx=&gt;parameter );</pre>

## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.
Position	Target position	sARB_MOVE_POSITION_REF	---*1		Sets the parameters used to define the robot target position for the movement command.
Blending	Blending	BOOL	TRUE or FALSE	FALSE	Enables blending between the executed movement and the next movement executed.  TRUE: Blending enabled (Continuous Path) FALSE: No blending enabled
MotionParams	Motion parameters	sARB_MOTION_PARAMS_REF	---*1		Set the motion parameters.
MoveConfig	Motion Configuration	sARB_MOVE_CONFIG_REF	---*1		Sets the motion configuration settings.

\*1. See structure definition

## Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged by the robot controller.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is being executed.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error ID	WORD	*1	Specify error number.
ErrorIDEx	Extended error ID	DWORD	*1	Specify extended error number.

\*1. Refer to *Troubleshooting* on page 127 for details.

### ● Output Variable Update Timing

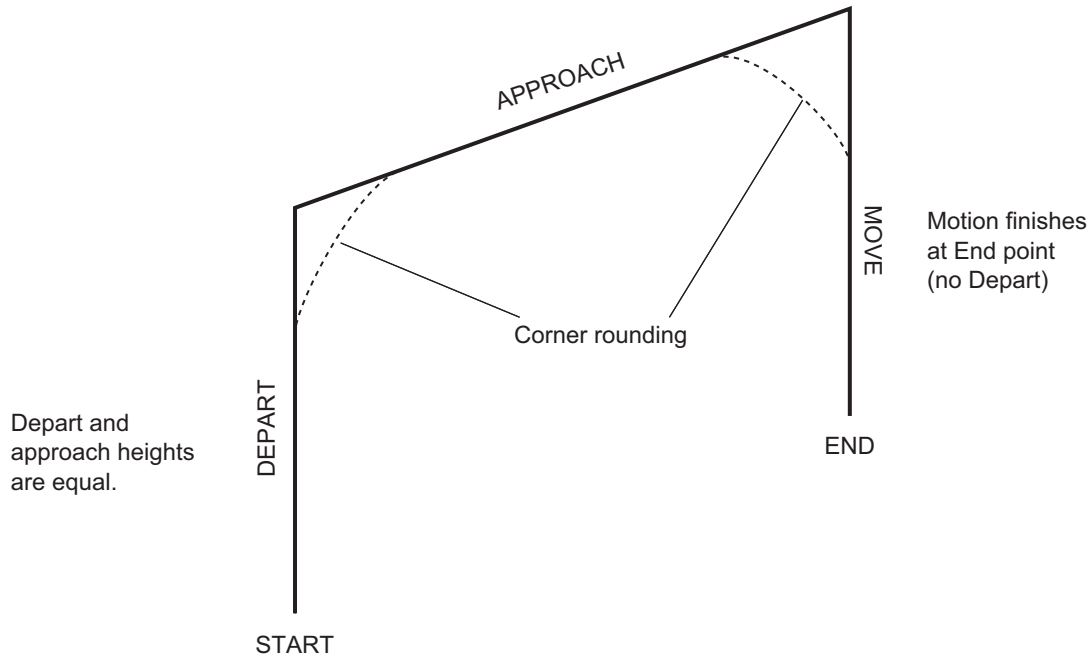
Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	<ul style="list-style-type: none"> <li>When instruction is acknowledge by the robot controller.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to FALSE.</li> <li>After one control period when <i>Execute</i> is FALSE.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>
Busy	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to TRUE.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>
Error	<ul style="list-style-type: none"> <li>When there is an error in the execution conditions or input parameters for the instruction.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to FALSE.</li> </ul>

## In-out variables

Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOT DATA_REF	---	Specify data needed for the robot controller.

## Function

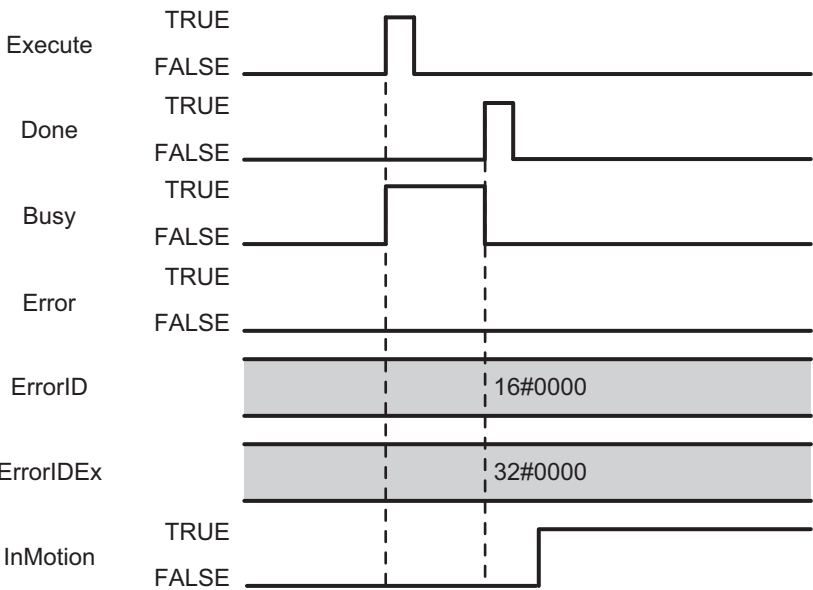
- This instruction moves the robot from the current position to the target position specified in the position input variable executing a three-motion pick-and-place operation as detailed in image below:



- Pick and Place is composed by 3 movements. A Depart from start position along Z axis and Approach to target position along X-Y coordinates and a Move down to end position along Z axis.
- If the Blending input is TRUE the corners will be rounded as shown in the picture above.
- Changing the arm configuration is not possible since Pick and Place is always a linear movement.
- The configuration in the Relative input variable can be overridden depending on the selection of the approach height (absolute or relative).
- The motion to the final position is always nulled following the precision specified in the CoarseNulling input variable inside the MotionConfig structure.
- Target position is provided by the Position input:
  - In case LocationNumber is higher than zero, this pre-defined position will be used.
  - In case LocationNumber is zero and PalletNumber is different than zero, the movement will be relative to the pallet. PalletNumber, PalletIndex and Position array will be used.
  - In the case that LocationNumber and PalletNumber are both equal to zero. Position array will be used as target position.
- Once the command has been completed (Done output is TRUE) another command can be executed. If the Blending input of the first one is TRUE, the second command will be blended.
- Pick and Place command always achieves its target position, even if the blending input is set to TRUE.
- Cancel Motion input can stop the movement once it has been commanded.

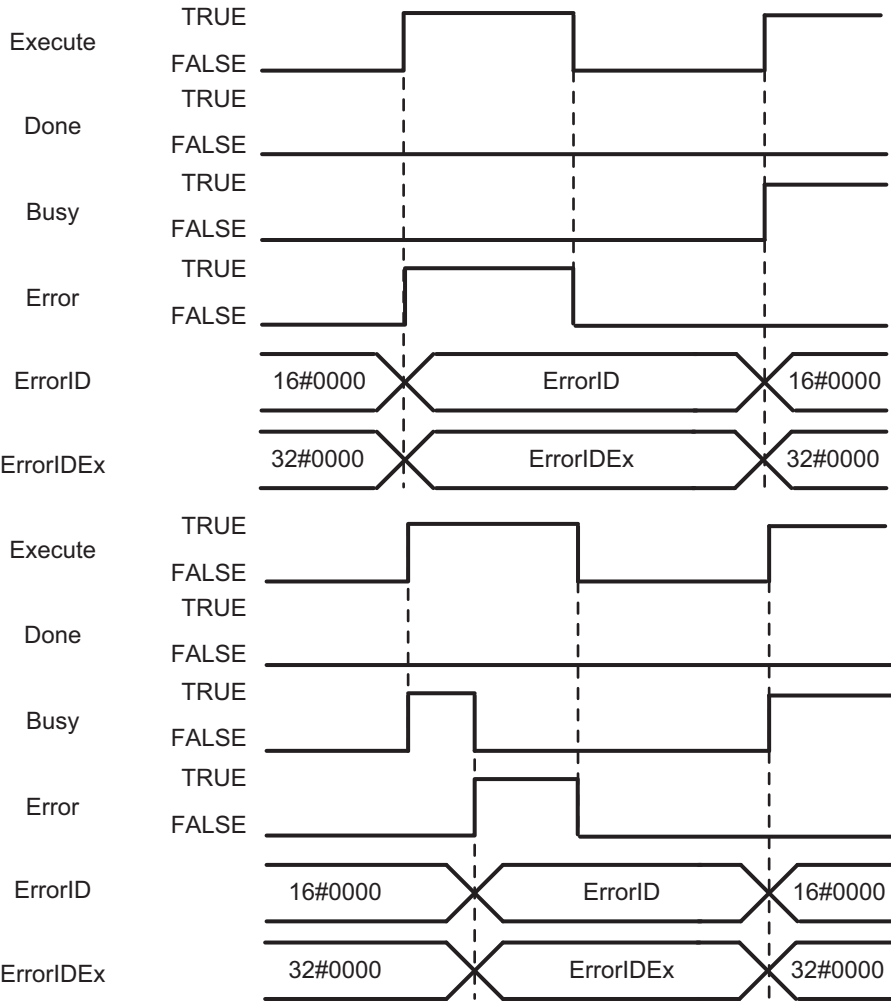
Timing charts

● Normal execution



● Error execution

If an error occurs during instruction execution, Error will change to TRUE. You can find out the cause of the error by referring to the value output by ErrorID and ErrorIDEx.





## Troubleshooting

Error code	Extended error ID	Status	Meaning	Correction
16#0000	16#00000000	The service ended normally.	---	---
16#3C3F	16#00000001	Robot controller error	Error occurs in the robot controller.	Check the RobotError variable output in ARB_RobotControl and perform corrections for the error.
	16#00000002	Communication error	Ethernet/IP communication error	Execute the instruction after the successful EtherNet/IP communication recovery.
	16#00000003	Invalid operation	The instruction could not be executed in this controller state. <i>See Execution Conditions Table on page 143.</i>	Execute the instruction after the function block is ready for execution.
	16#00000004	Re-execution disabled	Instruction cannot be re-executed	Wait for the Done variable output changes to TRUE and then execute the function block again.
	16#00000005	Multi-execution disabled	Multiple instructions that cannot be executed simultaneously were executed for the same target.	Wait for the Done variable output in the previously executed instructions changes to TRUE and then execute the function block.
	16#00000006	Location Number Out of Range	The value specified for the parameter LocationNumber in the input variable Position is out of range.	Set the LocationNumber within range.
	16#00000007	Pallet Number Out of Range	The value specified for the parameter PalletNumber in the input variable Position is out of range.	Set the PalletNumber within range.
	16#0000000B	Speed Out of Range	The value specified for the parameter Speed in the input variable MotionParams is out of range.	Set the Speed within range.
	16#0000000C	Acceleration Out of Range	The value specified for the parameter Acceleration in the input variable MotionParams is out of range.	Set the Acceleration within range.
	16#0000000D	Deceleration Out of Range	The value specified for the parameter Deceleration in the input variable MotionParams is out of range.	Set the Deceleration within range.
	16#0000000E	Acceleration Type Selection Out of Range	The value specified for the parameter AccelerationType in the input variable MotionParams is out of range.	Set the AccelerationType within range.

Error code	Extended error ID	Status	Meaning	Correction
16#3C3F	16#0000000F	Speed Limit Out of Range	The value specified for the parameter SpeedLimit in the input variable MotionParams is out of range.	Set the SpeedLimit within range.
	16#00000010	Pallet Index Out of Range	The value specified for the parameter PalletIndex in the input variable Position is out of range.	Set the PalletIndex within range.
	16#00000012	Illegal Motion Stop request	Cancel Motion or Stop on Input executed during execution of MoveCommand. Current Movement not started yet, so it cannot be stopped.	Wait for completion of the Current Movement and then execute the instruction again.
	16#00000013	Emergency Stop state	Emergency Stop input has been set during the execution of the instruction	Reset the Emergency Stop input.

# ARB\_Jog

The ARB\_Jog instruction moves the specified joint or axis of the robot.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_Jog	Jog	FB		<pre>ARB_Jog_instance( Invalid parameter Enable:=parameter, JogMode:=parameter, PositiveXOrJ1:=parameter, NegativeXOrJ1:=parameter, PositiveYOrJ2:=parameter, NegativeYOrJ2:=parameter, PositiveZOrJ3:=parameter, NegativeZOrJ3:=parameter, PositiveRxOrJ4:=parameter, NegativeRxOrJ4:=parameter, PositiveRyOrJ5:=parameter, NegativeRyOrJ5:=parameter, PositiveRzOrJ6:=parameter, NegativeRzOrJ6:=parameter, MotionParams:=parameter, Enabled =&gt;parameter, Busy =&gt;parameter, Error =&gt;parameter, ErrorID=&gt;parameter, ErrorIDEx=&gt;parameter, Jogging=&gt;parameter );</pre>

## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Enable	Enable	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Enable changes to TRUE.
JogMode	Jog Mode	INT	0 to 2	0	Specify the Jog Mode. 0 = World Mode 1 = Tool Mode 2 = Joint Mode
PositiveXOrJ1	Positive Jog X/J1	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the positive direction of X or Joint 1. When it changes to FALSE, the robot stops in the positive direction of X or Joint 1.
NegativeXOrJ1	Negative Jog X/J1	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the negative direction of X or Joint 1. When it changes to FALSE, the robot stops in the negative direction of X or Joint 1.
PositiveYOrJ2	Positive Jog Y/J2	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the positive direction of Y or Joint 2. When it changes to FALSE, the robot stops in the positive direction of Y or Joint 2.
NegativeYOrJ2	Negative Jog Y/J2	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the negative direction of Y or Joint 2. When it changes to FALSE, the robot stops in the negative direction of Y or Joint 2.
PositiveZOrJ3	Positive Jog Z/J3	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the positive direction of Z or Joint 3. When it changes to FALSE, the robot stops in the positive direction of Z or Joint 3.
NegativeZOrJ3	Negative Jog Z/J3	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the negative direction of Z or Joint 3. When it changes to FALSE, the robot stops in the negative direction of Z or Joint 3.
PositiveRrOrJ4	Positive Jog Rx/J4	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the positive direction of Rx or Joint 4. When it changes to FALSE, the robot stops in the positive direction of Rx or Joint 4.

Name	Meaning	Data type	Valid range	Default	Description
NegativeRxOrJ4	Negative Jog Rx/J4	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the negative direction of Rx or Joint 4. When it changes to FALSE, the robot stops in the negative direction of Rx or Joint 4.
PositiveRyOrJ5	Positive Jog Ry/J5	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the positive direction of Ry or Joint 5. When it changes to FALSE, the robot stops in the positive direction of Ry or Joint 5.
NegativeRyOrJ5	Negative Jog Ry/J5	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the negative direction of Ry or Joint 5. When it changes to FALSE, the robot stops in the negative direction of Ry or Joint 5.
PositiveRzOrJ6	Positive Jog Rz/J6	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the positive direction of Rz or Joint 6. When it changes to FALSE, the robot stops in the positive direction of Rz or Joint 6.
NegativeRzOrJ6	Negative Jog Rz/J6	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the negative direction of Rz or Joint 6. When it changes to FALSE, the robot stops in the negative direction of Rz or Joint 6.
MotionParams	Motion Parameters	sARB_MOTION_PARAMETERS	---	---	Set the motion parameters of jog function

## Output Variables

Name	Meaning	Data type	Valid range	Description
Enabled	Enabled	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged by the robot controller.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the function is being executed.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error ID	WORD	*1	Specify error number.
ErrorIDEx	Extended error ID	DWORD	*1	Specify extended error number.
Jogging	Jogging	BOOL	TRUE or FALSE	TRUE when the robot is jogging

\*1. Refer to *Troubleshooting* on page 136 for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Enabled	<ul style="list-style-type: none"> <li>When instruction is acknowledged by the robot controller.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Enable</i> changes to FALSE.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>
Busy	<ul style="list-style-type: none"> <li>When <i>Enable</i> is TRUE.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Enable</i> changes to FALSE.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>
Error	<ul style="list-style-type: none"> <li>When there is an error in the execution conditions or input parameters for the instruction.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Enable</i> changes to FALSE.</li> </ul>
Jogging	<ul style="list-style-type: none"> <li>When robot is in jog mode state</li> </ul>	<ul style="list-style-type: none"> <li>When robot is not in jog mode state.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>

## In-out variables

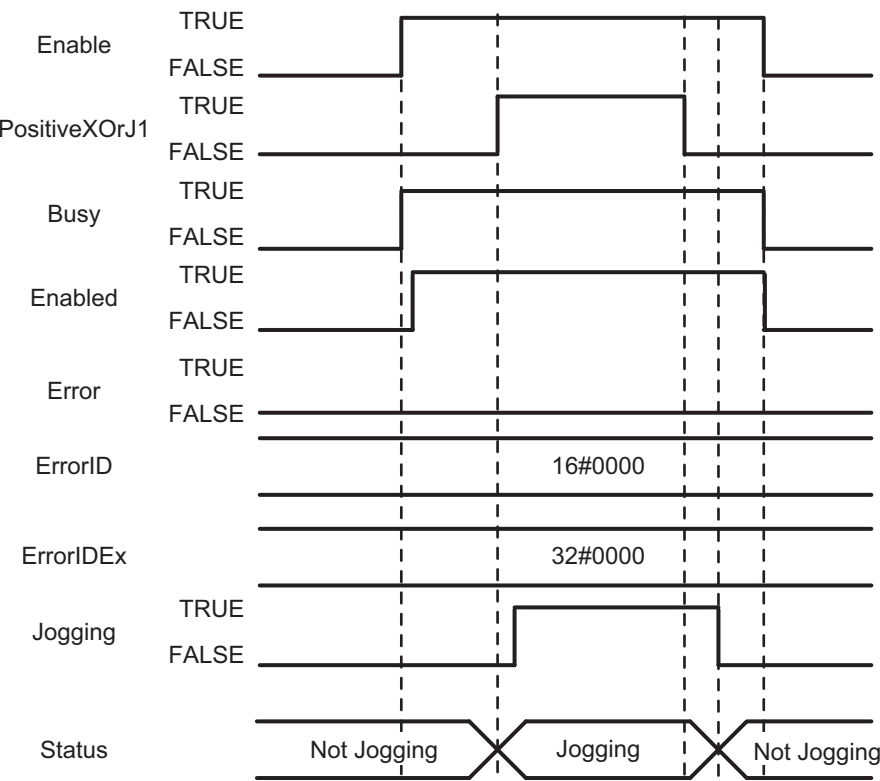
Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOT DATA_REF	---	Specify data needed for the robot controller.

## Function

- The Jog function moves the individual robot axes in jog mode.
- When enable input changes to true the selected jog mode is set (World Mode, Tool Mode or Joint Mode). The point of the robot to be commanded and the coordinate system to base the movement is defined by the Jog Mode.
- If the Jog Mode is set to Joint the jog movement is linked to each robot joint J1 to J6. While, if Jog Mode is set to World or Tool the jog movement is linked to Cartesian coordinates (X, Y, Z) and rotations around them (Rx, Ry, Rz).
- When enable input changes to true and jogging starts to be done in some direction, the movement will continue until enable input changes to false. At that moment, the robot would start to decelerate until it stops.
- Jogging can be only done in one direction at a time.
- Jog command cannot be blended with any other motion.
- Jog command cannot be stopped by the CancelMotion option (see ARB\_RobotControl).

Timing charts

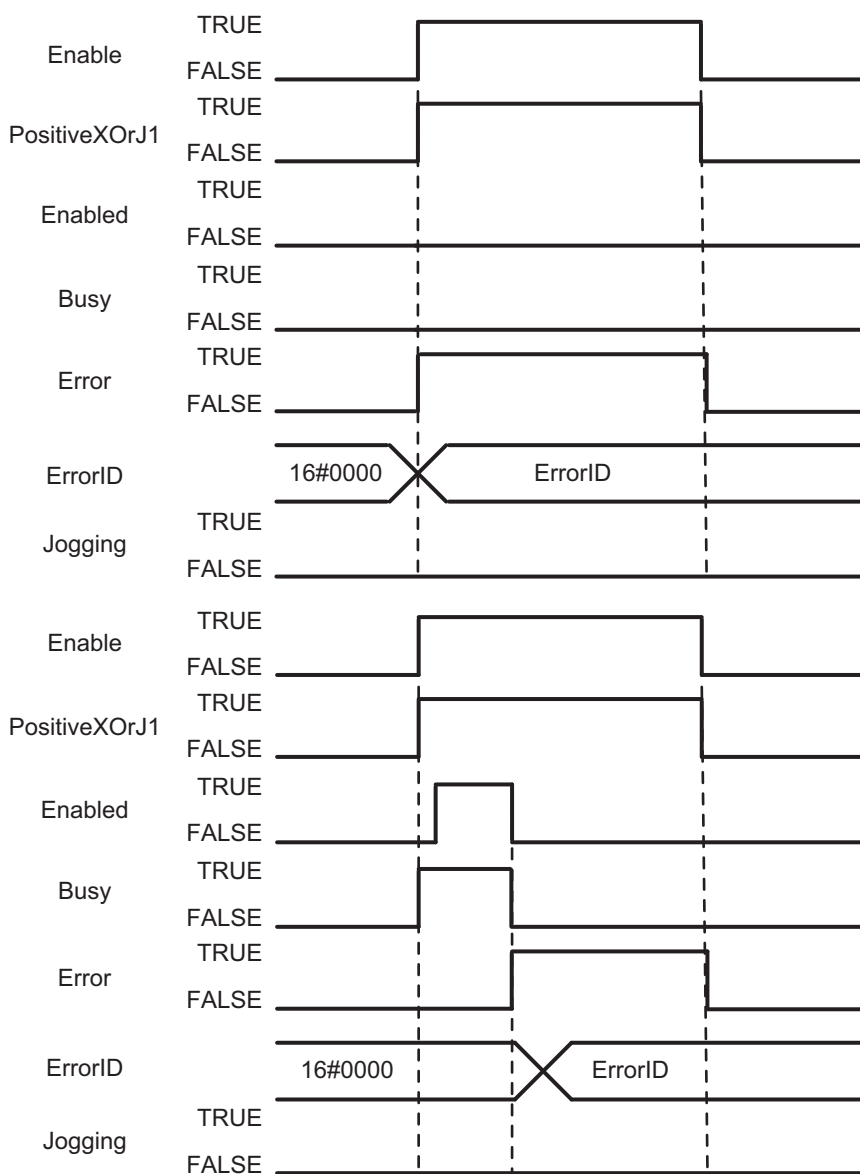
● Normal execution





## ● Error execution

If an error occurs during instruction execution, Error will change to TRUE. You can find out the cause of the error by referring to the value output by ErrorID.



## Troubleshooting

Error code	Extended error ID	Status	Meaning	Correction
16#0000	16#00000000	The service ended normally.	---	---
16#3C40	16#00000001	Robot controller error	Error occurs in the robot controller.	Check the RobotError variable output in ARB_RobotControl and perform corrections for the error.
	16#00000002	Communication error	Ethernet/IP communication error	Execute the instruction after the successful EtherNet/IP communication recovery.
	16#00000003	Invalid operation	The instruction could not be executed in this controller state. <i>See Execution Conditions Table on page 143.</i>	Execute the instruction after the function block is ready for execution.
	16#00000004	Re-execution disabled	Instruction cannot be re-executed	Wait for the Done variable output changes to TRUE and then execute the function block again.
	16#0000000A	Jog Mode Selection Out of Range	The value specified for the input variable JogMode is out of range.	Set the JogMode within range.
	16#0000000B	Speed Out of Range	The value specified for the parameter Speed in the input variable MotionParams is out of range.	Set the Speed within range.
	16#0000000C	Acceleration Out of Range	The value specified for the parameter Acceleration in the input variable MotionParams is out of range.	Set the Acceleration within range.
	16#0000000D	Deceleration Out of Range	The value specified for the parameter Deceleration in the input variable MotionParams is out of range.	Set the Deceleration within range.
	16#0000000E	Acceleration Type Selection Out of Range	The value specified for the parameter AccelerationType in the input variable MotionParams is out of range.	Set the AccelerationType within range.
	16#0000000F	Speed Limit Out of Range	The value specified for the parameter SpeedLimit in the input variable MotionParams is out of range.	Set the SpeedLimit within range.
	16#00000011	More than one direction selected	More than one direction input is true at the same time.	Input only one jog signal at a time.
	16#00000013	Emergency Stop state	Emergency Stop input has been set during the execution of the instruction	Reset the Emergency Stop input.

# ARB\_AlignToolCommand

The ARB\_AlignTool instruction rotates the tool to be aligned with World Coordinate System of the robot.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ARB_AlignToolCommand	AlignToolCommand	FB		<pre>ARB_AlignToolCommand_instance (   RobotData :=parameter,   Execute :=parameter,   Blending :=parameter,   MotionParams :=parameter,   AlignConfig :=parameter,   Done =&gt;parameter,   Busy =&gt;parameter,   Error =&gt;parameter,   ErrorID=&gt;parameter,   ErrorIDEx=&gt;parameter );</pre>

## Variables

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Enable</i> changes to TRUE.
Blending	Blending	BOOL	TRUE or FALSE	FALSE	Enables blending between the executed movement and the next movement executed.  TRUE: Blending enabled (Continuous Path) FALSE: No blending enabled
MotionParams	Motion parameters	sARB_MOTION_PARAMS_REF	---*1	0	Set the motion parameters.
AlignConfig	Align configuration	sARB_ALIGN_CONFIG_REF	---*1	0	Set the align configuration related with the robot movement operation.

\*1. See structure definition

## Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged by the robot controller.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is being executed.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error ID	WORD	*1	Specify error number.
ErrorIDEx	Extended error ID	DWORD	*1	Specify extended error number.

\*1. Refer to *Troubleshooting* on page 142 for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	<ul style="list-style-type: none"> <li>When instruction is acknowledged by the robot controller.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one control period when <i>Execute</i> is FALSE.</li> <li>When <i>Error</i> changes to TRUE</li> </ul>
Busy	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to TRUE.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE.</li> </ul>
Error	<ul style="list-style-type: none"> <li>When there is an error in the execution conditions or input parameters for the instruction.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> changes to FALSE.</li> </ul>

## In-Out Variables

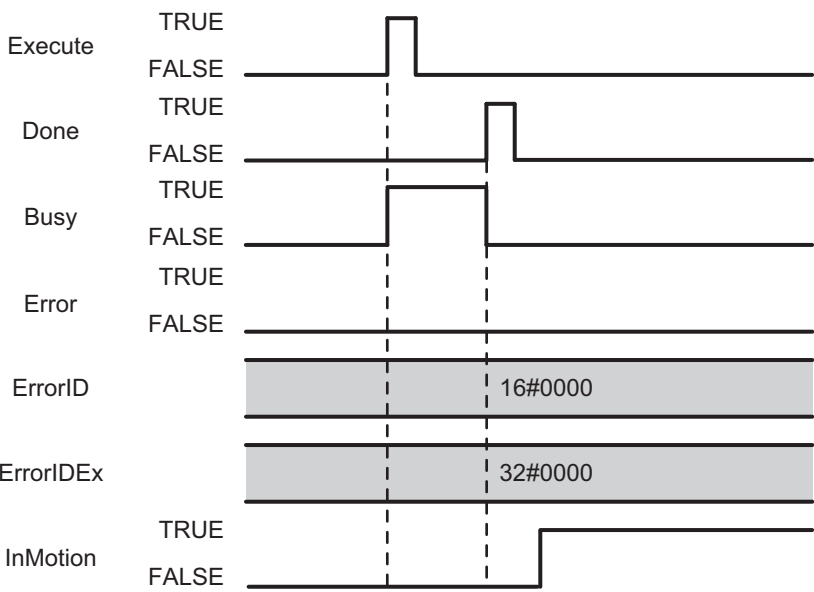
Name	Meaning	Data type	Valid range	Description
RobotData	Robot controller data	sARB_ROBOT_DATA_REF	---	Specify data needed for the robot controller.

## Function

- This instruction moves the robot to align robot tool Z-axis with the nearest world axis. All joints of the robot could be moved when alignment movement is executed.
- This function only affects 6-axis robots (Viper)
- Once the command has been completed (Done output is TRUE) another command can be executed. If the Blending input of the first one is TRUE, the second command will be blended.
- Align command always achieves its target position, even if the blending input is set to TRUE.
- Cancel Motion input can stop the movement once it has been commanded.

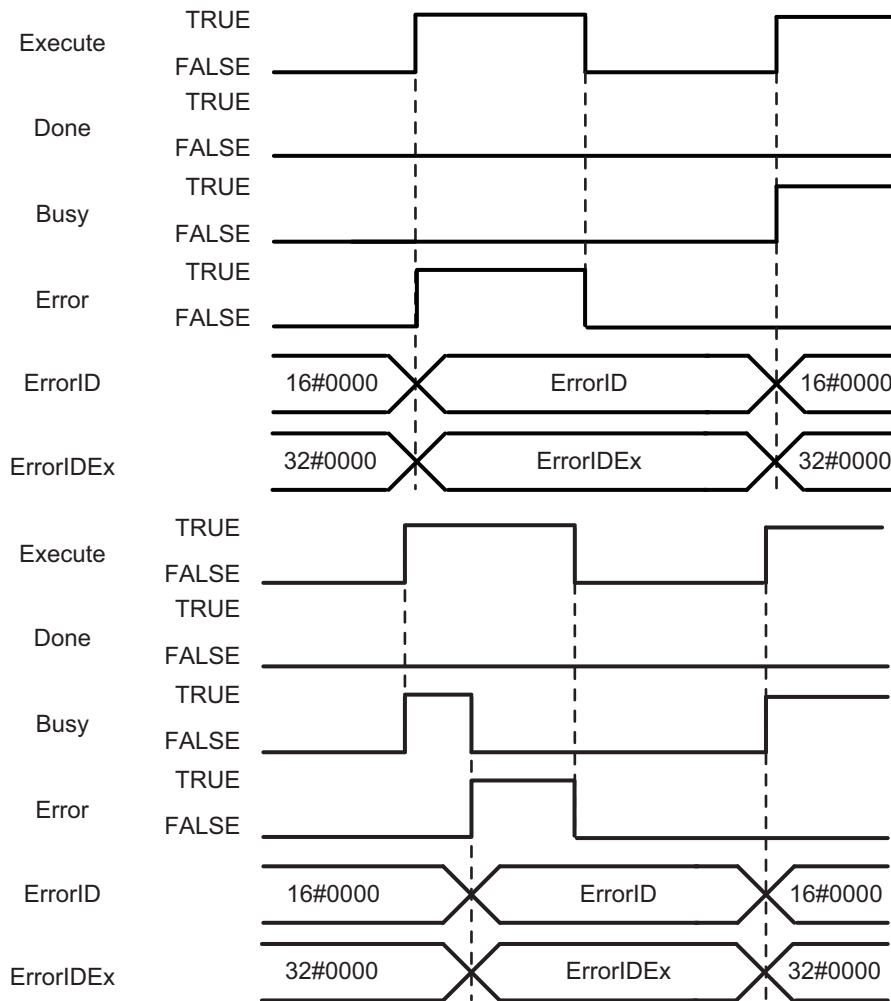
Timing charts

● Normal execution



## ● Error execution

If an error occurs during instruction execution, Error will change to TRUE. You can find out the cause of the error by referring to the value output by ErrorID and ErrorIDEx.



## Troubleshooting

Error code	Extended error ID	Status	Meaning	Correction
16#0000	16#00000000	The service ended normally.	---	---
16#3C41	16#00000001	Robot controller error	Error occurs in the robot controller.	Check the RobotError variable output in ARB_RobotControl and perform corrections for the error.
	16#00000002	Communication error	Ethernet/IP communication error	Execute the instruction after the successful EtherNet/IP communication recovery.
	16#00000003	Invalid operation	The instruction could not be executed in this controller state. * See <i>Execution Conditions Table</i> on page 143.	Execute the instruction after the function block is ready for execution.
	16#00000004	Re-execution disabled	Instruction cannot be re-executed	Wait for the Done variable output changes to TRUE and then execute the function block again.
	16#00000005	Multi-execution disabled	Multiple instructions that cannot be executed simultaneously were executed for the same target.	Wait for the Done variable output in the previously executed instructions changes to TRUE and then execute the function block.
	16#0000000B	Speed Out of Range	The value specified for the parameter Speed in the input variable MotionParams is out of range.	Set the Speed within range.
	16#0000000C	Acceleration Out of Range	The value specified for the parameter Acceleration in the input variable MotionParams is out of range.	Set the Acceleration within range.
	16#0000000D	Deceleration Out of Range	The value specified for the parameter Deceleration in the input variable MotionParams is out of range.	Set the Deceleration within range.
	16#0000000E	Acceleration Type Selection Out of Range	The value specified for the parameter AccelerationType in the input variable MotionParams is out of range.	Set the AccelerationType within range.
	16#0000000F	Speed Limit Out of Range	The value specified for the parameter SpeedLimit in the input variable MotionParams is out of range.	Set the SpeedLimit within range.
	16#00000012	Illegal Motion Stop request	Cancel Motion or Stop on Input executed during execution of MoveCommand. Current Movement not started yet, so it cannot be stopped.	Wait for completion of the Current Movement and then execute the instruction again.
	16#00000013	Emergency Stop state	Emergency Stop input has been set during the execution of the instruction	Reset the Emergency Stop input.



# Execution Conditions Table

Before and during execution of each command following conditions will be checked. In case one of the conditions is not fulfilled an error will occur.

	Connected <sup>*1</sup>	Cancel Motion <sup>*2</sup>	Stop On Input <sup>*3</sup>	Power State <sup>*4</sup>	Calibrated State <sup>*5</sup>	Error State <sup>*6</sup>	Estop State <sup>*7</sup>
SetToolTransform	TRUE	---	---	---	---	FALSE	---
ResetToolTransform	TRUE	---	---	---	---	FALSE	---
DefinePosition	TRUE	---	---	---	---	FALSE	---
DefinePallet	TRUE	---	---	---	---	FALSE	---
ResetRobotError	TRUE	---	---	---	---	---	---
RobotControl	---	---	---	---	---	---	---
TeachPosition	TRUE	---	---	---	---	FALSE	---
InputOutputSignals	TRUE	---	---	---	---	---	---
TeachPendantControl	TRUE	---	---	---	---	---	---
ReadLatch	TRUE	---	---	---	---	---	---
MoveCommand	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
PickAndPlaceCommand	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
Jog	TRUE	---	---	TRUE	TRUE	FALSE	FALSE
AlignToolCommand	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE

\*1. The RobotControl output variable in the ARB\_RobotControl function block

\*2. The CancelMotion input variable in the ARB\_RobotControl function block

\*3. The StopOnInput input variable in the ARB\_RobotControl function block

\*4. The RobotState.PowerState output variable in the ARB\_RobotControl function block

\*5. The RobotState.CalibratedState output variable in the ARB\_RobotControl function block

\*6. The RobotState.ErrorState output variable in the ARB\_RobotControl function block

\*7. The RobotState.EStopState output variable in the ARB\_RobotControl function block

- **For the condition of Connected:**

In case of NOT connected state the error will be always "16#00000002: Communication error"

- **For the other conditions of the table:**

In case conditions are not fulfilled in the instruction execution error will be "16#00000003: Invalid operation"

In case conditions are not fulfilled during execution errors will be different depending on the error cause:

- Power State, Calibrated State, and Error State:  
"16#00000001: OAT controller error"
- Cancel Motion and Stop on Input:  
"16#00000012: Illegal Motion Request"
- E-Stop State:  
"16#00000013: Emergency Stop State"

# Multi-execution of Function Blocks

---

Multi-execution is possible for the following instructions: They can be executed while other instructions are executed.

- ARB\_TeachPendantControl
- ARB\_InputOutputSignals

Multi-execution is disabled for the following instructions: Error will happen in case two or more instructions of this group are executed at same time.

- ARB\_Jog
- ARB\_MoveCommand
- ARB\_PickAndPlaceCommand
- ARB\_DefineLocation
- ARB\_DefinePallet
- ARB\_SetToolTransform
- ARB\_ResetToolTransform
- ARB\_ResetRobotError
- ARB\_TeachPosition
- ARB\_AlignToolCommand

The instruction ARB\_RobotControl control several robot actions. Some of the actions can be executed while other instructions are being executed:

- Power
- CancelMotion
- StopOnInput

Limitation of multi-execution exist in the instruction ARB\_ReadLatch, it cannot be multi-executed for a correct update of the latch signals received.

# Re-execution of Function Blocks

---

An error occurs when re-executing the same instance of any function block before it has been completed successfully ('Done' output is raised) or by error ('Error output is raised). This is only applicable for the 'Execute' type function blocks.

# Errors from Controller

The specific errors of the robot controller are listed in the following table:

Error	Description
-3001	*E-stop due to lost communications with PLC*
-3007	*Invalid command: More than one command bit enabled*
-3008	*Only one axis bit can be set while jogging*
-3009	*Cannot mix joint-coordinate location and pallet*
-3010	*Cannot move relative to a pallet*
-3011	*Invalid location number* Location n
-3012	*Invalid pallet number* Pallet n
-3013	*Invalid speed parameter* Speed n
-3014	*Invalid acceleration/deceleration parameter* Value n
-3015	*Undefined location* Location n
-3016	*Undefined pallet* Pallet n
-3019	*ePLC Connect start-up error* Code n Here are the possible causes: invalid eV+ software version, invalid eV+ software license, or invalid ePLC connection application license.
-3020	*Invalid approach parameter for this robot*
-3021	*Pallet locations are linear*
-3022	*Invalid pallet index* Position n
-3024	*Adept ACE Control mode enabled*



# 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 151.

\*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 151.

\*2. It is provided in English and Japanese.



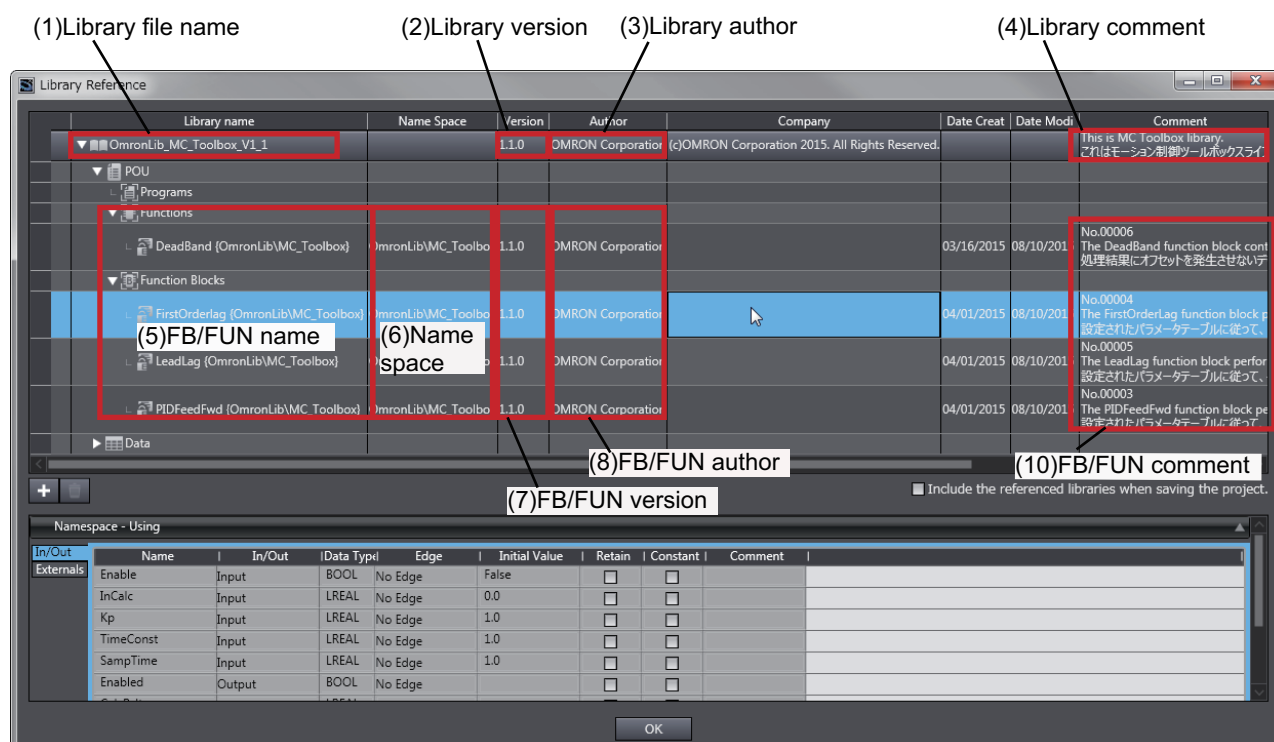
## Referring to Attributes of Libraries, Function Blocks and Functions

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

- Library Reference Dialog Box
- Toolbox Pane
- Ladder Editor

### (a) Library Reference Dialog Box

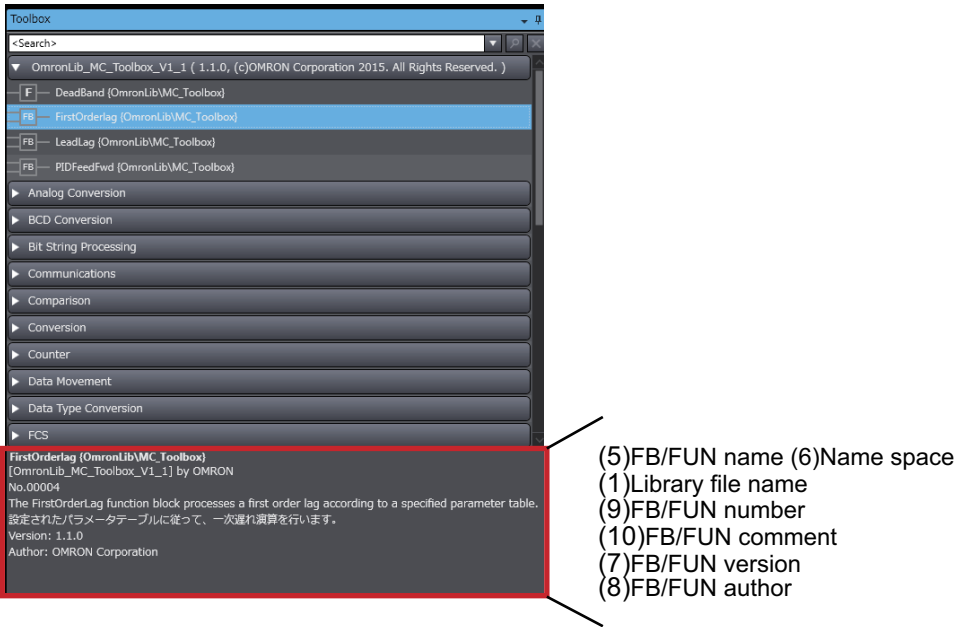
When you refer to the libraries, the library information is displayed at the locations shown below.



(b) Toolbox Pane

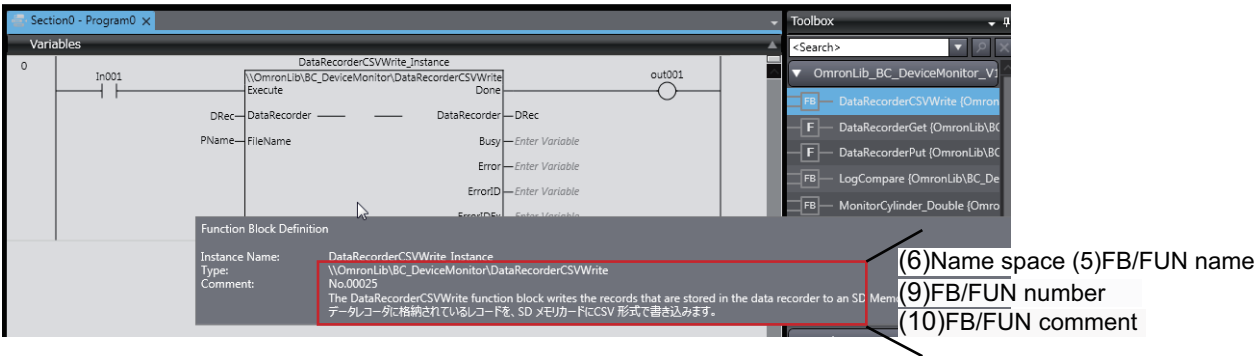
Select a function block and function to display its library information at the bottom of the Toolbox Pane.

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



(c) Ladder Editor

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



# Referring to Function Block and Function Source Codes

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

User function blocks and user functions can be created based on the copies of these source codes.

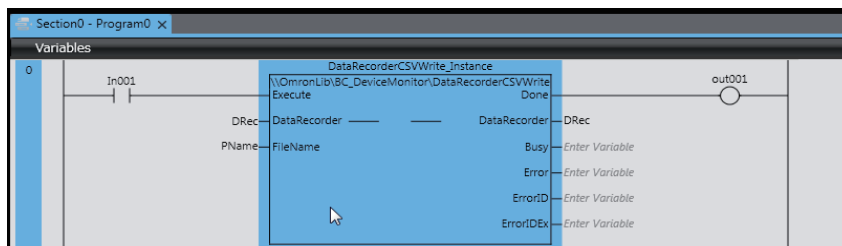
The following are the examples of items that you may need to customize.

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

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

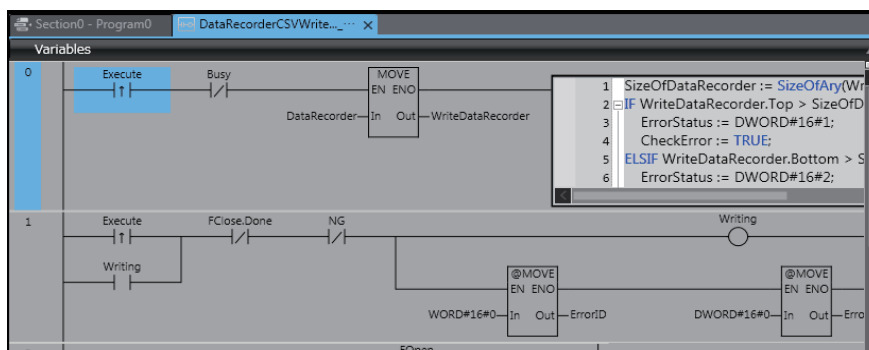
Use the following procedure to refer to the source codes of function blocks and functions.

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



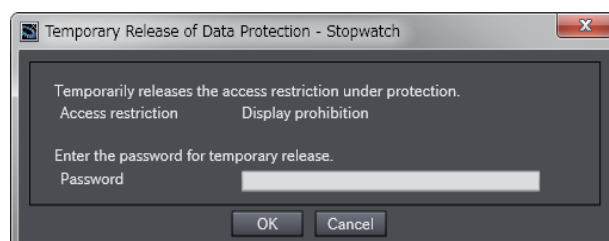
- 2 Double-click or right-click and select **To Lower Layer** from the menu.

The source code is displayed.



## Precautions for Correct Use

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







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

**Contact: [industrial.omron.eu](mailto:industrial.omron.eu)**

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

**Cat. No. W575-E1-01**

0416