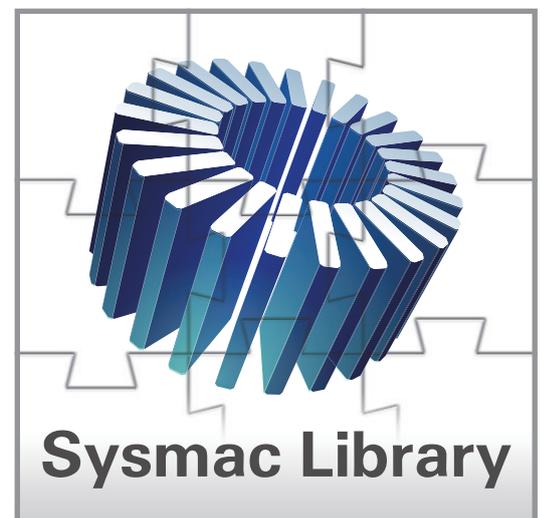


OMRON

Sysmac Library

User's Manual for SLMP Communications Library

SYSMAC-XR017



SYSMAC
always in control

W597-E1-03

NOTE

1. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.
2. No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice.
3. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Trademarks

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

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

Copyrights

- Microsoft product screen shots reprinted with permission from Microsoft Corporation.

Introduction

Thank you for purchasing an NJ/NX-series CPU Unit, PC for NY-series production.

This manual contains information that is necessary to use Function block for SLMP Communications Library (hereafter, sometimes abbreviated to FB). Please read this manual and make sure you understand the functionality and performance of the product before you attempt to use it in a control system. This manual provides function block specifications. It does not describe application restrictions or combination restrictions for Controllers, Units, and components.

Make sure to read the user's manual for each product before use.

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

Features of the Library

The SLMP Communication Library is a collection of software function objects for communicating with Mitsubishi Electric sequencers via SLMP.

Various FB types are provided for easy incorporation of the OMRON Machine Automation Controller in networks configured by the Mitsubishi Electric Sequencer.

This enables following to:

- A machine automation controller can monitor and control a system configured with Mitsubishi Electric sequencers.
- A machine automation controller can notify the own state to a system configured with Mitsubishi Electric sequencers.

Intended Audience

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

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

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

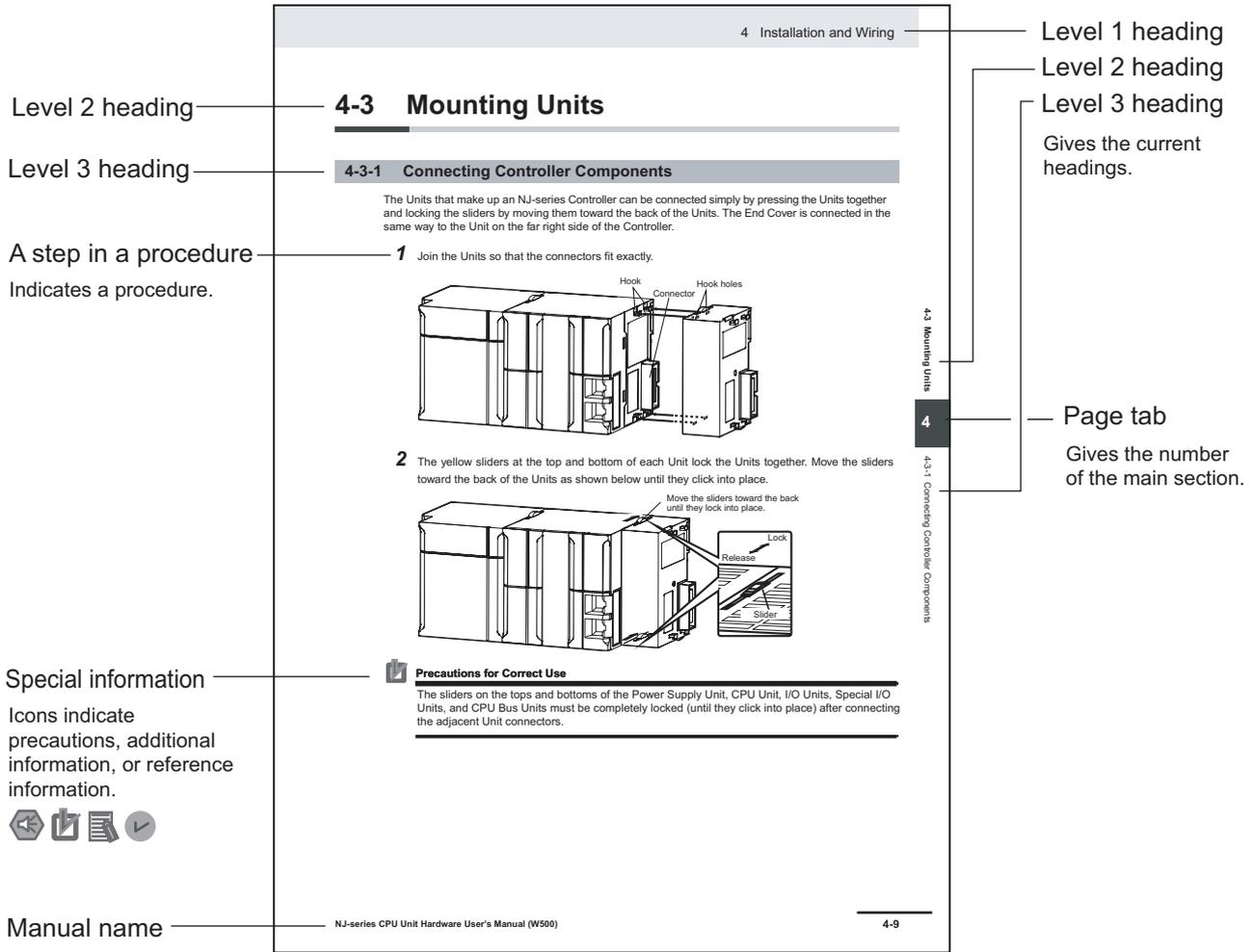
Applicable Products

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

Manual Structure

Page Structure

The following page structure is used in this manual.



Note This illustration is provided only as a sample. It may not literally appear in this manual.

Special Information

Special information in this manual is classified as follows:



Precautions for Safe Use

Precautions on what to do and what not to do to ensure safe usage of the product.



Precautions for Correct Use

Precautions on what to do and what not to do to ensure proper operation and performance.



Additional Information

Additional information to read as required.

This information is provided to increase understanding and make operation easier.



Version Information

Information on differences in specification and functionality for following items.

- This library
- CPU Units and the industrial-use PC
- Sysmac Studio

Sections in this Manual

1	Sysmac Library Usage Procedure	1
2	SLMP Communications Library	2
3	Common Specifications of Function Blocks	3
4	Individual Specifications of Function Blocks	4
A	Appendix	A
I	Index	I

CONTENTS

Introduction	1
Features of the Library.....	1
Intended Audience.....	1
Applicable Products	1
Manual Structure.....	2
Page Structure.....	2
Special Information	3
Terms and Conditions Agreement.....	9
Warranty, Limitations of Liability	9
Application Considerations	10
Disclaimers	10
Statement of security responsibilities for assumed use cases and against threats.....	11
Safety Precautions.....	12
Definition of Precautionary Information.....	12
Symbols	12
WARNING.....	13
Precautions for Correct Use	14
Using the Library.....	14
Using Sample Programming.....	14
Operation	14
Related Manuals.....	15
Revision History.....	20
Sections in this Manual	5

Section 1 Sysmac Library Usage Procedure

1-1 Procedure to Use Sysmac Library Installed Using the Installer.....	1-2
1-1-1 Using a Newly Installed Sysmac Library	1-2
1-1-2 Using an Upgraded Sysmac Library	1-4
1-2 How to use Sysmac Library in the CPU Unit or Industrial PC	1-7
1-3 Steps to display the Instruction Reference	1-9

Section 2 SLMP Communications Library

2-1 Overview	2-2
2-1-1 Device Read.....	2-2
2-1-2 Device Write	2-3
2-1-3 Send Command	2-3
2-1-4 Support for MC Protocol.....	2-3
2-2 Usage Method	2-4
2-2-1 Preliminary Preparations for SLMP Communications	2-4
2-2-2 Preparing / Terminating Socket	2-4
2-2-3 Precautions for FB Execution.....	2-4
2-3 Set Values	2-5

2-3-1	Device Code (Device Access for Sequencer CPU Made by Mitsubishi Electric)	2-5
-------	---	-----

Section 3 Common Specifications of Function Blocks

3-1	Common Variables	3-2
3-1-1	Definition of Input Variables and Output Variables	3-2
3-1-2	Execute-type Function Blocks	3-3
3-1-3	Enable-type Function Blocks	3-5
3-2	Precautions	3-8
3-2-1	Nesting	3-8
3-2-2	Instruction Options	3-8
3-2-3	Re-execution of Function Blocks	3-8

Section 4 Individual Specifications of Function Blocks

SLMP_DeviceRead	4-2
Function Block and Function Information	4-2
Input Variables	4-2
Output Variables	4-3
Input-Output Variables	4-4
Function	4-4
Timing Chart	4-7
Additional Information	4-7
Precautions for Correct Use	4-8
Troubleshooting	4-8
Sample Programming	4-10
SLMP_DeviceWriteWord	4-12
Function Block and Function Information	4-12
Input Variables	4-12
Output Variables	4-13
Input-Output Variables	4-14
Function	4-14
Timing Chart	4-16
Additional Information	4-17
Precautions for Correct Use	4-17
Troubleshooting	4-18
Sample Programming	4-19
SLMP_DeviceWriteBool	4-21
Function Block and Function Information	4-21
Input Variables	4-21
Output Variables	4-22
Input-Output Variables	4-23
Function	4-23
Timing Chart	4-24
Additional Information	4-25
Precautions for Correct Use	4-25
Troubleshooting	4-26
Sample Programming	4-27
SLMP_SendCmd	4-30
Function Block and Function Information	4-30
Input variables	4-30
Output variables	4-31
In-out variables	4-31
Structures	4-32
Function	4-36
Timing Charts	4-37
Additional Information	4-38
Precautions for Correct Use	4-38
Troubleshooting	4-38

Sample program	4-40
SLMP_DeviceRead_UDP	4-43
Function Block and Function Information	4-43
Input variables	4-43
Output variables	4-44
Input-Output Variables	4-45
Function	4-45
Timing Chart	4-48
Additional Information	4-49
Precautions for Correct Use	4-49
Troubleshooting	4-49
Sample program	4-51
SLMP_DeviceWriteWord_UDP	4-53
Function Block and Function Information	4-53
Input variables	4-53
Output variables	4-54
In-out variables	4-54
Function	4-55
Timing Chart	4-57
Additional Information	4-58
Precautions for Correct Use	4-58
Troubleshooting	4-59
Sample program	4-60
SLMP_DeviceWriteBool_UDP	4-62
Function Block and Function Information	4-62
Input variables	4-62
Output variables	4-63
Input-Output Variables	4-64
Function	4-64
Timing Chart	4-65
Additional Information	4-66
Precautions for Correct Use	4-66
Troubleshooting	4-67
Sample program	4-69
SLMP_SendCmd_UDP	4-71
Function Block and Function Information	4-71
Input variables	4-71
Output variables	4-72
In-out variables	4-72
Structures	4-73
Function	4-77
Timing Charts	4-78
Additional Information	4-79
Precautions for Correct Use	4-79
Troubleshooting	4-79
Sample program	4-81

Appendix

A-1 Referring to Library Information.....	A-2
A-1-1 Library Attributes, and FB or FUN Attributes	A-2
A-1-2 Referring to Attributes of Libraries, Function Blocks, and Functions	A-3

Index

Terms and Conditions Agreement

Warranty, Limitations of Liability

Warranties

- **Exclusive Warranty**

Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

- **Limitations**

OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.

Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

- **Buyer Remedy**

Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

See <https://www.omron.com/global/> or contact your Omron representative for published information.

Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY

WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY OR IN LARGE QUANTITIES WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron's test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may

be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Statement of security responsibilities for assumed use cases and against threats

OMRON SHALL NOT BE RESPONSIBLE AND/OR LIABLE FOR ANY LOSS, DAMAGE, OR EXPENSES DIRECTLY OR INDIRECTLY RESULTING FROM THE INFECTION OF OMRON PRODUCTS, ANY SOFTWARE INSTALLED THEREON OR ANY COMPUTER EQUIPMENT, COMPUTER PROGRAMS, NETWORKS, DATABASES OR OTHER PROPRIETARY MATERIAL CONNECTED THERETO BY DISTRIBUTED DENIAL OF SERVICE ATTACK, COMPUTER VIRUSES, OTHER TECHNOLOGICALLY HARMFUL MATERIAL AND/OR UNAUTHORIZED ACCESS.

It shall be the users sole responsibility to determine and use adequate measures and checkpoints to satisfy the users particular requirements for (i) antivirus protection, (ii) data input and output, (iii) maintaining a means for reconstruction of lost data, (iv) preventing Omron Products and/or software installed thereon from being infected with computer viruses and (v) protecting Omron Products from unauthorized access.

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 this library on the NJ/NX-series CPU Unit, PC for NY-series production.

The safety precautions that are provided are extremely important for safety. Always read and heed the information provided in all safety precautions.

The following notation is used.

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

Symbols

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

WARNING

Caution

Read all related manuals carefully before you use this library.



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



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



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



Perform the test run by holding an emergency stop switch in hand or otherwise prepare for rapid motor operation in an application to control the motor.

Also perform the test run by using parameters for which the motor does not rapidly accelerate or decelerate before you gradually adjust the parameters.



In heating or cooling applications, perform the test run by using parameters for which rapid temperature changes will not occur before you gradually adjust the parameters.



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



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



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



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



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.
- The multi-execution (buffer mode) cannot be performed 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 a function or function block with an Enabled output variable, if the value of Enabled is FALSE, do not use the processing result of the function or function block as a command value to the control target.
- In the function block with Execute, do not perform re-execution by the same instance. The output value of the function block will return to the default value.

Related Manuals

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

Manual name	Man. No.	Model	Application	Description
NX-series CPU Unit Hardware User's Manual	W535	NX701-□□□□	Learning the basic specifications of the NX701 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NX701 system is provided along with the following information on the CPU Unit. <ul style="list-style-type: none"> • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NX-series NX502 CPU Unit Hardware User's Manual	W629	NX502-□□□□	Learning the basic specifications of the NX502 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NX502 system is provided along with the following information on the CPU Unit. <ul style="list-style-type: none"> • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NX-series NX102 CPU Unit Hardware User's Manual	W593	NX102-□□□□	Learning the basic specifications of the NX102 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NX102 system is provided along with the following information on the CPU Unit. <ul style="list-style-type: none"> • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection

Manual name	Man. No.	Model	Application	Description
NX-series NX1P2 CPU Unit Hardware User's Manual	W578	NX1P2-□□□□	Learning the basic specifications of the NX1P2 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NX1P2 system is provided along with the following information on the CPU Unit. <ul style="list-style-type: none"> • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NJ-series CPU Unit Hardware User's Manual	W500	NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning the basic specifications of the NJ-series CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NJ-series system is provided along with the following information on the CPU Unit. <ul style="list-style-type: none"> • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NY-series IPC Machine Controller Industrial Panel PC Hardware User's Manual	W557	NY532-□□□□	Learning the basic specifications of the NY-series Industrial Panel PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NY-series system is provided along with the following information on the Industrial Panel PC. <ul style="list-style-type: none"> • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection

Manual name	Man. No.	Model	Application	Description
NY-series IPC Machine Controller Industrial Box PC Hardware User's Manual	W556	NY512-□□□□	Learning the basic specifications of the NY-series Industrial Box PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NY-series system is provided along with the following information on the Industrial Box PC. <ul style="list-style-type: none"> • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NJ/NX-series CPU Unit Software User's Manual	W501	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning how to program and set up an NJ/NX-series CPU Unit. Mainly software information is provided.	The following information is provided on a Controller built with an NJ/NX-series CPU Unit. <ul style="list-style-type: none"> • CPU Unit operation • CPU Unit features • Initial settings • Programming based on IEC 61131-3 language specifications
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Software User's Manual	W558	NY532-□□□□ NY512-□□□□	Learning how to program and set up the Controller functions of an NY-series Industrial PC.	The following information is provided on the NY-series Controller functions. <ul style="list-style-type: none"> • Controller operation • Controller features • Controller settings • Programming based on IEC 61131-3 language specifications
NJ/NX-series Instructions Reference Manual	W502	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning detailed specifications on the basic instructions of an NJ/NX-series CPU Unit.	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NY-series Instructions Reference Manual	W560	NY532-□□□□ NY512-□□□□	Learning detailed specifications on the basic instructions of an NY-series Industrial PC.	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NJ/NX-series CPU Unit Motion Control User's Manual	W507	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about motion control settings and programming concepts.	The settings and operation of the CPU Unit and programming concepts for motion control are described.

Manual name	Man. No.	Model	Application	Description
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Motion Control User's Manual	W559	NY532-□□□□ NY512-□□□□	Learning about motion control settings and programming concepts of an NY-series Industrial PC.	The settings and operation of the Controller and programming concepts for motion control are described.
NJ/NX-series Motion Control Instructions Reference Manual	W508	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the specifications of the motion control instructions.	The motion control instructions are described.
NY-series Motion Control Instructions Reference Manual	W561	NY532-□□□□ NY512-□□□□	Learning about the specifications of the motion control instructions of an NY-series Industrial PC.	The motion control instructions are described.
NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual	W506	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Using the built-in EtherNet/IP port on an NJ/NX-series CPU Unit.	Information on the built-in EtherNet/IP port is provided. Information is provided on the basic setup, tag data links, and other features.
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP™ Port User's Manual	W563	NY532-□□□□ NY512-□□□□	Using the built-in EtherNet/IP port in an NY-series Industrial PC.	Information on the built-in EtherNet/IP port is provided. Information is provided on the basic setup, tag data links, and other features.
NJ/NY-series NC Integrated Controller User's Manual	O030	NJ501-5300 NY532-5400	Performing numerical control with NJ/NY-series Controllers.	Describes the functionality to perform the numerical control.
NJ/NY-series G code Instructions Reference Manual	O031	NJ501-5300 NY532-5400	Learning about the specifications of the G code/M code instructions.	The G code/M code instructions are described.
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC -SE2□□□	Learning about the operating procedures and functions of the Sysmac Studio.	Describes the operating procedures of the Sysmac Studio.

Manual name	Man. No.	Model	Application	Description
CNC Operator Operation Manual	O032	SYSMAC-RTNC0□□□D	Learning an introduction of the CNC Operator and how to use it.	An introduction of the CNC Operator, installation procedures, basic operations, connection operations, and operating procedures for main functions are described.

Revision History

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

Cat. No.	W597-E1-03
-----------------	-------------------

↑
Revision code

Revision code	Date	Revised content
01	April 2018	Original production
02	January 2019	Added the target model number.
03	April 2024	<ul style="list-style-type: none"> • Added target models • Added the description of following FBs: <ul style="list-style-type: none"> a) SLMP_SendCmd b) SLMP_DeviceRead_UDP c) SLMP_DeviceWriteWord_UDP d) SLMP_DeviceWriteBool_UDP e) SLMP_SendCmd_UDP • Corrected mistakes.

1

Sysmac Library Usage Procedure

The section describes the procedure to use Sysmac Library installed using the installer, and Sysmac Library in the CPU unit or Industrial PC.

1-1	Procedure to Use Sysmac Library Installed Using the Installer.....	1-2
1-1-1	Using a Newly Installed Sysmac Library	1-2
1-1-2	Using an Upgraded Sysmac Library.....	1-4
1-2	How to use Sysmac Library in the CPU Unit or Industrial PC	1-7
1-3	Steps to display the Instruction Reference	1-9

1-1 Procedure to Use Sysmac Library Installed Using the Installer

This section describes the procedure to use Sysmac Library installed using the installer. There are two ways to use libraries.

- Using a newly installed Sysmac Library
- Using an upgraded Sysmac Library

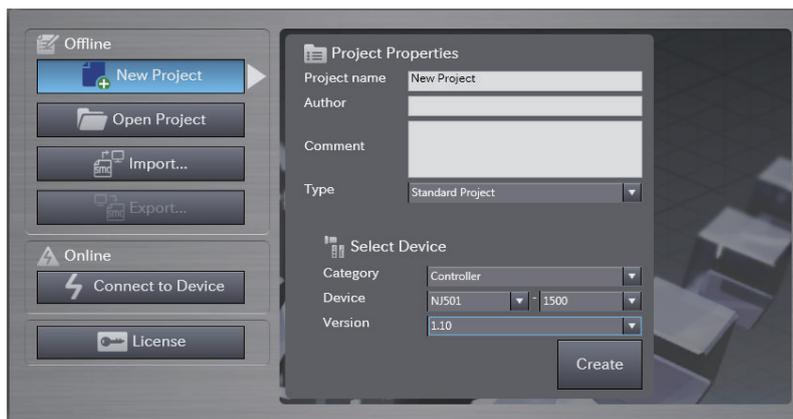


Version Information

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

1-1-1 Using a Newly Installed Sysmac Library

- 1 Start the Sysmac Studio and open a project using Sysmac Library, or create a new one.

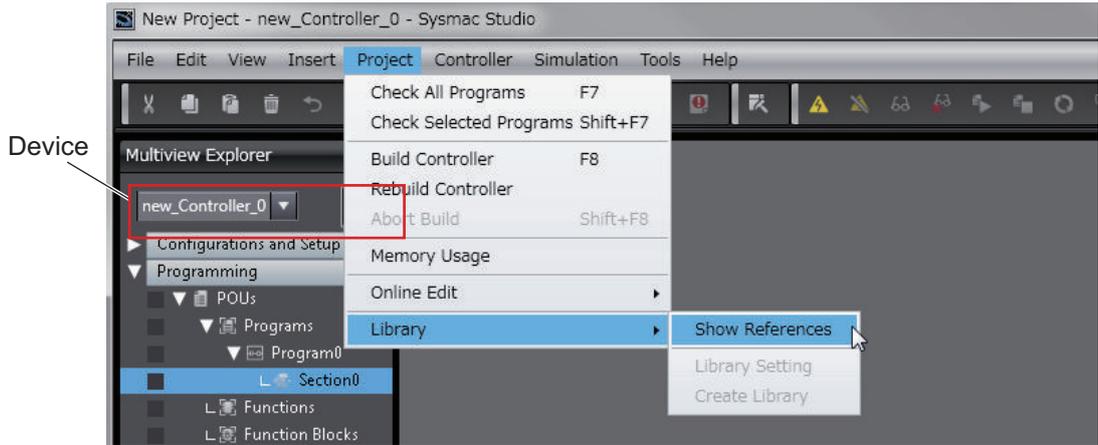


Precautions for Correct Use

If you create a new project, be sure to configure the settings as follows to enable use of the Sysmac Library. Without the settings below, you cannot proceed to Step 2 and later steps.

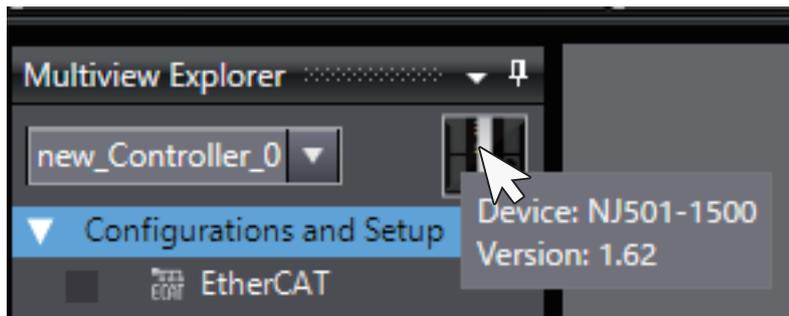
- Set the project type to Standard Project or Library Project.
- Set the device category to Controller.
- For the setting of Controller and Version in the Select Device section, refer to .

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

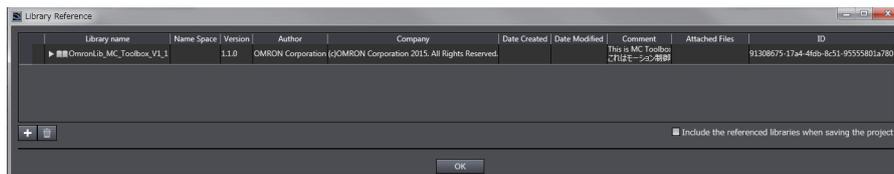


Precautions for Correct Use

If you have registered multiple devices in the project, make sure that the currently selected device is Controller (NJ/NX-series CPU Unit or NY-series Industrial PC). If the Controller is not selected, the menu for browsing the library will not appear. You can confirm the selected device by hovering your mouse cursor over the icon displayed in Multi View Explorer.



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



Sysmac Library is read into the project.

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

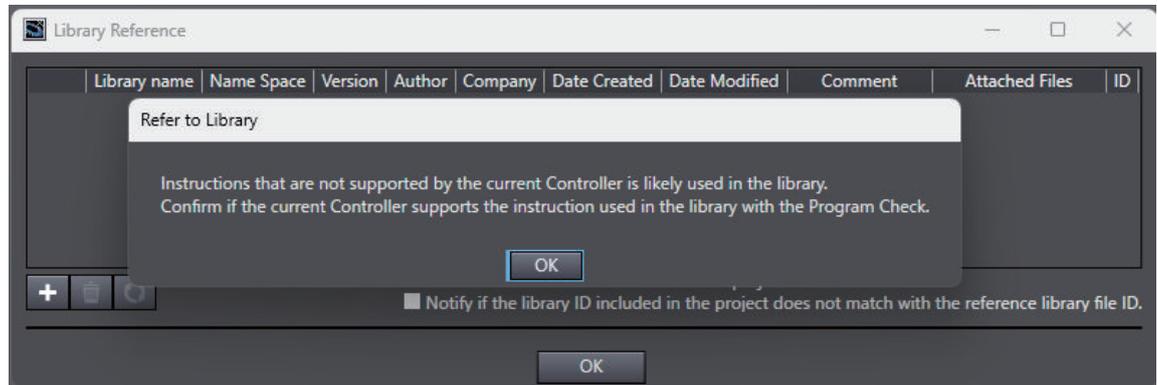
For the procedure for adding and setting libraries in the above screen, refer to *Sysmac Studio Version 1 Operation Manual (W504)*.



Precautions for Correct Use

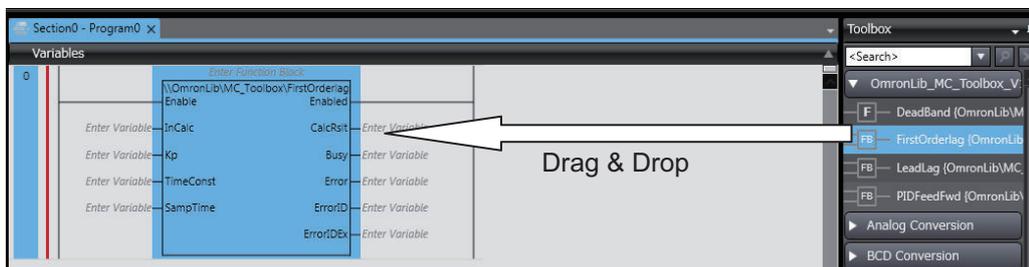
Depending on the type of the Controller you are using, the following dialog will be displayed when adding the Sysmac Library. In this case, after adding the Sysmac Library, and then run Check all programs, and confirm that there are no errors in Build Tab Page about result of program check. There is no problem if there are no errors.

From the Main Menu, select **project - check all programs**.

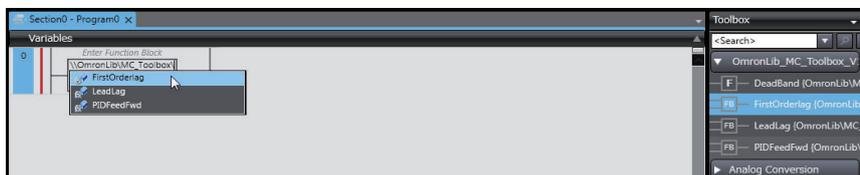


4 Insert the Sysmac Library's function blocks and functions into the circuit using one of the following two methods.

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

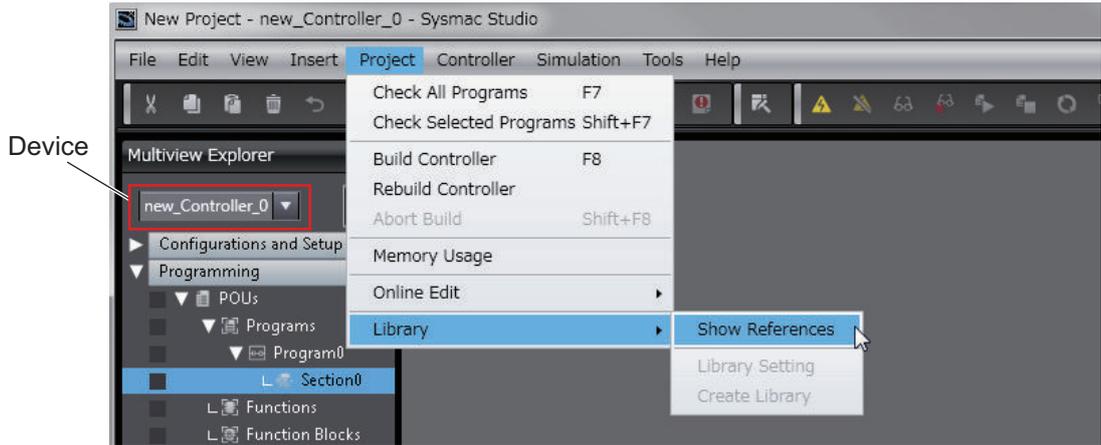


- Right-click the Ladder Editor, select **Insert Function Block** in the menu, and enter the fully qualified name (%%namespacename%FBname).



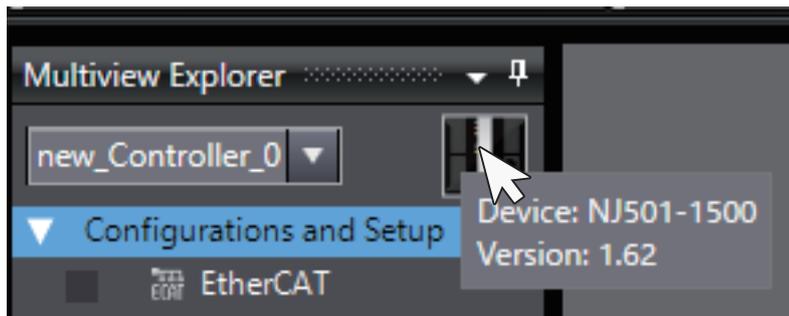
1-1-2 Using an Upgraded Sysmac Library

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

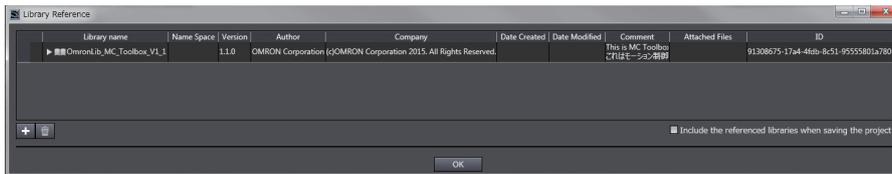


Precautions for Correct Use

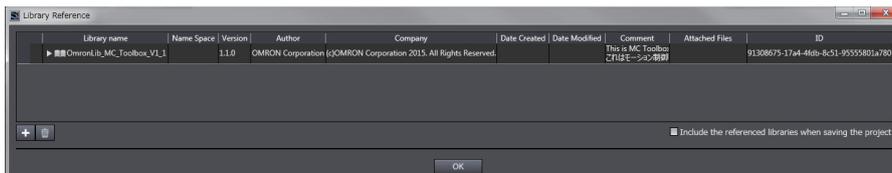
If you have registered multiple devices in the project, make sure that the currently selected device is Controller (NJ/NX-series CPU Unit or NY-series Industrial PC). If the Controller is not selected, the menu for browsing the library will not appear. You can confirm the selected device by hovering your mouse cursor over the icon displayed in Multi View Explorer.



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



4 Add New version Sysmac Library to the list and click **OK**.

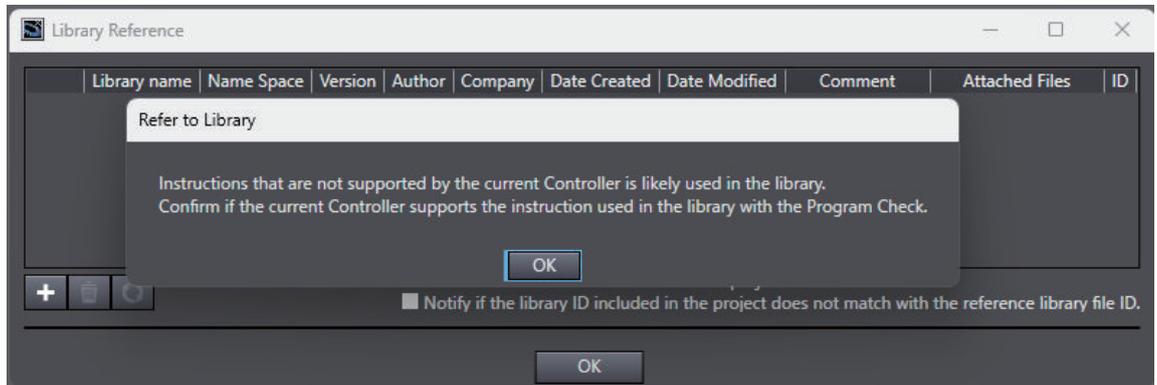




Precautions for Correct Use

Depending on the type of the Controller you are using, the following dialog will be displayed when adding the Sysmac Library. In this case, after adding the Sysmac Library, and then run Check all programs, and confirm that there are no errors in Build Tab Page about result of program check. There is no problem if there are no errors.

From the Main Menu, select **project - check all programs**.



1-2 How to use Sysmac Library in the CPU Unit or Industrial PC

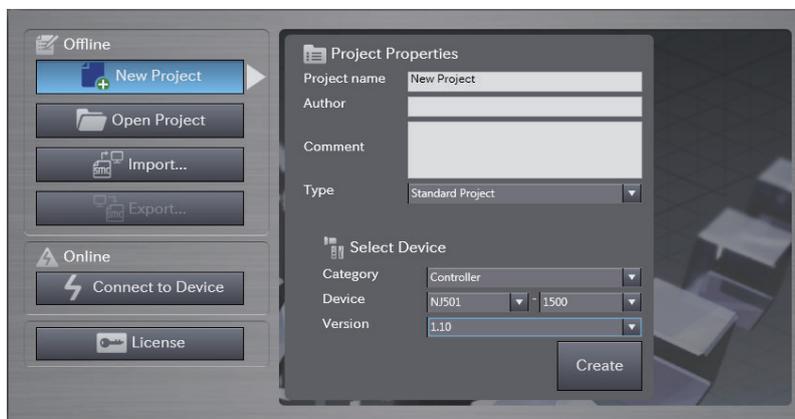
Even when Sysmac Library is not installed on your computer, you can use Sysmac Library by uploading it from the CPU Unit or Industrial PC to your computer.

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

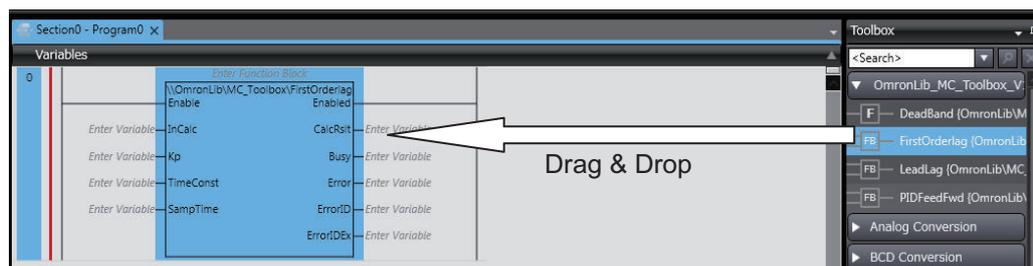
✓ Version Information

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

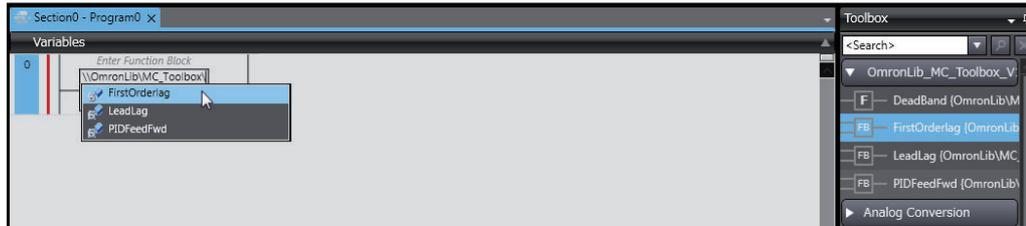
- 1 Start the Sysmac Studio and create a new project in which you want to use Sysmac Library.



- 2 Connect online to the CPU Unit or Industrial PC.
- 3 Upload the POUs in which Sysmac Library is used.
Now, when you select the Ladder Editor or ST Editor, the function blocks and functions included in the Sysmac Library used in the uploaded 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 Ladder Editor, select **Insert Function Block** in the menu, and enter the fully qualified name (¥¥namespace¥¥FBname).



Precautions for Correct Use

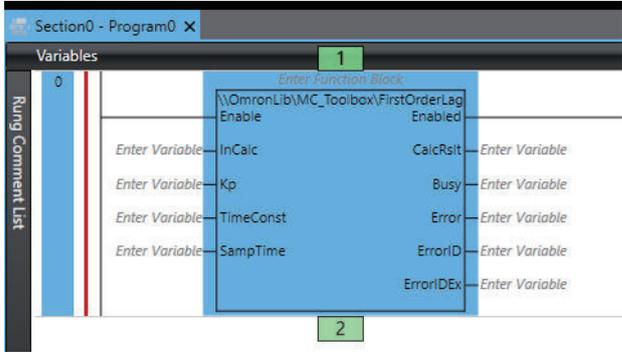
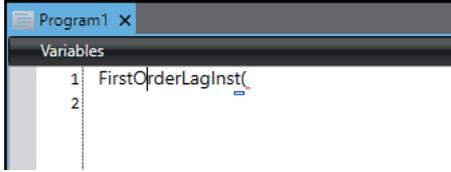
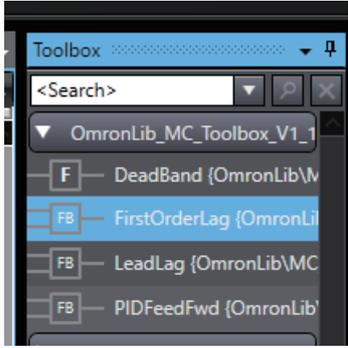
- The Sysmac Studio installs Sysmac Library library files to the specified folder on the computer if they are not present. However, the Sysmac Studio does not install libraries to the specified folder on the computer if they are present.
The specified folder here means the folder in which library files are installed by the installer.
- Note that uploading Sysmac Library from a CPU Unit or Industrial PC does not install the manual and help files for Sysmac Library, unlike installation using the installer. Please install the manual and help files using the installer if you need them.

1-3 Steps to display the Instruction Reference

You can display the Instruction Reference of Sysmac Library as well as the instructions in the NJ/NX/NY series. The contents of the Instruction Reference are the same as the explanations in this manual.

Display the instruction word reference using the following procedure.

- 1 Press the **F1** Key in the Ladder Editor, ST Editor, or Instruction Navigation.

case	step
Ladder Editor	<p>While selecting the Sysmac Library instruction for which you want to display help, press the F1 Key.</p> 
ST Editor	<p>While cursor on the Sysmac Library instruction^{*1} for which you want to display help, press the F1 Key.</p> 
Instruction Navigator	<p>While selecting the Sysmac Library instruction for which you want to display help, press the F1 Key.</p> 

*1. For function block type instructions, an instance of the instruction must be created and the cursor must be on the instruction instance.

Displays the Instruction Reference of Sysmac Library.

Function block name	Name	FB/FUN	Graphic expression	ST expression
FirstOrderLag	First Order Lag	FB		<pre>FB_FirstOrderLag_instance (Enable, InCalc, Kp, TimeConst, SampTime, Enabled, CalcRst, ErrorID, ErrorDEx);</pre>



Additional Information

The Instruction Reference can be displayed only when Sysmac Library is installed using the installer.

2

SLMP Communications Library

This section describes the shared specifications of each FB in the SLMP Communications Library.

2-1	Overview	2-2
2-1-1	Device Read	2-2
2-1-2	Device Write	2-3
2-1-3	Send Command	2-3
2-1-4	Support for MC Protocol	2-3
2-2	Usage Method.....	2-4
2-2-1	Preliminary Preparations for SLMP Communications	2-4
2-2-2	Preparing / Terminating Socket	2-4
2-2-3	Precautions for FB Execution.....	2-4
2-3	Set Values	2-5
2-3-1	Device Code (Device Access for Sequencer CPU Made by Mitsubishi Electric)	2-5

2-1 Overview

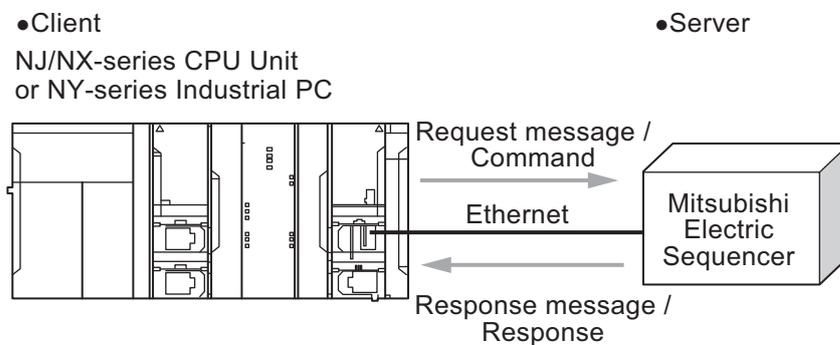
The SLMP Communication Library is a collection of software function objects for communicating with Mitsubishi Electric sequencers via SLMP.

Various FB types are provided for easy incorporation of the OMRON Machine Automation Controller in networks configured by the Mitsubishi Electric Sequencer.

This enables following to:

- A machine automation controller can monitor and control a system configured with Mitsubishi Electric sequencers.
- A machine automation controller can notify the own state to a system configured with Mitsubishi Electric sequencers.

An example system configuration is following figure.



This library provides three types of FBs (Device Read, Device Write, and Send Command). Each FBs are provided both TCP and UDP socket services. The specifications are as follows:



Precautions for Correct Use

UDP is a protocol that does not controls to ensure communication reliability, such as confirming the arrival of transmitted data.

To improve the reliability of communications when you use UDP sockets, create a user program that performs arrival confirmation and retransmission processing. Alternatively, consider using TCP sockets.



Version Information

- You can use the Send Command FB with version 2.0 or later of this library.
- You can use the method using UDP socket service with version 2.0 or later of this library.

2-1-1 Device Read

You can use this function to read values for the Connected Mitsubishi Electric sequencer's Device.

You can specify the Device type, first Device number, and number of Device points.

Select the FB to use according to the socket service method.

Types	FB using the TCP socket service	FB using the UDP socket service
Device Read	SLMP_DeviceRead	SLMP_DeviceRead_UDP

2-1-2 Device Write

You can use this function to write values for the specified Mitsubishi Electric sequencer's Device. You can specify the Device type, first Device number, and number of Device points. Select the FB to use according to writing Device type and the socket service method.

Types	FB using the TCP socket service	FB using the UDP socket service
Device write in WORD units	SLMP_DeviceWriteWord	SLMP_DeviceWriteWord_UDP
Device write in Bit units	SLMP_DeviceWriteBool	SLMP_DeviceWriteBool_UDP

2-1-3 Send Command

Users can send any commands to Mitsubishi Electric sequencers by creating request headers and request data.

You can specify whether or not to response.

This FB implements ST type (Single-Transmission) PDUs in 3E frames for client/server (single request) type and client-server (response not required) type communication models in SLMP.

Select the FB to use according to the socket service method.

Types	FB using the TCP socket service	FB using the UDP socket service
Send Command	SLMP_SendCmd	SLMP_SendCmd_UDP



Additional Information

For commands that can be sent to Mitsubishi Electric sequencers, see the destination sequencer's manual.

2-1-4 Support for MC Protocol

Since SLMP has the same message format as the MC protocol's QnA interchangeable 3E frame and 4E frame, this library can also be used for Mitsubishi Electric Sequencers supporting the MC protocol.

2-2 Usage Method

This section describes the usage method common to the FB provided in this library.

2-2-1 Preliminary Preparations for SLMP Communications

When performing SLMP communications, you must perform the setting to enable SLMP communications on the Mitsubishi Electric Sequencer side.

2-2-2 Preparing / Terminating Socket

When you use this library, pre-processes (preparing Socket, clearing the receive buffer) for performing Ethernet communications with Mitsubishi Electric Sequencer, and post-processing (terminating Socket) are required.

Perform communications preparations in the user program, and execute FB in conditions where communications with the Mitsubishi Electric Sequencer are enabled.

After completion of FB execution, perform the post-processing at a suitable timing.

Refer to the EtherNet/IP communications instructions below in the *NJ/NX-series Instructions Reference Manual (W508)* or *NY-series Instructions Reference Manual (W560)* for details on function blocks for Socket control.

Process	FB using the TCP socket service	FB using the UDP socket service
Preparing Socket	SktTCPConnect	SktUDPCreate
Clear the reception buffer	SktClearBuf	SktClearBuf
Terminating Socket	SktClose	SktClose

2-2-3 Precautions for FB Execution

The FBs provided in this library uses a handle that is created in preparing the socket.

To send multiple requests with one handle, send the next request after the previous request has been successfully sent.

2-3 Set Values

2-3-1 Device Code (Device Access for Sequencer CPU Made by Mitsubishi Electric)

The supported Devices for Device Read and Device Write in this library are as follows.
Use the device code listed in () to use 0001/0000 in the sub-command of the command message.
The shaded Devices with "N" in the *Support* column are not supported.

Support	DeviceCode	Device	Type	SLMP Device Code ASCII	SLMP Device Code Binary	Device No. range
N	0	Special relay (SM)	Bit	SM** (SM)	0x0091 (0x91)	Decimal
N	1	Special register (SD)	Word	SD** (SD)	0x00A9 (0xA9)	Decimal
Y	2	Input (X)	Bit	X*** (X*)	0x009C (0x9C)	Hexadecimal
Y	3	Output (Y)	Bit	Y*** (Y*)	0x009D (0x9D)	Hexadecimal
Y	4	Internal relay (M)	Bit	M*** (M*)	0x0090 (0x90)	Decimal
Y	5	Latch relay (L)	Bit	L*** (L*)	0x0092 (0x92)	Decimal
N	6	Annunciator (F)	Bit	F*** (F*)	0x0093 (0x93)	Decimal
N	7	Edge relay (V)	Bit	V*** (V*)	0x0094 (0x94)	Decimal
Y	8	Link relay (B)	Bit	B*** (B*)	0x00A0 (0xA0)	Hexadecimal
Y	9	Data register (D)	Word	D*** (D*)	0x00A8 (0xA8)	Decimal
Y	10	Link register (W)	Word	W*** (W*)	0x00B4 (0xB4)	Hexadecimal
N	11	Timer (T)	Contact (TS)	Bit	TS** (TS)	0x00C1 (0xC1)
N	12		Coil (TC)	Bit	TC** (TC)	0x00C0 (0xC0)
N	13		Current value (TN)	Word	TN** (TN)	0x00C2 (0xC2)
N	14	Long timer (LT)	Contact (LTS)	Bit	LTS* (—)	0x0051 (—)
N	15		Coil (LTC)	Bit	LTC* (—)	0x0050 (—)
N			Current value (LTN)	Double word	LTN* (—)	0x0052 (—)

Support	DeviceCode	Device	Type	SLMP Device Code ASCII	SLMP Device Code Binary	Device No. range	
N	17	Retentive timer (ST)	Contact (STS)	Bit	STS* (SS)	0x00C7 (0xC7)	Decimal
N	18		Coil (STC)	Bit	STC* (SC)	0x00C6 (0xC6)	Decimal
N	19		Current value (STN)	Word	STN* (SN)	0x00C8 (0xC8)	Decimal
N	20	Long retentive timer (LST)	Contact (LSTS)	Bit	LSTS (—)	0x0059 (—)	Decimal
N	21		Coil (LSTC)	Bit	LSTC (—)	0x0058 (—)	Decimal
N	22		Current value (LSTN)	Double word	LSTN (—)	0x005A (—)	Decimal
N	23	Counter (C)	Contact (CS)	Bit	CS** (CS)	0x00C4 (0xC4)	Decimal
N	24		Coil (CC)	Bit	CC** (CC)	0x00C3 (0xC3)	Decimal
N	25		Current value (CN)	Word	CN** (CN)	0x00C5 (0xC5)	Decimal
N	26	Long counter (LC)	Contact (LCS)	Bit	LCS* (—)	0x0055 (—)	Decimal
N	27		Coil (LCC)	Bit	LCC* (—)	0x0054 (—)	Decimal
N	28		Current value (LCN)	Double word	LCN* (—)	0x0056 (—)	Decimal
N	29	Link special relay (SB)	Bit	SB** (SB)	0x00A1 (0xA1)	Hexadecimal	
N	30	Link special register (SW)	Word	SW** (SW)	0x00B5 (0xB5)	Hexadecimal	
N	31	Direct access input (DX)	Bit	DX** (DX)	0x00A2 (0xA2)	Hexadecimal	
N	32	Direct access output (DY)	Bit	DY** (DY)	0x00A3 (0xA3)	Hexadecimal	
N	33	Index register (Z)	Word	Z*** (Z*)	0x00CC (0xCC)	Decimal	
N	34	Long index register (LZ)	Double word	LZ** (—)	0x0062 (—)	Decimal	
N	35	File register (R)	Word	R*** (R*)	0x00AF (0xAF)	Decimal	
Y	36	File register (ZR)	Word	ZR** (ZR)	0x00B0 (0xB0)	Hexadecimal	
Y	37	Extended data register (D)	Word	— (D*)	— (0xA8)	Decimal	
Y	38	Extended link register (W)	Word	— (W*)	— (0xB4)	Hexadecimal	

Support	DeviceCode	Device	Type	SLMP Device Code ASCII	SLMP Device Code Binary	Device No. range
N	39	Refresh data register (RD)	Word	RD** (—)	0x002C (—)	Decimal
N	40	Link direct device				
N	41	Module access device				
N	42	CPU buffer memory access device				

The table below shows whether read/write is enabled or not for a device type.

Types	FB	Bit device	Word device
Device Read (Word access)	SLMP_DeviceRead	Enabled	Enabled
	SLMP_DeiceRead_UDP	Enabled	Enabled
Device Read (Bit access)	SLMP_DeviceRead	Enabled	Disabled*1
	SLMP_DeiceRead_UDP	Enabled	Disabled*1
Device Write (Word access)	SLMP_DeviceWriteWord	Enabled	Enabled
	SLMP_DeiceWriteWord_UDP	Enabled	Enabled
Device Write (Bit access)	SLMP_DeviceWriteBool	Enabled	Disabled*1
	SLMP_DeiceWriteBool_UDP	Enabled	Disabled*1

*1. If you execute this combination, the FB will produce an error (illegal device code).

3

Common Specifications of Function Blocks

This section describes the shared specifications of each FB in the Sysmac Library.

3-1	Common Variables	3-2
3-1-1	Definition of Input Variables and Output Variables	3-2
3-1-2	Execute-type Function Blocks	3-3
3-1-3	Enable-type Function Blocks	3-5
3-2	Precautions	3-8
3-2-1	Nesting	3-8
3-2-2	Instruction Options	3-8
3-2-3	Re-execution of Function Blocks	3-8

3-1 Common Variables

This section describes the specifications of variables (EN, Execute, Enable, Abort, ENO, Done, CalcRslt, Enabled, Busy, CommandAborted, Error, ErrorID, and ErrorIDEx) that are used for more than one function or function block. The specifications are described separately for functions, for execute-type function blocks, and for enable-type function blocks.

3-1-1 Definition of Input Variables and Output Variables

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

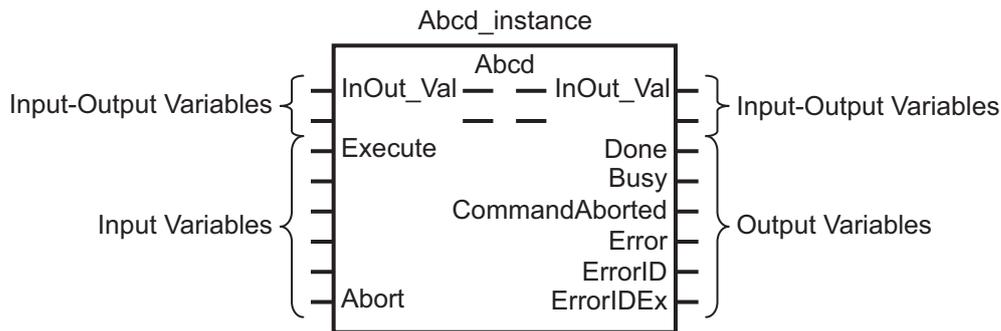
Variable	I/O	Data type	Function/function block type to use			Meaning	Definition
			Function block		Function		
			Execute-type	Enable-type			
EN	Input	BOOL			OK	Execute	The processing is executed while the variable is TRUE.
Execute		BOOL	OK			Execute	The processing is executed when the variable changes to TRUE.
Enable		BOOL		OK		Run	The processing is executed while the variable is TRUE.
Abort		BOOL	OK			Abort	The processing is aborted. You can select the aborting method.

Variable	I/O	Data type	Function/function block type to use			Meaning	Definition
			Function block		Function		
			Execute-type	Enable-type			
ENO	Output	BOOL			OK	Done	The variable changes to TRUE when the processing ends normally. It is FALSE when the processing ends in an error, the processing is in progress, or the execution condition is not met.
Done		BOOL	OK			Done	The variable changes to TRUE when the processing ends normally. It is FALSE when the processing ends in an error, the processing is in progress, or the execution condition is not met.
Busy		BOOL	OK	OK		Executing	The variable is TRUE when the processing is in progress. Turns to FALSE while the process is not being executed.
CalcRslt		LREAL		OK		Calculation Result	The calculation result is output.
Enabled		BOOL		OK		Enabled	The variable is TRUE when the output is enabled. It is used to calculate the control amount for motion control, temperature control, etc.
Command Aborted		BOOL	OK			Command Aborted	The variable changes to TRUE when the processing is aborted. It changes to FALSE when the processing is executed the next time again.
Error		BOOL	OK	OK		Error	This variable is TRUE while there is an error. It is FALSE when the processing ends normally, the processing is in progress, or the execution condition is not met.
ErrorID		WORD	OK	OK		Error Code	An error code is output.
ErrorIDEx		DWORD	OK	OK		Expansion Error Code	An expansion error code is output.

3-1-2 Execute-type Function Blocks

- Processing starts when Execute changes to TRUE.

- When Execute changes to TRUE, Busy also changes to TRUE. When processing is completed normally, Busy changes to FALSE and Done changes to TRUE.
- When continuously executing function blocks of the same instance, change the next Execute to TRUE for at least one task period after Done changes to FALSE in the previous execution.
- If the function block has a CommandAborted (Instruction Aborted) output variable and processing is aborted, CommandAborted changes to TRUE and Busy changes to FALSE.
- If an error occurs in the function block, Error changes to TRUE and Busy changes to FALSE.
- For function blocks that output the result of calculations for motion control and temperature control, you can use the BOOL input variable Abort to abort the FB process. When Abort changes to TRUE, CommandAborted changes to TRUE and the execution of the function block is aborted.

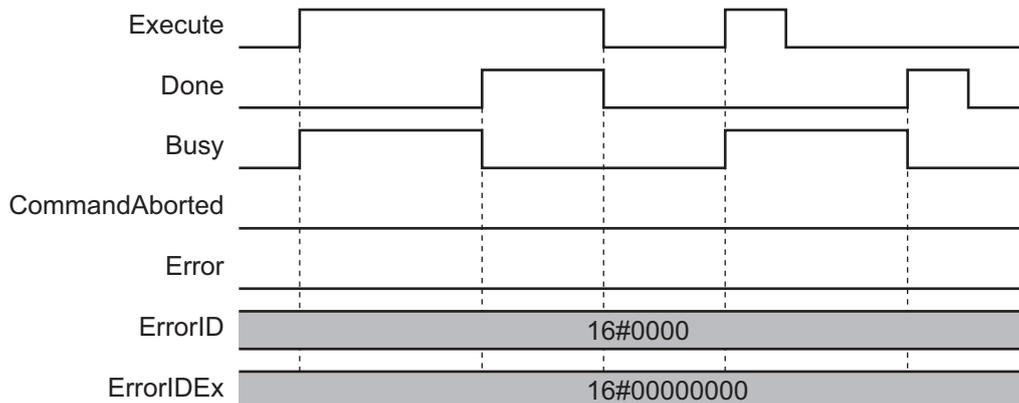


- If Execute is TRUE and Done, CommandAborted, or Error changes to TRUE, Done, CommandAborted, or Error changes to FALSE when Execute is changed to FALSE.
- If Execute is FALSE and Done, CommandAborted, or Error changes to TRUE, Done, CommandAborted, or Error changes to TRUE for only one task period.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

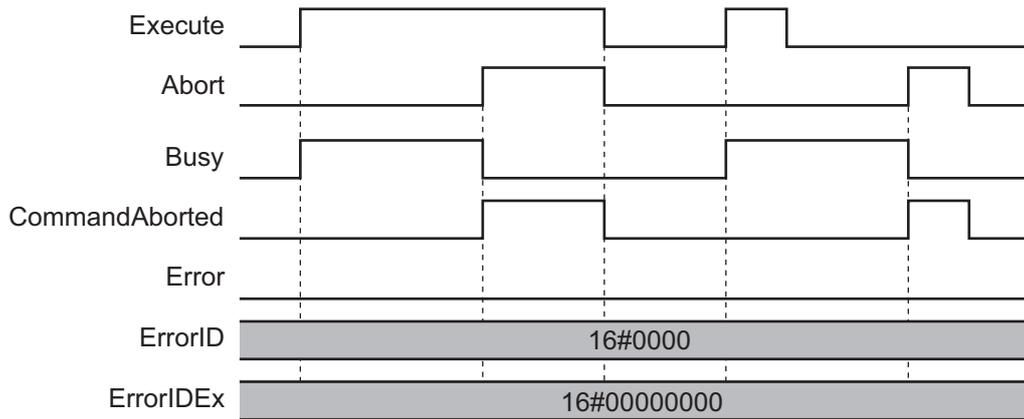
Timing Chart

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

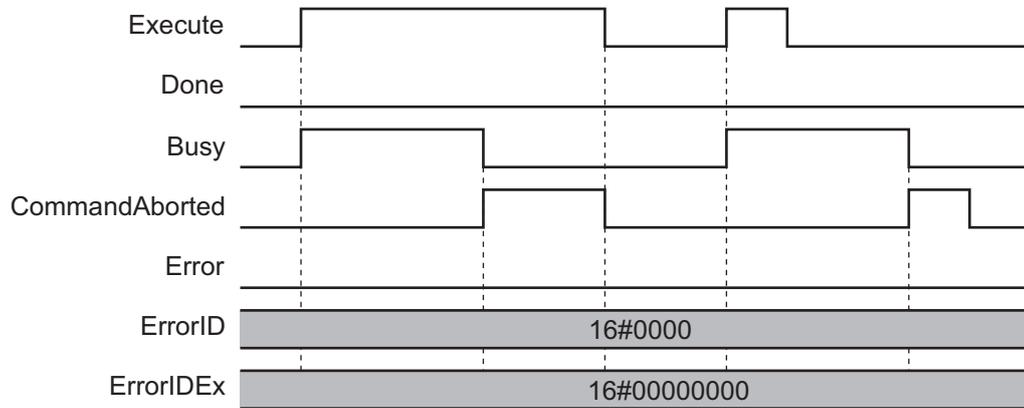
● Normal End



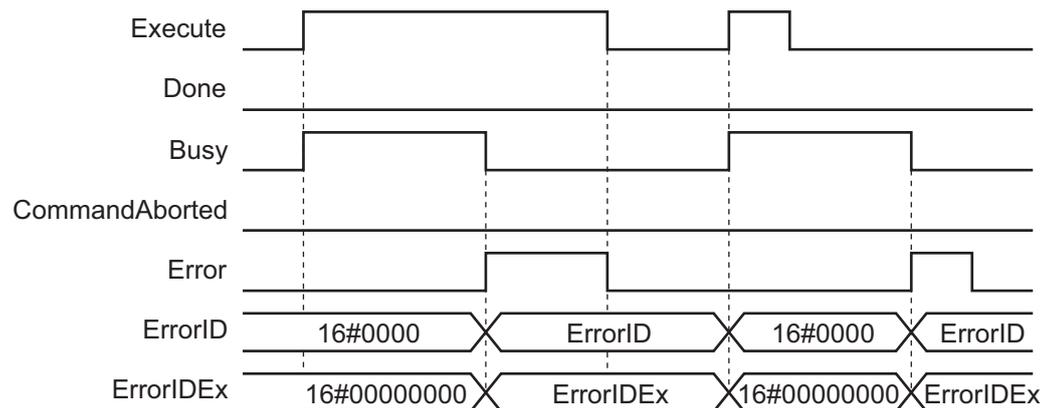
● **Canceled Execution**



● **Aborted Execution**



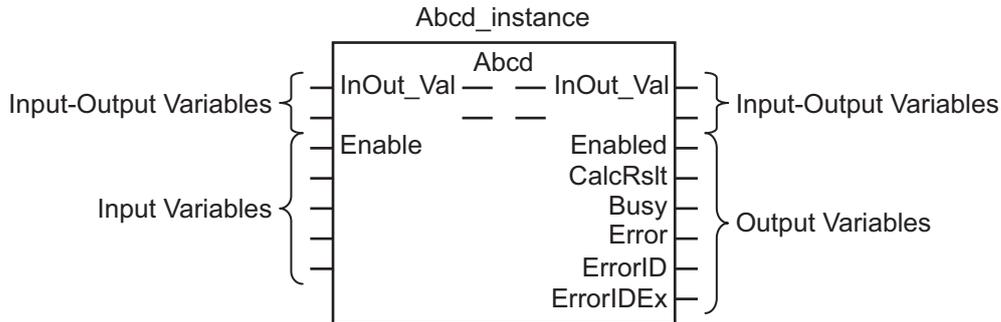
● **Errors**



3-1-3 Enable-type Function Blocks

- Processing is executed while Enable is TRUE.
- When Enable changes to TRUE, Busy also changes to TRUE. Enabled is TRUE during calculation of the output value.

- If an error occurs in the function block, Error changes to TRUE and Busy and Enabled change to FALSE. When Enable changes to FALSE, Enabled, Busy, and Error change to FALSE.

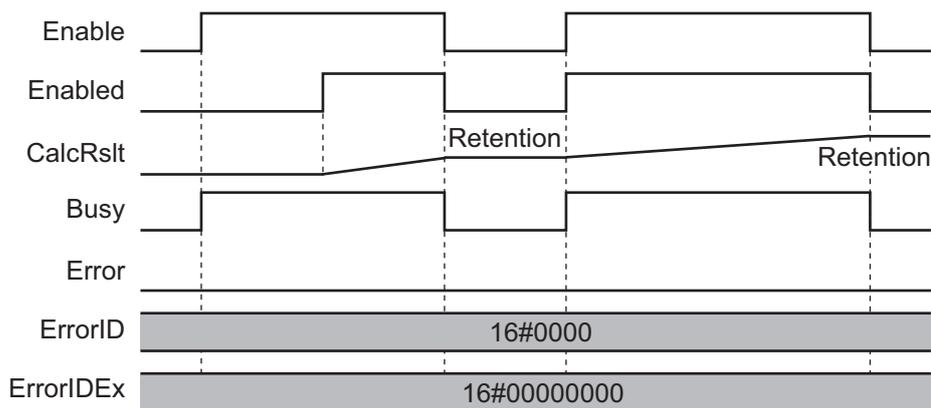


- If an error occurs in the function block, the relevant error code and expansion error code are set in ErrorID (Error Code) and ErrorIDEx (Expansion Error Code). The error codes are retained even after Error changes to FALSE, but ErrorID is set to 16#0000 and ErrorIDEx is set to 16#0000 0000 when Execute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

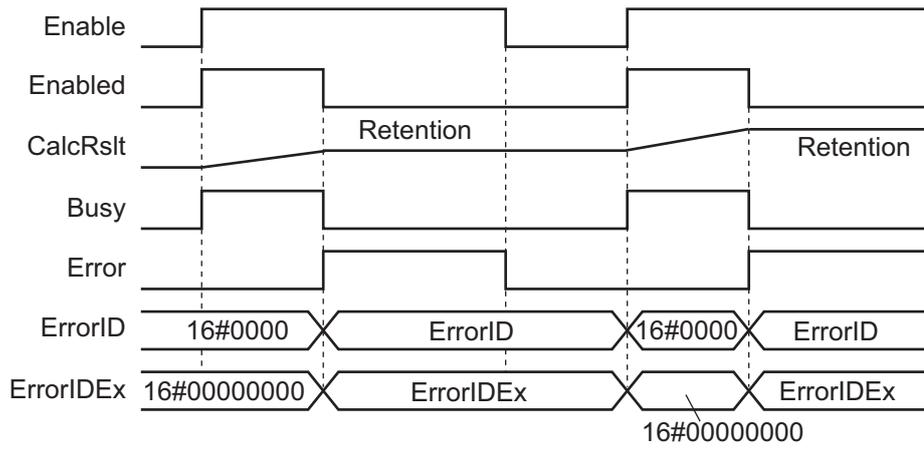
Timing Charts

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

● Normal End



● **Errors**



3-2 Precautions

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

3-2-1 Nesting

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

Refer to *NJ/NX-series CPU Unit Software User's Manual (W501)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Software User's Manual (W558)* for details on the nesting function block.

3-2-2 Instruction Options

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

3-2-3 Re-execution of Function Blocks

Execute-type function blocks cannot be re-executed by the same instance.

If you do so, the output value will be the initial value.

Refer to *NJ/NX-series CPU Unit Motion Control User's Manual (W507)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Motion Control User's Manual (W559)* for details on re-execution.

4

Individual Specifications of Function Blocks

This section describes the individual specifications of each FB in the SLMP Communications Library.

SLMP_DeviceRead	4-2
SLMP_DeviceWriteWord.....	4-12
SLMP_DeviceWriteBool	4-21
SLMP_SendCmd.....	4-30
SLMP_DeviceRead_UDP	4-43
SLMP_DeviceWriteWord_UDP	4-53
SLMP_DeviceWriteBool_UDP	4-62
SLMP_SendCmd_UDP	4-71

SLMP_DeviceRead

Through SLMP communication using TCP socket service, to read Device of Mitsubishi Electric Sequencer.

Function block name	Name	FB/FUN	Graphic expression	ST expression
SLMP_DeviceRead	Internal memory Batch read (TCP)	FB	<p style="text-align: center;">SLMP_DeviceRead_instance</p>	<pre>SLMP_DeviceRead_instance(Execute:=, Socket:=, Mode:=, BitAccess:=, DeviceCode:=, Offset:=, Size:=, TimeOut:=, Done=>, Busy=>, Error=>, ErrorID=>, ErrorIDEx=>, ReadSize=>, ReadDat:=);</pre>

Function Block and Function Information

Item	Description
Library file name	OmronLib_SLMP_Comm_Vx_x.slr (x shows the unit version.)
Namespace	OmronLib\SLMP_Comm
Function block and function number	00169
Source code published/not published	Not Published

Input Variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Execute	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	—	FALSE
Socket	Socket	_sSOCKET*1	Socket	—	—	—

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Mode	Mode	BOOL	SLMP frame communicating code TRUE: ASCII mode FALSE: BINARY mode	TRUE, FALSE	—	FALSE
BitAccess	Bit access specification	BOOL	The units of read Device TRUE: Bit unit FALSE: Word unit	TRUE, FALSE	—	FALSE
Device-Code	Device code	UINT	Input the Device Code for the read target.*2	Follow the specifications of the Connected station	—	UINT#0
Offset	First device No.	UDINT	Input the first number of the read Device.*3	Follow the specifications of the Connected station	—	UDINT#0
Size	Number of device points	UINT	Input the number of read Device.*3	Follow the specifications of the Connected station	—	UINT#0
TimeOut	Timeout time	UINT	Input the timeout time for FB. If "0" is set, the timeout time is 2.0 s.	Depends on data type.	0.1s	UINT#0

*1. Refer to the SktTCPConnect in the *NJ/NX-series Instructions Reference Manual (W508)* or *NY-series Instructions Reference Manual (W560)* for details of the Data type.

*2. Refer to *2-3-1 Device Code (Device Access for Sequencer CPU Made by Mitsubishi Electric)* on page 2-5 for details.

*3. Refer to *Function* on page 4-4 for details.

Output Variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Done	Done	BOOL	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE, FALSE	—	—
Busy	Executing	BOOL	TRUE: Executing. FALSE: Not executing.	TRUE, FALSE	—	—
Error	Error	BOOL	TRUE: Monitoring in progress. FALSE: Normal end, executing, or execution condition not met	TRUE, FALSE	—	—
ErrorID	Error Code	WORD	This is the error ID for an error end. The value is 16#0 for a normal end.	*1	—	—
ErrorIDEx	Expansion Error Code	DWORD	This is the error ID for an Expansion Unit Hardware Error. The value is 16#0 for a normal end.	*1	—	—

Variable	Meaning	Data type	Description	Valid range	Unit	Default
ReadSize	Read data size	UINT	The actual read-out data size is outputted.	Depends on data type.	Bytes	—

*1. Refer to the *Troubleshooting* on page 4-8 for details.

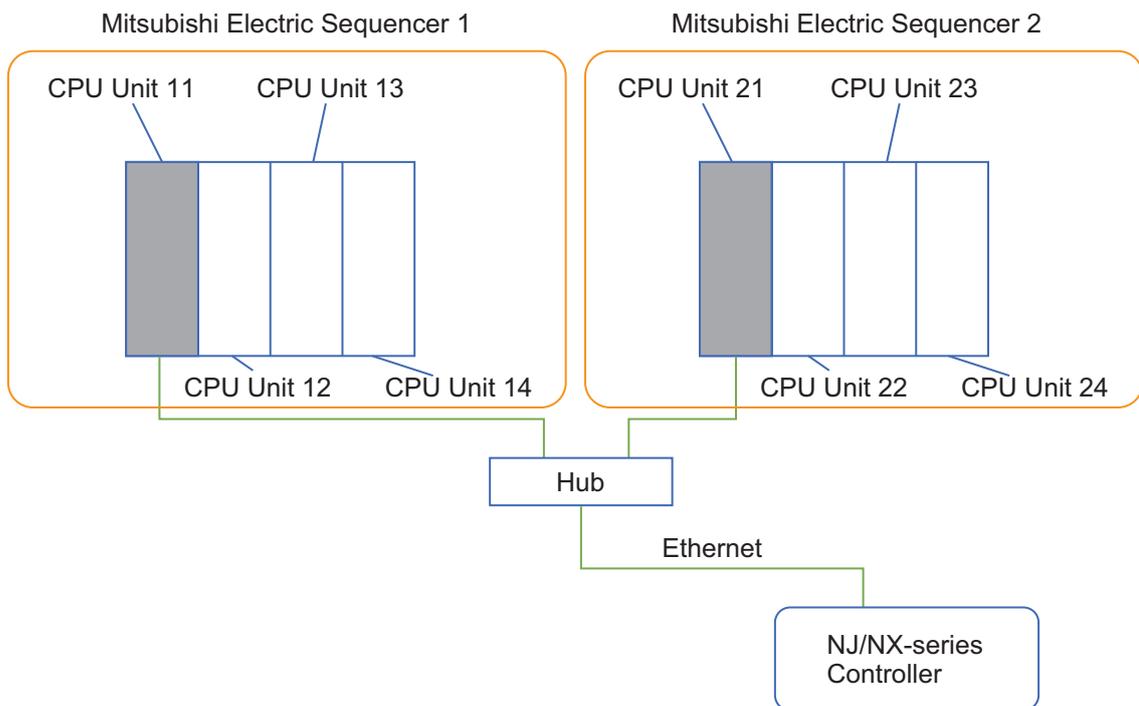
Input-Output Variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
ReadDat	Read data	ARRAY[*] OF BYTE *1	The actual read-out data is outputted. *2*3	Depends on data type.	—	—

- *1. You can input array variables of any length BYTE type. Array length is in the range from 1 to 1920. Arrays can start with either element 0 or with element n. You cannot input BYTE type variables of the base data type.
- *2. It is outputted in the communicating code specified in *Mode*. Conversion from BINARY to ASCII and conversion from ASCII to BINARY are not executed.
- *3. If the value of the output variable *ReadSize* exceeds the number of elements in the array, the excess data is discarded.

Function

- This FB sends an ST (Single-Transmission) type request message for SLMP communication.
- The supported communication model by ST type is client/server (single request) type.
- Use SLMP communications to read the sequencer device made by Mitsubishi Electric on the connected station.
- In the image below, the Mitsubishi Electric Sequencers that can read-out are CPU Unit 11 and CPU Unit 21. CPU Units 12 to 14 and CPU Units 22 to 24 do not support read-out.



- Reads out the amount of the input variable *Size* from the input variable *Offset* from the Device specified by the input variable *DeviceCode*.

- The SLMP request message is based on the input variable *Mode* and *BitAccess*, and created as Read (Command: 0401, Sub-command: 0001/0000).
- The read result is outputted to the in-out variable *ReadDat* with the SLMP response data "resData" array as is. The output size is outputted to the output variable *ReadSize* in byte units .
- Since the data type of the input variable *Offset* is UDINT, the base of the data is decimal number, decimal data. When specifying a Device with a hexadecimal number, convert the input value to a decimal number or enter a hexadecimal constant (literal), etc.
- The possible input range for the input variable *Size*, based on the Input variable *Mode* and input variable *BitAccess*, and the output variable *ReadSize* output range are as shown below.
When *Size* is set to 0, the command will be completed without sending command (the output variable *Done* is TRUE). In that case, 0 is outputted to *ReadSize*, and the in-out variable *ReadDat* is not updated.

Variable name	Conditions		
	BitAccess	Mode	
		FALSE (BINARY mode)	TRUE (ASCII mode)
Size	TRUE (When read in Bit units)	1 - 3840 (points)	1 - 1920 (points)
ReadSize		1 - 1920 (Bytes)	1 - 1920 (Bytes)
Size	FALSE (When read in Word units)	1 - 960 (points)	1 - 480 (points)
ReadSize		2 - 1920 (Bytes)	4 - 1920 (Bytes)

Example of 8 Bit Read-Out from M100 to M107 in Bit Units

In case of M100 to M107 are 0/0/0/1/0/0/1/1

- BINARY mode:

Size 8

ReadDat	[0]	[1]	[2]	[3]
	0x00	0x01	0x00	0x11
	M100/M101	M102/M103	M104/M105	M106/M107

ReadSize 4

- ASCII mode:

Size 8

ReadDat	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
	0	0	0	1	0	0	1	1
	0x30	0x30	0x30	0x31	0x30	0x30	0x31	0x31
	M100		...					M107

ReadSize 8

Example of 2 Word Read-Out from M100 to M131 (Bit Device) in Word Units

In case of M100 to M131 are 0010/1100/0100/1000/0100/0000/0000/0000

- BINARY mode:

Size 2

ReadDat	[0]	[1]	[2]	[3]
	0x34	0x12	0x02	0x00
	M107...M100	M115...M108	M123...M116	M131...M124

ReadSize 4

- ASCII mode:

Size 2

ReadDat	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
	1	2	3	4	0	0	0	2
	0x31	0x32	0x33	0x34	0x30	0x30	0x30	0x32
	M115	M100	M131	M116		

ReadSize 8

Example of 3 Word Read-Out from D100 to D102 (Word Device) in Word Units

In case of D100 is 0x1234, D101 is 0x0002, D102 is 0x1DEF

- BINARY mode:

Size 3

ReadDat	[0]	[1]	[2]	[3]	[4]	[5]
	0x34	0x12	0x02	0x00	0xEF	0x1D
	D100		D101		D102	

ReadSize 6

- ASCII mode:

Size 3

ReadDat	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
	1	2	3	4	0	0	0	2	1	D	E	F
	0x31	0x32	0x33	0x34	0x30	0x30	0x30	0x32	0x31	0x44	0x45	0x46
	D100			D101			D102					

ReadSize 12

Timing Chart

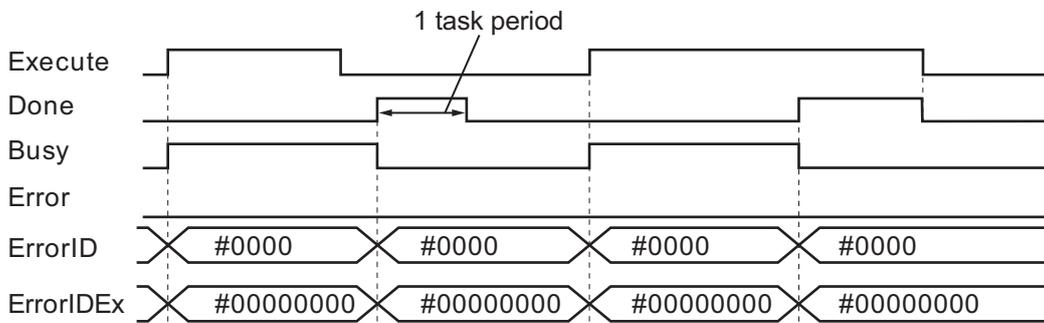
The timing charts are shown below.

- *Busy* (Executing) changes to TRUE when *Execute* changes to TRUE.
- When a response message from the Mitsubishi Electric Sequencer is received normally, and End-Code=0, *Done* changes to TRUE.
- If an error occurs when execution of the function block is in progress, *Error* changes to TRUE and *Busy* (Executing) changes to FALSE.

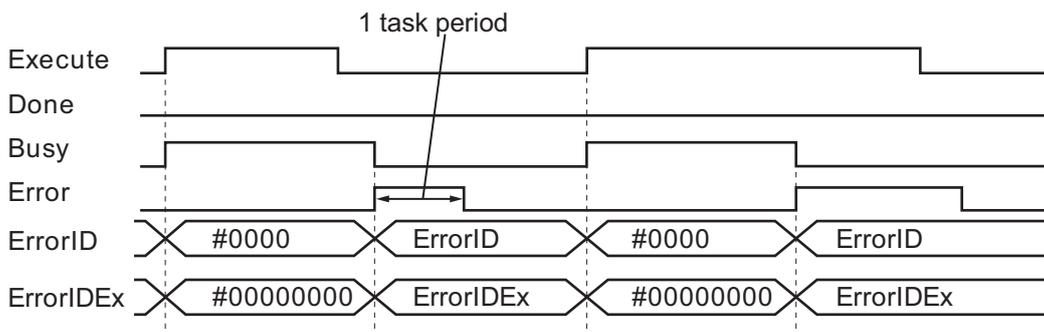
You can find out the cause of the error by accessing the values output to *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code).

- If *Execute* changes to FALSE before execution of the FB is ended, *Done* and *Error* are TRUE only for one task period.
- If *Execute* remains TRUE even after execution of the function block is ended, the output values of *Done* and *Error* are retained.

- Timing Chart for Normal End



- Timing Chart for Error End



Additional Information

- For this FB, use the socket service function. Refer to *NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual (W506)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP™ Port User's Manual (W563)* for details of the socket service function.

Precautions for Correct Use

- Execution of this instruction is continued until processing is completed even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is completed. Use this to confirm normal completion of processing.
- This FB executes one SktTCPRcv instruction and one SktTCPSend instruction internally. For the number of simultaneous executions, refer to the *NJ/NX-series Instructions Reference Manual (W502)* or *NY-series Instructions Reference Manual (W560)* SktTCPRcv instruction and SktTCPSend instruction.
- When the same socket is shared by FBs of multiple SLMP communications instructions, multiple SLMP communications instructions cannot be executed simultaneously. SLMP communications instructions cannot distinguish responses to commands sent by themselves from responses to commands sent by other SLMP communications instructions. Therefore, wait until the execution of one SLMP communications instruction is complete and then execute the next SLMP communications instruction.



Precautions for Correct Use

Refer to 2-2 *Usage Method* on page 2-4 for details.

Troubleshooting

Error code	Expansion error code	Error name	Description	Corrective action
16#0000	16#00000000	Normal end	—	—

Error code	Expansion error code	Error name	Description	Corrective action	
16#0400	16#00000000	Input Value Out of Range	An input parameter for an instruction exceeded the valid range for an input variable. Or, division by an integer of 0 occurred in division or remainder calculations.	Refer to the description of <i>Machine Automation Controller Troubleshooting Manual (W503)</i> for the event code with 5401 appended to the upper 4 digits of the Error code. For example, if the Error code is 16#0400, refer to the description of event code 54010400 hex.	
16#0406	16#00000000	Illegal Data Position Specified	A memory address or data size that was specified for the instruction is not suitable.		
16#0407	16#00000000	Data Range Exceeded	The results of instruction processing exceeded the data area range of the output parameter.		
16#0419	16#00000000	Incorrect Data Type	A data type that cannot be used for an instruction is specified for an input or in-out variable.		
16#2003	16#00000000	Socket Status Error	The status was not suitable for execution of the socket service instruction.		
16#2006	16#00000000	Socket Timeout	A timeout occurred for a socket service instruction.		
16#2007	16#00000000	Socket Handle Out of Range	The handle that is specified for the socket service instruction is not correct.		
16#2008	16#00000000	Socket Communications Resource Overflow	The maximum resources that you can use for socket service instructions at the same time was exceeded.		
16#3CEA	16#00000001	Illegal device code	This function block specified an unsupported Device Code.		Check the FB input variable <i>DeviceCode</i> .
16#3CEA	16#00000002	Illegal first device No.	This FB specified an unsupported first Device No.		Check the FB input variable <i>Offset</i> .
16#3CEA	16#00000003	Illegal number of device points	Using a combination of the input variables <i>BitAccess</i> and <i>Size</i> , the read data size exceeded the controller specification upper limit value.	Check the FB input variables <i>BitAccess</i> and <i>Size</i> .	
16#3CEA	16#00000004	ReadDat upper limit exceeded	The array size allocated to in-out variable <i>ReadDat</i> exceeds the upper limit.	Confirm the number of elements in the array allocated to the In-out variable <i>ReadDat</i> in this FB is 1 to 1920.	

Error code	Expansion error code	Error name	Description	Corrective action
16#3CEA	16#00000010	Response re-ception timeout	A response message could not be received within the time specified in input variable <i>TimeOut</i> .	Check the network status and communications target Mitsubishi Electric Sequencer status. Or, increase the value of the input variable <i>TimeOut</i> of this FB.
16#3CEA	16#FFFF0001 - 16#FFFFFFFF	SLMP end code	An error response message was received.	Refer to <i>End codes</i> in the SLMP Specifications Overview for troubleshooting. For information on how to obtain the SLMP Specifications Overview, contact the CC-Link Partner Association.

Sample Programming

From our website (http://www.ia.omron.com/sysmac_library/), you can download sample programs using SLMP_DeviceRead. You can use with MELSEC iQ-R Series, iQ-F Series, Q Series, and L Series.



Precautions for Correct Use

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

The overview of the sample programming is as follows:

Conditions

Item	Description
Process	Read-Out 8 Bit from M100 to M107 in Bit Units
Destination IP Address	192.168.250.39
Destination TCP port number	5100
Source TCP port number.	Automatically assigned from free ports
Communication code	BINARY
Timeout time	2 seconds (default)

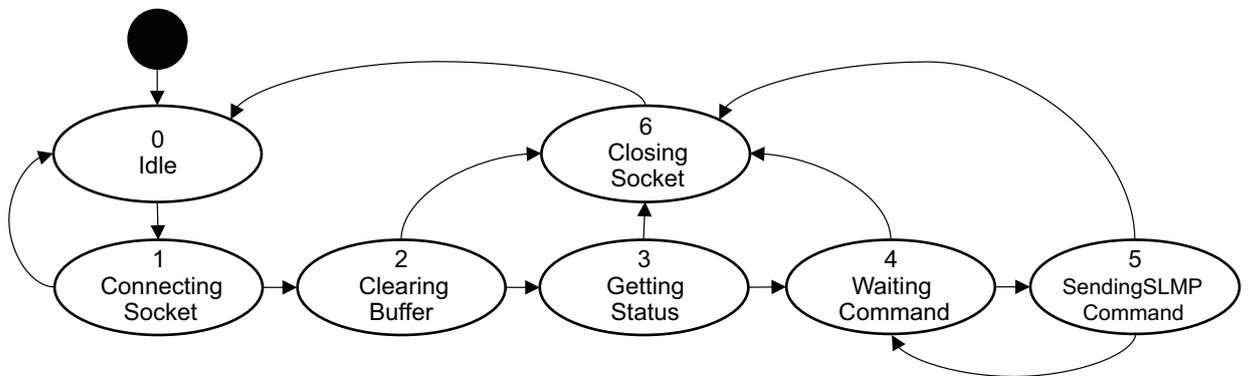
Process

- 1 Execute the SktTCPConnect instruction, and establish the TCP connection.

- 2** Execute the SktClearBuf command, and clear the TCP socket receiving buffer.
- 3** Confirm that no other communications FB execution is in progress.
- 4** Start up the SLMP_DeviceRead_instance.
- 5** After completing FB execution, check the read data.
- 6** Execute the SktClose instruction, and close the TCP connection.

State transition of sample program

The state transition of the sample programming is as follows:



No	State name	Processing
0	Idle	This is the state waiting for the request of establishment TCP connection. The TCP connection has not been established and is in a waiting state.
1	ConnectingSocket	This is the state establishes a TCP connection with the communication destination station.
2	ClearingBuffer	This is the state to clear the receive buffer of the TCP socket.
3	GettingStatus	This is the state checking the state of TCP connection.
4	WatingCommand	This is the state waiting for a request to send a command for SLMP communication or a request to disconnect the TCP connection. The TCP connection has been established and is in a waiting state.
5	SendingSLMPCommand	This is the state sends SLMP communication commands and receives responses to the destination station.
6	ClosingSocket	This is the state closing the TCP connection.

SLMP_DeviceWriteWord

Through SLMP communication using TCP socket service, to write Device of Mitsubishi Electric Sequencer in word units.

Function block name	Name	FB/FUN	Graphic expression	ST expression
SLMP_DeviceWriteWord	Internal memory batch write (Word, TCP)	FB	<p style="text-align: center;">SLMP_DeviceWriteWord_instance</p>	<pre>SLMP_DeviceWriteWord_instance(Execute:=, Socket:=, Mode:=, DeviceCode:=, Offset:=, Size:=, TimeOut:=, Done=>, Busy=>, Error=>, ErrorID=>, ErrorIDEx=>, WriteDat:=);</pre>

Function Block and Function Information

Item	Description
Library file name	OmronLib_SLMP_Comm_Vx_x.slr (x shows the unit version.)
Namespace	OmronLib\SLMP_Comm
Function block and function number	00170
Source code published/not published	Not Published

Input Variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Execute	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	—	FALSE
Socket	Socket	_sSOCKET*1	Socket	—	—	—
Mode	Mode	BOOL	SLMP frame communicating code TRUE: ASCII mode FALSE: BINARY mode	TRUE, FALSE	—	FALSE

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Device-Code	Device code	UINT	Input the Device Code for the write target.* ²	Follow the specifications of the Connected station	—	UINT#0
Offset	First device No.	UDINT	Input the first number of the write Device.* ³	Follow the specifications of the Connected station	—	UDINT#0
Size	Number of device points	UINT	Input the number of write Device.* ³	Follow the specifications of the Connected station	—	UINT#0
TimeOut	Timeout time	UINT	Input the timeout time for FB. If "0" is set, the timeout time is 2.0 s.	Depends on data type.	0.1s	UINT#0

*1. Refer to the SktTCPConnect in the *NJ/NX-series Instructions Reference Manual (W508)* or *NY-series Instructions Reference Manual (W560)* for details of the Data type.

*2. Refer to *2-3-1 Device Code (Device Access for Sequencer CPU Made by Mitsubishi Electric)* on page 2-5 for details.

*3. Refer to *Function* on page 4-14 for details.

Output Variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Done	Done	BOOL	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE, FALSE	—	—
Busy	Executing	BOOL	TRUE: Executing. FALSE: Not executing.	TRUE, FALSE	—	—
Error	Error	BOOL	TRUE: Error end. FALSE: Normal end, executing, or execution condition not met	TRUE, FALSE	—	—
ErrorID	Error Code	WORD	This is the error ID for an error end. The value is 16#0 for a normal end.	*1	—	—
ErrorIDEx	Expansion Error Code	DWORD	This is the error ID for an Expansion Unit Hardware Error. The value is 16#0 for a normal end.	*1	—	—

*1. Refer to the *Troubleshooting* on page 4-18 for details.

Input-Output Variables

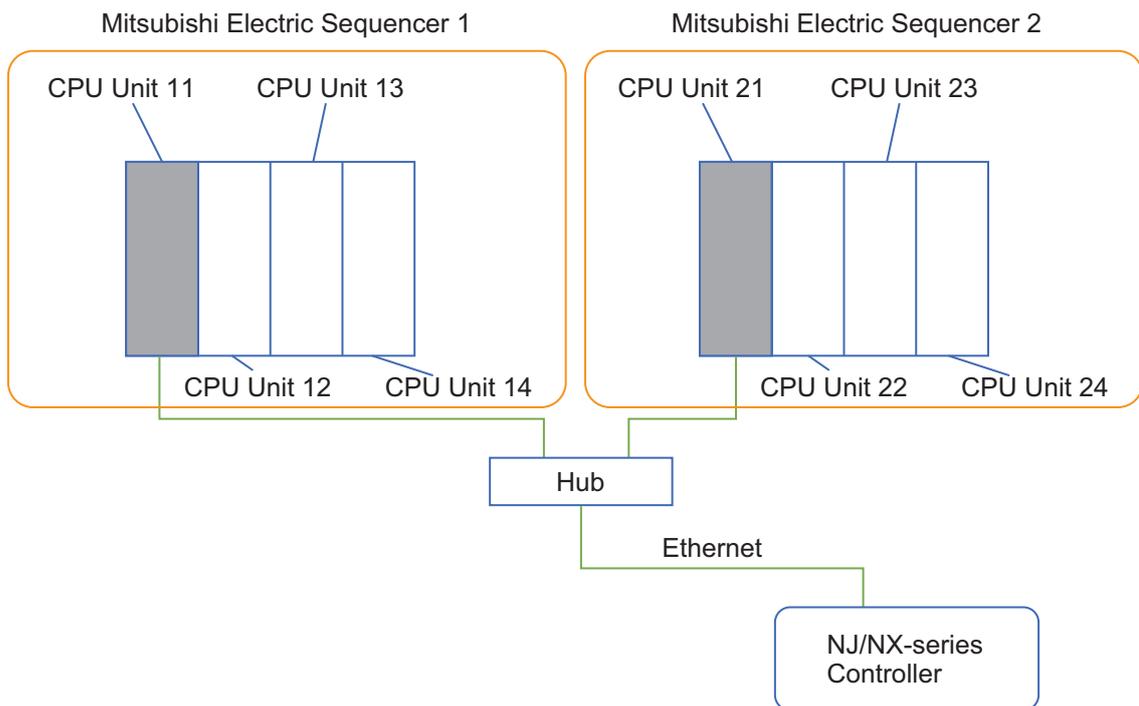
Variable	Meaning	Data type	Description	Valid range	Unit	Default
WriteDat	Write data	ARRAY[*] OF WORD*1	Input the write data. *2	Depends on data type.	—	—

*1. You can input array variables of any length WORD type. The number of elements in the array ranges from 1 to 960 (when in BINARY mode) and 1 to 480 (when in ASCII mode). WORD arrays can start with either element 0 or with element n. You cannot input WORD type variables of the base data type.

*2. You should input in the communicating code specified in input variable *Mode*. Conversion from BINARY to ASCII and conversion from ASCII to BINARY are not executed.

Function

- This FB sends an ST (Single-Transmission) type request message for SLMP communication.
- The supported communication model by ST type is client/server (single request) type.
- Use SLMP communications to write in word units to the sequencer device made by Mitsubishi Electric on the connected station.
- In the image below, the Mitsubishi Electric Sequencers that can write-out are CPU Unit 11 and CPU Unit 21. CPU Units 12 to 14 and CPU Units 22 to 24 do not support write-out.



- Writes the amount of the input variable *Size* from the input variable *Offset* to the Device specified by the input variable *DeviceCode*.
- The SLMP request message is based on the input variable *Mode*, and created as Write (Command: 1401, Sub-command: 0000).
- Write data is stored in the in-out variable *WriteDat*.
- Since the data type of the input variable *Offset* is UDINT, the base of the data is decimal number, decimal data. When specifying a Device with a hexadecimal number, convert the input value to a decimal number or enter a hexadecimal constant (literal), etc.

- The possible input range for the input variable *Size*, based on the input variable *Mode* and input variable *BitAccess*, and the SLMP request message write data length are as shown below. When *Size* is set to 0, the command will be completed without sending command (the output variable *Done* is TRUE).

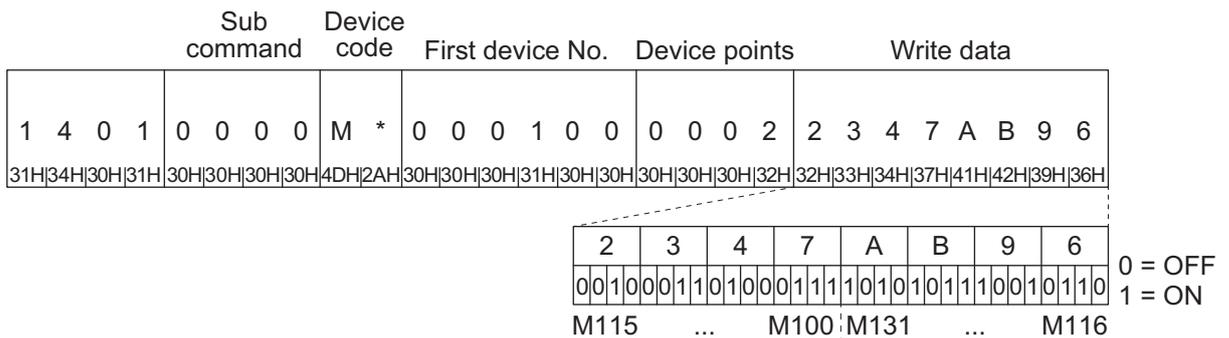
Item	Conditions	
	Mode	
	FALSE (BINARY mode)	TRUE (ASCII mode)
Size	1 - 960 (points)	1 - 480 (points)
SLMP request message write data length	2 - 1920 Bytes	4 - 1920 Bytes

Communicating Example (when writing in word units (bit device))

Write the value from M100 to M131 (2 word part).

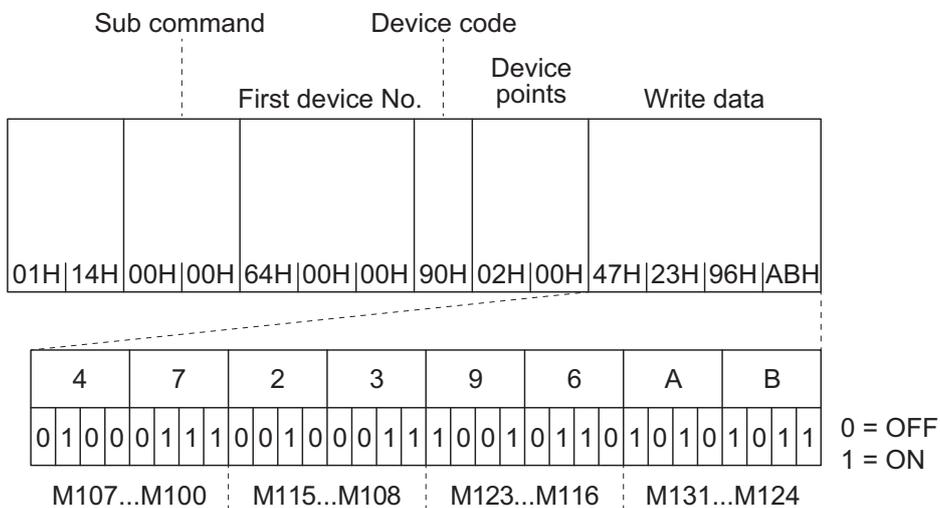
● When Data Communications are in ASCII Code

(Request data)



● When Data Communications are in Binary Code

(Request data)

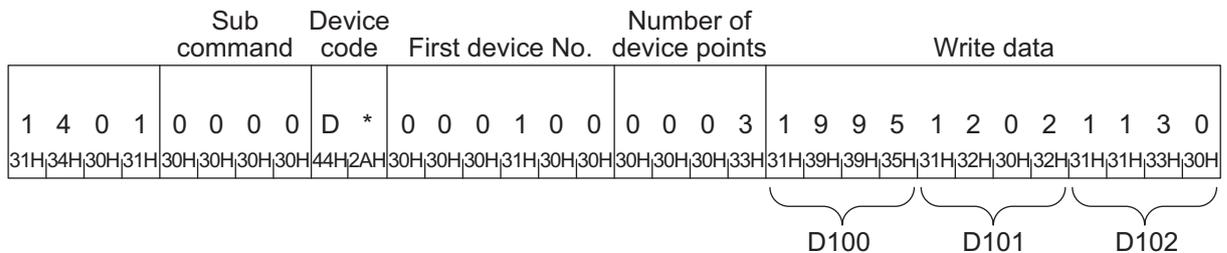


Communicating Example (when writing in word units (word device))

Write the value from D100 to D102 (2 word part).

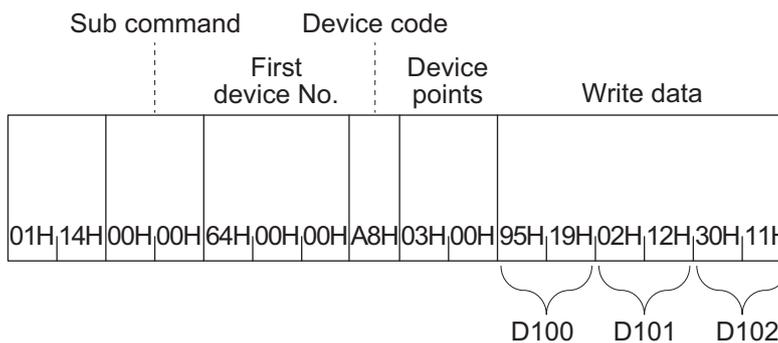
● When Data Communications are in ASCII Code

(Request data)



● When Data Communications are in Binary Code

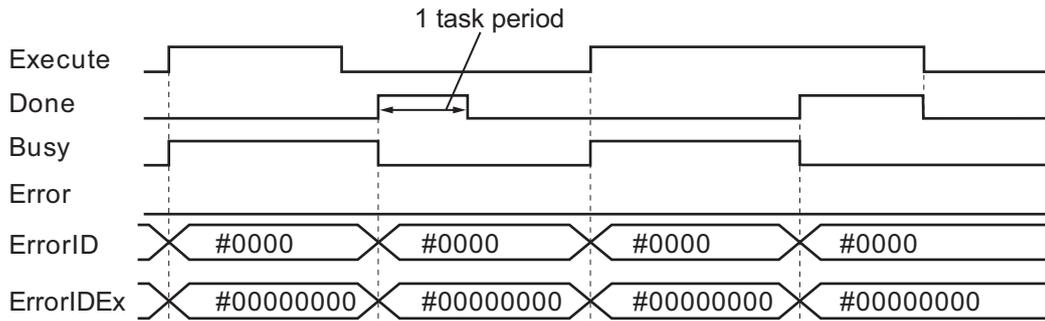
(Request data)



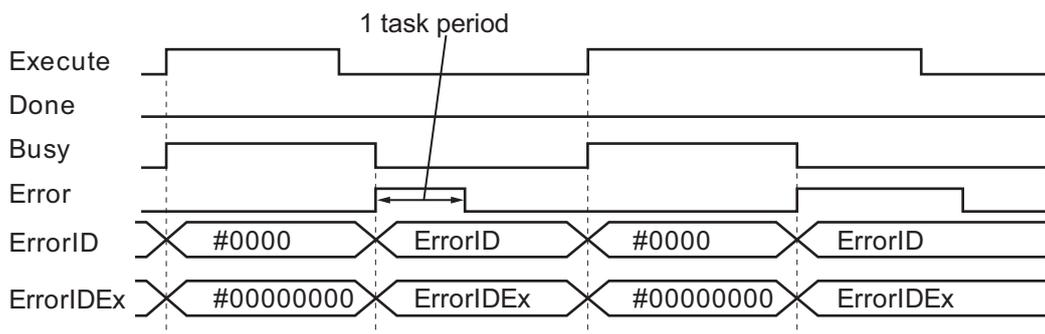
Timing Chart

The timing charts are shown below.

- *Busy* (Executing) changes to TRUE when *Execute* changes to TRUE.
- When a response message from the Mitsubishi Electric Sequencer is received normally, and End-Code=0, *Done* changes to TRUE.
- If an error occurs when execution of the function block is in progress, *Error* changes to TRUE and *Busy* (Executing) changes to FALSE.
You can find out the cause of the error by accessing the values output to *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code).
- If *Execute* changes to FALSE before execution of the FB is ended, *Done* and *Error* are TRUE only for one task period.
- If *Execute* remains TRUE even after execution of the function block is ended, the output values of *Done* and *Error* are retained.
- Timing Chart for Normal End



- Timing Chart for Error End



Additional Information

- For this FB, use the socket service function. Refer to *NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual (W506)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP™ Port User's Manual (W563)* for details of the socket service function.

Precautions for Correct Use

- Execution of this instruction is continued until processing is completed even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is completed. Use this to confirm normal completion of processing.
- This FB executes one SktTCPRcv instruction and one SktTCPSend instruction internally. For the number of simultaneous executions, refer to the *NJ/NX-series Instructions Reference Manual (W502)* or *NY-series Instructions Reference Manual (W560)* SktTCPRcv instruction and SktTCPSend instruction.
- When the same socket is shared by FBs of multiple SLMP communications instructions, multiple SLMP communications instructions cannot be executed simultaneously. SLMP communications instructions cannot distinguish responses to commands sent by themselves from responses to commands sent by other SLMP communications instructions. Therefore, wait until the execution of one SLMP communications instruction is complete and then execute the next SLMP communications instruction.



Precautions for Correct Use

- Refer to *2-2 Usage Method* on page 2-4 for details.
- If "Write during RUN" is prohibited on the Mitsubishi Electric Sequencer side, confirm that the Mitsubishi Electric Sequencer is in STOP state, and then execute.

Troubleshooting

Error code	Expansion error code	Error name	Description	Corrective action
16#0000	16#00000000	Normal end	—	—
16#0400	16#00000000	Input Value Out of Range	An input parameter for an instruction exceeded the valid range for an input variable. Or, division by an integer of 0 occurred in division or remainder calculations.	Refer to the description of <i>Machine Automation Troubleshooting Manual (W503)</i> for the event code with <i>5401</i> appended to the upper 4 digits of the Error code. For example, if the Error code is 16#0400, refer to the description of event code 54010400 hex.
16#0406	16#00000000	Illegal Data Position Specified	A memory address or data size that was specified for the instruction is not suitable.	
16#0407	16#00000000	Data Range Exceeded	The results of instruction processing exceeded the data area range of the output parameter.	
16#0419	16#00000000	Incorrect Data Type	A data type that cannot be used for an instruction is specified for an input or in-out variable.	
16#2003	16#00000000	Socket Status Error	The status was not suitable for execution of the socket service instruction.	
16#2006	16#00000000	Socket Timeout	A timeout occurred for a socket service instruction.	
16#2007	16#00000000	Socket Handle Out of Range	The handle that is specified for the socket service instruction is not correct.	
16#2008	16#00000000	Socket Communications Resource Overflow	The maximum resources that you can use for socket service instructions at the same time was exceeded.	
16#3CEB	16#00000001	Illegal device code	This function block specified an unsupported Device Code.	Check the FB input variable <i>DeviceCode</i> .
16#3CEB	16#00000002	Illegal first device No.	This FB specified an unsupported first Device No.	Check the FB input variable <i>Offset</i> .
16#3CEB	16#00000003	Illegal number of device points	Using the input variable <i>Size</i> , the write data size exceeded the controller specification upper limit value.	Check the FB input variable <i>Size</i> .
16#3CEB	16#00000004	Insufficient WriteDat size	Array size allocated to <i>WriteDat</i> is less than the write data length specified in <i>Mode</i> , <i>Size</i> .	Check the FB input variables <i>Mode</i> and <i>Size</i> . Or check the number of elements in the array to assign to the In-out variable <i>WriteDat</i> in this FB.

Error code	Expansion error code	Error name	Description	Corrective action
16#3CEB	16#00000005	WriteDat upper limit exceeded	The array size allocated to In-out variables <i>WriteDat</i> exceeds the upper limit.	Check that the array size allocated to In-out variables <i>WriteDat</i> is 1 - 960 (when in BINARY mode), 1 - 480 (when in ASCII mode).
16#3CEB	16#00000010	Response reception timeout	A response message could not be received within the time specified in input variable <i>TimeOut</i> .	Check the network status and communications target Mitsubishi Electric Sequencer status. Or, increase the value of the input variable <i>TimeOut</i> of this FB.
16#3CEB	16#FFFF0001 - 16#FFFFFFFF	SLMP end code	An error response message was received.	Refer to <i>End codes</i> in the SLMP Specifications Overview for troubleshooting. For information on how to obtain the SLMP Specifications Overview, contact the CC-Link Partner Association.

Sample Programming

From our website (http://www.ia.omron.com/sysmac_library/), you can download sample programs using SLMP_DeviceWriteWord. You can use with MELSEC iQ-R Series, iQ-F Series, Q Series, and L Series.



Precautions for Correct Use

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

The overview of the sample programming is as follows:

Conditions

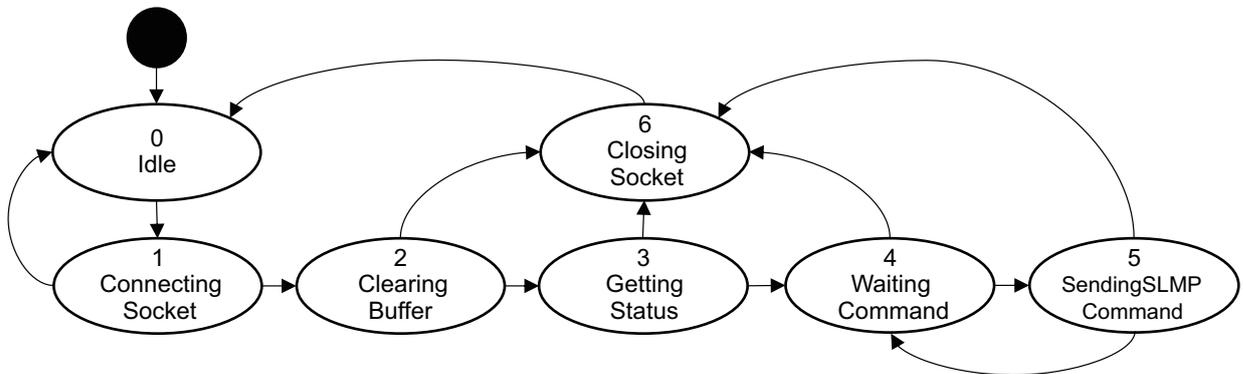
Item	Description
Process	Write the value from M100 to M131 (2 words).
Destination IP Address	192.168.250.39
Destination TCP port number	5100
Source TCP port number.	Automatically assigned from free ports
Communication code	BINARY
Timeout time	2 seconds (default)

Process

- 1** Execute the SktTCPConnect instruction, and establish the TCP connection.
- 2** Execute the SktClearBuf command, and clear the TCP socket receiving buffer.
- 3** Confirm that no other communications FB execution is in progress.
- 4** Start up the SLMP_DeviceWriteWord_instance.
- 5** Execute the SktClose instruction, and close the TCP connection.

State transition of sample program

The state transition of the sample programming is as follows:



No	State name	Processing
0	Idle	This is the state waiting for the request of establishment TCP connection. The TCP connection has not been established and is in a waiting state.
1	ConnectingSocket	This is the state establishes a TCP connection with the communication destination station.
2	ClearingBuffer	This is the state to clear the receive buffer of the TCP socket.
3	GettingStatus	This is the state checking the state of TCP connection.
4	WaitingCommand	This is the state waiting for a request to send a command for SLMP communication or a request to disconnect the TCP connection. The TCP connection has been established and is in a waiting state.
5	SendingSLMPCommand	This is the state sends SLMP communication commands and receives responses to the destination station.
6	ClosingSocket	This is the state closing the TCP connection.

SLMP_DeviceWriteBool

Through SLMP communication using TCP socket service, to write Device of Mitsubishi Electric Sequencer in bit units.

Function block name	Name	FB/FUN	Graphic expression	ST expression
SLMP_DeviceWriteBool	Internal memory Batch writing (Bit, TCP)	FB		<pre>SLMP_DeviceWriteBool_instance(Execute:=, Socket:=, Mode:=, DeviceCode:=, Offset:=, Size:=, TimeOut:=, Done=>, Busy=>, Error=>, ErrorID=>, ErrorIDEx=>, WriteDat:=);</pre>

Function Block and Function Information

Item	Description
Library file name	OmronLib_SLMP_Comm_Vx_x.slr (x shows the unit version.)
Namespace	OmronLib\SLMP_Comm
Function block and function number	00171
Source code published/not published	Not Published

Input Variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Execute	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	—	FALSE
Socket	Socket	_sSOCKET ¹	Socket	—	—	—
Mode	Mode	BOOL	SLMP frame communicating code TRUE: ASCII mode FALSE: BINARY mode	TRUE, FALSE	—	FALSE

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Device-Code	Device code	UINT	Input the Device Code for the write target.* ²	Follow the specifications of the Connected station	—	UINT#0
Offset	First device No.	UDINT	Input the first number of the write Device.* ³	Follow the specifications of the Connected station	—	UDINT#0
Size	Number of device points	UINT	Input the number of write Device.* ³	Follow the specifications of the Connected station	—	UINT#0
TimeOut	Timeout time	UINT	Input the timeout time for FB. If "0" is set, the timeout time is 2.0 s.	Depends on data type.	0.1s	UINT#0

*1. Refer to the SktTCPConnect in the *NJ/NX-series Instructions Reference Manual (W508)* or *NY-series Instructions Reference Manual (W560)* for details of the Data type.

*2. Refer to *2-3-1 Device Code (Device Access for Sequencer CPU Made by Mitsubishi Electric)* on page 2-5 for details.

*3. Refer to *Function* on page 4-23 for details.

Output Variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Done	Done	BOOL	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE, FALSE	—	—
Busy	Executing	BOOL	TRUE: Executing. FALSE: Not executing.	TRUE, FALSE	—	—
Error	Error	BOOL	TRUE: Error end. FALSE: Normal end, executing, or execution condition not met	TRUE, FALSE	—	—
ErrorID	Error Code	WORD	This is the error ID for an error end. The value is 16#0 for a normal end.	*1	—	—
ErrorIDEx	Expansion Error Code	DWORD	This is the error ID for an Expansion Unit Hardware Error. The value is 16#0 for a normal end.	*1	—	—

*1. Refer to the *Troubleshooting* on page 4-26 for details.

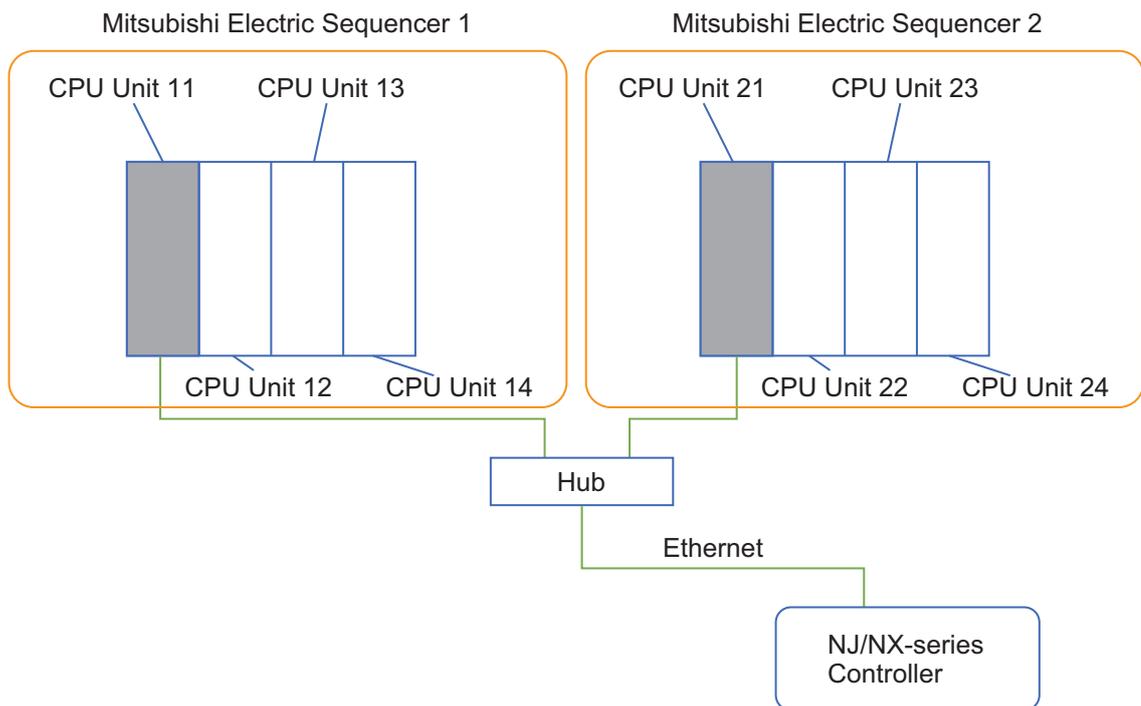
Input-Output Variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
WriteDat	Write data	ARRAY[*] OF BOOL *1	Input the write data. *2	Depends on data type.	—	—

- *1. You can input array variables of any length BOOL type. Array length is in the range of 1 - 3840 (when in BINARY mode), 1 - 1920 (when in ASCII mode). Arrays can start with either element 0 or with element n. You cannot input BOOL type variables of the base data type.
- *2. You should input in the communicating code specified in input variable *Mode*. Conversion from BINARY to ASCII and conversion from ASCII to BINARY are not executed.

Function

- This FB sends an ST (Single-Transmission) type request message for SLMP communication.
- The supported communication model by ST type is client/server (single request) type.
- Use SLMP communications to write in bit units to the sequencer device made by Mitsubishi Electric on the connected station.
- In the image below, the Mitsubishi Electric Sequencers that can write-out are CPU Unit 11 and CPU Unit 21. CPU Units 12 to 14 and CPU Units 22 to 24 do not support write-out.



- Writes the amount of the input variable *Size* from the input variable *Offset* to the Device specified by the input variable *DeviceCode*.
- The SLMP request message is based on the input variable *Mode*, and created as Write (Command: 1401, Sub-command: 0001).
- Write data is stored in the in-out variable *WriteDat*.
- Since the data type of the input variable *Offset* is UDINT, the base of the data is decimal number, decimal data. When specifying a Device with a hexadecimal number, convert the input value to a decimal number or enter a hexadecimal constant (literal), etc.

- The input range for the input variable *Size*, based on the Input variable *Mode*, and the SLMP request message write data length, are as shown below.
When *Size* is set to 0, the command will be completed without sending command (the output variable *Done* is TRUE).

Item	Conditions	
	Mode	
	FALSE (BINARY mode)	TRUE (ASCII mode)
Size	1 - 3840 (points)	1 - 1920 (points)
SLMP request message write data length	1 - 1920 Bytes	1 - 1920 Bytes

Communicating Example (when writing in bit units)

Write the value from M100 to M107.

● When Data Communications are in ASCII Code

(Request data)

Sub command	Device code	First device No.	Device points	Write data
1 4 0 1	0 0 0 1	M *	0 0 0 1 0 0	0 0 0 8
31H 34H 30H 31H 30H 30H 30H 31H 4DH 2AH 30H 30H 30H 31H 30H 30H 30H 30H 30H 30H 38H 31H 31H 30H 30H 31H 31H 30H 30H				
				M100 ... M107

0 = OFF
1 = ON

● When Data Communications are in Binary Code

(Request data)

Sub command	Device code	First device No.	Device points	Write data
01H 14H 01H 00H 64H 00H 00H 90H 08H 00H 11H 00H 11H 00H				
				M100...M107

0 = OFF
1 = ON

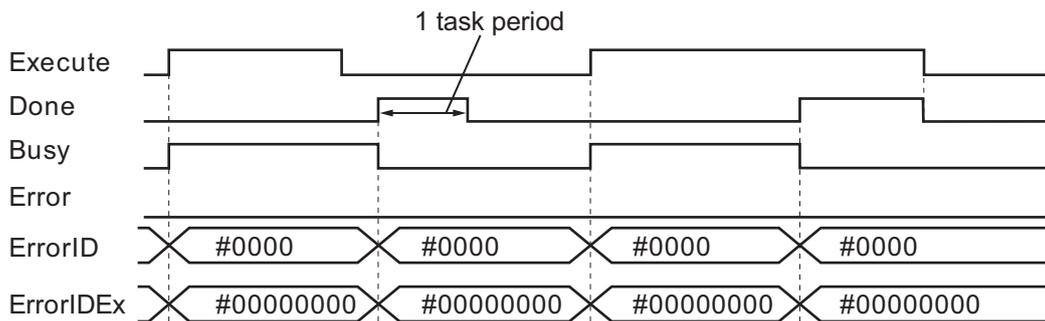
Timing Chart

The timing charts are shown below.

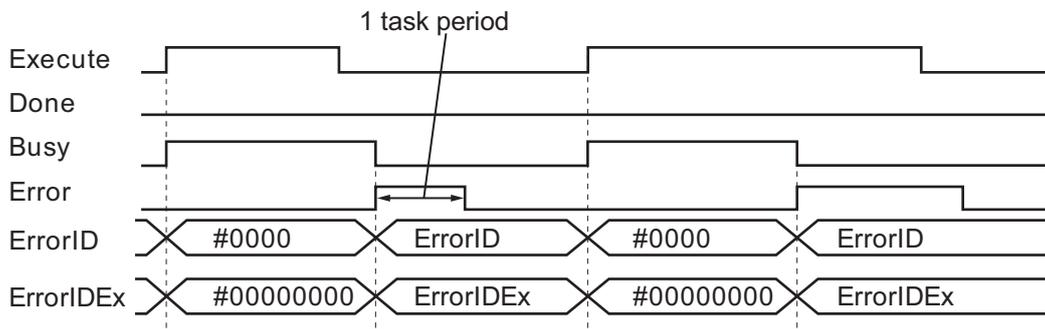
- Busy* (Executing) changes to TRUE when *Execute* changes to TRUE.
- When a response message from the Mitsubishi Electric Sequencer is received normally, and End-Code=0, *Done* changes to TRUE.

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

- Timing Chart for Normal End



- Timing Chart for Error End



Additional Information

- For this FB, use the socket service function. Refer to *NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual (W506)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP™ Port User's Manual (W563)* for details of the socket service function.

Precautions for Correct Use

- Execution of this instruction is continued until processing is completed even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is completed. Use this to confirm normal completion of processing.
- This FB executes one SktTCPRcv instruction and one SktTCPSend instruction internally. For the number of simultaneous executions, refer to the *NJ/NX-series Instructions Reference Manual (W502)* or *NY-series Instructions Reference Manual (W560)* SktTCPRcv instruction and SktTCPSend instruction.

- When the same socket is shared by FBs of multiple SLMP communications instructions, multiple SLMP communications instructions cannot be executed simultaneously. SLMP communications instructions cannot distinguish responses to commands sent by themselves from responses to commands sent by other SLMP communications instructions. Therefore, wait until the execution of one SLMP communications instruction is complete and then execute the next SLMP communications instruction.



Precautions for Correct Use

- Refer to 2-2 *Usage Method* on page 2-4 for details.
- If "Write during RUN" is prohibited on the Mitsubishi Electric Sequencer side, confirm that the Mitsubishi Electric Sequencer is in STOP state, and then execute.

Troubleshooting

Error code	Expansion error code	Error name	Description	Corrective action
16#0000	16#00000000	Normal end	—	—
16#0400	16#00000000	Input Value Out of Range	An input parameter for an instruction exceeded the valid range for an input variable. Or, division by an integer of 0 occurred in division or remainder calculations.	Refer to the description of <i>Machine Automation Troubleshooting Manual (W503)</i> for the event code with 5401 appended to the upper 4 digits of the Error code. For example, if the Error code is 16#0400, refer to the description of event code 54010400 hex.
16#0406	16#00000000	Illegal Data Position Specified	A memory address or data size that was specified for the instruction is not suitable.	
16#0407	16#00000000	Data Range Exceeded	The results of instruction processing exceeded the data area range of the output parameter.	
16#0419	16#00000000	Incorrect Data Type	A data type that cannot be used for an instruction is specified for an input or in-out variable.	
16#2003	16#00000000	Socket Status Error	The status was not suitable for execution of the socket service instruction.	
16#2006	16#00000000	Socket Timeout	A timeout occurred for a socket service instruction.	
16#2007	16#00000000	Socket Handle Out of Range	The handle that is specified for the socket service instruction is not correct.	
16#2008	16#00000000	Socket Communications Resource Overflow	The maximum resources that you can use for socket service instructions at the same time was exceeded.	

Error code	Expansion error code	Error name	Description	Corrective action
16#3CEC	16#00000001	Illegal device code	This function block specified an unsupported Device Code.	Check the FB input variable <i>DeviceCode</i> .
16#3CEC	16#00000002	Illegal first device No.	This FB specified an unsupported first Device No.	Check the FB input variable <i>Offset</i> .
16#3CEC	16#00000003	Illegal number of device points	Using the input variable <i>Size</i> , the write data size exceeded the controller specification upper limit value.	Check the FB input variable <i>Size</i> .
16#3CEC	16#00000004	Insufficient WriteDat size	Array size allocated to <i>WriteDat</i> is less than the write data length specified in <i>Mode</i> , <i>Size</i> .	Check the FB input variables <i>Mode</i> and <i>Size</i> . Or check the number of elements in the array to assign to the In-out variable <i>WriteDat</i> in this FB.
16#3CEC	16#00000005	WriteDat upper limit exceeded	The array size allocated to In-out variables <i>WriteDat</i> exceeds the upper limit.	Confirm the array size allocated to in-out variable <i>WriteDat</i> is 1 - 3840 (when in BINARY mode), 1 - 1920 (when in ASCII mode).
16#3CEC	16#00000010	Response reception timeout	A response message could not be received within the time specified in input variable <i>TimeOut</i> .	Check the network status and communications target Mitsubishi Electric Sequencer status. Or, increase the value of the input variable <i>TimeOut</i> of this FB.
16#3CEC	16#FFFF0001 - 16#FFFFFFFF	SLMP end code	An error response message was received.	Refer to <i>End codes</i> in the SLMP Specifications Overview for troubleshooting. For information on how to obtain the SLMP Specifications Overview, contact the CC-Link Partner Association.

Sample Programming

From our website (http://www.ia.omron.com/sysmac_library/), you can download sample programs using SLMP_DeviceWriteBool. You can use with MELSEC iQ-R Series, iQ-F Series, Q Series, and L Series.



Precautions for Correct Use

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

The overview of the sample programming is as follows:

Conditions

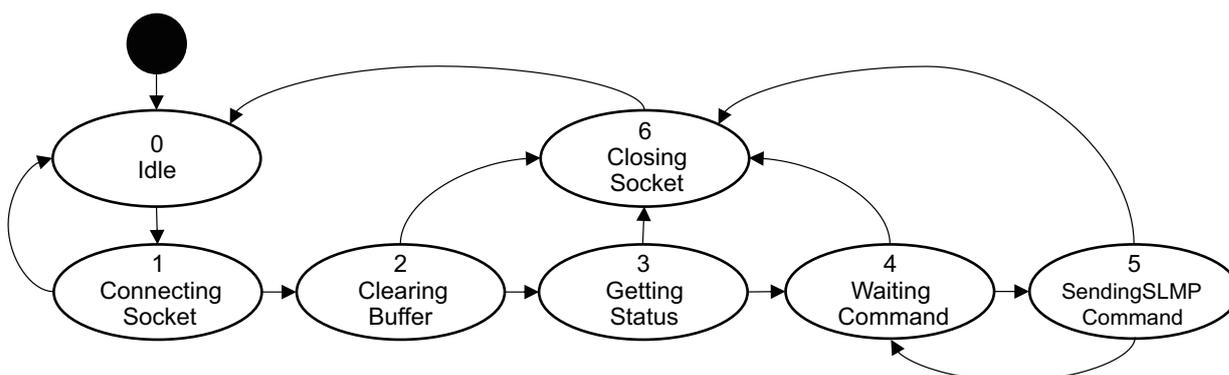
Item	Description
Process	Write values from M100 to M107
Destination IP Address	192.168.250.39
Destination TCP port number	5100
Source TCP port number.	Automatically assigned from free ports
Communication code	BINARY
Timeout time	2 seconds (default)

Process

- 1** Execute the SktTCPConnect instruction, and establish the TCP connection.
- 2** Execute the SktClearBuf command, and clear the TCP socket receiving buffer.
- 3** Confirm that no other communications FB execution is in progress.
- 4** Start up the SLMP_DeviceWriteBool_instance.
- 5** Execute the SktClose instruction, and close the TCP connection.

State transition of sample program

The state transition of the sample programming is as follows:



No	State name	Processing
0	Idle	This is the state waiting for the request of establishment TCP connection. The TCP connection has not been established and is in a waiting state.
1	ConnectingSocket	This is the state establishes a TCP connection with the communication destination station.
2	ClearingBuffer	This is the state to clear the receive buffer of the TCP socket.
3	GettingStatus	This is the state checking the state of TCP connection.
4	WaitingCommand	This is the state waiting for a request to send a command for SLMP communication or a request to disconnect the TCP connection. The TCP connection has been established and is in a waiting state.
5	SendingSLMPCommand	This is the state sends SLMP communication commands and receives responses to the destination station.
6	ClosingSocket	This is the state closing the TCP connection.

SLMP_SendCmd

Through SLMP communication using TCP socket service, any request message is sent to the Mitsubishi Electric sequencer on the Connected station and a response message is received.

Function block name	Name	FB/FUN	Graphic expression	ST expression
SLMP_SendCmd	Send SLMP Command (TCP)	FB		<pre>SLMP_SendCmd_instance(Execute:=, Socket:=, Mode:=, NoResp:=, Timeout:=, Done=>, Busy=>, Error=>, ErrorID=>, ErrorIDEx=>, ReqHeader:=, ReqDat:=, RespHeader:=, RespDat:=, RespDatSize=>);</pre>

Function Block and Function Information

Item	Description
Library file name	OmronLib_SLMP_Comm_Vx_x.slr (x shows the unit version.)
Namespace	OmronLib\SLMP_Comm
Function block and function number	00172
Source code published/not published	Not Published



Version Information

You can use this FB with version 2.0 or later of this library.

Input variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Execute	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	—	FALSE
Socket	Socket	_sSOCKET*1	Socket	—	—	—

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Mode	Mode	BOOL	SLMP frame communicating code TRUE: ASCII mode FALSE: BINARY mode	TRUE, FALSE	—	FALSE
NoResp	response is not required	BOOL	Whether or not to response to the request message TRUE: It means the request message is not require the response. FALSE: It means the request message is require the response.	TRUE, FALSE	—	FALSE
TimeOut	Timeout time	UINT	Input the timeout time for FB. If "0" is set, the timeout time is 2.0 s.	Depends on data type.	0.1s	UINT#0

*1. Refer to the SktTCPConnect in the *NJ/NX-series Instructions Reference Manual (W508)* or *NY-series Instructions Reference Manual (W560)* for details of the Data type.

Output variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Done	Done	BOOL	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE, FALSE	—	—
Busy	Executing	BOOL	TRUE: Executing. FALSE: Not executing.	TRUE, FALSE	—	—
Error	Error	BOOL	TRUE: Monitoring in progress. FALSE: Normal end, executing, or execution condition not met	TRUE, FALSE	—	—
ErrorID	Error Code	WORD	This is the error ID for an error end. The value is 16#0 for a normal end.	*1	—	—
ErrorIDex	Expansion Error Code	DWORD	This is the error ID for an Expansion Unit Hardware Error. The value is 16#0 for a normal end.	*1	—	—
RespDat-Size	Response data size	UINT	The data size of the response data "resData" in the received response message is outputted.	Depends on data type.	Bytes	—

*1. Refer to the *Troubleshooting* on page 4-38 for details.

In-out variables

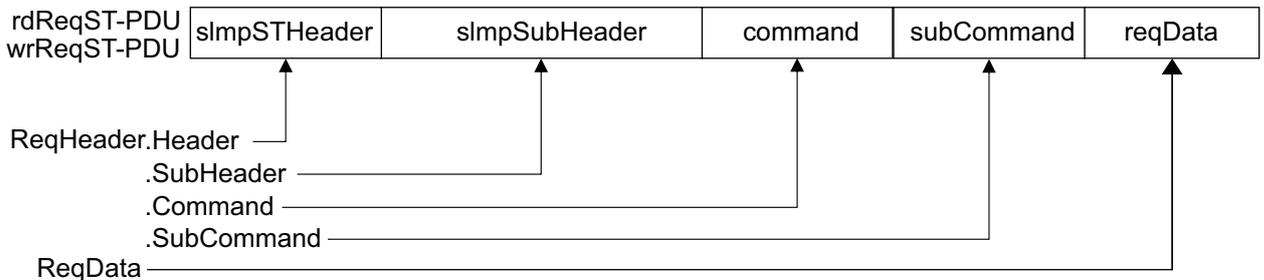
Variable	Name	Data type	Description	Valid range	Unit	Default
ReqHeader	Request header	sRdWrReq STHeader	Stores the header information of the request message.	—	—	—

Variable	Name	Data type	Description	Valid range	Unit	Default
ReqDat	Request data	ARRAY[*] OF BYTE* ¹	Input the request data "reqData" for the request message. * ³	—	—	—
RespHeader	Response header	sRdWrResSTHeader	The header information of the response message is stored.	—	—	—
RespDat	Response data	ARRAY[*] OF BYTE* ²	The size of the response data "resData" in the response message is outputted. * ⁴ * ⁵	—	—	—

- *1. You can input array variables of any length BYTE type. Arrays can start with either element 0 or with element n. Make sure that the array is long enough to store the request data. Refer to the *Function* on page 4-36 for details. You cannot input BYTE type variables of the base data type.
- *2. You can input array variables of any length BYTE type. Arrays can start with either element 0 or with element n. Make sure that the array is long enough to store the response data for request message. Refer to the *Function* on page 4-36 for details. You cannot input BYTE type variables of the base data type.
- *3. You should input in the communicating code specified in input variable *Mode*. Conversion from BINARY to ASCII and conversion from ASCII to BINARY are not executed.
- *4. It is outputted in the communicating code specified in *Mode*. Conversion from BINARY to ASCII and conversion from ASCII to BINARY are not executed.
- *5. If the value of the output variable *RespDatSize* exceeds the number of elements in the array, the excess data is discarded.

Structures

The data type of the in-out variable *ReqHeader* in this FB is the structure `OmronLib\SLMPLib\lsRdWrReqSTHeader`. Each member is reflected in the SLMP frame as shown in the following figure.



The specifications are as follows:

Variable	Name	Data type	Description	Valid range	Unit	Default
ReqHeader	Request header	OmronLib \SLMPLib \lsRdWrReqSTHeader	Stores the header information of the request message to be sent.	—	—	—
Header	Header	WORD	Input the type of communication frame. Input 16#5000.	16#5000	—	16#0000
Sub-Header	Sub header	sSubHeader1	See structure type sSubHeader1.	—	—	—

Variable	Name	Data type	Description	Valid range	Unit	Default
Command	Command	WORD	Input the command to be sent. For commands that can be sent to Mitsubishi Electric sequencers, see the destination sequencer's manual.	16#0000 to 16#FFFF	—	16#0000
Sub-Command	Sub command	WORD	Input the Sub-command to be sent. For commands that can be sent to Mitsubishi Electric sequencers, see the destination sequencer's manual.	16#0000 to 16#FFFF	—	16#0000

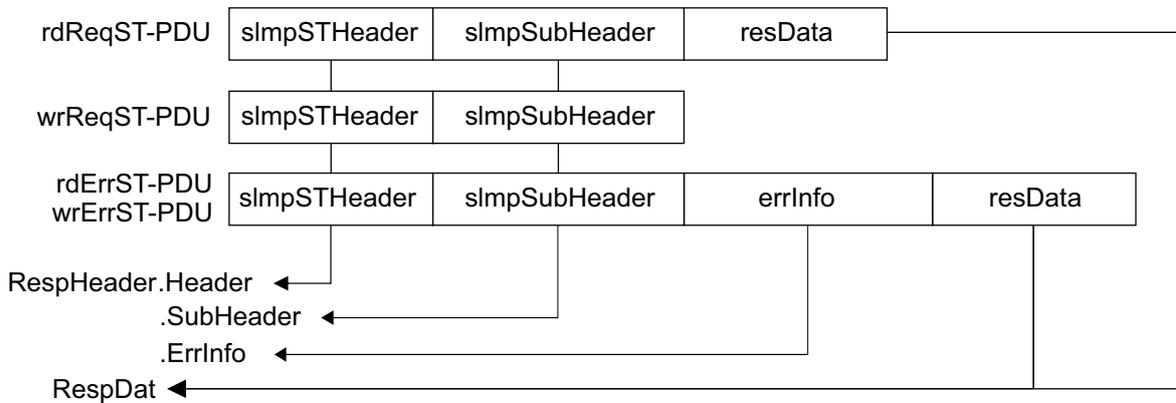
The data type of the in-out variable *ReqHeader.SubHeader* in this FB is the structure `OmronLib \SLMPLib\SubHeader1`. The specifications are as follows:

Variable	Name	Data type	Description	Valid range	Unit	Default
ReqHeader.Sub-Header	Sub header	OmronLib \SLMPLib \SubHeader1	Stores the sub header information of the request message to be sent.	—	—	—
	NetNo	BYTE	Input the network number of the Connected station.	Follow the specifications of the Connected station	—	16#00
	NodeNo	BYTE	Input the station number of the Connected station.	Follow the specifications of the Connected station	—	16#00
	DstProc-No	WORD	Input the processor number of the Connected station.	Follow the specifications of the Connected station	—	16#0000
	Reserved1	BYTE	—	16#00	—	16#00
	DI	UINT	Input the Byte size of the SLMP request message from the "timer" to the end of the "reqData" (request data). Input +6 value for BINARY Mode*1, +12 value for ASCII Mode*2 in the data size of "reqData".	Depends on data type.	—	0
	Timer	UINT	Input the time to wait for a response from the Connected station. The SLMP specification recommends settings in the range of 1 to 40.	0:Infinite wait 1-65535: Wait time	250 ms	0

*1. "timer" (2 Bytes) + "command" (2 Bytes) + " subCommand" (2 Bytes), total 6 Bytes

*2. "timer" (4 Bytes) + "command" (4 Bytes) + " subCommand" (4 Bytes), total 12 Bytes

The data type of the in-out variable *RespHeader* in this FB is the structure *OmronLib\SLMPLib\lsRdWrResSTHeader*. The value of the SLMP frame is reflected for each member, as shown in the following figure.



The specifications are as follows:

Variable	Name	Data type	Description	Valid range	Unit	Default
RespHeader	Response header	OmronLib\SLMPLib\lsRdWrResSTHeader	The header information of the received response message is stored. In SLMP, "errInfo" is not appended to the normal response message, but it is always present in this structure.	—	—	—
	Header	WORD	The type of communication frame is output. 16#D000 fixed.	16#D000	—	16#0000
	Sub-Header	sSubHeader2	See structure type sSubHeader2.	—	—	—
	ErrInfo	sErrInfo	Error information is outputted when the in-out variable <i>RespHeader.SubHeader.ErrCode</i> is other than 16#0000. See structure type sErrInfo.	—	—	—

The data type of the in-out variable *RespHeader.SubHeader* in this FB is the structure *OmronLib\SLMPLib\lsSubHeader2*. The specifications are as follows:

Variable	Name	Data type	Description	Valid range	Unit	Default
RespHeader.SubHeader	Sub header	OmronLib\SLMPLib\lsSubHeader2	The header information of the received response message is stored.	—	—	—
	NetNo	BYTE	The network number of the Connected station is outputted.	Follow the specifications of the Connected station	—	16#00
	NodeNo	BYTE	The station number of the Connected station is outputted.	Follow the specifications of the Connected station	—	16#00

Variable	Name	Data type	Description	Valid range	Unit	Default
DstProc-No	Processor number of request destination station	WORD	The processor number of the Connected station is outputted.	Follow the specifications of the Connected station	—	16#0000
Re-served1	System reserved	BYTE	—	16#00	—	16#00
DI	Request data length	UINT	The data size of the SLMP response message from the "endCode" (end code) to the "FCS", the end of the PDU, is outputted in Byte units.*1	Depends on data type.	—	0
End-Code	End code	WORD	An error code is outputted.	16#0000: Processing successful 16#0001 to 16#FFFFFF : Error	—	16#0000

*1. The meanings are shown in the following table.

Response type	Communication code	Value (Byte)
Normal response	BINARY	Data size of "resData" + 2
	ASCII	Data size of "resData" + 4
Error response	BINARY	Data size of "resData" + 11
	ASCII	Data size of "resData" + 22



Precautions for Correct Use

The value of "dl" in the SLMP response message is outputted to the in-out variable *RespHeader.SubHeader.DI* in this FB. When responding normally, the size of the in-out variable *RespHeader.ErrInfo* is not included.

You should judge the result of the execution of the request message by the in-out variable *RespHeader.SubHeader.EndCode* of this FB.

The data type of the in-out variable *RespHeader.ErrInfo* in this FB is the structure `OmronLib\SLMPLIB\ErrInfo`. The specifications are as follows:

Variable	Name	Data type	Description	Valid range	Unit	Default
RespHeader.ErrInfo	Error information of response message	OmronLib\SLMPLib\ErrInfo	The error information of the received response message is stored. When the response message is normal, 0 is stored in all members.	—	—	—
NetNo	Network number of request source station	BYTE	The network number of the requesting station is outputted.	Follow the specifications of the Connected station	—	16#00
NodeNo	Station number of request source station	BYTE	The station number of the requesting station is outputted.	Follow the specifications of the Connected station	—	16#00

Variable	Name	Data type	Description	Valid range	Unit	Default
DstProc-No	Processor number of request source station	WORD	The processor number of the requesting station is outputted.	Follow the specifications of the Connected station	—	16#0000
Re-served1	System reserved	BYTE	-	16#00	—	16#00
Command	Command	WORD	The requested command is outputted.	16#0000 to 16#FFFF	—	16#0000
Sub-Command	Sub command	WORD	The requested subcommand is outputted.	16#0000 to 16#FFFF	—	16#0000

Function

- This FB sends an ST (Single-Transmission) type request message for SLMP communication.
- The supported communication models by ST type are client/server (single request) type and client/server (response not required) type.
- This FB creates and sends a request message from the In-out variable *ReqHeader* and *ReqDat*. After sending, wait for the response from the server side, and store the received response message in the in-out variable *RespHeader* and *RespDat*.
- When ASCII mode is selected by the input variable *Mode*, this FB creates a message that converts the contents of the in-out variable *ReqHeader* to ASCII code. However, the contents of the In-out variable *ReqDat* are not converted. Therefore, enter the data created by ASCII code in *ReqDat*. This FB does not check the contents of *ReqDat*.
- The maximum Byte size of the request data "reqData" that can be sent by this FB and the response data "resData" that can be received are shown in the table below.

Communicating code of SLMP frame	Request data "reqData"	The size of the response data "resData" in case of normal response message		The size of the response data "resData" in case of error response message
		rdResST-PDU	wrResST-PDU	
BINARY	1985 (2000 - 15 ^{*1})	1989 (2000 - 11 ^{*2})	0	1980 (2000 - 20 ^{*3})
ASCII	1970 (2000 - 30 ^{*1})	1978 (2000 - 22 ^{*2})	0	1960 (2000 - 40 ^{*3})

*1. Sum of rdReqST-PDU or wrReqST-PDU other than request data "reqData"

*2. Sum of rdResST-PDU other than response data "resData"

*3. Sum of rdErrST-PDU or wrErrST-PDU other than response data "resData"

- This FB is as follows when the data size of the received response message (response header + response data) exceeds 2000 Bytes. Use this FB so that the data size of the response message does not exceed 2000 Bytes.
 - Reception processing is suspended when the received data exceeds 2000 Bytes.
 - Receive data exceeding 2000 Bytes is discarded.
 - A part of the response message may remain in the receive buffer of socket.
- When the response data size in the response message exceeds the data size of the in-out variable *RespDat*, only the size of *RespDat* is stored. Therefore, the number of elements of the BYTE

- array variable used for *RespDat* should be equal to or greater than the maximum size of the response data "resData".
- When input variable *NoResp* means "no response ", this FB ends upon completion of sending a request message.



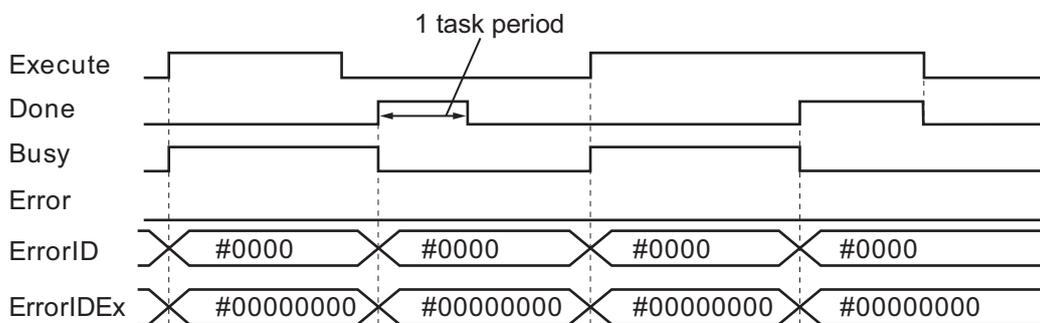
Precautions for Correct Use

- Set the input variable *TimeOut* to a value greater than the in-out variable *ReqHeader.SubHeader.Timer*. In addition, set each value according to the communication path and processing details.
Even if this FB terminates error due to timeout according to *TimeOut* when the *ReqHeader.SubHeader.Timer* is set to a value greater than or equal to the *TimeOut* or "0" (infinite wait), the Mitsubishi Electric sequencer side continues the processing. Because of this FB cannot receive response data.
- This FB is designed to be able to send and receive request and response messages of up to 2000 Bytes. However, depending on the specifications of the target SLMP device, up to 2000 Bytes may not be able to be sent and received. Use this FB after checking the specifications of the target SLMP device.

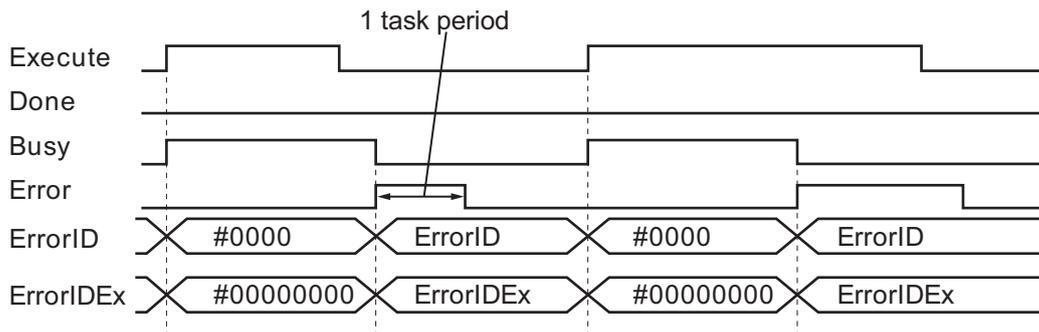
Timing Charts

The timing chart is shown below.

- Busy* (Executing) changes to TRUE when *Execute* changes to TRUE.
- When a response message from the Mitsubishi Electric Sequencer is received normally, and End-Code=0, *Done* changes to TRUE. If *NoResp* (no response required) is TRUE, then *Done* will be TRUE upon successful transmission of the request message.
- If an error occurs when execution of the function block is in progress, *Error* changes to TRUE and *Busy* (Executing) changes to FALSE.
You can find out the cause of the error by accessing the values output to *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code).
- If *Execute* changes to FALSE before execution of the FB is ended, *Done* and *Error* are TRUE only for one task period.
- If *Execute* remains TRUE even after execution of the function block is ended, the output values of *Done* and *Error* are retained.
- Timing Chart for Normal End



- Timing Chart for Error End



Additional Information

- For this FB, use the socket service function. Refer to *NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual (W506)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP™ Port User's Manual (W563)* for details of the socket service function.

Precautions for Correct Use

- Execution of this instruction is continued until processing is completed even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is completed. Use this to confirm normal completion of processing.
- This FB executes one SktTCPRcv instruction and one SktTCPSend instruction internally. For the number of simultaneous executions, refer to the *NJ/NX-series Instructions Reference Manual (W502)* or *NY-series Instructions Reference Manual (W560)* SktTCPRcv instruction and SktTCPSend instruction.
- When the same socket is shared by FBs of multiple SLMP communications instructions, multiple SLMP communications instructions cannot be executed simultaneously. SLMP communications instructions cannot distinguish responses to commands sent by themselves from responses to commands sent by other SLMP communications instructions. Therefore, wait until the execution of one SLMP communications instruction is complete and then execute the next SLMP communications instruction.

Troubleshooting

Error code	Expansion error code	Error name	Description	Corrective action
16#0000	16#00000000	Normal end	—	—

Error code	Expansion error code	Error name	Description	Corrective action	
16#0400	16#00000000	Input Value Out of Range	An input parameter for an instruction exceeded the valid range for an input variable. Or, division by an integer of 0 occurred in division or remainder calculations.	Refer to the description of <i>Machine Automation Controller Troubleshooting Manual (W503)</i> for the event code with 5401 appended to the upper 4 digits of the Error code. For example, if the Error code is 16#0400, refer to the description of event code 54010400 hex.	
16#0406	16#00000000	Illegal Data Position Specified	A memory address or data size that was specified for the instruction is not suitable.		
16#0407	16#00000000	Data Range Exceeded	The results of instruction processing exceeded the data area range of the output parameter.		
16#0419	16#00000000	Incorrect Data Type	A data type that cannot be used for an instruction is specified for an input or in-out variable.		
16#2003	16#00000000	Socket Status Error	The status was not suitable for execution of the socket service instruction.		
16#2006	16#00000000	Socket Timeout	A timeout occurred for a socket service instruction.		
16#2007	16#00000000	Socket Handle Out of Range	The handle that is specified for the socket service instruction is not correct.		
16#2008	16#00000000	Socket Communications Resource Overflow	The maximum resources that you can use for socket service instructions at the same time was exceeded.		
16#3CED	16#00000004	Too short ReqDat	One of the following. <ul style="list-style-type: none"> The number of elements in the array assigned to the in-out variable <i>ReqDat</i> is less than the requested data length specified by the in-out variable <i>ReqHeader.SubHeader.DI</i>. The in-out variable <i>RespHeader.SubHeader.DI</i> is set to a value out of specification of this FB. 		Confirm the in-out variable <i>ReqHeader.SubHeader.DI</i> in this FB. Or confirm the number of elements in the array to assign to the In-out variable <i>ReqDat</i> in this FB.

Error code	Expansion error code	Error name	Description	Corrective action
16#3CED	16#00000005	Request message length limit over	The length of the request statement, construct constructed according to the value of the in-out variable <i>ReqHeader.SubHeader.DI</i> exceeds the maximum limit that can be produce with this FB, function block. *1	Confirm the in-out variable <i>ReqHeader.SubHeader.DI</i> in this FB.
16#3CED	16#00000010	Response reception timeout	A response message could not be received within the time specified in input variable <i>TimeOut</i> .	Check the network status and communications target Mitsubishi Electric Sequencer status. Or, increase the value of the input variable <i>TimeOut</i> of this FB.
16#3CED	16#FFFF0001 - 16#FFFFFFFF	SLMP end code	An error response message was received.	Refer to <i>End codes</i> in the SLMP Specifications Overview for troubleshooting. For information on how to obtain the SLMP Specifications Overview, contact the CC-Link Partner Association.

*1. Refer to the *Function* on page 4-36 for details.

Sample program

From our website (http://www.ia.omron.com/sysmac_library/), you can download sample programs using SLMP_SendCmd. You can use with MELSEC iQ-R Series, iQ-F Series, Q Series, and L Series.



Precautions for Correct Use

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

The overview of the sample programming is as follows:

Conditions

Item	Description
Process	Read non-consecutive Devices D0, M100, Y160 with "Random reading" command
Destination IP Address	192.168.250.39

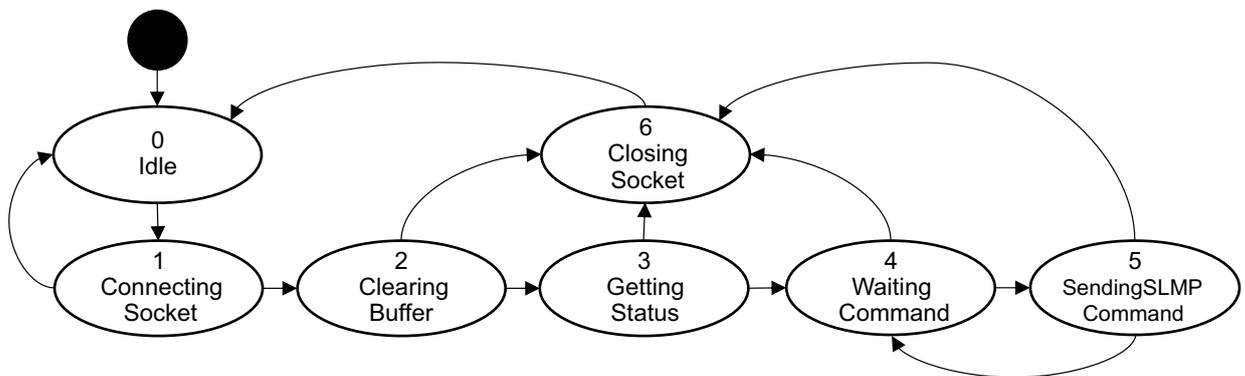
Item	Description
Destination TCP port number	5100
Source TCP port number.	Automatically assigned from free ports
Communication code	BINARY
Timeout time	2 seconds (default)

Process

- 1** Execute the SktTCPConnect instruction, and establish the TCP connection.
- 2** Execute the SktClearBuf command, and clear the TCP socket receiving buffer.
- 3** Confirm that no other communications FB execution is in progress.
- 4** Launch the SLMP_SendCmd FB-instance.
- 5** After completing FB execution, check the response data.
- 6** Execute the SktClose instruction, and close the TCP connection.

State transition of sample program

The state transition of the sample programming is as follows:



No	State name	Processing
0	Idle	This is the state waiting for the request of establishment TCP connection. The TCP connection has not been established and is in a waiting state.
1	ConnectingSocket	This is the state establishes a TCP connection with the communication destination station.
2	ClearingBuffer	This is the state to clear the receive buffer of the TCP socket.
3	GettingStatus	This is the state checking the state of TCP connection.
4	WatingCommand	This is the state waiting for a request to send a command for SLMP communication or a request to disconnect the TCP connection. The TCP connection has been established and is in a waiting state.

No	State name	Processing
5	SendingSLMPCCommand	This is the state sends SLMP communication commands and receives responses to the destination station.
6	ClosingSocket	This is the state closing the TCP connection.

SLMP_DeviceRead_UDP

Through SLMP communication using UDP socket service, to read Device of Mitsubishi Electric Sequencer.

Function block name	Name	FB/FUN	Graphic expression	ST expression
SLMP_DeviceRead_UDP	Internal memory Batch read (UDP)	FB		<pre>SLMP_DeviceRead_UDP_instance(Execute:=, Socket:=, Mode:=, BitAccess:=, DeviceCode:=, Offset:=, Size:=, TimeOut:=, Done=>, Busy=>, Error=>, ErrorID=>, ErrorIDEx=>, ReadSize=>, ReadDat:=);</pre>

Function Block and Function Information

Item	Description
Library file name	OmronLib_SLMP_Comm_Vx_x.slr (x shows the unit version.)
Namespace	OmronLib\SLMP_Comm
Function block and function number	00246
Source code published/not published	Not Published



Version Information

You can use this FB with version 2.0 or later of this library.

Input variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Execute	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	—	FALSE

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Socket	Socket	_sSOCKET* ¹	Socket	—	—	—
Mode	Mode	BOOL	SLMP frame communicating code TRUE: ASCII mode FALSE: BINARY mode	TRUE, FALSE	—	FALSE
BitAccess	Bit access specification	BOOL	The units of read Device TRUE: Bit unit FALSE: Word unit	TRUE, FALSE	—	FALSE
Device-Code	Device code	UINT	Input the Device Code for the read target.* ²	Follow the specifications of the Connected station	—	UINT#0
Offset	First device No.	UDINT	Input the first number of the read Device.* ³	Follow the specifications of the Connected station	—	UDINT#0
Size	Number of device points	UINT	Input the number of read Device.* ³	Follow the specifications of the Connected station	—	UINT#0
TimeOut	Timeout time	UINT	Input the timeout time for FB. If "0" is set, the timeout time is 2.0 s.	Depends on data type.	0.1s	UINT#0

*1. Refer to the SktUDPCreate in the *NJ/NX-series Instructions Reference Manual (W508)* or *NY-series Instructions Reference Manual (W560)* for details of the Data type.

*2. Refer to *2-3-1 Device Code (Device Access for Sequencer CPU Made by Mitsubishi Electric)* on page 2-5 for details.

*3. Refer to *Function* on page 4-45 for details.

Output variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Done	Done	BOOL	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE, FALSE	—	—
Busy	Executing	BOOL	TRUE: Executing. FALSE: Not executing.	TRUE, FALSE	—	—
Error	Error	BOOL	TRUE: Monitoring in progress. FALSE: Normal end, executing, or execution condition not met	TRUE, FALSE	—	—
ErrorID	Error Code	WORD	This is the error ID for an error end. The value is 16#0 for a normal end.	*1	—	—

Variable	Meaning	Data type	Description	Valid range	Unit	Default
ErrorIDEx	Expansion Error Code	DWORD	This is the error ID for an Expansion Unit Hardware Error. The value is 16#0 for a normal end.	*1	—	—
ReadSize	Read data size	UINT	The actual read-out data size is outputted.	Depends on data type.	Bytes	—

*1. Refer to the *Troubleshooting* on page 4-49 for details.

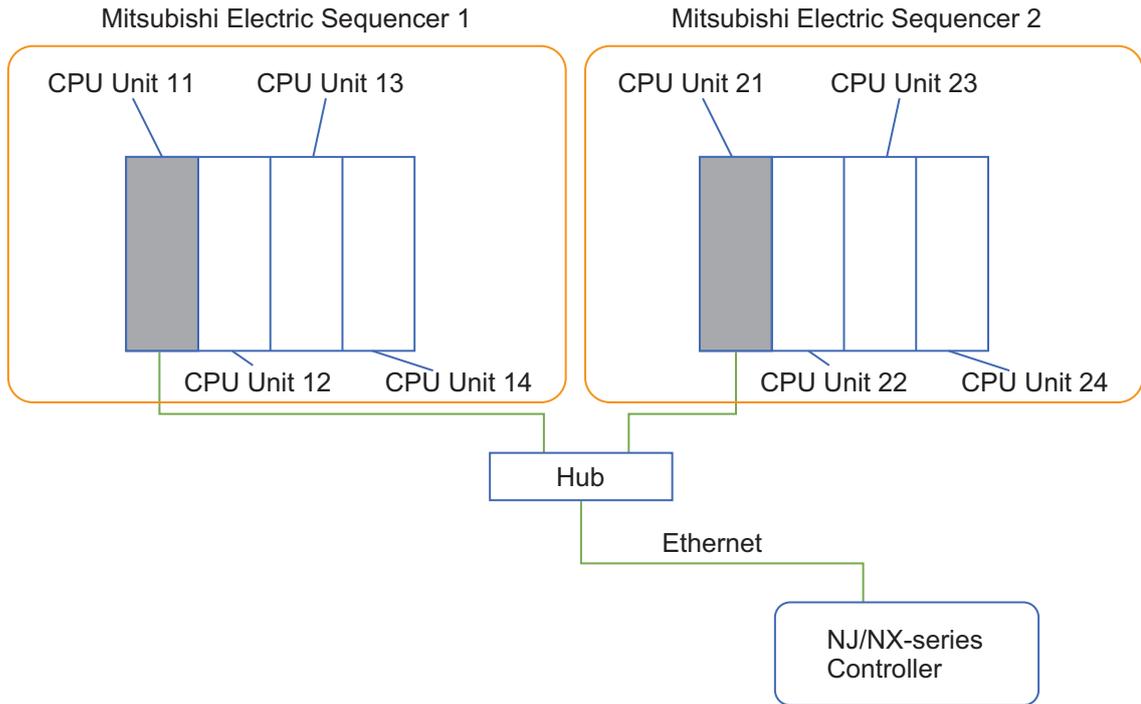
Input-Output Variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
ReadDat	Read data	ARRAY[*] OF BYTE *1	The actual read-out data is outputted. *2*3	Depends on data type.	—	—

- *1. You can input array variables of any length BYTE type. Array length is in the range from 1 to 1920. Arrays can start with either element 0 or with element n. You cannot input BYTE type variables of the base data type.
- *2. It is outputted in the communicating code specified in *Mode*. Conversion from BINARY to ASCII and conversion from ASCII to BINARY are not executed.
- *3. If the value of the output variable *ReadSize* exceeds the number of elements in the array, the excess data is discarded.

Function

- This FB sends an ST (Single-Transmission) type request message for SLMP communication.
- The supported communication model by ST type is client/server (single request) type.
- Use SLMP communications to read the sequencer device made by Mitsubishi Electric on the connected station.
- In the image below, the Mitsubishi Electric Sequencers that can read-out are CPU Unit 11 and CPU Unit 21. CPU Units 12 to 14 and CPU Units 22 to 24 do not support read-out.



- Reads out the amount of the input variable *Size* from the input variable *Offset* from the Device specified by the input variable *DeviceCode*.
 - The SLMP request message is based on the input variable *Mode* and *BitAccess*, and created as Read (Command: 0401, Sub-command: 0001/0000).
 - The read result is outputted to the in-out variable *ReadDat* with the SLMP response data "resData" array as is. The output size is outputted to the output variable *ReadSize* in byte units .
 - Since the data type of the input variable *Offset* is UDINT, the base of the data is decimal number, decimal data. When specifying a Device with a hexadecimal number, convert the input value to a decimal number or enter a hexadecimal constant (literal), etc.
 - The possible input range for the input variable *Size*, based on the Input variable *Mode* and input variable *BitAccess*, and the output variable *ReadSize* output range are as shown below.
- When *Size* is set to 0, the command will be completed without sending command (the output variable *Done* is TRUE). In that case, 0 is outputted to *ReadSize*, and the in-out variable *ReadDat* is not updated.

Variable name	Conditions		
	BitAccess	Mode	
		FALSE (BINARY mode)	TRUE (ASCII mode)
Size	TRUE	1 - 3840 (points)	1 - 1920 (points)
ReadSize	(When read in Bit units)	1 - 1920 (Bytes)	1 - 1920 (Bytes)
Size	FALSE	1 - 960 (points)	1 - 480 (points)
ReadSize	(When read in Word units)	2 - 1920 (Bytes)	4 - 1920 (Bytes)

Example of 8 Bit Read-Out from M100 to M107 in Bit Units

- In case of M100 to M107 are 0/0/0/1/0/0/1/1
- BINARY mode:

Size 8

ReadDat	[0]	[1]	[2]	[3]
	0x00	0x01	0x00	0x11
	M100/M101	M102/M103	M104/M105	M106/M107

ReadSize 4

- ASCII mode:

Size 8

ReadDat	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
	0	0	0	1	0	0	1	1
	0x30	0x30	0x30	0x31	0x30	0x30	0x31	0x31
	M100			...				M107

ReadSize 8

Example of 2 Word Read-Out from M100 to M131 (Bit Device) in Word Units

In case of M100 to M131 are 0010/1100/0100/1000/0100/0000/0000/0000

- BINARY mode:

Size 2

ReadDat	[0]	[1]	[2]	[3]
	0x34	0x12	0x02	0x00
	M107...M100	M115...M108	M123...M116	M131...M124

ReadSize 4

- ASCII mode:

Size 2

ReadDat	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
	1	2	3	4	0	0	0	2
	0x31	0x32	0x33	0x34	0x30	0x30	0x30	0x32
	M115	M100	M131			M116

ReadSize 8

Example of 3 Word Read-Out from D100 to D102 (Word Device) in Word Units

In case of D100 is 0x1234, D101 is 0x0002, D102 is 0x1DEF

- BINARY mode:

Size 3

ReadDat	[0]	[1]	[2]	[3]	[4]	[5]
	0x34	0x12	0x02	0x00	0xEF	0x1D
	D100		D101		D102	

ReadSize 6

- ASCII mode:

Size 3

ReadDat	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
	1	2	3	4	0	0	0	2	1	D	E	F
	0x31	0x32	0x33	0x34	0x30	0x30	0x30	0x32	0x31	0x44	0x45	0x46
	D100			D101				D102				

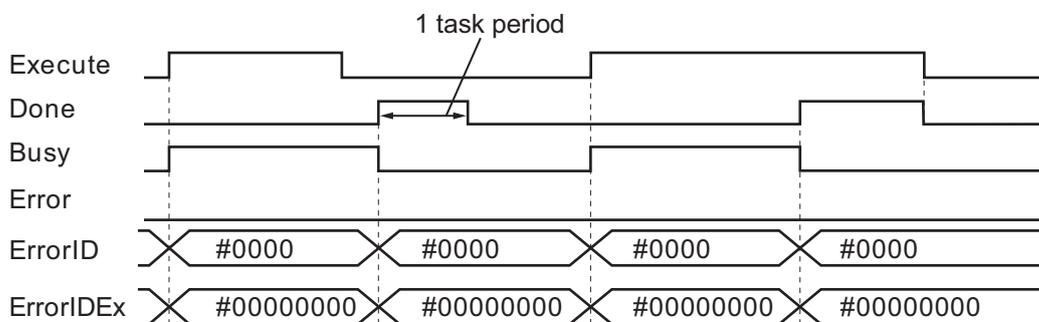
ReadSize 12

Timing Chart

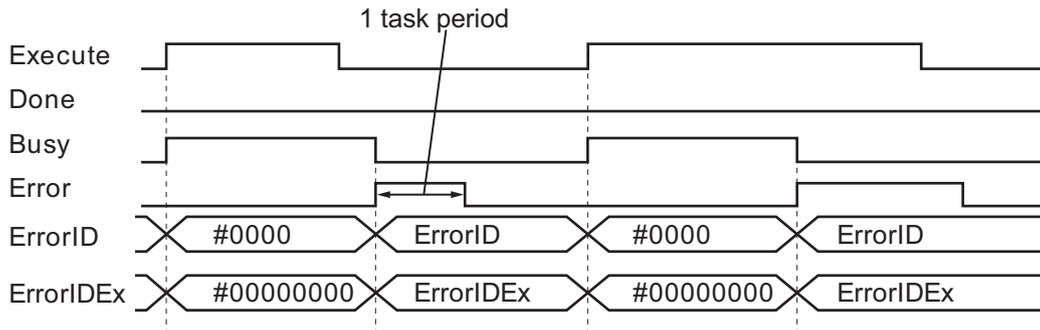
The timing charts are shown below.

- *Busy* (Executing) changes to TRUE when *Execute* changes to TRUE.
- When a response message from the Mitsubishi Electric Sequencer is received normally, and End-Code=0, *Done* changes to TRUE.
- If an error occurs when execution of the function block is in progress, *Error* changes to TRUE and *Busy* (Executing) changes to FALSE.
You can find out the cause of the error by accessing the values output to *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code).
- If *Execute* changes to FALSE before execution of the FB is ended, *Done* and *Error* are TRUE only for one task period.
- If *Execute* remains TRUE even after execution of the function block is ended, the output values of *Done* and *Error* are retained.

- Timing Chart for Normal End



- Timing Chart for Error End



Additional Information

- For this FB, use the socket service function. Refer to *NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual (W506)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP™ Port User's Manual (W563)* for details of the socket service function.

Precautions for Correct Use

- Execution of this instruction is continued until processing is completed even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is completed. Use this to confirm normal completion of processing.
- This FB executes one SktUDPRcv instruction and one SktUDPSend instruction internally. For the number of simultaneous executions, refer to the SktTCPRcv and SktTCPSend in *NJ/NX-series Instructions Reference Manual (W502)* or *NY-series Instructions Reference Manual (W560)*.
- When the same socket is shared by FBs of multiple SLMP communications instructions, multiple SLMP communications instructions cannot be executed simultaneously. SLMP communications instructions cannot distinguish responses to commands sent by themselves from responses to commands sent by other SLMP communications instructions. Therefore, wait until the execution of one SLMP communications instruction is complete and then execute the next SLMP communications instruction.
- You should specify the values mean destination to *Socket.DstAdr.PortNo* and *Socket.DstAdr.IpAdr*.



Precautions for Correct Use

Refer to 2-2 *Usage Method* on page 2-4 for details.

Troubleshooting

Error code	Expansion error code	Error name	Description	Corrective action
16#0000	16#00000000	Normal end	—	—

Error code	Expansion error code	Error name	Description	Corrective action
16#0400	16#00000000	Input Value Out of Range	An input parameter for an instruction exceeded the valid range for an input variable. Or, division by an integer of 0 occurred in division or remainder calculations.	Refer to the description of <i>Machine Automation Controller Troubleshooting Manual (W503)</i> for the event code with 5401 appended to the upper 4 digits of the Error code. For example, if the Error code is 16#0400, refer to the description of event code 54010400 hex.
16#0406	16#00000000	Illegal Data Position Specified	A memory address or data size that was specified for the instruction is not suitable.	
16#0407	16#00000000	Data Range Exceeded	The results of instruction processing exceeded the data area range of the output parameter.	
16#0419	16#00000000	Incorrect Data Type	A data type that cannot be used for an instruction is specified for an input or in-out variable.	
16#2002	16#00000000	Address Resolution Failed	Address resolution failed for a remote node with the host name that was specified in the instruction.	
16#2003	16#00000000	Socket Status Error	The status was not suitable for execution of the socket service instruction.	
16#2006	16#00000000	Socket Timeout	A timeout occurred for a socket service instruction.	
16#2007	16#00000000	Socket Handle Out of Range	The handle that is specified for the socket service instruction is not correct.	
16#2008	16#00000000	Socket Communications Resource Overflow	The maximum resources that you can use for socket service instructions at the same time was exceeded.	
16#3D27	16#00000001	Illegal device code	This function block specified an unsupported Device Code.	
16#3D27	16#00000002	Illegal first device No.	This FB specified an unsupported first Device No.	Check the FB input variable <i>Offset</i> .
16#3D27	16#00000003	Illegal number of device points	Using a combination of the input variables <i>BitAccess</i> and <i>Size</i> , the read data size exceeded the controller specification upper limit value.	Check the FB input variables <i>BitAccess</i> and <i>Size</i> .

Error code	Expansion error code	Error name	Description	Corrective action
16#3D27	16#00000004	ReadDat upper limit exceeded	The array size allocated to in-out variable <i>ReadDat</i> exceeds the upper limit.	Confirm the number of elements in the array allocated to the In-out variable <i>ReadDat</i> in this FB is 1 to 1920.
16#3D27	16#00000010	Response reception timeout	A response message could not be received within the time specified in input variable <i>TimeOut</i> .	Check the network status and communications target Mitsubishi Electric Sequencer status. Or, increase the value of the input variable <i>TimeOut</i> of this FB.
16#3D27	16#FFFF0001 - 16#FFFFFFFF	SLMP end code	An error response message was received.	Refer to <i>End codes</i> in the SLMP Specifications Overview for troubleshooting. For information on how to obtain the SLMP Specifications Overview, contact the CC-Link Partner Association.

Sample program

From our website (http://www.ia.omron.com/sysmac_library/), you can download sample programs using SLMP_DeviceRead_UDP. You can use with MELSEC iQ-R Series, iQ-F Series, Q Series, and L Series.



Precautions for Correct Use

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

The overview of the sample programming is as follows:

Conditions

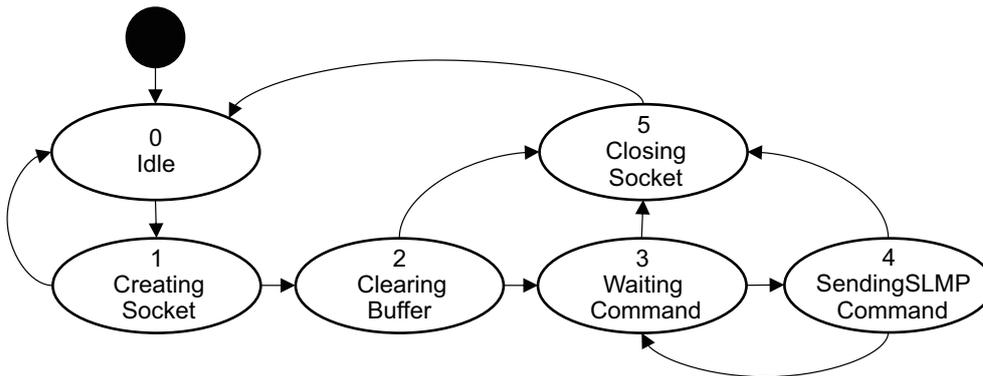
Item	Description
Process	Read-Out 8 Bit from M100 to M107 in Bit Units
Destination IP Address	192.168.250.39
Destination UDP port number	5100
Source UDP port number	60000
Communication code	BINARY
Timeout time	2 seconds (default)

Process

- 1** Execute the SktUDPCreate instruction, and create a UDP socket.
Then, set the values that specify the destination in *DstAdr.PortNo* and *DstAdr.IpAdr* of the created UDP socket.
- 2** Execute the SktClearBuf instruction, and clear the receive buffer of TCP socket.
- 3** Confirm that no other communications FB execution is in progress.
- 4** Launch the SLMP_DeviceRead_UDP FB-instance.
- 5** After completing FB execution, check the read data.
- 6** Execute the SktClose instruction, and close the UDP socket.

State transition of sample program

The state transition of the sample programming is as follows:



No	State name	Processing
0	Idle	This is the state waiting for the request of creating UDP socket. The UDP socket has not been created and is in a waiting state.
1	CreatingSocket	This is the state to create a UDP socket.
2	ClearingBuffer	This is the state to clear the receive buffer of the UDP socket.
3	WatingCommand	This is the state waiting for a request to send a command for SLMP communication or a request to close the UDP socket. The UDP socket has been created and is in a waiting state.
4	SendingSLMPCommand	This is the state sends SLMP communication commands and receives responses to the destination station.
5	ClosingSocket	This is the state to close the UDP socket.

SLMP_DeviceWriteWord_UDP

Through SLMP communication using UDP socket service, to write Device of Mitsubishi Electric Sequencer in word units.

Function block name	Name	FB/FUN	Graphic expression	ST expression
SLMP_DeviceWriteWord_UDP	Internal memory batch write (Word, UDP)	FB		<pre>SLMP_DeviceWriteWord_UDP_instance(Execute:=, Socket:=, Mode:=, DeviceCode:=, Offset:=, Size:=, TimeOut:=, Done=>, Busy=>, Error=>, ErrorID=>, ErrorIDEx=>, WriteDat:=);</pre>

Function Block and Function Information

Item	Description
Library file name	OmronLib_SLMP_Comm_Vx_x.slr (x shows the unit version.)
Namespace	OmronLib\SLMP_Comm
Function block and function number	00247
Source code published/not published	Not Published



Version Information

You can use this FB with version 2.0 or later of this library.

Input variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Execute	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	—	FALSE
Socket	Socket	_sSOCKET*1	Socket	—	—	—

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Mode	Mode	BOOL	SLMP frame communicating code TRUE: ASCII mode FALSE: BINARY mode	TRUE, FALSE	—	FALSE
Device-Code	Device code	UINT	Input the Device Code for the read target.*2	Follow the specifications of the Connected station	—	UINT#0
Offset	First device No.	UDINT	Input the first number of the read Device.*3	Follow the specifications of the Connected station	—	UDINT#0
Size	Number of device points	UINT	Input the number of read Device.*3	Follow the specifications of the Connected station	—	UINT#0
TimeOut	Timeout time	UINT	Input the timeout time for FB. If "0" is set, the timeout time is 2.0 s.	Depends on data type.	0.1 s	UINT#0

*1. Refer to the SkUDPCreate in the *NJ/NX-series Instructions Reference Manual (W508)* or *NY-series Instructions Reference Manual (W560)* for details of the Data type.

*2. Refer to *2-3-1 Device Code (Device Access for Sequencer CPU Made by Mitsubishi Electric)* on page 2-5 for details.

*3. Refer to *Function* on page 4-55 for details.

Output variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Done	Done	BOOL	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE, FALSE	—	—
Busy	Executing	BOOL	TRUE: Executing. FALSE: Not executing.	TRUE, FALSE	—	—
Error	Error	BOOL	TRUE: Error end. FALSE: Normal end, executing, or execution condition not met	TRUE, FALSE	—	—

*1. Refer to the *Troubleshooting* on page 4-59 for details.

In-out variables

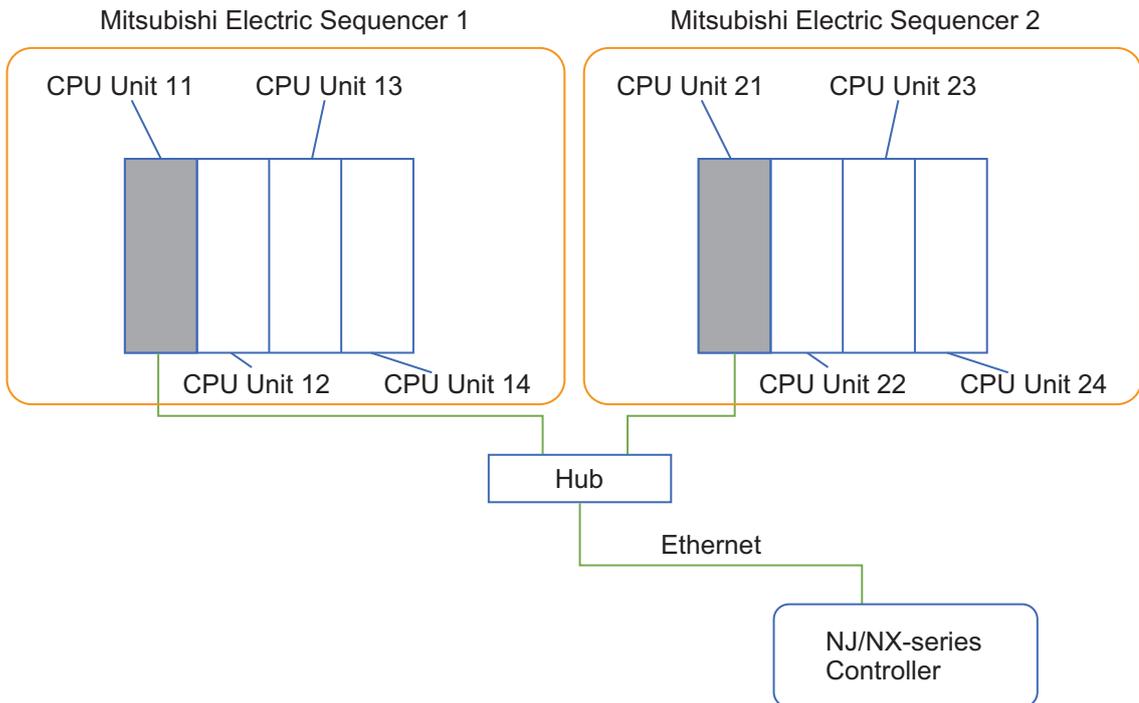
Variable	Meaning	Data type	Description	Valid range	Unit	Default
WriteDat	Write data	ARRAY[*] OF WORD*1	Input the write data.*2	Depends on data type.	—	—

*1. You can input array variables of any length WORD type. The number of elements in the array ranges from 1 to 960 (when in BINARY mode) and 1 to 480 (when in ASCII mode). WORD arrays can start with either element 0 or with element n. You cannot input WORD type variables of the base data type.

- *2. You should input in the communicating code specified in input variable *Mode*. Conversion from BINARY to ASCII and conversion from ASCII to BINARY are not executed.

Function

- This FB sends an ST (Single-Transmission) type request message for SLMP communication.
- The supported communication model by ST type is client/server (single request) type.
- Use SLMP communications to write in word units to the sequencer device made by Mitsubishi Electric on the connected station.
- In the image below, the Mitsubishi Electric Sequencers that can write-out are CPU Unit 11 and CPU Unit 21. CPU Units 12 to 14 and CPU Units 22 to 24 do not support write-out.



- Writes the amount of the input variable *Size* from the input variable *Offset* to the Device specified by the input variable *DeviceCode*.
- The SLMP request message is based on the input variable *Mode*, and created as Write (Command: 1401, Sub-command: 0000).
- Write data is stored in the in-out variable *WriteDat*.
- Since the data type of the input variable *Offset* is UDINT, the base of the data is decimal number, decimal data. When specifying a Device with a hexadecimal number, convert the input value to a decimal number or enter a hexadecimal constant (literal), etc.
- The possible input range for the input variable *Size*, based on the input variable *Mode* and input variable *BitAccess*, and the SLMP request message write data length are as shown below. When *Size* is set to 0, the command will be completed without sending command (the output variable *Done* is TRUE).

Item	Conditions	
	Mode	
	FALSE (BINARY mode)	TRUE (ASCII mode)
Size	1 - 960 (points)	1 - 480 (points)

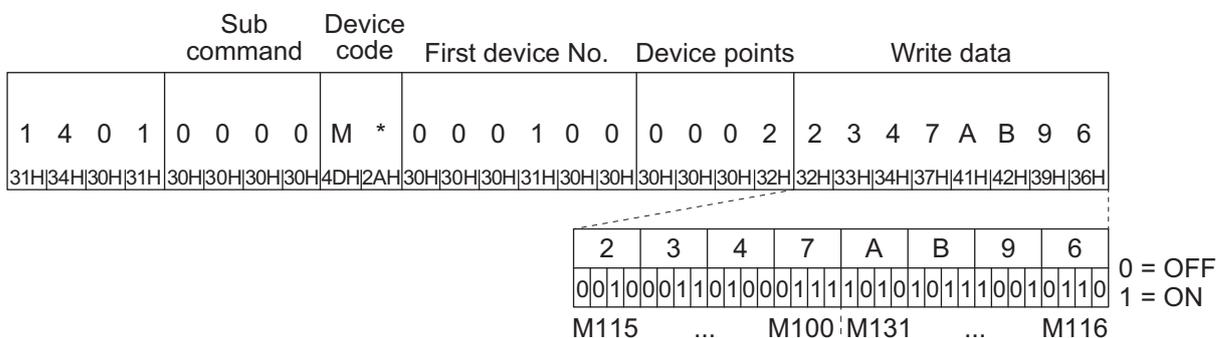
Item	Conditions	
	Mode	
	FALSE (BINARY mode)	TRUE (ASCII mode)
SLMP request message write data length	2 - 1920 Bytes	4 - 1920 Bytes

Communicating Example (when writing in word units (bit device))

Write the value from M100 to M131 (2 word part).

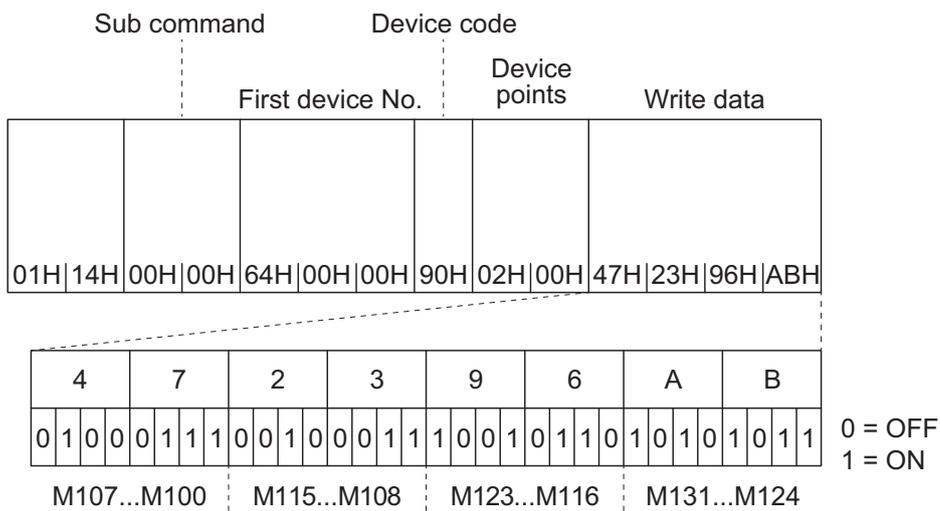
● When Data Communications are in ASCII Code

(Request data)



● When Data Communications are in Binary Code

(Request data)

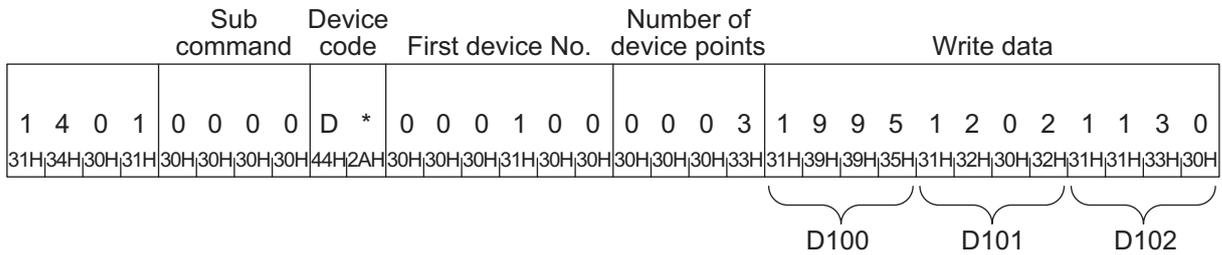


Communicating Example (when writing in word units (word device))

Write the value from D100 to D102 (2 word part).

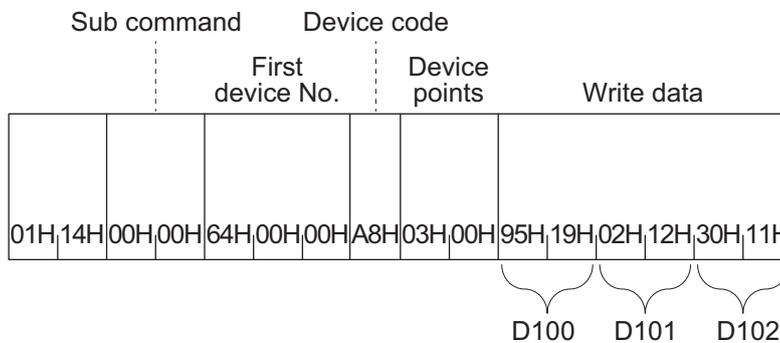
● When Data Communications are in ASCII Code

(Request data)



● When Data Communications are in Binary Code

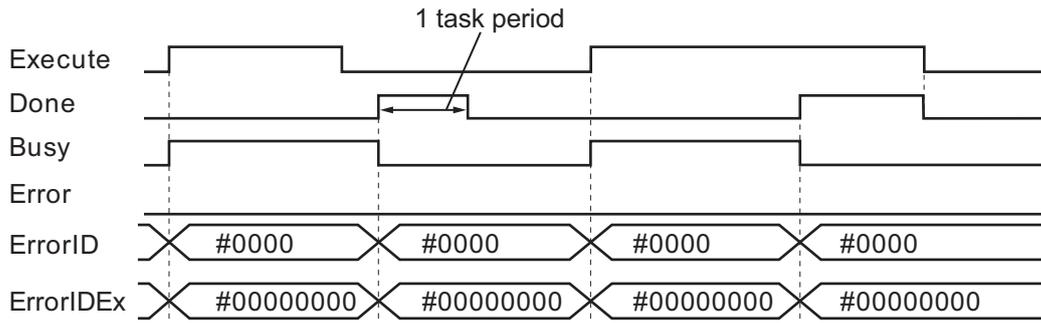
(Request data)



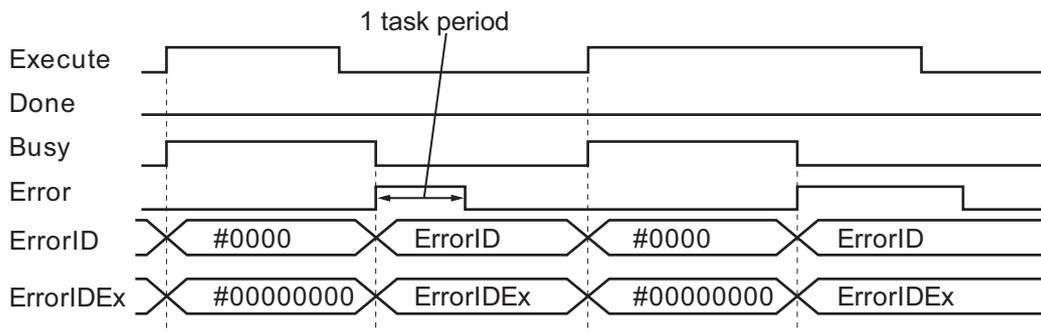
Timing Chart

The timing charts are shown below.

- *Busy* (Executing) changes to TRUE when *Execute* changes to TRUE.
- When a response message from the Mitsubishi Electric Sequencer is received normally, and End-Code=0, *Done* changes to TRUE.
- If an error occurs when execution of the function block is in progress, *Error* changes to TRUE and *Busy* (Executing) changes to FALSE.
You can find out the cause of the error by accessing the values output to *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code).
- If *Execute* changes to FALSE before execution of the FB is ended, *Done* and *Error* are TRUE only for one task period.
- If *Execute* remains TRUE even after execution of the function block is ended, the output values of *Done* and *Error* are retained.
- Timing Chart for Normal End



• Timing Chart for Error End



Additional Information

- For this FB, use the socket service function. Refer to *NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual (W506)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP™ Port User's Manual (W563)* for details of the socket service function.

Precautions for Correct Use

- Execution of this instruction is continued until processing is completed even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is completed. Use this to confirm normal completion of processing.
- This FB executes one SktUDPRcv instruction and one SktUDPSend instruction internally. For the number of simultaneous executions, refer to the SktTCPRcv and SktTCPSend in *NJ/NX-series Instructions Reference Manual (W502)* or *NY-series Instructions Reference Manual (W560)*.
- When the same socket is shared by FBs of multiple SLMP communications instructions, multiple SLMP communications instructions cannot be executed simultaneously. SLMP communications instructions cannot distinguish responses to commands sent by themselves from responses to commands sent by other SLMP communications instructions. Therefore, wait until the execution of one SLMP communications instruction is complete and then execute the next SLMP communications instruction.
- You should specify the values mean destination to *Socket.DstAdr.PortNo* and *Socket.DstAdr.IpAdr*.



Precautions for Correct Use

- Refer to *2-2 Usage Method* on page 2-4 for details.
- If "Write during RUN" is prohibited on the Mitsubishi Electric Sequencer side, confirm that the Mitsubishi Electric Sequencer is in STOP state, and then execute.

Troubleshooting

Error code	Expansion error code	Error name	Description	Corrective action
16#0000	16#00000000	Normal end	—	—
16#0400	16#00000000	Input Value Out of Range	An input parameter for an instruction exceeded the valid range for an input variable. Or, division by an integer of 0 occurred in division or remainder calculations.	Refer to the description of <i>Machine Automation Controller Troubleshooting Manual (W503)</i> for the event code with 5401 appended to the upper 4 digits of the Error code. For example, if the Error code is 16#0400, refer to the description of event code 54010400 hex.
16#0406	16#00000000	Illegal Data Position Specified	A memory address or data size that was specified for the instruction is not suitable.	
16#0407	16#00000000	Data Range Exceeded	The results of instruction processing exceeded the data area range of the output parameter.	
16#0419	16#00000000	Incorrect Data Type	A data type that cannot be used for an instruction is specified for an input or in-out variable.	
16#2002	16#00000000	Address Resolution Failed	Address resolution failed for a remote node with the host name that was specified in the instruction.	
16#2003	16#00000000	Socket Status Error	The status was not suitable for execution of the socket service instruction.	
16#2006	16#00000000	Socket Timeout	A timeout occurred for a socket service instruction.	
16#2007	16#00000000	Socket Handle Out of Range	The handle that is specified for the socket service instruction is not correct.	
16#2008	16#00000000	Socket Communications Resource Overflow	The maximum resources that you can use for socket service instructions at the same time was exceeded.	
16#3D28	16#00000001	Illegal device code	This function block specified an unsupported Device Code.	Check the FB input variable <i>DeviceCode</i> .
16#3D28	16#00000002	Illegal first device No.	This FB specified an unsupported first Device No.	Check the FB input variable <i>Offset</i> .
16#3D28	16#00000003	Illegal number of device points	Using the input variable <i>Size</i> , the write data size exceeded the controller specification upper limit value.	Check the FB input variable <i>Size</i> .

Error code	Expansion error code	Error name	Description	Corrective action
16#3D28	16#00000004	Insufficient WriteDat size	Array size allocated to <i>WriteDat</i> is less than the write data length specified in <i>Mode</i> , <i>Size</i> .	Check the FB input variables <i>Mode</i> and <i>Size</i> . Or check the number of elements in the array to assign to the In-out variable <i>WriteDat</i> in this FB.
16#3D28	16#00000005	WriteDat upper limit exceeded	The array size allocated to In-out variables <i>WriteDat</i> exceeds the upper limit.	Check that the array size allocated to In-out variables <i>WriteDat</i> is 1 - 960 (when in BINARY mode), 1 - 480 (when in ASCII mode).
16#3D28	16#00000010	Response reception timeout	A response message could not be received within the time specified in input variable <i>TimeOut</i> .	Check the network status and communications target Mitsubishi Electric Sequencer status. Or, increase the value of the input variable <i>TimeOut</i> of this FB.
16#3D28	16#FFFF0001 - 16#FFFFFFFF	SLMP end code	An error response message was received.	Refer to <i>End codes</i> in the SLMP Specifications Overview for troubleshooting. For information on how to obtain the SLMP Specifications Overview, contact the CC-Link Partner Association.

Sample program

From our website (http://www.ia.omron.com/sysmac_library/), you can download sample programs using SLMP_DeviceWriteWord_UDP. You can use with MELSEC iQ-R Series, iQ-F Series, Q Series, and L Series.



Precautions for Correct Use

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

The overview of the sample programming is as follows:

Conditions

Item	Description
Process	Write the value from M100 to M131 (2 words).
Destination IP Address	192.168.250.39

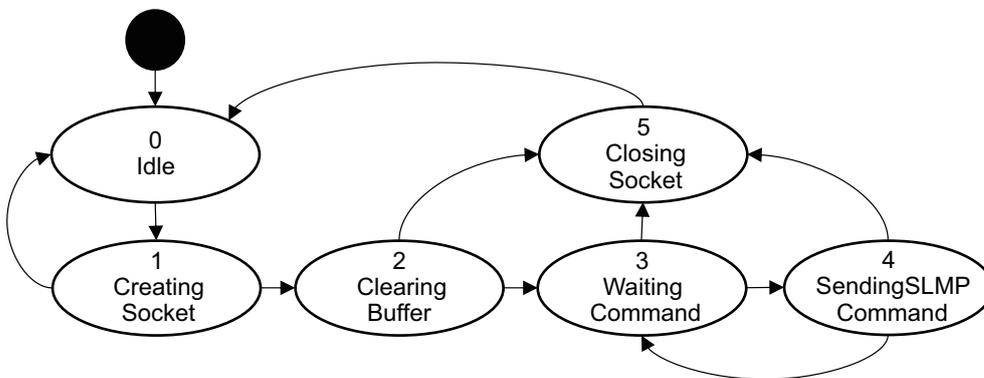
Item	Description
Destination UDP port number	5100
Source UDP port number	60000
Communication code	BINARY
Timeout time	2 seconds (default)

Process

- 1** Execute the SktUDPCreate instruction, and create a UDP socket.
Then, set the values that specify the destination in *DstAdr.PortNo* and *DstAdr.IpAdr* of the created UDP socket.
- 2** Execute the SktClearBuf instruction, and clear the receive buffer of TCP socket.
- 3** Confirm that no other communications FB execution is in progress.
- 4** Launch the SLMP_DeviceWriteWord_UDP FB-instance.
- 5** Execute the SktClose instruction, and close the UDP socket.

State transition of sample program

The state transition of the sample programming is as follows:



No	State name	Processing
0	Idle	This is the state waiting for the request of creating UDP socket. The UDP socket has not been created and is in a waiting state.
1	CreatingSocket	This is the state to create a UDP socket.
2	ClearingBuffer	This is the state to clear the receive buffer of the UDP socket.
3	WaitingCommand	This is the state waiting for a request to send a command for SLMP communication or a request to close the UDP socket. The UDP socket has been created and is in a waiting state.
4	SendingSLMPCommand	This is the state sends SLMP communication commands and receives responses to the destination station.
5	ClosingSocket	This is the state to close the UDP socket.

SLMP_DeviceWriteBool_UDP

Through SLMP communication using UDP socket service, to write Device of Mitsubishi Electric Sequencer in bit units.

Function block name	Name	FB/FUN	Graphic expression	ST expression
SLMP_DeviceWriteBool_UDP	Internal memory Batch writing (Bit, UDP)	FB		<pre>SLMP_DeviceWriteBool_UDP_instance(Execute:=, Socket:=, Mode:=, DeviceCode:=, Offset:=, Size:=, TimeOut:=, Done=>, Busy=>, Error=>, ErrorID=>, ErrorIDEx=>, WriteDat:=);</pre>

Function Block and Function Information

Item	Description
Library file name	OmronLib_SLMP_Comm_Vx_x.slr (x shows the unit version.)
Namespace	OmronLib\SLMP_Comm
Function block and function number	00248
Source code published/not published	Not Published



Version Information

You can use this FB with version 2.0 or later of this library.

Input variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Execute	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	—	FALSE
Socket	Socket	_sSOCKET*1	Socket	—	—	—

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Mode	Mode	BOOL	SLMP frame communicating code TRUE: ASCII mode FALSE: BINARY mode	TRUE, FALSE	—	FALSE
Device-Code	Device code	UINT	Input the Device Code for the read target.*2	Follow the specifications of the Connected station	—	UINT#0
Offset	First device No.	UDINT	Input the first number of the read Device.*3	Follow the specifications of the Connected station	—	UDINT#0
Size	Number of device points	UINT	Input the number of read Device.*3	Follow the specifications of the Connected station	—	UINT#0
TimeOut	Timeout time	UINT	Input the timeout time for FB. If "0" is set, the timeout time is 2.0 s.	Depends on data type.	0.1s	UINT#0

*1. Refer to the SktUDPCreate in the *NJ/NX-series Instructions Reference Manual (W508)* or *NY-series Instructions Reference Manual (W560)* for details of the Data type.

*2. Refer to *2-3-1 Device Code (Device Access for Sequencer CPU Made by Mitsubishi Electric)* on page 2-5 for details.

*3. Refer to *Function* on page 4-64 for details.

Output variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Done	Done	BOOL	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE, FALSE	—	—
Busy	Executing	BOOL	TRUE: Executing. FALSE: Not executing.	TRUE, FALSE	—	—
Error	Error	BOOL	TRUE: Error end. FALSE: Normal end, executing, or execution condition not met	TRUE, FALSE	—	—
ErrorID	Error Code	WORD	This is the error ID for an error end. The value is 16#0 for a normal end.	*1	—	—
ErrorIDEx	Expansion Error Code	DWORD	This is the error ID for an Expansion Unit Hardware Error. The value is 16#0 for a normal end.	*1	—	—

*1. Refer to the *Troubleshooting* on page 4-67 for details.

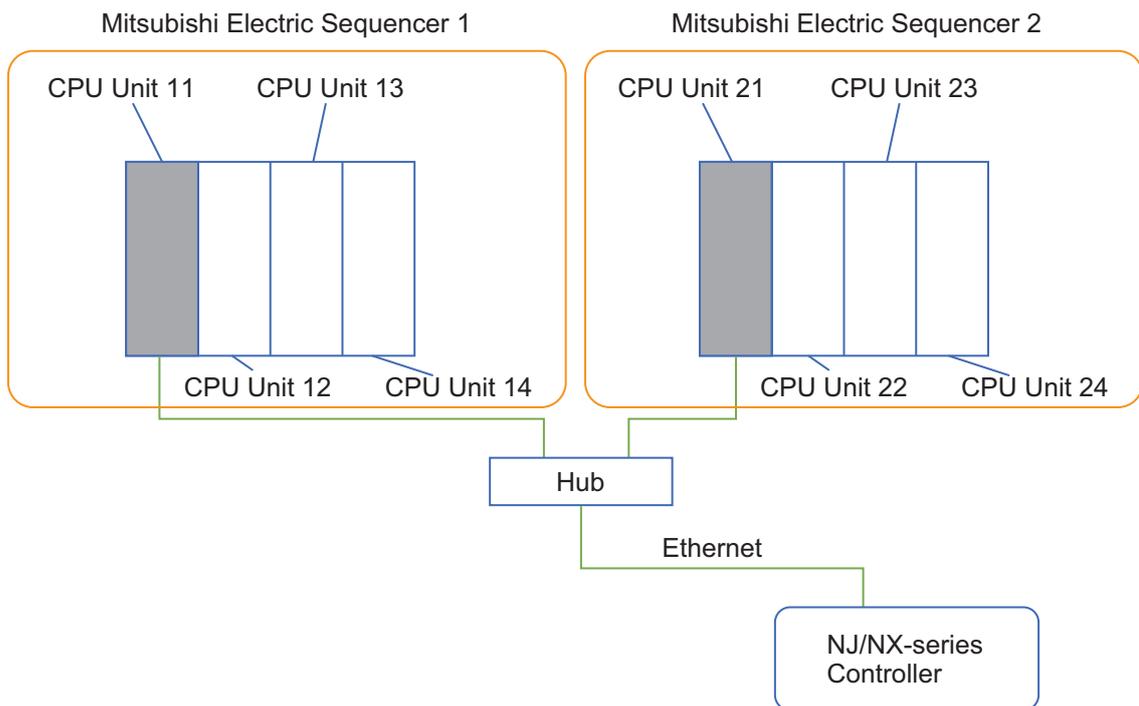
Input-Output Variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
WriteDat	Write data	ARRAY[*] OF BOOL *1	Input the write data. *2	Depends on data type.	—	—

- *1. You can input array variables of any length BOOL type. Array length is in the range of 1 - 3840 (when in BINARY mode), 1 - 1920 (when in ASCII mode). Arrays can start with either element 0 or with element n. You cannot input BOOL type variables of the base data type.
- *2. You should input in the communicating code specified in input variable *Mode*. Conversion from BINARY to ASCII and conversion from ASCII to BINARY are not executed.

Function

- This FB sends an ST (Single-Transmission) type request message for SLMP communication.
- The supported communication model by ST type is client/server (single request) type.
- Use SLMP communications to write in bit units to the sequencer device made by Mitsubishi Electric on the connected station.
- In the image below, the Mitsubishi Electric Sequencers that can write-out are CPU Unit 11 and CPU Unit 21. CPU Units 12 to 14 and CPU Units 22 to 24 do not support write-out.



- Writes the amount of the input variable *Size* from the input variable *Offset* to the Device specified by the input variable *DeviceCode*.
- The SLMP request message is based on the input variable *Mode*, and created as Write (Command: 1401, Sub-command: 0001).
- Write data is stored in the in-out variable *WriteDat*.
- Since the data type of the input variable *Offset* is UDINT, the base of the data is decimal number, decimal data. When specifying a Device with a hexadecimal number, convert the input value to a decimal number or enter a hexadecimal constant (literal), etc.

- The input range for the input variable *Size*, based on the Input variable *Mode*, and the SLMP request message write data length, are as shown below.
When *Size* is set to 0, the command will be completed without sending command (the output variable *Done* is TRUE).

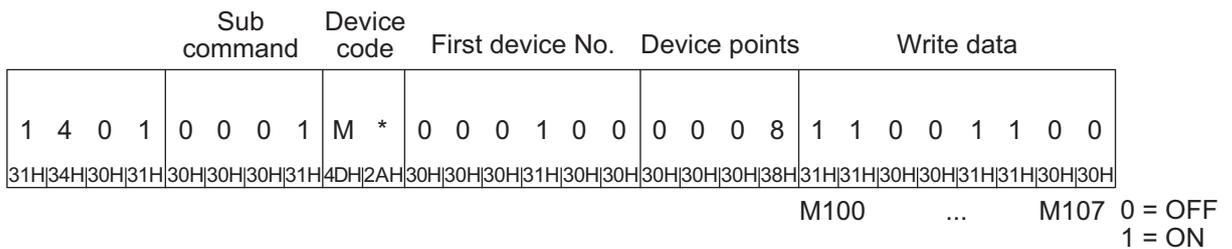
Item	Conditions	
	Mode	
	FALSE (BINARY mode)	TRUE (ASCII mode)
Size	1 - 3840 (points)	1 - 1920 (points)
SLMP request message write data length	1 - 1920 Bytes	1 - 1920 Bytes

Communicating Example (when writing in bit units)

Write the value from M100 to M107.

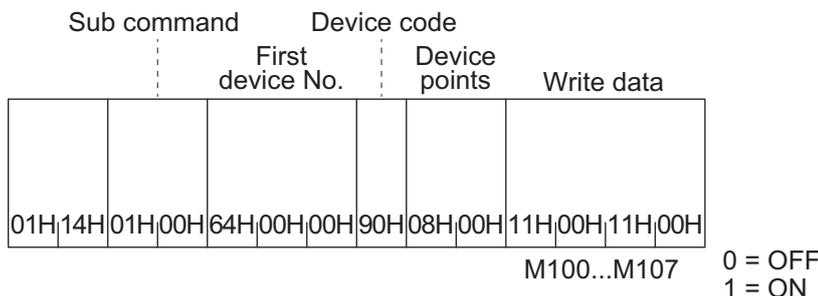
When Data Communications are in ASCII Code

(Request data)



When Data Communications are in Binary Code

(Request data)



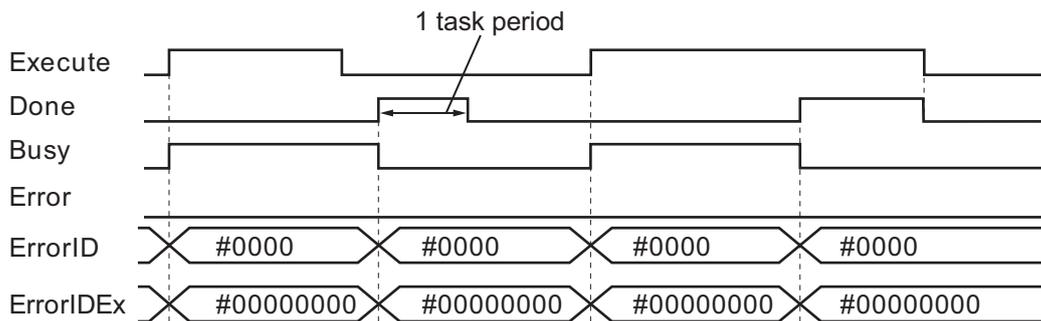
Timing Chart

The timing charts are shown below.

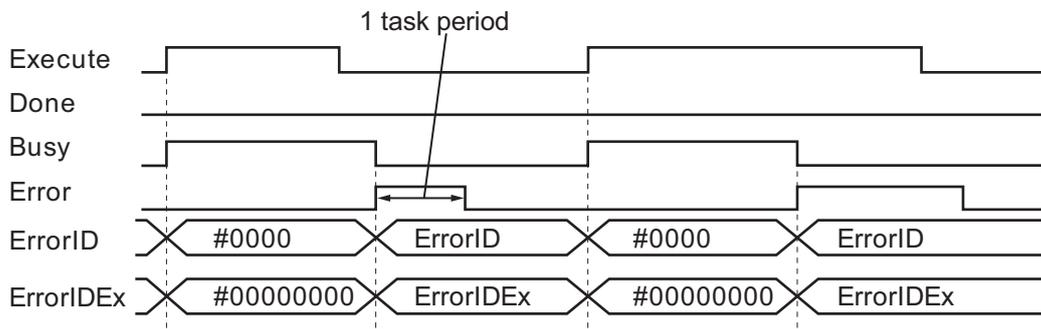
- Busy* (Executing) changes to TRUE when *Execute* changes to TRUE.
- When a response message from the Mitsubishi Electric Sequencer is received normally, and End-Code=0, *Done* changes to TRUE.

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

- Timing Chart for Normal End



- Timing Chart for Error End



Additional Information

- For this FB, use the socket service function. Refer to *NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual (W506)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP™ Port User's Manual (W563)* for details of the socket service function.

Precautions for Correct Use

- Execution of this instruction is continued until processing is completed even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is completed. Use this to confirm normal completion of processing.
- This FB executes one SktUDPRcv instruction and one SktUDPSend instruction internally. For the number of simultaneous executions, refer to the SktTCPRcv and SktTCPSEND in *NJ/NX-series Instructions Reference Manual (W502)* or *NY-series Instructions Reference Manual (W560)*.
- When the same socket is shared by FBs of multiple SLMP communications instructions, multiple SLMP communications instructions cannot be executed simultaneously. SLMP communications

instructions cannot distinguish responses to commands sent by themselves from responses to commands sent by other SLMP communications instructions. Therefore, wait until the execution of one SLMP communications instruction is complete and then execute the next SLMP communications instruction.

- You should specify the values mean destination to *Socket.DstAdr.PortNo* and *Socket.DstAdr.IpAdr*.



Precautions for Correct Use

- Refer to *2-2 Usage Method* on page 2-4 for details.
- If "Write during RUN" is prohibited on the Mitsubishi Electric Sequencer side, confirm that the Mitsubishi Electric Sequencer is in STOP state, and then execute.

Troubleshooting

Error code	Expansion error code	Error name	Description	Corrective action
16#0000	16#00000000	Normal end	—	—

Error code	Expansion error code	Error name	Description	Corrective action
16#0400	16#00000000	Input Value Out of Range	An input parameter for an instruction exceeded the valid range for an input variable. Or, division by an integer of 0 occurred in division or remainder calculations.	Refer to the description of <i>Machine Automation Controller Troubleshooting Manual (W503)</i> for the event code with 5401 appended to the upper 4 digits of the Error code. For example, if the Error code is 16#0400, refer to the description of event code 54010400 hex.
16#0406	16#00000000	Illegal Data Position Specified	A memory address or data size that was specified for the instruction is not suitable.	
16#0407	16#00000000	Data Range Exceeded	The results of instruction processing exceeded the data area range of the output parameter.	
16#0419	16#00000000	Incorrect Data Type	A data type that cannot be used for an instruction is specified for an input or in-out variable.	
16#2002	16#00000000	Address Resolution Failed	Address resolution failed for a remote node with the host name that was specified in the instruction.	
16#2003	16#00000000	Socket Status Error	The status was not suitable for execution of the socket service instruction.	
16#2006	16#00000000	Socket Timeout	A timeout occurred for a socket service instruction.	
16#2007	16#00000000	Socket Handle Out of Range	The handle that is specified for the socket service instruction is not correct.	
16#2008	16#00000000	Socket Communications Resource Overflow	The maximum resources that you can use for socket service instructions at the same time was exceeded.	
16#3D29	16#00000001	Illegal device code	This function block specified an unsupported Device Code.	
16#3D29	16#00000002	Illegal first device No.	This FB specified an unsupported first Device No.	Check the FB input variable <i>Offset</i> .
16#3D29	16#00000003	Illegal number of device points	Using the input variable <i>Size</i> , the write data size exceeded the controller specification upper limit value.	Check the FB input variable <i>Size</i> .

Error code	Expansion error code	Error name	Description	Corrective action
16#3D29	16#00000004	Insufficient WriteDat size	Array size allocated to <i>WriteDat</i> is less than the write data length specified in <i>Mode</i> , <i>Size</i> .	Check the FB input variables <i>Mode</i> and <i>Size</i> . Or check the number of elements in the array to assign to the In-out variable <i>WriteDat</i> in this FB.
16#3D29	16#00000005	WriteDat upper limit exceeded	The array size allocated to In-out variables <i>WriteDat</i> exceeds the upper limit.	Confirm the array size allocated to in-out variable <i>WriteDat</i> is 1 - 3840 (when in BINARY mode), 1 - 1920 (when in ASCII mode).
16#3D29	16#00000010	Response reception timeout	A response message could not be received within the time specified in input variable <i>TimeOut</i> .	Check the network status and communications target Mitsubishi Electric Sequencer status. Or, increase the value of the input variable <i>TimeOut</i> of this FB.
16#3D29	16#FFFF0001 - 16#FFFFFFFF	SLMP end code	An error response message was received.	Refer to <i>End codes</i> in the SLMP Specifications Overview for troubleshooting. For information on how to obtain the SLMP Specifications Overview, contact the CC-Link Partner Association.

Sample program

From our website (http://www.ia.omron.com/sysmac_library/), you can download sample programs using SLMP_DeviceWriteBool_UDP. You can use with MELSEC iQ-R Series, iQ-F Series, Q Series, and L Series.



Precautions for Correct Use

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

The overview of the sample programming is as follows:

Conditions

Item	Description
Process	Write values from M100 to M107
Destination IP Address	192.168.250.39

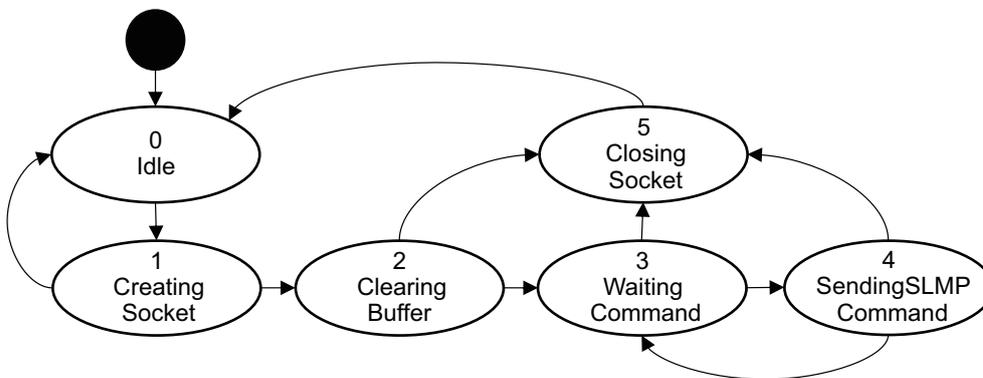
Item	Description
Destination UDP port number	5100
Source UDP port number	60000
Communication code	BINARY
Timeout time	2 seconds (default)

Process

- 1** Execute the SktUDPCreate instruction, and create a UDP socket. Then, set the values that specify the destination in *DstAdr.PortNo* and *DstAdr.IpAdr* of the created UDP socket.
- 2** Execute the SktClearBuf instruction, and clear the receive buffer of TCP socket.
- 3** Confirm that no other communications FB execution is in progress.
- 4** Launch the SLMP_DeviceWriteBool_UDP FB-instance.
- 5** Execute the SktClose instruction, and close the UDP socket.

State transition of sample program

The state transition of the sample programming is as follows:



No	State name	Processing
0	Idle	This is the state waiting for the request of creating UDP socket. The UDP socket has not been created and is in a waiting state.
1	CreatingSocket	This is the state to create a UDP socket.
2	ClearingBuffer	This is the state to clear the receive buffer of the UDP socket.
3	WaitingCommand	This is the state waiting for a request to send a command for SLMP communication or a request to close the UDP socket. The UDP socket has been created and is in a waiting state.
4	SendingSLMPCommand	This is the state sends SLMP communication commands and receives responses to the destination station.
5	ClosingSocket	This is the state to close the UDP socket.

SLMP_SendCmd_UDP

Through SLMP communication using UDP socket service, any request message is sent to the Mitsubishi Electric sequencer on the Connected station and a response message is received.

Function block name	Name	FB/FUN	Graphic expression	ST expression
SLMP_SendCmd_UDP	Send SLMP Command (UDP)	FB	<p style="text-align: center;">SLMP_SendCmd_UDP_instance</p>	<pre>SLMP_SendCmd_UDP_instance(Execute:=, Socket:=, Mode:=, NoResp:=, TimeOut:=, Done=>, Busy=>, Error=>, ErrorID=>, ErrorIDEx=>, ReqHeader:=, ReqDat:=, RespHeader:=, RespDat:=, RespDatSize=>);</pre>

Function Block and Function Information

Item	Description
Library file name	OmronLib_SLMP_Comm_Vx_x.slr (x shows the unit version.)
Namespace	OmronLib\SLMP_Comm
Function block and function number	00249
Source code published/not published	Not Published



Version Information

You can use this FB with version 2.0 or later of this library.

Input variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Execute	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	—	FALSE
Socket	Socket	_sSOCKET ¹	Socket	—	—	—

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Mode	Mode	BOOL	SLMP frame communicating code TRUE: ASCII mode FALSE: BINARY mode	TRUE, FALSE	—	FALSE
NoResp	response is not required	BOOL	Whether or not to response to the request message TRUE: It means the request message is not require the response. FALSE: It means the request message is require the response.	TRUE, FALSE	—	FALSE
TimeOut	Timeout time	UINT	Input the timeout time for FB. If "0" is set, the timeout time is 2.0 s.	Depends on data type.	0.1s	UINT#0

*1. Refer to the SktUDPCreate in the *NJ/NX-series Instructions Reference Manual (W508)* or *NY-series Instructions Reference Manual (W560)* for details of the Data type.

Output variables

Variable	Meaning	Data type	Description	Valid range	Unit	Default
Done	Done	BOOL	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE, FALSE	—	—
Busy	Executing	BOOL	TRUE: Executing. FALSE: Not executing.	TRUE, FALSE	—	—
Error	Error	BOOL	TRUE: Monitoring in progress. FALSE: Normal end, executing, or execution condition not met	TRUE, FALSE	—	—
ErrorID	Error Code	WORD	This is the error ID for an error end. The value is 16#0 for a normal end.	*1	—	—
ErrorIDEx	Expansion Error Code	DWORD	This is the error ID for an Expansion Unit Hardware Error. The value is 16#0 for a normal end.	*1	—	—
RespDat-Size	Response data size	UINT	The data size of the response data "resData" in the received response message is outputted.	Depends on data type.	Bytes	—

*1. Refer to the *Troubleshooting* on page 4-79 for details.

In-out variables

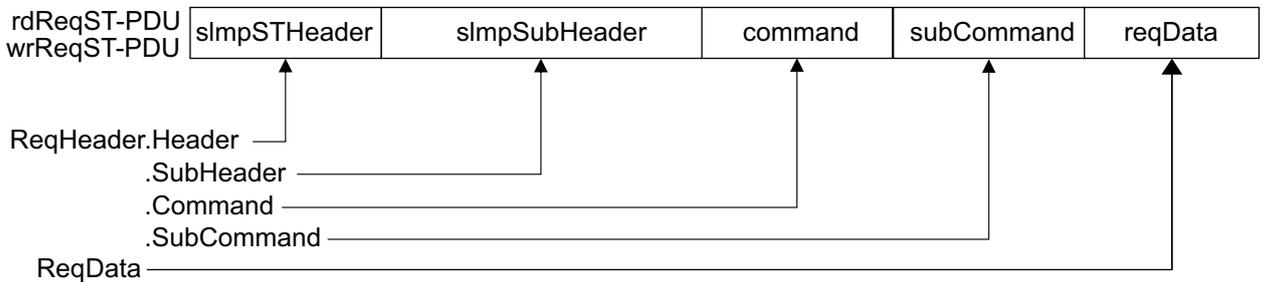
Variable	Name	Data type	Description	Valid range	Unit	Default
ReqHeader	Request header	sRdWrReq STHeader	Stores the header information of the request message.	—	—	—
ReqDat	Request data	ARRAY[*] OF BYTE*1	Input the request data for the request message. *3	—	—	—

Variable	Name	Data type	Description	Valid range	Unit	Default
RespHeader	Response header	sRdWrResSTHeader	The header information of the response message is stored.	—	—	—
RespDat	Response data	ARRAY[*] OF BYTE*2	The response data in the response message is outputted. *4*5	—	—	—

- *1. You can input array variables of any length BYTE type. Arrays can start with either element 0 or with element n. Make sure that the array is long enough to store the request data.
Refer to the *Function* on page 4-77 for details.
You cannot input BYTE type variables of the base data type.
- *2. You can input array variables of any length BYTE type. Arrays can start with either element 0 or with element n. Make sure that the array is long enough to store the response data for request message.
Refer to the *Function* on page 4-77 for details.
You cannot input BYTE type variables of the base data type.
- *3. You should input in the communicating code specified in input variable *Mode*. Conversion from BINARY to ASCII and conversion from ASCII to BINARY are not executed.
- *4. It is outputted in the communicating code specified in *Mode*. Conversion from BINARY to ASCII and conversion from ASCII to BINARY are not executed.
- *5. If the value of the output variable *RespDatSize* exceeds the number of elements in the array, the excess data is discarded.

Structures

The data type of the in-out variable *ReqHeader* in this FB is the structure OmronLib\SLMPLib\lsRdWrReqSTHeader. Each member is reflected in the SLMP frame as shown in the following figure.



The specifications are as follows:

Variable	Name	Data type	Description	Valid range	Unit	Default
ReqHeader	Request header	OmronLib \SLMPLib \lsRdWrReq STHeader	Stores the header information of the request message to be sent.	—	—	—
	Header	WORD	Input the type of communication frame. Input 16#5000.	16#5000	—	16#0000
	Sub-Header	sSubHeader1	See structure type sSubHeader1.	—	—	—
	Command	WORD	Input the command to be sent. For commands that can be sent to Mitsubishi Electric sequencers, see the destination sequencer's manual.	16#0000 to 16#FFFF	—	16#0000

Variable	Name	Data type	Description	Valid range	Unit	Default
Sub-Command	Sub command	WORD	Input the Sub-command to be sent. For commands that can be sent to Mitsubishi Electric sequencers, see the destination sequencer's manual.	16#0000 to 16#FFFF	—	16#0000

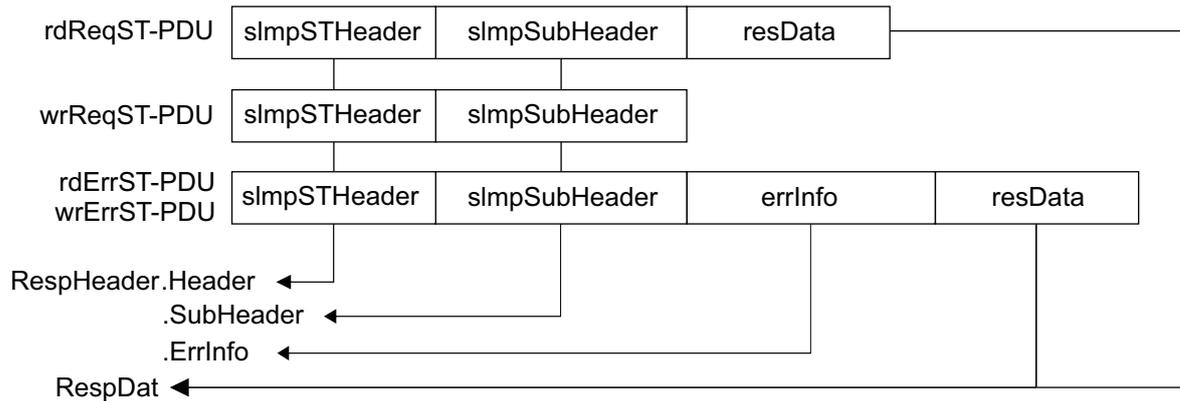
The data type of the in-out variable *ReqHeader.SubHeader* in this FB is the structure `OmronLib\SLMPLib\SubHeader1`. The specifications are as follows:

Variable	Name	Data type	Description	Valid range	Unit	Default
ReqHeader.SubHeader	Sub header	OmronLib\SLMPLib\SubHeader1	Stores the sub header information of the request message to be sent.	—	—	—
NetNo	Network number of request destination station	BYTE	Input the network number of the Connected station.	Follow the specifications of the Connected station	—	16#00
NodeNo	Station number of request destination station	BYTE	Input the station number of the Connected station.	Follow the specifications of the Connected station	—	16#00
DstProcNo	Processor number of request destination station	WORD	Input the processor number of the Connected station.	Follow the specifications of the Connected station	—	16#0000
Reserved1	System reserved	BYTE	—	16#00	—	16#00
DI	Request data length	UINT	Input the Byte size of the SLMP request message from the "timer" to the end of the "reqData" (request data). Input +6 value for BINARY Mode ^{*1} , +12 value for ASCII Mode ^{*2} in the data size of "reqData".	Depends on data type.	—	0
Timer	Timer	UINT	Input the time to wait for a response from the Connected station. The SLMP specification recommends settings in the range of 1 to 40.	0:Infinite wait 1-65535: Wait time	250 ms	0

*1. "timer" (2 Bytes) + "command" (2 Bytes) + " subCommand" (2 Bytes), total 6 Bytes

*2. "timer" (4 Bytes) + "command" (4 Bytes) + " subCommand" (4 Bytes), total 12 Bytes

The data type of the in-out variable *RespHeader* in this FB is the structure `OmronLib\SLMPLib\RdWrResSTHeader`. The value of the SLMP frame is reflected for each member, as shown in the following figure.



The specifications are as follows:

Variable	Name	Data type	Description	Valid range	Unit	Default
RespHeader	Response header	OmronLib \SLMPLib \sRdWrRes STHeader	The header information of the received response message is stored. In SLMP, "errInfo" is not appended to the normal response message, but it is always present in this structure.	—	—	—
	Header	WORD	The type of communication frame is output. 16#D000 fixed.	16#D000	—	16#0000
	Sub-Header	sSubHeader2	See structure type sSubHeader2.	—	—	—
	ErrInfo	sErrInfo	Error information is outputted when the in-out variable <i>RespHeader.SubHeader.ErrCode</i> is other than 16#0000. See structure type sErrInfo.	—	—	—

The data type of the in-out variable *RespHeader.SubHeader* in this FB is the structure OmronLib \SLMPLib\sSubHeader2. The specifications are as follows:

Variable	Name	Data type	Description	Valid range	Unit	Default
RespHeader.Sub-Header	Sub header	OmronLib \SLMPLib \sSubHeader2	The header information of the received response message is stored.	—	—	—
	NetNo	BYTE	The network number of the Connected station is outputted.	Follow the specifications of the Connected station	—	16#00
	NodeNo	BYTE	The station number of the Connected station is outputted.	Follow the specifications of the Connected station	—	16#00

Variable	Name	Data type	Description	Valid range	Unit	Default
DstProc-No	Processor number of request destination station	WORD	The processor number of the Connected station is outputted.	Follow the specifications of the Connected station	—	16#0000
Re-served1	System reserved	BYTE	—	16#00	—	16#00
DI	Request data length	UINT	The data size of the SLMP response message from the "endCode" (end code) to the "FCS", the end of the PDU, is outputted in Byte units.*1	Depends on data type.	—	0
End-Code	End code	WORD	An error code is outputted.	16#0000: Processing successful 16#0001 to 16#FFFFFF: Error	—	16#0000

*1. The meanings are shown in the following table.

Response type	Communication code	Value (Byte)
Normal response	BINARY	Data size of "resData" + 2
	ASCII	Data size of "resData" + 4
Error response	BINARY	Data size of "resData" + 11
	ASCII	Data size of "resData" + 22



Precautions for Correct Use

The value of "dl" in the SLMP response message is outputted to the in-out variable *RespHeader.SubHeader.DI* in this FB. When responding normally, the size of the in-out variable *RespHeader.ErrInfo* is not included.

You should judge the result of the execution of the request message by the in-out variable *RespHeader.SubHeader.EndCode* of this FB.

The data type of the in-out variable *RespHeader.ErrInfo* in this FB is the structure `OmronLib\SLMPLIB\ErrInfo`. The specifications are as follows:

Variable	Name	Data type	Description	Valid range	Unit	Default
RespHeader.ErrInfo	Error information of response message	OmronLib\SLMPLib\ErrInfo	The error information of the received response message is stored. When the response message is normal, 0 is stored in all members.	—	—	—
NetNo	Network number of request source station	BYTE	The network number of the requesting station is outputted.	Follow the specifications of the Connected station	—	16#00
NodeNo	Station number of request source station	BYTE	The station number of the requesting station is outputted.	Follow the specifications of the Connected station	—	16#00

Variable	Name	Data type	Description	Valid range	Unit	Default
DstProc-No	Processor number of request source station	WORD	The processor number of the requesting station is outputted.	Follow the specifications of the Connected station	—	16#0000
Re-served1	System reserved	BYTE	-	16#00	—	16#00
Command	Command	WORD	The requested command is outputted.	16#0000 to 16#FFFF	—	16#0000
Sub-Command	Sub command	WORD	The requested subcommand is outputted.	16#0000 to 16#FFFF	—	16#0000

Function

- This FB sends an ST (Single-Transmission) type request message for SLMP communication.
- The supported communication models by ST type are client/server (single request) type and client/server (response not required) type.
- This FB creates and sends a request message from the In-out variable *ReqHeader* and *ReqDat*. After sending, wait for the response from the server side, and store the received response message in the in-out variable *RespHeader* and *RespDat*.
- When ASCII mode is selected by the input variable *Mode*, this FB creates a message that converts the contents of the in-out variable *ReqHeader* to ASCII code. However, the contents of the In-out variable *ReqDat* are not converted. Therefore, enter the data created by ASCII code in *ReqDat*. This FB does not check the contents of *ReqDat*.
- The maximum Byte size of the request data "reqData" that can be sent by this FB and the response data "resData" that can be received are shown in the table below.

Communicating code of SLMP frame	Request data "reqData"	The size of the response data "resData" in case of normal response message		The size of the response data "resData" in case of error response message
		rdResST-PDU	wrResST-PDU	
BINARY	1985 (2000 - 15* ¹)	1989 (2000 - 11* ²)	0	1980 (2000 - 20* ³)
ASCII	1970 (2000 - 30* ¹)	1978 (2000 - 22* ²)	0	1960 (2000 - 40* ³)

*1. Sum of rdReqST-PDU or wrReqST-PDU other than request data "reqData"

*2. Sum of rdResST-PDU other than response data "resData"

*3. Sum of rdErrST-PDU or wrErrST-PDU other than response data "resData"

- This FB is as follows when the data size of the received response message (response header + response data) exceeds 2000 Bytes. Use this FB so that the data size of the response message does not exceed 2000 Bytes.
 - Reception processing is suspended when the received data exceeds 2000 Bytes.
 - Receive data exceeding 2000 Bytes is discarded.
 - A part of the response message may remain in the receive buffer of socket.
- When the response data size in the response message exceeds the data size of the in-out variable *RespDat*, only the size of *RespDat* is stored. Therefore, the number of elements of the BYTE

array variable used for *RespDat* should be equal to or greater than the maximum size of the response data "resData".

- When input variable *NoResp* means "no response ", this FB ends upon completion of sending a request message.



Precautions for Correct Use

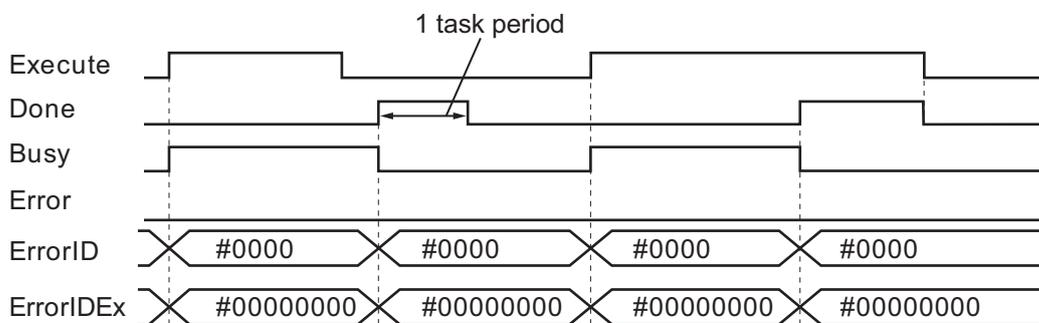
- Set the input variable *TimeOut* to a value greater than the in-out variable *ReqHeader.SubHeader.Timer*. In addition, set each value according to the communication path and processing details.
Even if this FB terminates error due to timeout according to *TimeOut* when the *ReqHeader.SubHeader.Timer* is set to a value greater than or equal to the *TimeOut* or "0" (infinite wait), the Mitsubishi Electric sequencer side continues the processing. Because of this FB cannot receive response data.
- This FB is designed to be able to send and receive request and response messages of up to 2000 Bytes. However, depending on the specifications of the target SLMP device, up to 2000 Bytes may not be able to be sent and received. Use this FB after checking the specifications of the target SLMP device.

Timing Charts

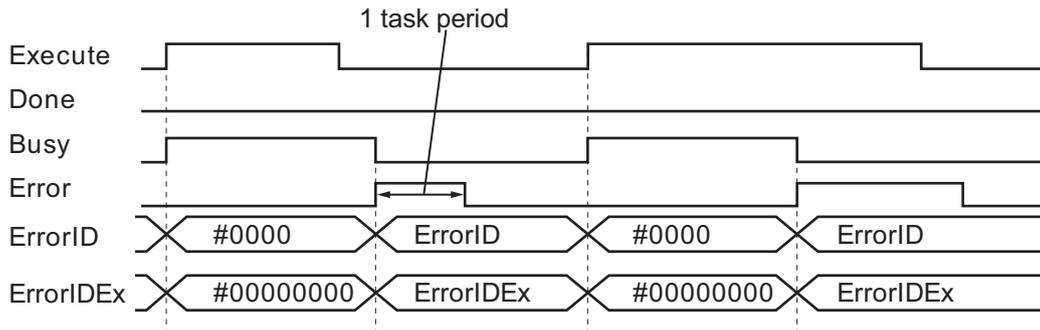
The timing chart is shown below.

- *Busy* (Executing) changes to TRUE when *Execute* changes to TRUE.
- When a response message from the Mitsubishi Electric Sequencer is received normally, and End-Code=0, *Done* changes to TRUE. If *NoResp* (no response required) is TRUE, then *Done* will be TRUE upon successful transmission of the request message.
- If an error occurs when execution of the function block is in progress, *Error* changes to TRUE and *Busy* (Executing) changes to FALSE.
You can find out the cause of the error by accessing the values output to *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code).
- If *Execute* changes to FALSE before execution of the FB is ended, *Done* and *Error* are TRUE only for one task period.
- If *Execute* remains TRUE even after execution of the function block is ended, the output values of *Done* and *Error* are retained.

- Timing Chart for Normal End



- Timing Chart for Error End



Additional Information

- For this FB, use the socket service function. Refer to *NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual (W506)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP™ Port User's Manual (W563)* for details of the socket service function.

Precautions for Correct Use

- Execution of this instruction is continued until processing is completed even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is completed. Use this to confirm normal completion of processing.
- This FB executes one SktUDPRcv instruction and one SktUDPSend instruction internally. For the number of simultaneous executions, refer to the SktTCPRcv and SktTCPSend in *NJ/NX-series Instructions Reference Manual (W502)* or *NY-series Instructions Reference Manual (W560)*.
- When the same socket is shared by FBs of multiple SLMP communications instructions, multiple SLMP communications instructions cannot be executed simultaneously. SLMP communications instructions cannot distinguish responses to commands sent by themselves from responses to commands sent by other SLMP communications instructions. Therefore, wait until the execution of one SLMP communications instruction is complete and then execute the next SLMP communications instruction.
- You should specify the values mean destination to *Socket.DstAdr.PortNo* and *Socket.DstAdr.IpAdr*.

Troubleshooting

Error code	Expansion error code	Error name	Description	Corrective action
16#0000	16#00000000	Normal end	—	—

Error code	Expansion error code	Error name	Description	Corrective action
16#0400	16#00000000	Input Value Out of Range	An input parameter for an instruction exceeded the valid range for an input variable. Or, division by an integer of 0 occurred in division or remainder calculations.	Refer to the description of <i>Machine Automation Controller Troubleshooting Manual (W503)</i> for the event code with 5401 appended to the upper 4 digits of the Error code. For example, if the Error code is 16#0400, refer to the description of event code 54010400 hex.
16#0406	16#00000000	Illegal Data Position Specified	A memory address or data size that was specified for the instruction is not suitable.	
16#0407	16#00000000	Data Range Exceeded	The results of instruction processing exceeded the data area range of the output parameter.	
16#0419	16#00000000	Incorrect Data Type	A data type that cannot be used for an instruction is specified for an input or in-out variable.	
16#2002	16#00000000	Address Resolution Failed	Address resolution failed for a remote node with the host name that was specified in the instruction.	
16#2003	16#00000000	Socket Status Error	The status was not suitable for execution of the socket service instruction.	
16#2006	16#00000000	Socket Timeout	A timeout occurred for a socket service instruction.	
16#2007	16#00000000	Socket Handle Out of Range	The handle that is specified for the socket service instruction is not correct.	
16#2008	16#00000000	Socket Communications Resource Overflow	The maximum resources that you can use for socket service instructions at the same time was exceeded.	
16#3D2A	16#00000004	Too short ReqDat	One of the following. <ul style="list-style-type: none"> The number of elements in the array assigned to the in-out variable <i>ReqDat</i> is less than the requested data length specified by the in-out variable <i>ReqHeader.SubHeader.DI</i>. The in-out variable <i>RespHeader.SubHeader.DI</i> is set to a value out of specification of this FB. 	

Error code	Expansion error code	Error name	Description	Corrective action
16#3D2A	16#00000005	Request message length limit over	The length of the request statement, construct constructed according to the value of the in-out variable <i>ReqHeader.SubHeader.Dl</i> exceeds the maximum limit that can be produce with this FB, function block. *1	Confirm the in-out variable <i>ReqHeader.SubHeader.Dl</i> in this FB.
16#3D2A	16#00000010	Response reception timeout	A response message could not be received within the time specified in input variable <i>TimeOut</i> .	Check the network status and communications target Mitsubishi Electric Sequencer status. Or, increase the value of the input variable <i>TimeOut</i> of this FB.
16#3D2A	16#FFFF0001 - 16#FFFFFFFF	SLMP end code	An error response message was received.	Refer to <i>End codes</i> in the SLMP Specifications Overview for troubleshooting. For information on how to obtain the SLMP Specifications Overview, contact the CC-Link Partner Association.

*1. Refer to the *Function* on page 4-77 for details.

Sample program

From our website (http://www.ia.omron.com/sysmac_library/), you can download sample programs using SLMP_SendCmd_UDP. You can use with MELSEC iQ-R Series, iQ-F Series, Q Series, and L Series.



Precautions for Correct Use

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

The overview of the sample programming is as follows:

Conditions

Item	Description
Process	Read non-consecutive Devices D0, M100, Y160 with "Random reading" command
Destination IP Address	192.168.250.39

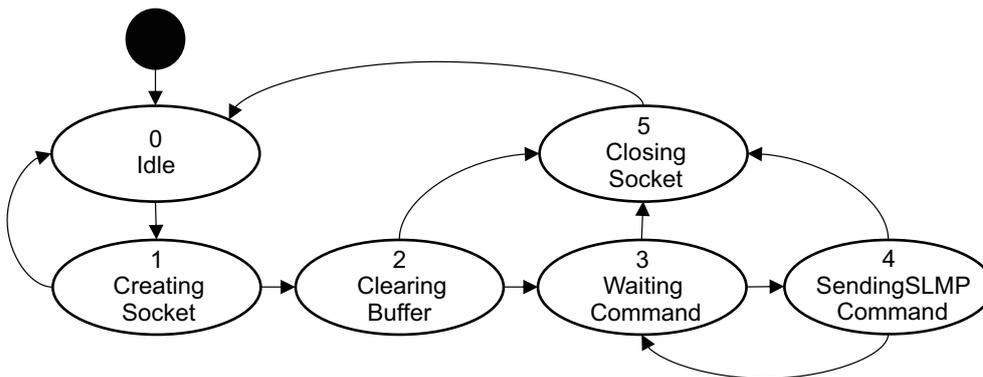
Item	Description
Destination UDP port number	5100
Source UDP port number	60000
Communication code	BINARY
Timeout time	2 seconds (default)

Process

- 1** Execute the SktUDPCreate instruction, and create a UDP socket. Then, set the values that specify the destination in *DstAdr.PortNo* and *DstAdr.IpAdr* of the created UDP socket.
- 2** Execute the SktClearBuf instruction, and clear the receive buffer of TCP socket.
- 3** Confirm that no other communications FB execution is in progress.
- 4** Launch the SLMP_SendCmd_UDP FB-instance.
- 5** After completing FB execution, check the response data.
- 6** Execute the SktClose instruction, and close the UDP socket.

State transition of sample program

The state transition of the sample programming is as follows:



No	State name	Processing
0	Idle	This is the state waiting for the request of creating UDP socket. The UDP socket has not been created and is in a waiting state.
1	CreatingSocket	This is the state to create a UDP socket.
2	ClearingBuffer	This is the state to clear the receive buffer of the UDP socket.
3	WatingCommand	This is the state waiting for a request to send a command for SLMP communication or a request to close the UDP socket. The UDP socket has been created and is in a waiting state.
4	SendingSLMPCommand	This is the state sends SLMP communication commands and receives responses to the destination station.

No	State name	Processing
5	ClosingSocket	This is the state to close the UDP socket.



Appendix

This section describes information that is convenient to know, such as library information reference methods, FB or FUN source code reference methods, etc.

A-1	Referring to Library Information	A-2
A-1-1	Library Attributes, and FB or FUN Attributes	A-2
A-1-2	Referring to Attributes of Libraries, Function Blocks, and Functions.....	A-3



A-1 Referring to Library Information

When you make an inquiry to OMRON about a library, you can refer to the library information to identify the library to ask about.

The library information is useful in identifying the target library among the libraries provided by OMRON or created by the user.

The library information consists of the attributes of the library and the attributes of function blocks and functions contained in the library.

- Attributes of libraries
Information for identifying the library itself
- Attributes of function blocks and functions
Information for identifying the function block and function contained in the library

Use the Sysmac Studio to access the library information.

A-1-1 Library Attributes, and FB or FUN Attributes

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

Library Attributes

No.*1	Attribute	Description
(1)	Library file name	The name of the library file
(2)	Library version	The version of the library
(3)	Author	The name of the creator of the library
(4)	Comment	The description of the library*2

*1. These numbers correspond to the numbers shown on the screen images in the next section, *A-1-2 Referring to Attributes of Libraries, Function Blocks, and Functions* on page A-3.

*2. It is provided in English and Japanese.

Attributes of Function Blocks and Functions

No.*1	Attribute	Description
(5)	FB/FUN name	The name of the function block or function
(6)	Name space	The name of the name space for the function block or function
(7)	FB/FUN version	The version of the function block or function
(8)	Author	The name of the creator of the function block or function
(9)	FB/FUN number	The function block number or function number
(10)	Comment	The description of the function block or function *2

*1. These numbers correspond to the numbers shown on the screen images in the next section, *A-1-2 Referring to Attributes of Libraries, Function Blocks, and Functions* on page A-3.

*2. It is provided in English and Japanese.

A-1-2 Referring to Attributes of Libraries, Function Blocks, and Functions

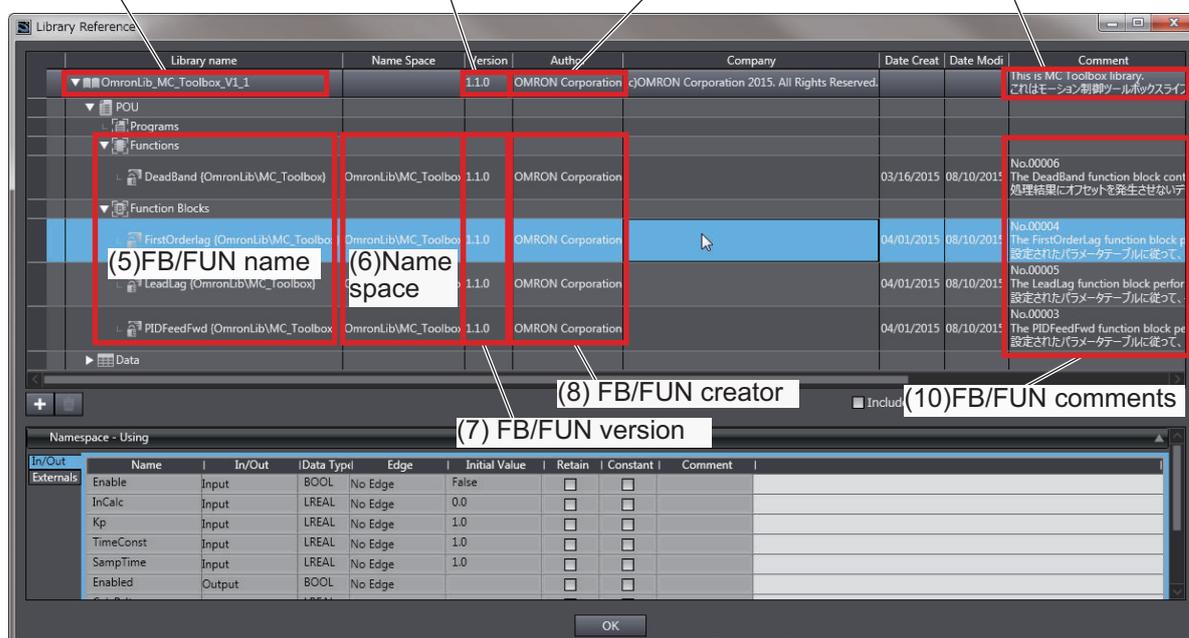
You can refer to the library attributes of library information, and FB or FUN attributes at the following Sysmac Studio locations.

- Library Reference Dialog Box
- Toolbox
- Programming screen

Library Reference Dialog Box

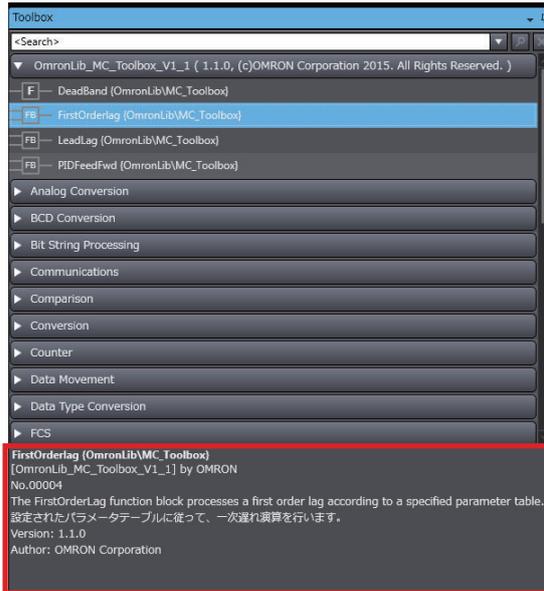
When you refer to the libraries, the library information is displayed at the locations shown below.

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



Toolbox

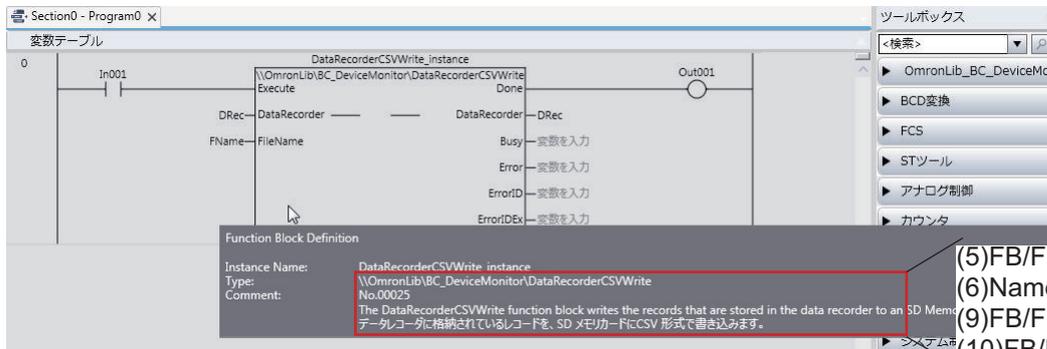
Select a function block or function to display its library information at the bottom of the Toolbox Pane. The text "by OMRON" which is shown on the right of the library name (1) indicates that this library was provided by OMRON.



- (5)FB/FUN name (6)Name space
- (1)Library file name
- (9)FB/FUN number
- (10)FB/FUN comment
- (7)FB/FUN version
- (8)FB/FUN author

Programming Screen

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



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



Index



Index

S

SLMP_DeviceRead.....	4-2
SLMP_DeviceRead_UDP.....	4-43
SLMP_DeviceWriteBool.....	4-21
SLMP_DeviceWriteBool_UDP.....	4-62
SLMP_DeviceWriteWord.....	4-12
SLMP_DeviceWriteWord_UDP.....	4-53
SLMP_SendCmd.....	4-30
SLMP_SendCmd_UDP.....	4-71

OMRON Corporation Industrial Automation Company

Kyoto, JAPAN

Contact : www.ia.omron.com

Regional Headquarters

OMRON EUROPE B.V.

Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands

Tel: (31) 2356-81-300 Fax: (31) 2356-81-388

OMRON ELECTRONICS LLC

2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.

Tel: (1) 847-843-7900 Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.

438B Alexandra Road, #08-01/02 Alexandra
Technopark, Singapore 119968

Tel: (65) 6835-3011 Fax: (65) 6835-3011

OMRON (CHINA) CO., LTD.

Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,

PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-6023-0333 Fax: (86) 21-5037-2388

Authorized Distributor:

©OMRON Corporation 2018-2024 All Rights Reserved.
In the interest of product improvement,
specifications are subject to change without notice.

Cat. No. W597-E1-03 0424