

Machine Automation Controller NJ-series

General-purpose Serial Connection Guide (RS-232C) OMRON Corporation

Ultra Small Multi-code Reader
(V400-R2 Series)

Network
Connection
Guide

About Intellectual Property Rights and Trademarks

Microsoft product screen shots reprinted with permission from Microsoft Corporation.

Windows is a registered trademark of Microsoft Corporation in the USA and other countries.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

Sysmac is a trademark or registered trademark of OMRON Corporation in Japan and other countries for OMRON factory automation products.

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

Table of Contents

1. Related Manuals	1
2. Terms and Definitions	2
3. Remarks	3
4. Overview	5
5. Applicable Devices and Device Configuration	6
5.1. Applicable Devices.....	6
5.2. Device Configuration.....	7
6. Serial Communications Settings	9
6.1. Serial Communications Settings	9
6.2. Cable Wiring Diagram	10
6.3. Example of Connection Check.....	12
7. Connection Procedure	13
7.1. Work Flow	13
7.2. Setting UP the Code Reader.....	15
7.3. Setting Up the Controller.....	16
7.4. Checking the Serial Communications	34
8. Initialization Method	40
8.1. Initializing the Controller.....	40
8.2. Initializing the Code Reader	41
9. Program.....	42
9.1. Overview	42
9.2. Communications sequence.....	46
9.3. Error Detection Processing	47
9.4. Variables	48
9.5. ST Program.....	51
9.6. Protocol macro data	56
9.7. Timing Charts	63
9.8. Error processing.....	64
10. Revision History	66

1. Related Manuals

The table below lists the manuals related to this document.

To ensure system safety, make sure to always read and heed the information provided in all Safety Precautions, Precautions for Safe Use, and Precaution for Correct Use of manuals for each device which is used in the system.

Cat. No.	Model	Manual name
W500	NJ501-□□□□ NJ301-□□□□	NJ-series CPU Unit Hardware User's Manual
W501	NJ501-□□□□ NJ301-□□□□	NJ-series CPU Unit Software User's Manual
W494	CJ1W-SCU□2	CJ-series Serial Communications Units Operation Manual for NJ-series CPU Unit
W504	SYSMAC-SE2□□□□	Sysmac Studio Version 1 Operation Manual
W344	CXONE-AL□□C-V4 /AL□□D-V4	CX-Protocol Operation Manual
W502	NJ501-□□□□ NJ301-□□□□	NJ-series Instructions Reference Manual
Z333	V400-R2 Series	Ultra Small Multi-code Reader V400-R2 Series User's Manual



2. Terms and Definitions

Term	Explanation and Definition
Protocol macro	A protocol macro is a function that stores a data send/receive procedure (protocols) in a Serial Communications Board or Serial Communications Unit to exchange data with general-purpose external devices by executing the ExecPMCR instruction on the CPU Unit.
Protocol	A unit of independent communication processing with a specific general-purpose device. A protocol includes a data send/receive procedure. A protocol consists of multiple sequences.
Sequence	A unit of the independent communication processing which can be started by executing the ExecPMCR instruction of a program. A sequence that is started will execute steps registered in its own sequence.
Step	A unit to execute any one of the followings: message send processing, message receive processing, message send/receive processing, clear receive buffer, or step wait. Up to 15 steps can be set per sequence.
Send message	A communication frame (command) sent to the external general-purpose device. A send message is read from the step in the sequence, and sent to the external general-purpose device.
Receive message	A communication frame (response) sent from the external general-purpose device. A receive message is read from the step in the sequence and is compared with data received from the general-purpose external device.
Matrix	A matrix is used when a general-purpose external device sends multiple types of communications frames (responses). More than one communication frame can be registered in one matrix.
Case	A unit to register multiple communication frames (response) to a matrix. One communication frame is registered as one case. Up to 15 types of cases can be registered per matrix.
AT (AT specification)	This attribute allows the user to specify an address assigned to a variable. An I/O port or an address in memory used for CJ-series Units can be specified.

3. Remarks

- (1) Understand the specifications of devices which are used in the system. Allow some margin for ratings and performance. Provide safety measures, such as installing safety circuit in order to ensure safety and minimize risks of abnormal occurrence.
- (2) To ensure system safety, always read and heed the information provided in all Safety Precautions, Precautions for Safe Use, and Precaution for Correct Use of manuals for each device used in the system.
- (3) The user is encouraged to confirm the standards and regulations that the system must conform to.
- (4) It is prohibited to copy, to reproduce, and to distribute a part or the whole of this document without the permission of OMRON Corporation.
- (5) The information contained in this document is current as of September 2013. It is subject to change without notice for improvement.

The following notation is used in this document.

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



Precautions for Safe Use

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



Precautions for Correct Use

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



Additional Information

Additional information to read as required.

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

Symbol



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.

4. Overview

This document describes the procedure for connecting OMRON Corporation's Ultra Small Multi-code Reader (V400-R2 series) (hereinafter referred to as the Code Reader) with NJ-series Machine Automation Controller (hereinafter referred to as the Controller) via serial communications, and describes the procedure for checking their connection.

Refer to the serial communications settings of the prepared Sysmac Studio project file and understand the setting method and key points to connect the devices via serial communications.

The user program in the Sysmac Studio project file and the protocol macro data in the CX-Protocol project file are used to check the serial connection by executing the "Reading Trigger" command on the Code Reader.

Prepare the latest Sysmac Studio project file and the CX-Protocol project file beforehand. To obtain the files, contact your OMRON representative.

Name	File name	Version
Sysmac Studio project file (extension: smc)	OMRON_V400-R2_PMCR232C_EV100.smc	Ver.1.00
CX-Protocol project file (extension: psw)	OMRON_V400-R2_PMCR_EV100.psw	Ver.1.00

*Hereinafter, the Sysmac Studio project file is referred to as the "project file"

The user program in the project file is referred to as the "program"

The CX-Protocol project file is called the "Protocol macro data"

Caution

This document aims to explain the wiring method and communications settings necessary to connect the corresponding devices and provide the setting procedure. The program used in this document is designed to check if the connection was properly established and is not designed to be constantly used at a site. Therefore, functionality and performances are not sufficiently taken into consideration. When you construct an actual system, please use the wiring method, communications settings and setting procedure described in this document as a reference and design a new program according to your application needs.



5. Applicable Devices and Device Configuration

5.1. Applicable Devices

The applicable devices are as follows:

Manufacturer	Name	Model
OMRON	NJ-series CPU Unit	NJ501-□□□□ NJ301-□□□□
OMRON	Serial Communications Unit	CJ1W-SCU□2
OMRON	Ultra Small Multi-code Reader	V400-R2CF□□□



Precautions for Correct Use

As applicable devices above, the devices with the models and versions listed in Section 5.2. are actually used in this document to describe the procedure for connecting devices and checking the connection.

You cannot use devices with versions lower than the versions listed in Section 5.2.

To use the above devices with versions not listed in Section 5.2 or versions higher than those listed in Section 5.2, check the differences in the specifications by referring to the manuals before operating the devices.

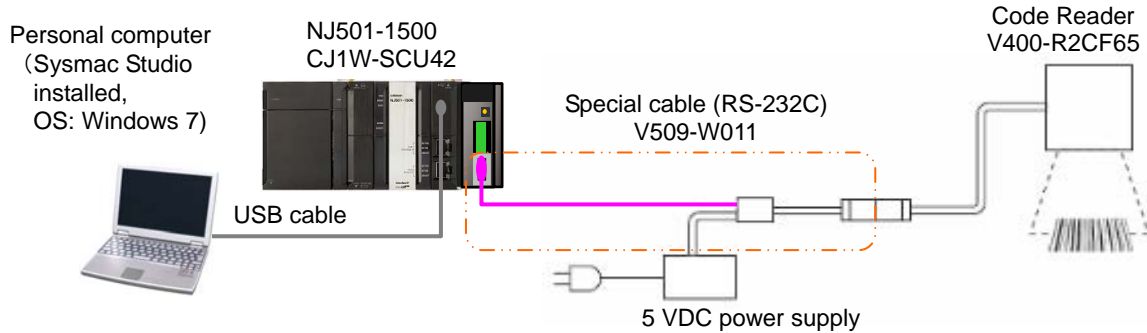


Additional Information

This document describes the procedure to establish the network connection. Except for the connection procedure, it does not provide information on operation, installation or wiring method. It also does not describe the functionality or operation of the devices. Refer to the manuals or contact your OMRON representative.

5.2. Device Configuration

The hardware components to reproduce the connection procedure of this document are as follows:



Manufacturer	Name	Model	Version
OMRON	Serial Communications Unit	CJ1W-SCU42	Ver.2.0
OMRON	NJ-series CPU Unit	NJ501-1500	Ver.1.05
OMRON	Power Supply Unit	NJ-PA3001	
OMRON	Sysmac Studio	SYSMAC-SE2[]	Ver.1.06
OMRON	CX-Protocol	(Included in Sysmac Studio.)	Ver.1.97
OMRON	Sysmac Studio project file	OMRON_V400-R2_PMC R232C_EV100.smc	Ver.1.00
OMRON	CX-Protocol project file	OMRON_V400-R2_PMC R_EV100.psw	Ver.1.00
-	Personal computer (OS: Windows7)	-	
-	USB cable (USB 2.0 type B connector)	-	
OMRON	Ultra Small Multi-code Reader	V400-R2CF65	
OMRON	Special cable (RS-232C)	V509-W011	
-	5 VDC power supply	-	



Precautions for Correct Use

Prepare the latest project file and protocol macro data in advance.
To obtain the files, contact your OMRON representative.



Precautions for Correct Use

Update the Sysmac Studio and CX-Protocol to the versions specified in this section or higher versions using the auto update function. If a version not specified in this section is used, the procedures described in Section 7 and subsequent sections may not be applicable. In that case, use the equivalent procedures described in the Sysmac Studio Version 1 Operation Manual (Cat. No. W504) and the CX-Protocol Operation Manual (Cat. No. W344).



Additional Information

It may not be possible to reproduce the same operation with different devices or versions. Check the configuration, model and version. If they are different from your configuration, contact your OMRON representative.



Additional Information

For information on the special cable (V509-W011), refer to *Cable for programmable controller connection made by OMRON (V509-W011)* in *Specifications and External Dimension* in *Section 5 Appendix* of the *Ultra Small Multi-code Reader V400-R2 Series User's Manual* (Cat. No. Z333).



Additional Information

The system configuration in this document uses USB for the connection to the Controller. For information on how to install a USB driver, refer to *A-1 Driver Installation for Direct USB Cable Connection* of the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504).

6. Serial Communications Settings

This section describes the specifications such as communication parameters and wiring that are defined in this document.



Additional Information

To perform communications without using the settings described in this section, you need to modify the program. For information on the program, refer to *Section 9. Program*.

6.1. Serial Communications Settings

The serial communications settings are shown below.

Setting item	Serial Communications Unit	Code Reader
Device name	J01	-
Unit number	0	-
Communications (connection) port	Port 2 (RS-232C)	-
Serial communications mode	Protocol macro	-
Data length	8 bits	8 bits (Default)
Stop bit	1 bit	1 bit (Default)
Parity	None	None (Default)
Transmission rate (Baud rate)	9,600 bps (Default)	9,600 bps (Default)
Protocol macro transmissions	Full-duplex	-
Header	-	None (Default)
Footer	-	<CR> (Default)
Number of digit output	-	None (Default)
RS/CS control	-	None (no protocol system) (Default)
CS waiting time	-	Not limited (Default)
ACK/NAK waiting time	-	Not limited (Default)



Precautions for Correct Use

This document describes the setting procedure of the CJ1W-SCU42 Serial Communications Unit with unit number 0, communications (connection) port 2, and device name J01. To connect devices under different conditions, change the variable names and setting values used in the program by referring to 9. Program.

6.2. Cable Wiring Diagram

For details on the cable wiring, refer to *Section 3 Installation and Wiring* of the *CJ-series Serial Communications Units Operation Manual for NJ-series CPU Unit* (Cat. No. W494).

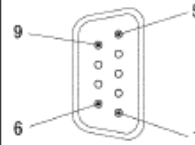
Check the connector configuration and pin assignment before wiring.

You do not need to make a cable when you use the special cable (V509-W011).

■ Connector configuration and pin assignment

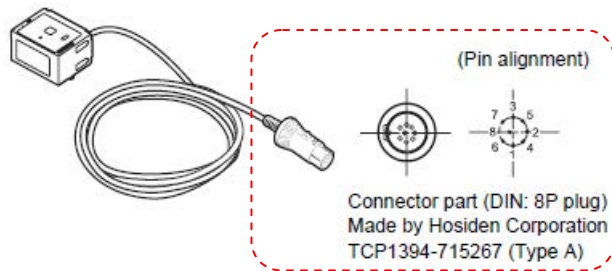
<CJ1W-SCU42> Applicable connector: D-sub 9 pin

Pin	Abbreviation	Signal name	I/O
1	FG	Shield	---
2	SD	Send data	Output
3	RD	Receive data	Input
4	RTS (RS)	Request to send	Output
5	CTS (CS)	Clear to send	Input
6	5V	Power supply	---
7	DSR (DR)	Data set ready	Input
8	DTR (ER)	Data terminal ready	Output
9	SG	Signal ground	---
Hood	FG	Shield	---

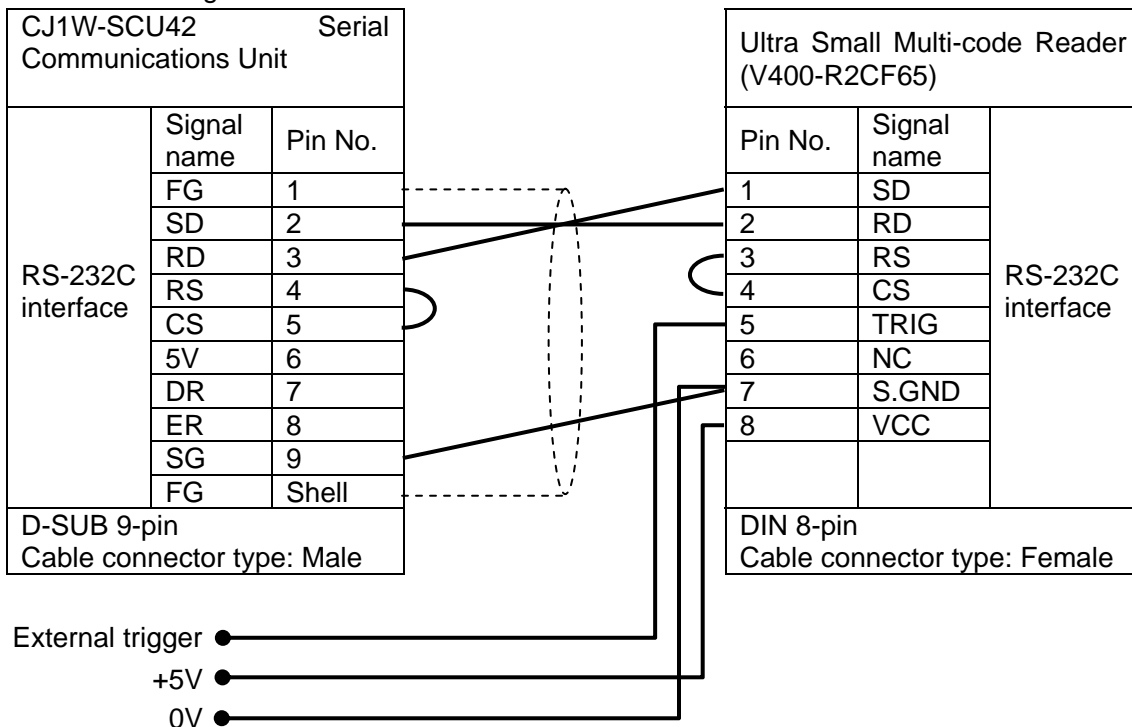


<V400-R2CF65> Applicable connector: DIN 8 pin

Pin No.	Signal name	Function
1	SD	Transmission data
2	RD	Received data
3	RS	Transmission request
4	CS	Transmission allowed
5	TRIG	External trigger signal
6	NC	Not connected
7	S.GND	0 V
8	VCC	Power supply



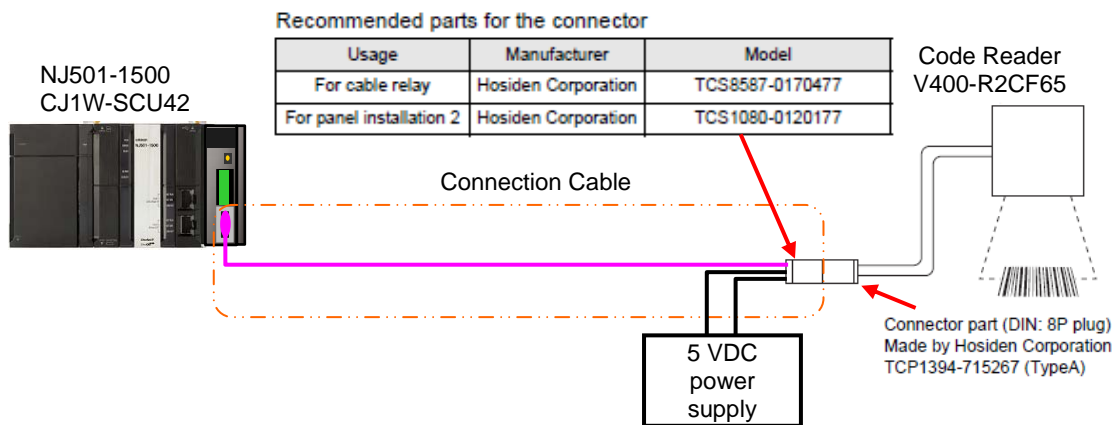
■ Cable/Pin assignment



*The external trigger is not used in this document. Insulate it from other cables.
*Connect +5V and 0V to the external power supply.

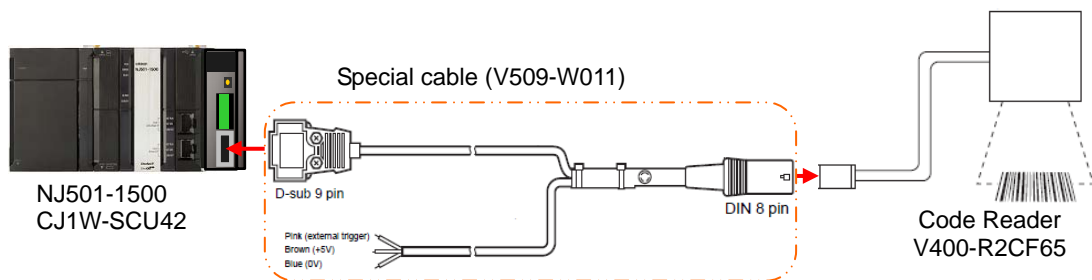
■ Recommended parts for the connector

When you make a connection cable, use the following connectors to connect it to the connector of the Code Reader.



■ Special cable

The special cable (V509-W011) is used in this document.



*The external trigger is not used in this document. Insulate it from other cables.
*Connect +5V and 0V to the external power supply.

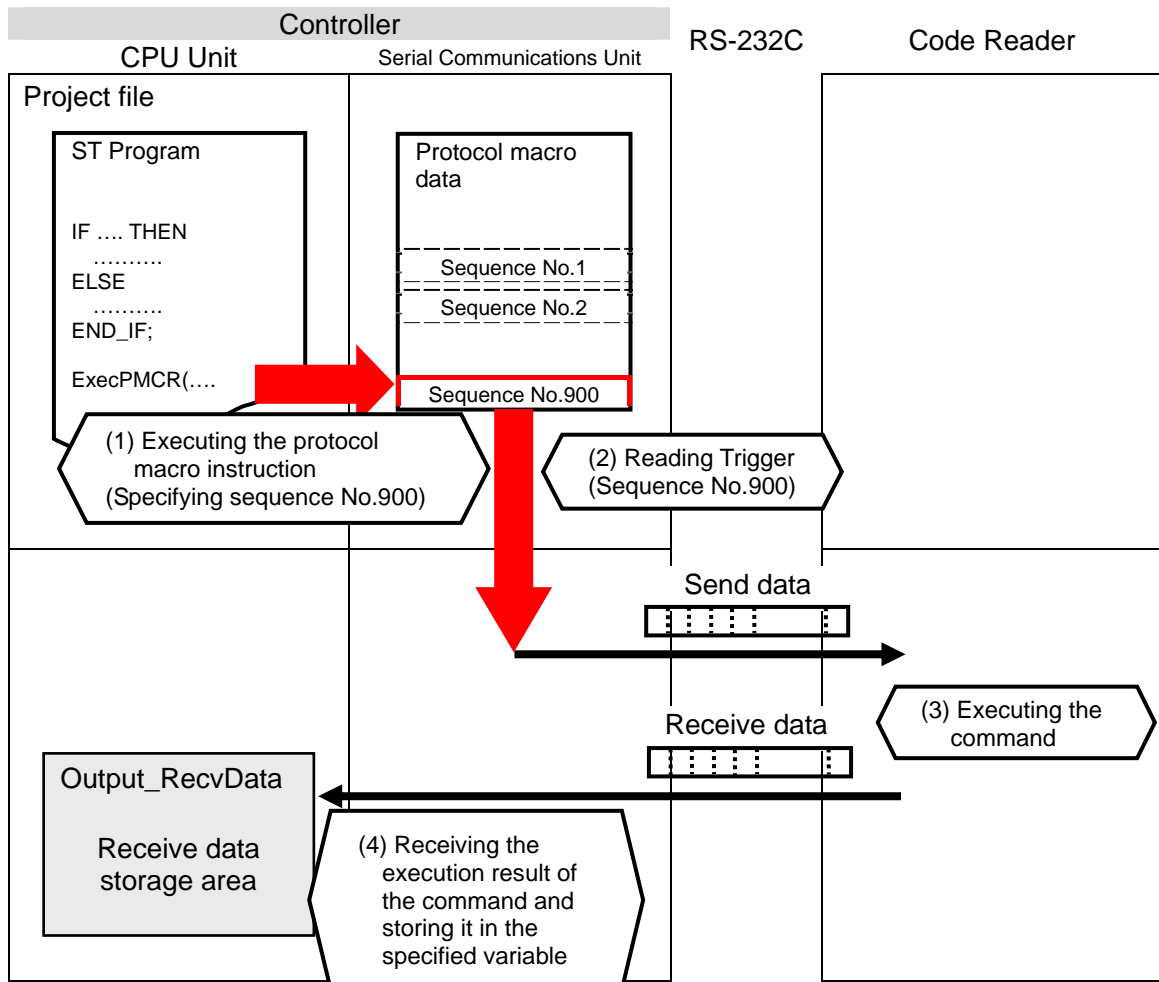
6.3. Example of Connection Check

This document shows an example of an ST (structured text) program and protocol macro data in which the Controller sends/receives the message to/from the Code Reader.

The Controller and the Code Reader send and receive the message of "Reading Trigger (sequence No. 900)". The following figure outlines the sequence operation.

- Overview of sequence operation

- (1) The ST program executes the protocol macro instruction (ExecPMCR instruction) for which sequence No. 900 of protocol macro data is specified
- (2) The Reading Trigger command (sequence No. 900) is selected from the protocol macro data, and it is sent to the Code Reader.
- (3) The Code Reader executes the command according to the send data from the Controller.
- (4) The Controller receives the result of the command executed by the Code Reader, and then stores in the specified variable.

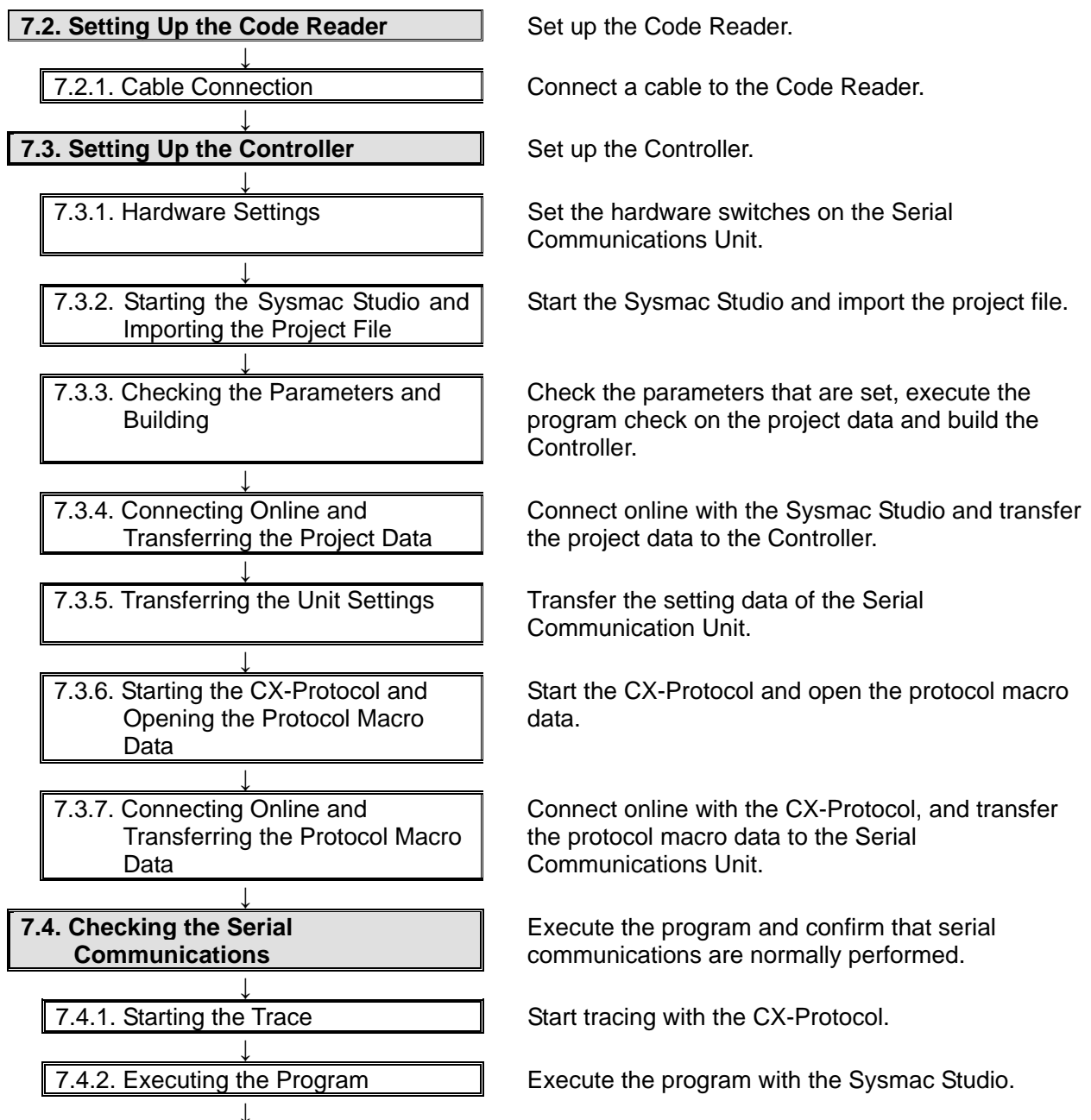


7. Connection Procedure

This section describes the procedure for connecting the Code Reader to the Controller via serial communications. This document explains the procedures for setting up the Controller and the Code Reader from the factory default setting. For the initialization, refer to *Section 8 Initialization Method*.

7.1. Work Flow

Take the following steps to connect the Code Reader to the Controller via serial communications.



7.4.3. Checking the Trace Data



7.4.4. Checking the Receive Data

Confirm that the correct data is sent and received by checking the trace data of the CX-Protocol.

Confirm that the correct data are written to the variables of the Controller with the Sysmac Studio.

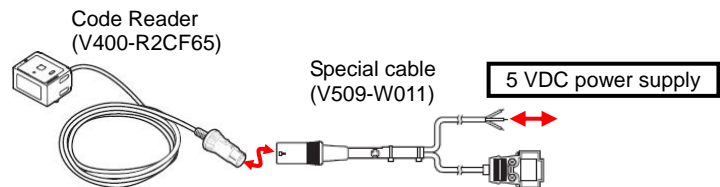
7.2. Setting UP the Code Reader

Set up the Code Reader.

7.2.1. Cable Connection

Connect a cable to the Code Reader.

- 1 Connect the special cable (V509-W011) to the Code Reader.
Connect the 5 VDC power supply to the special cable and turn ON the power supply.



Precautions for Correct Use

This document explains the procedures for setting up the Code Reader from the factory default setting. If the Code Reader was changed from the factory default setting, make sure to initialize by referring to Section 8.2. Initializing the Code Reader.



Additional Information

For details on the factory default setting of the Code Reader, refer to *Factory default setting in Return to the factory default setting in Menu Sheet/Command List in Section 4 Setting Method of the Ultra Small Multi-code Reader V400-R2 Series User's Manual (Cat. No. Z333).*

7.3. Setting Up the Controller

Set up the Controller.

7.3.1. Hardware Settings

Set the hardware switches on the Serial Communications Unit.



Precautions for Correct Use

Make sure that the power supply is OFF when you perform the setting up.

- 1 Make sure that the power supply to the PLC is OFF.

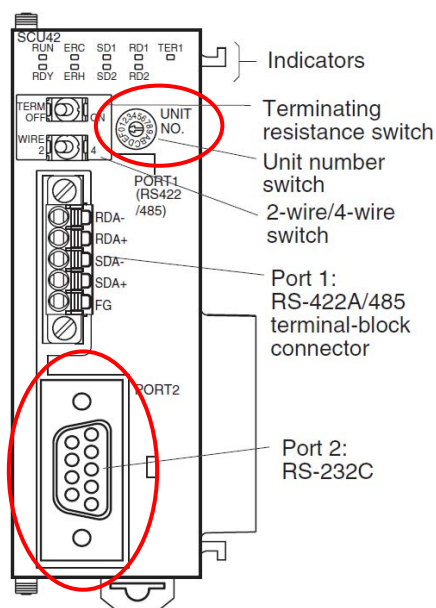
*If the power supply is turned ON, settings may not be applicable as described in the following procedure.

Check the hardware switches located on the front panel of the Serial Communications Unit by referring to the right figure

Check the connector for Port 2 (RS-232C) by referring to the right figure.

*This setting is required to use Port 2 of the Serial Communications Unit.

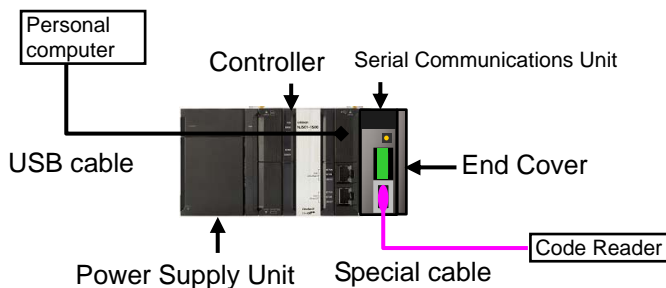
CJ1W-SCU42



- 2 Set the Unit No. Switch to 0. (The unit number is set to 0 as the factory default setting.)



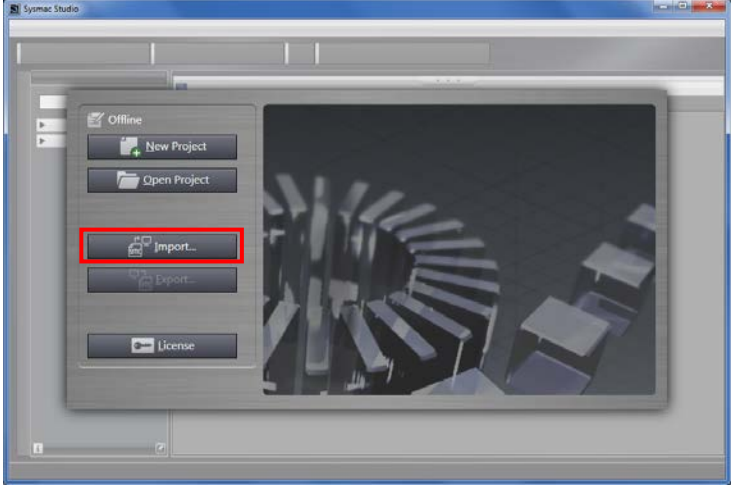
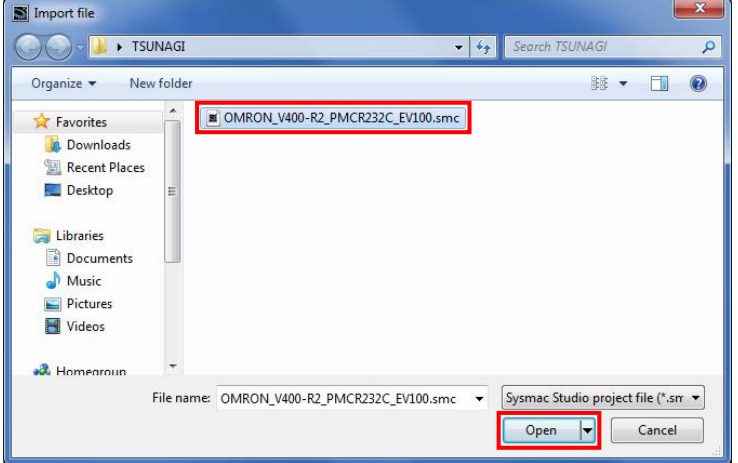
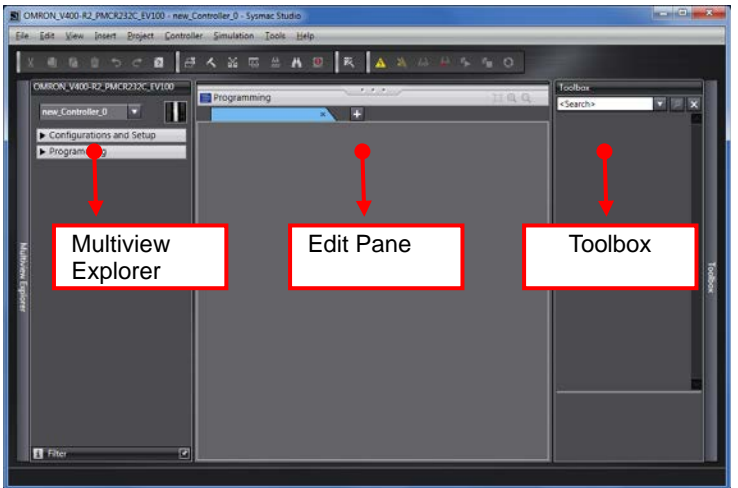
- 3 Connect the Serial Communications Unit to the Controller as shown on the right. Connect the special cable for the Code Reader to the Serial Communications Unit. Connect the personal computer to the Controller with a USB cable.



7.3.2. Starting the Sysmac Studio and Importing the Project File

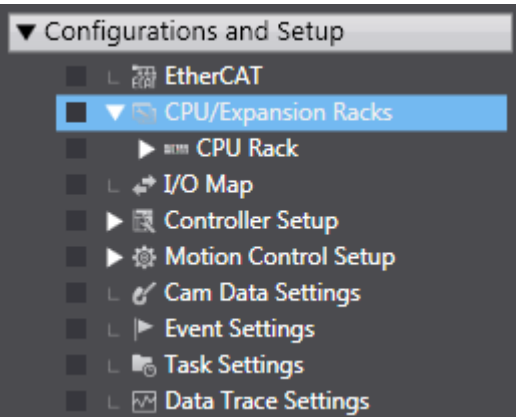
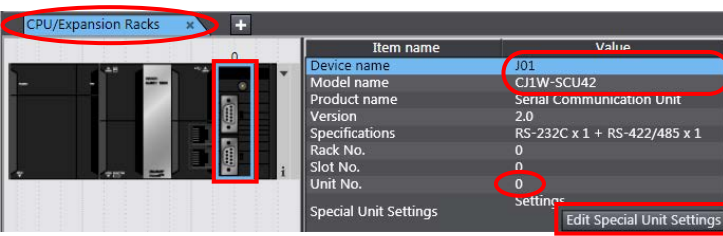
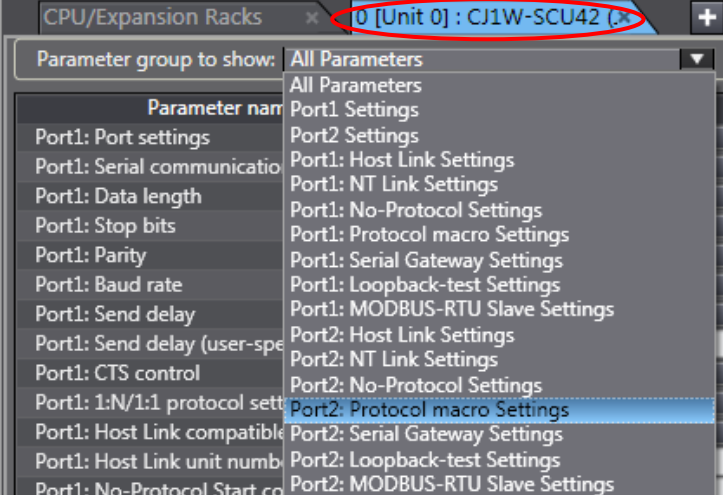
Start the Sysmac Studio and import the project file.

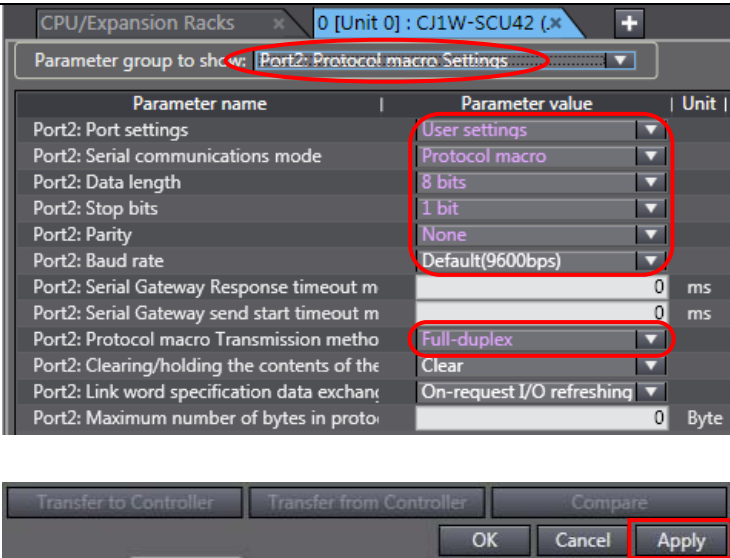
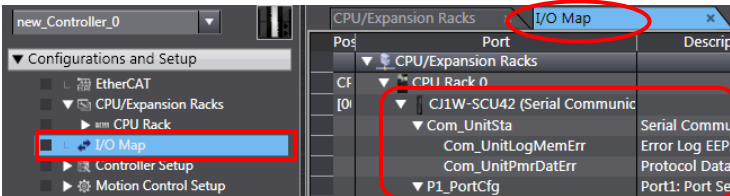
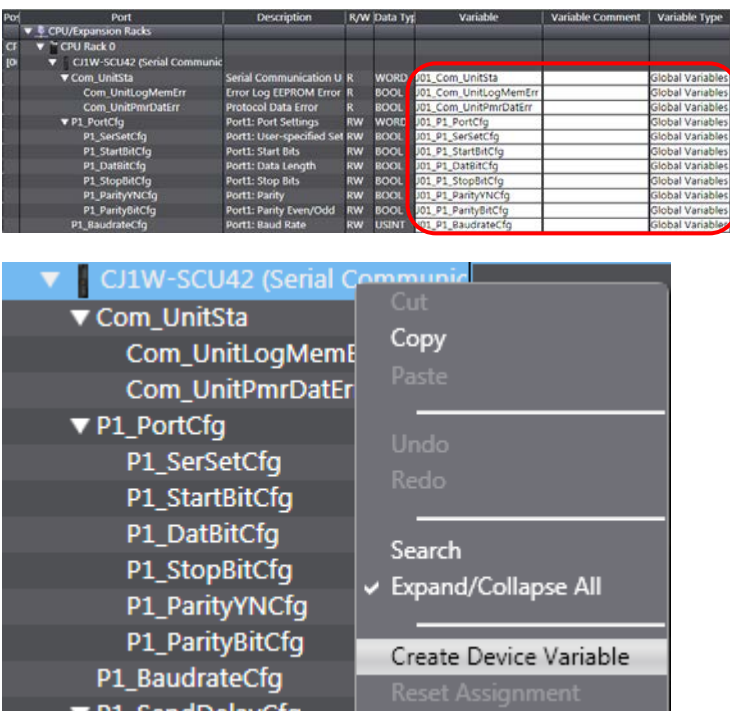
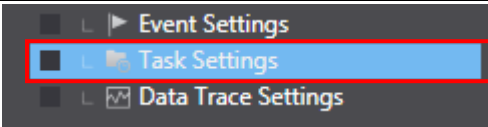
Install the Sysmac Studio and USB driver in the personal computer beforehand.

<p>1 Confirm that the personal computer and the Controller are connected with the USB cable and turn ON the power supply to the Controller.</p> <p>Start the Sysmac Studio. Click the Import Button.</p> <p>*If a confirmation dialog for an access right is displayed at start, select to start.</p>	
<p>2 The Import File Dialog Box is displayed. Select OMRON_V400-R2_PMCR232C_EV100.smc and click the Open Button.</p> <p>*Obtain the project file from OMRON.</p>	
<p>3 The OMRON_V400-R2_PMCR232C_EV100 project is displayed. The left pane is called Multiview Explorer, the right pane is called Toolbox and the middle pane is called Edit Pane.</p> <p>*If an error message is displayed stating "Failed to Load Descendants", change the version of the Sysmac Studio to the version specified in 5.2. Device Configuration or higher version.</p>	

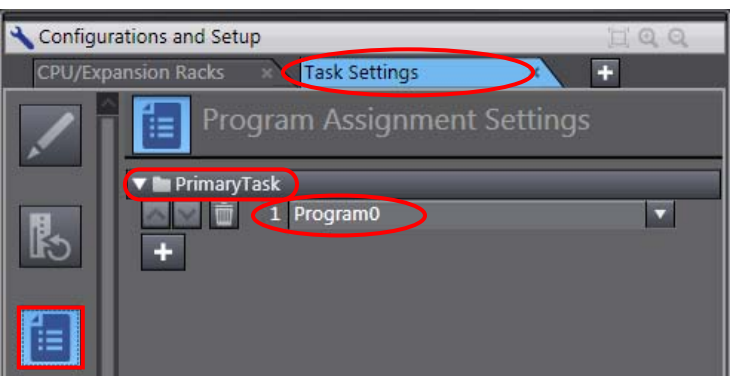
7.3.3. Checking the Parameters and Building

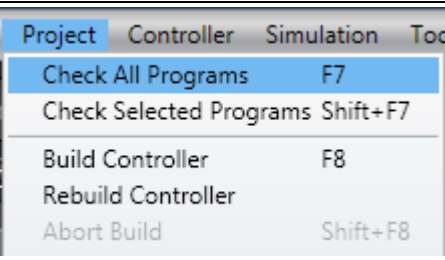
Check the parameters that are set, execute the program check on the project data and build the Controller.

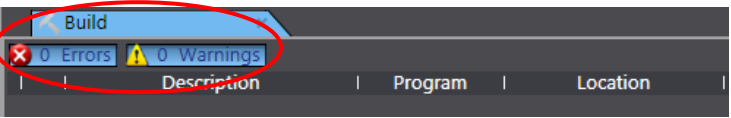
<p>1</p>	<p>Double-click CPU/Expansion Racks under Configurations and Setup in the Multiview Explorer.</p>	
<p>2</p>	<p>The CPU/Expansion Racks Tab is displayed in the Edit Pane. Select the Serial Communications Unit icon as shown on the right. Confirm that CJ1W-SCU42 is displayed, the Device name is J01, and the Unit No. is 0.</p> <p>*If the settings are different, change the values.</p> <p>Click Edit Special Unit Settings.</p>	
<p>3</p>	<p>The 0 [Unit 0]: Tab is displayed. Select <i>Port2: Protocol macro Settings</i> from the pull-down list of Parameter group to show.</p>	

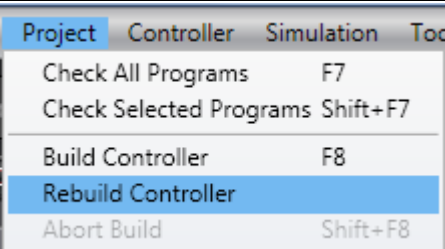
- 4 Parameter group to show is set to Port2: Protocol macro Settings. The setting items of the Port 2: Protocol macro Settings are shown. Confirm that the Port2: Port Settings is set to User settings and other settings are the same as those listed in Section 6.1.
- *If the settings are different from the above, change the values from the pull-down lists. After changing the values, click the **Apply** Button.
- 
- 5 Double-click **I/O Map** under **Configurations and Setup** on the Multiview Explorer. The I/O Map Tab is displayed and then the parameters for the Unit are listed.
- 
- 6 Confirm that data in the Variable Columns start with J01 and the Global Variable is set in each Variable Type Column.
- *If the settings are different from the above, right-click on **CJ1W-SCU42** and select **Create Device Variable**.
- 
- 7 Double-click **Task Settings** under **Configurations and Setup** in the Multiview Explorer.
- 

- 8 The Task Settings Tab is displayed in the Edit Pane. Click the **Program Assignment Settings** Button and confirm that Program0 is set under PrimaryTask.

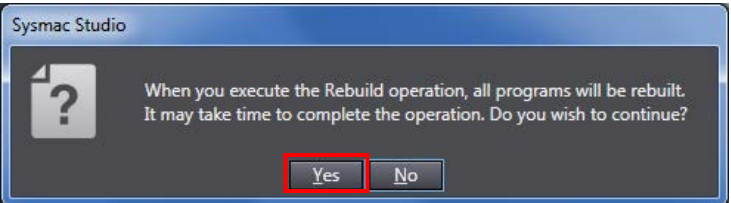

- 9 Select **Check All Programs** from the Project Menu.


- 10 Confirm that "0 Errors" and "0 Warnings" are displayed in the Build Tab Page under the Edit Pane.



- 11 Select **Rebuild Controller** from the Project Menu.

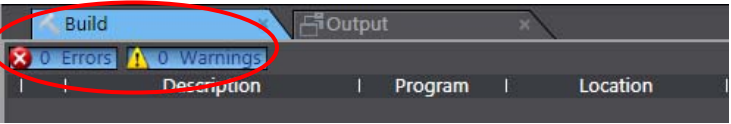


A confirmation dialog box is displayed. Check the contents and click the **Yes** Button.



A screen is displayed indicating the conversion is being performed.


- 12 Confirm that "0 Errors" and "0 Warnings" are displayed in the Build Tab Page.



7.3.4. Connecting Online and Transferring the Project Data

Connect online with the Sysmac Studio and transfer the project data to the Controller.

WARNING

Always confirm safety at the destination node before you transfer a user program, configuration data, setup data, device variables, or values in memory used for CJ-series Units from the Sysmac Studio.

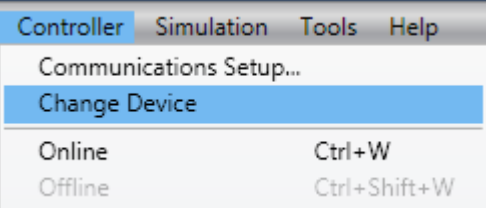
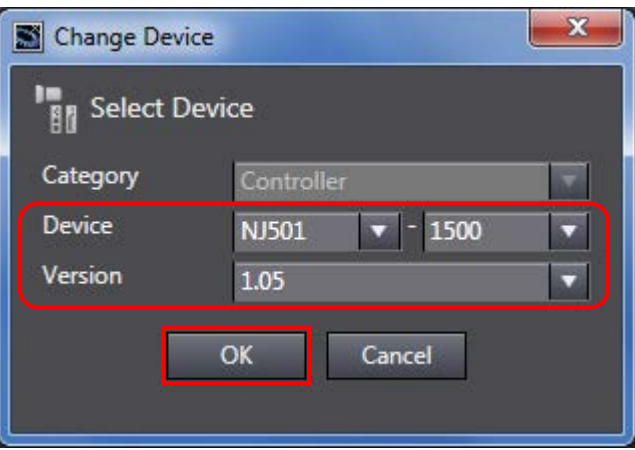
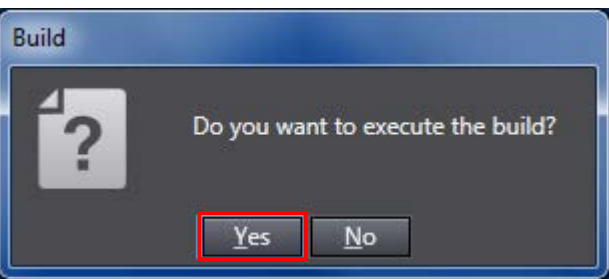
The devices or machines may perform unexpected operation regardless of the operating mode of the CPU Unit.



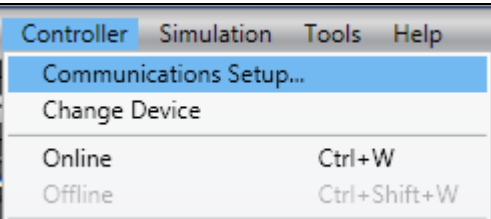
Caution

Always confirm safety before you reset the Controller or any components.

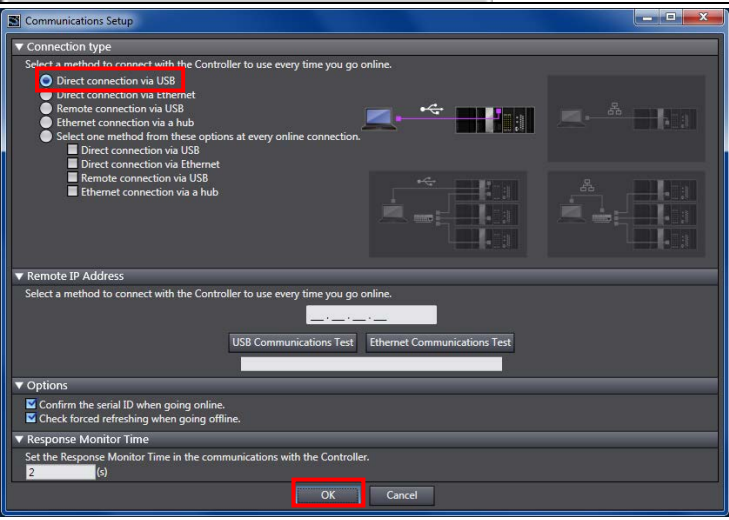


<p>1 Select Change Device from the Controller Menu.</p>	
<p>2 The Change Device Dialog Box is displayed. Confirm that the Device and Version are set as shown on the right and click the OK Button.</p> <p>*If the settings are not correct, select the setting items from the pull-down list.</p>	
<p>3 If the settings were changed in step 2, the Build Dialog Box is displayed. Check the contents and click the Yes Button.</p> <p>*This dialog box is not displayed if no change was made.</p>	

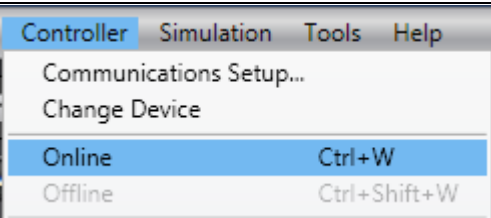
4 Select **Communications Setup** from the Controller Menu.



5 The Communications Setup Dialog Box is displayed. Select the *Direct connection via USB* Option for Connection Type. Click the **OK** Button.

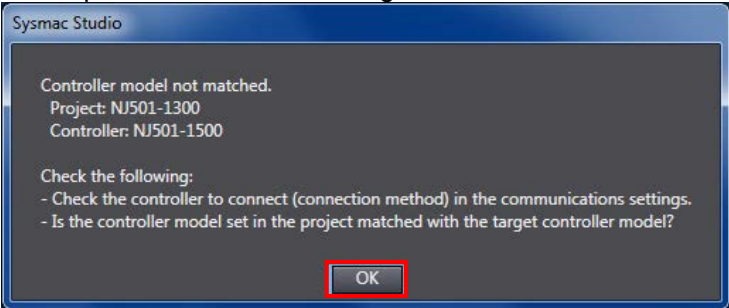


6 Select **Online** from the Controller Menu.

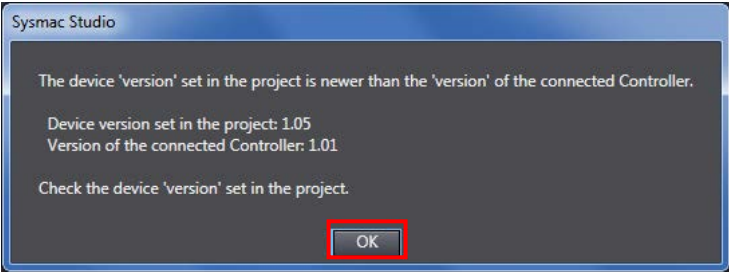


*If the dialog on the right is displayed, the model or version of the Controller does not match that of the project file. Review the Controller model, version and device settings of the project file, and then repeat the procedure from step 1 in this section. Close the dialog box by clicking the **OK** Button.

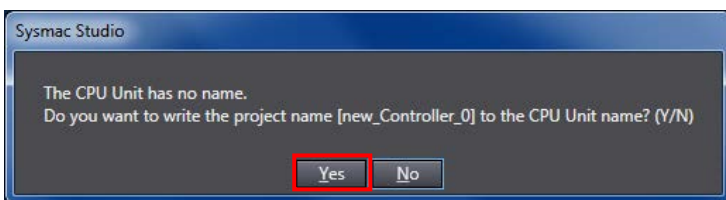
*Example of confirmation dialog box



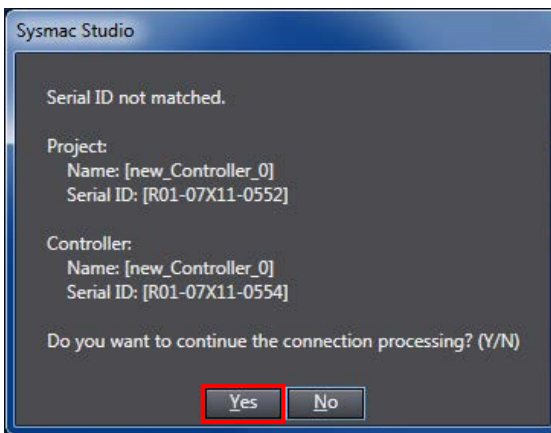
*The model and version displayed on the confirmation dialog box differ depending on the Controller used and the device setting of the project file.



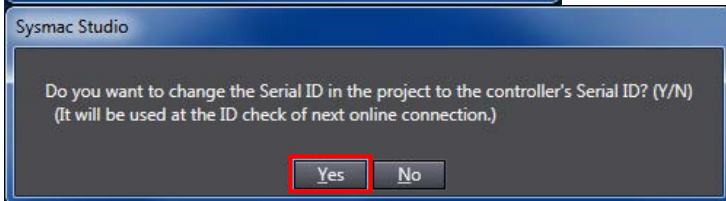
7 A confirmation dialog box is displayed as shown on the right. Check the contents and click the **Yes** Button.



*The displayed dialog depends on the status of the Controller used. Check the contents and click the **Yes** Button to proceed with the processing.



*The displayed serial ID differs depending on the device.



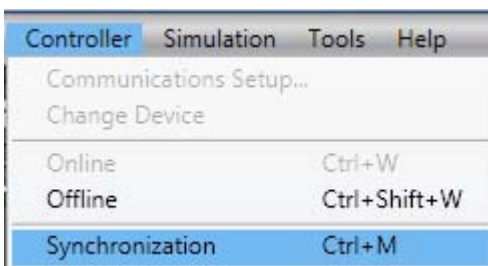
Additional Information

For details on online connections to a Controller, refer to *Section 5 Online Connections to a Controller* of the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504).

8 When an online connection is established, a yellow bar is displayed on the top of the Edit Pane.

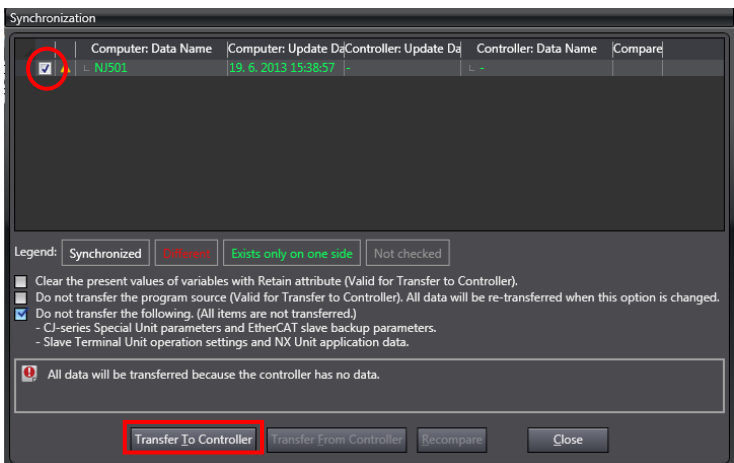


9 Select **Synchronization** from the Controller Menu.



10 The Synchronization Dialog Box is displayed. Confirm that the data to transfer (NJ501 in the right dialog) is selected. Then, click the **Transfer To Controller** Button.

*After executing the Transfer To Controller, the Sysmac Studio project data is transferred to the Controller and the data are compared.

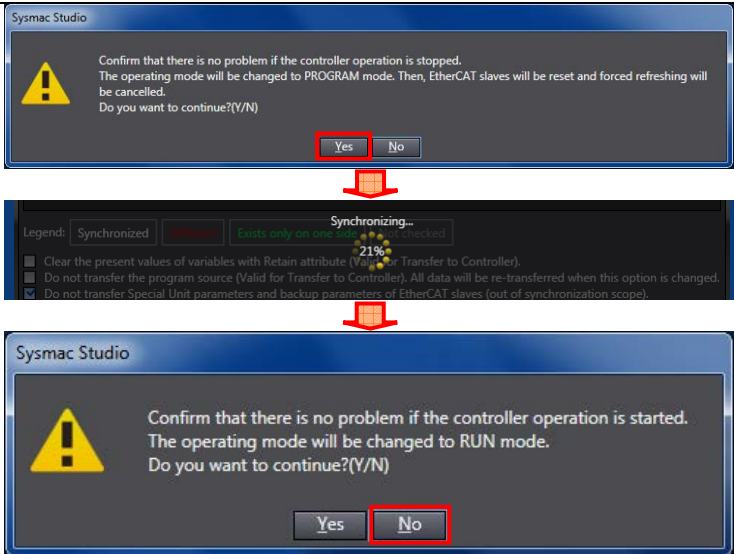


11 A confirmation dialog box is displayed. Confirm that there is no problem and click the **Yes** Button.

A screen stating "Synchronizing" is displayed.

A confirmation dialog box is displayed. Confirm that there is no problem and click the **No** Button.

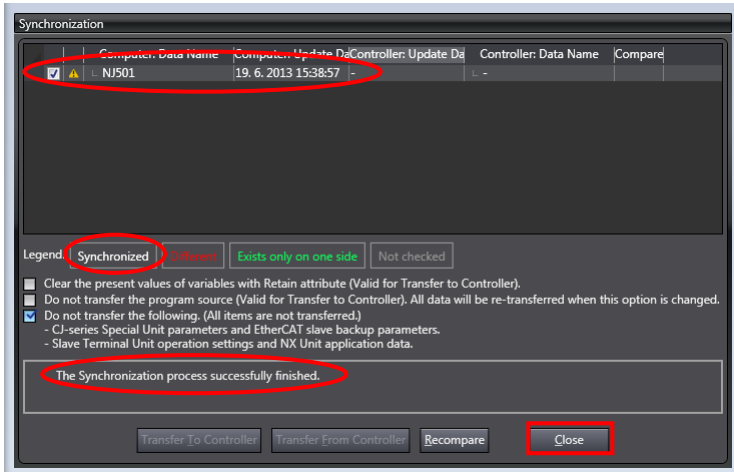
*Click the **No** Button to operate in PROGRAM mode in the following steps.



12 Confirm that the synchronized data is displayed with the color specified by "Synchronized", and that a message is displayed stating "The synchronization process successfully finished". If there is no problem, click the **Close** Button.

*A message stating "The synchronization process successfully finished" is displayed if the Sysmac Studio project data if the data in the Controller match.

*If the synchronization fails, check the wiring and repeat the procedure in this section.

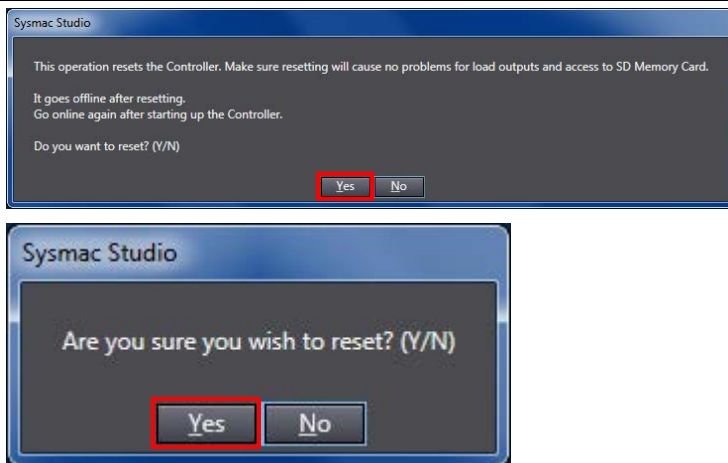


13 Select **Reset Controller** from the Controller Menu.

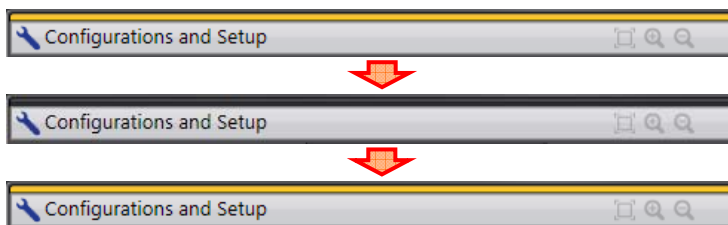
*When Mode is set to RUN Mode, Reset Controller cannot be selected. In this case, select **Mode - PROGRAM Mode** from the Controller Menu to change to PROGRAM mode and perform the procedure in this step.



14 A confirmation dialog box is displayed several times. Check the contents and click the **Yes** Button.


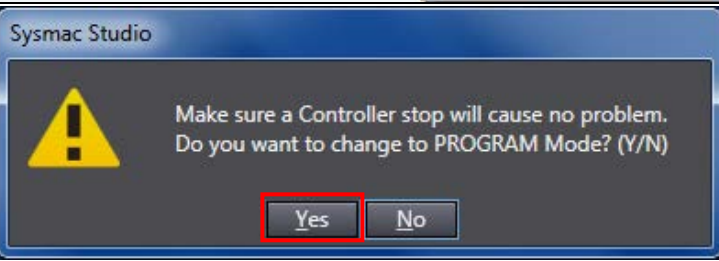
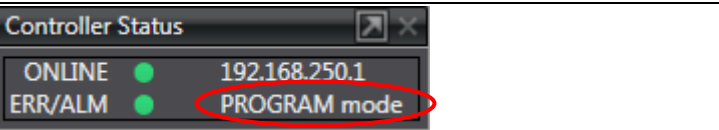
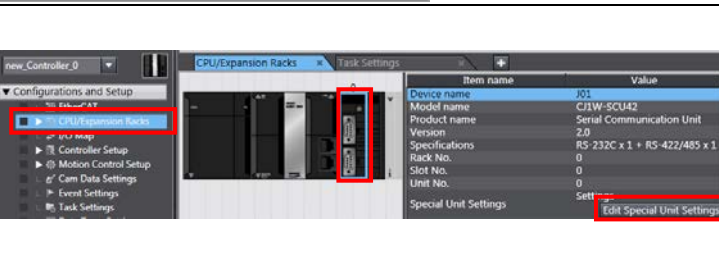
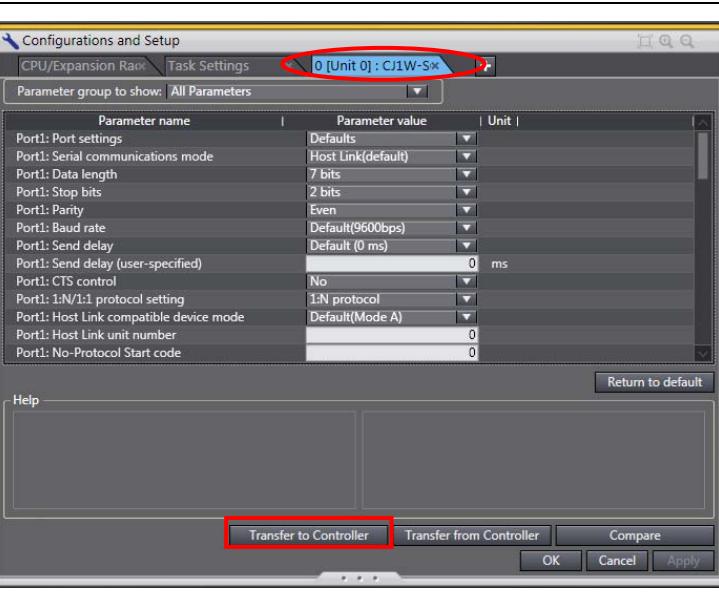


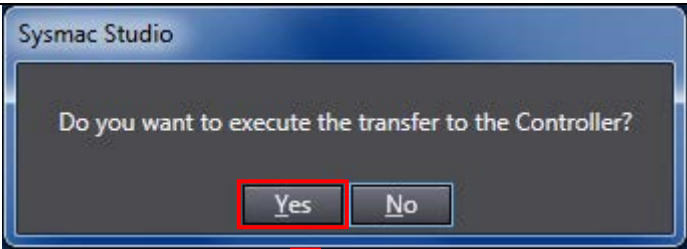
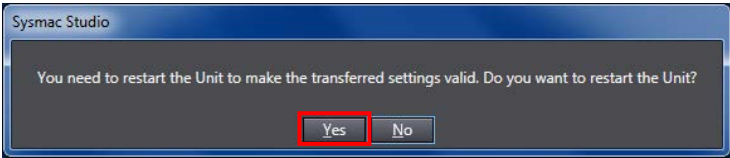
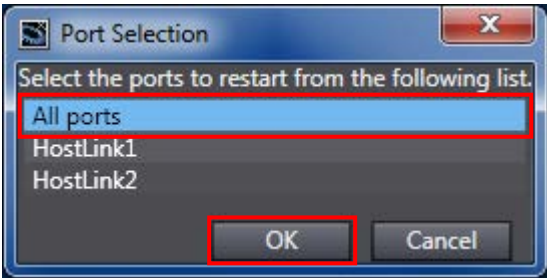
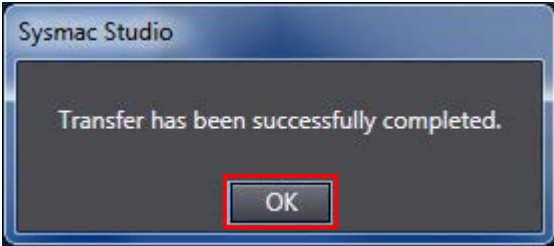
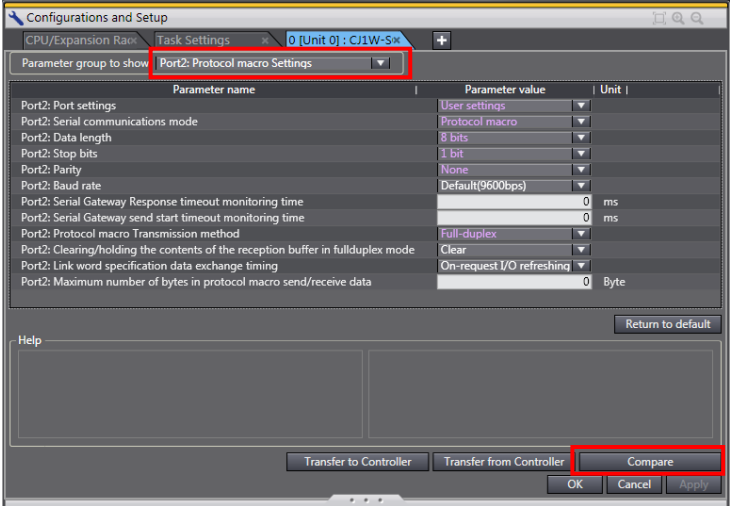
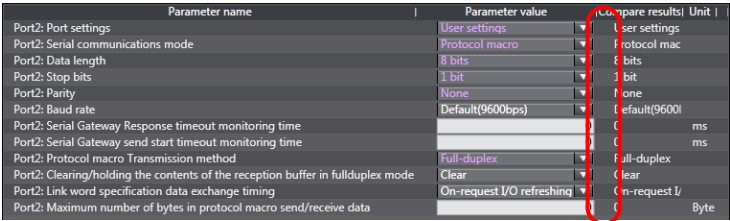
15 The Controller is reset and the Sysmac Studio goes offline. Then, the yellow bar on the top of the Edit Pane disappears. Go online again by following steps 6 to 8.



7.3.5. Transferring the Unit Settings

Transfer the setting data of the Serial Communication Unit.

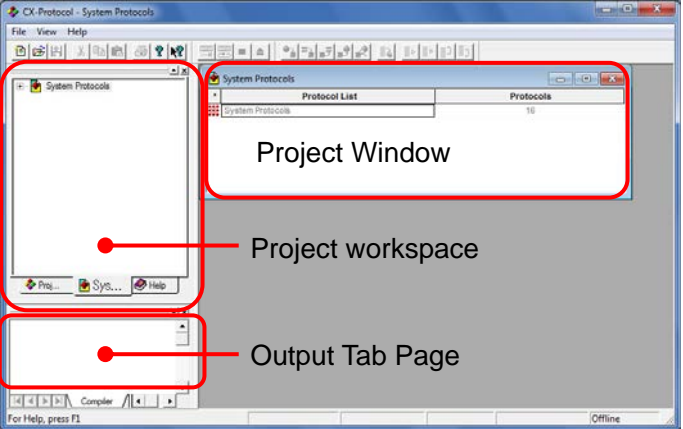
<p>1</p>	<p>Select Mode - PROGRAM Mode from the Controller Menu.</p>	
<p>2</p>	<p>A confirmation dialog box is displayed. Confirm that there is no problem and click the Yes Button.</p>	
<p>3</p>	<p>PROGRAM mode is displayed on the Controller Status Pane.</p>	
<p>4</p>	<p>Double-click CPU/Expansion Racks under Configurations and Setup in the Multiview Explorer. Select the Serial Communications Unit icon. Click Edit Special Unit Settings.</p>	
<p>5</p>	<p>The 0 [Unit 0]: Tab is displayed. Click the Transfer To Controller Button.</p>	

- 6 A confirmation dialog box is displayed. Check the contents and click the **Yes** Button.
- 
- A dialog box is displayed indicating transferring is being performed, and a confirmation dialog box is displayed. Check the contents and click the **Yes** Button.
- 
- 7 The Port Selection Dialog Box is displayed. Select *All ports* and click the **OK** Button.
- 
- 8 A confirmation dialog box is displayed. Check the contents and click the **OK** Button.
- 
- 9 Select *Port2: Protocol macro Settings* from the pull-down list of Parameter group to show. Click the **Compare** Button.
- 
- 10 Confirm that "≠" (mismatch) is not shown in the red frame on the right.
- 
- | Parameter name | Parameter value | Compare results | Unit |
|--|---------------------------|-----------------|---------------|
| Port2: Port settings | User settings | | User settings |
| Port2: Serial communications mode | Protocol macro | | Protocol mac |
| Port2: Data length | 8 bits | | 8 bits |
| Port2: Stop bits | 1 bit | | 1 bit |
| Port2: Parity | None | | None |
| Port2: Baud rate | Default(9600bps) | | Default(9600) |
| Port2: Serial Gateway Response timeout monitoring time | 0 | | ms |
| Port2: Serial Gateway send start timeout monitoring time | 0 | | ms |
| Port2: Protocol macro Transmission method | Full-duplex | | Full-duplex |
| Port2: Clearing/holding the contents of the reception buffer in full-duplex mode | Clear | | Clear |
| Port2: Link word specification data exchange timing | On-request I/O refreshing | | On-request I/ |
| Port2: Maximum number of bytes in protocol macro send/receive data | 0 | | Byte |

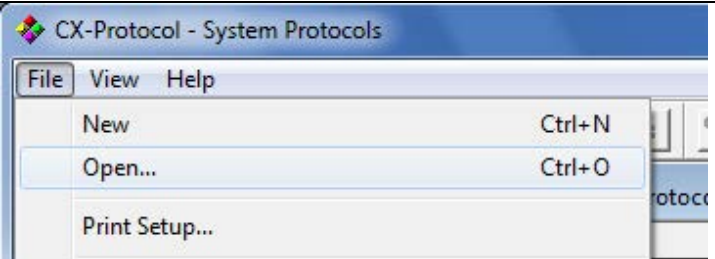
7.3.6. Starting the CX-Protocol and Opening the Protocol Macro Data

Start the CX-Protocol and open the protocol macro data.

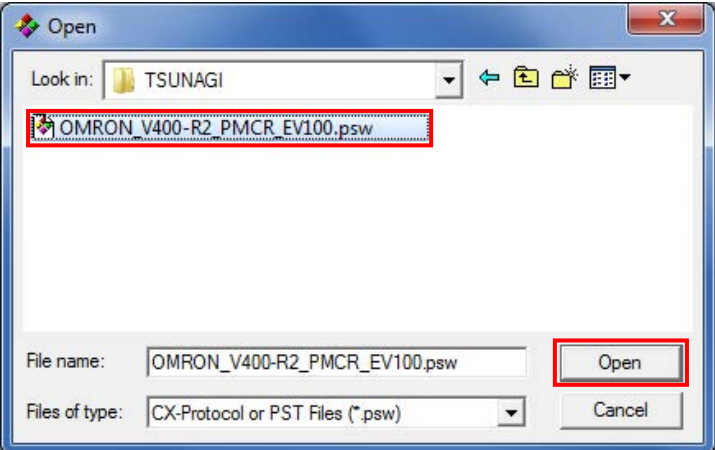
- 1 Start the CX-Protocol.

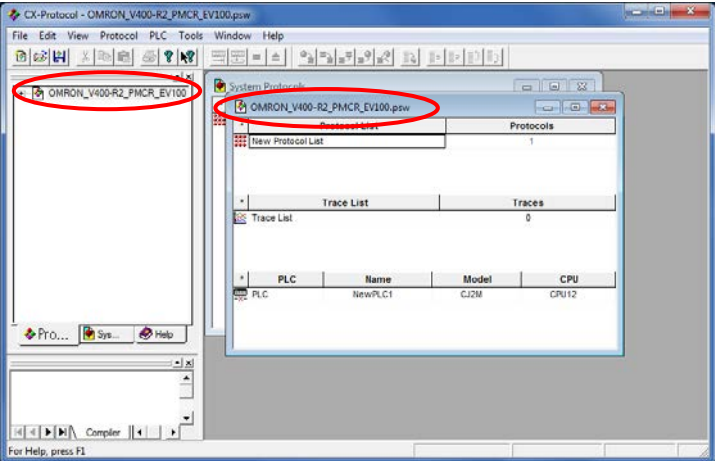


(CX-Protocol)
- 2 Select **Open** from the File Menu.


- 3 On the Open Dialog Box, select OMRON_V400-R2_PMCR_EV100.psw and click the **Open** Button.

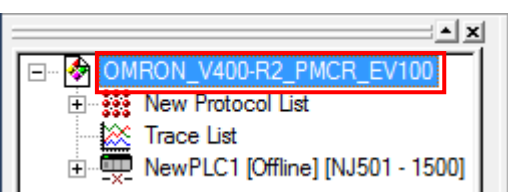
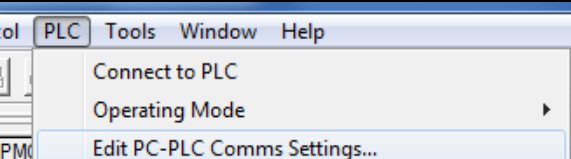
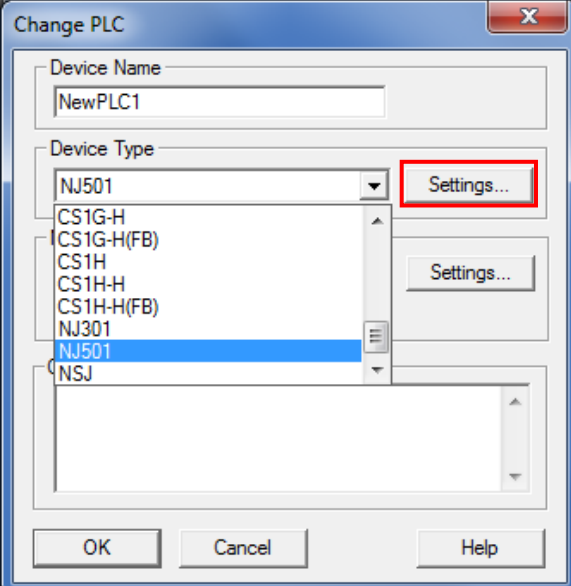
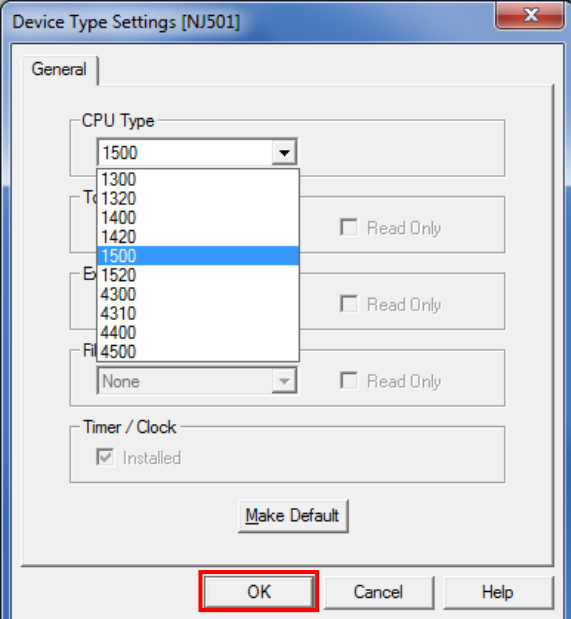
*Obtain the protocol macro data from OMRON.


- 4 The project workspace and the Project Window display the protocol macro data that was read.



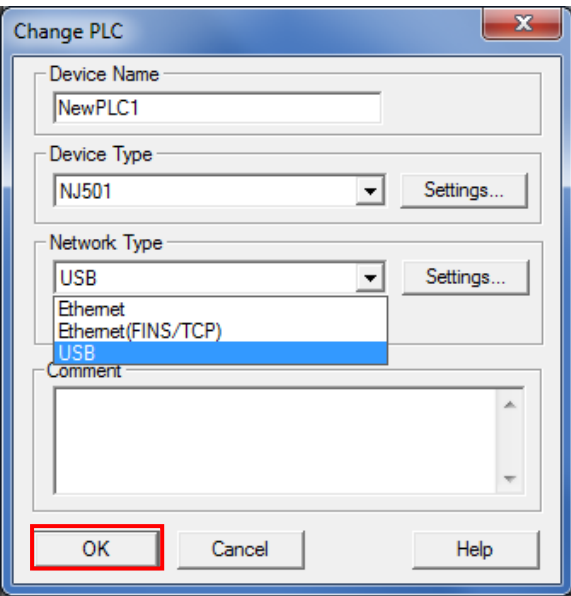
7.3.7. Connecting Online and Transferring the Protocol Macro Data

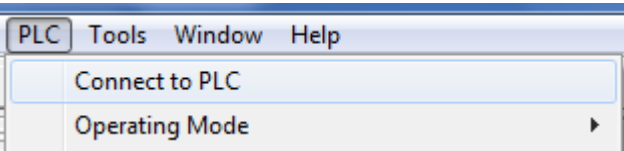
Connect online with the CX-Protocol, and transfer the protocol macro data to the Serial Communications Unit.

<p>1 Double-click OMRON_V400-R2_PMCR_EV100 on the project workspace to display a tree.</p>	
<p>2 Select Edit PC-PLC Comms Settings from the PLC Menu.</p>	
<p>3 The Change PLC Dialog Box is displayed. Select a device type to use from the pull-down list of the Device Type, and click the Settings Button.</p> <p>*NJ501 is used in this document.</p>	
<p>4 The Device Type Settings Dialog Box is displayed. Select the CPU type to use from the pull-down list, and click the OK Button.</p> <p>*1500 is used in this document.</p>	

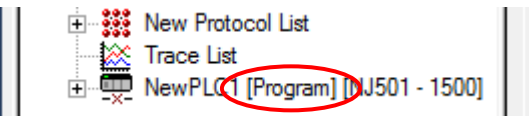
- 5 Confirm that the Network Type is set to USB on the Change PLC Dialog Box and click the **OK** Button.

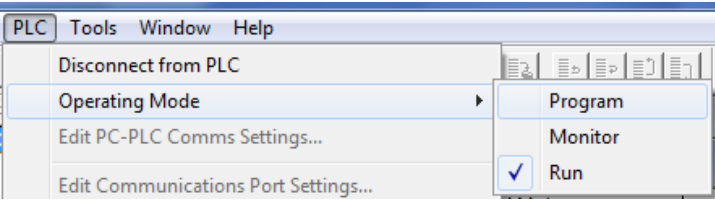
*If the Network Type is not set to USB, select *USB* from the pull-down list.

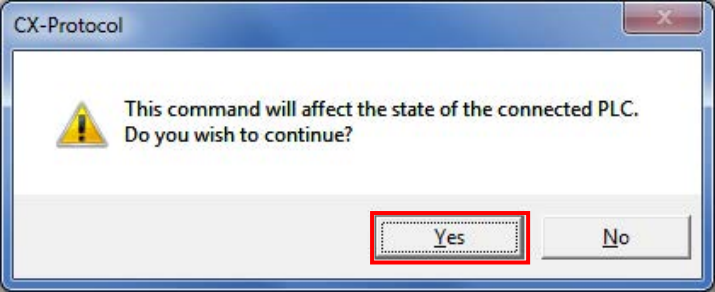
The 'Change PLC' dialog box is shown. It has a title bar with a close button. The 'Device Name' field contains 'NewPLC1'. The 'Device Type' dropdown is set to 'NJ501'. The 'Network Type' dropdown is set to 'USB', and its list is open, showing 'Ethernet', 'Ethernet(FINS/TCP)', and 'USB' (which is highlighted). There are 'Settings...' buttons next to both 'Device Type' and 'Network Type'. A 'Comment' text area is empty. At the bottom, the 'OK' button is highlighted with a red box, along with 'Cancel' and 'Help' buttons.
- 6 Select **Connect to PLC** from the PLC Menu.

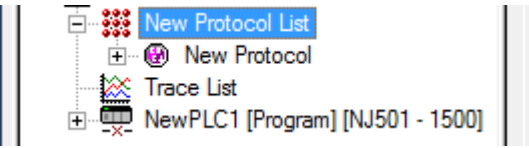
The 'PLC' menu is open, showing 'Connect to PLC' and 'Operating Mode'.
- 7 The PLC icon on the project workspace changes from Offline to Program. It means that the Controller is connected online.

*If Run is displayed, change it to Program by following steps 8 and 9.

The project workspace shows a tree view with 'New Protocol List', 'Trace List', and 'NewPLC1 [Program] [NJ501 - 1500]'. The 'NewPLC1' icon is circled in red.
- 8 If the operating mode of the Controller is Run, select **Operating Mode - Program** from the PLC Menu.

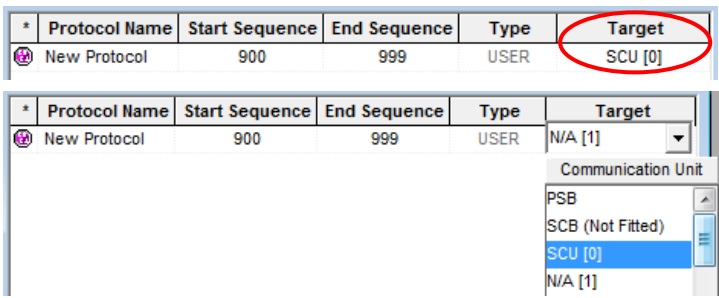
The 'PLC' menu is open, and the 'Operating Mode' sub-menu is also open, showing 'Program', 'Monitor', and 'Run' (which has a checkmark).
- 9 The dialog box on the right is displayed. Confirm that there is no problem and click the **Yes** Button. Confirm that the operating mode was changed to Program mode as shown in step 7.

The 'CX-Protocol' dialog box displays a warning: 'This command will affect the state of the connected PLC. Do you wish to continue?'. The 'Yes' button is highlighted with a red box.
- 10 Double-click the **New Protocol List** on the project workspace to display a tree.

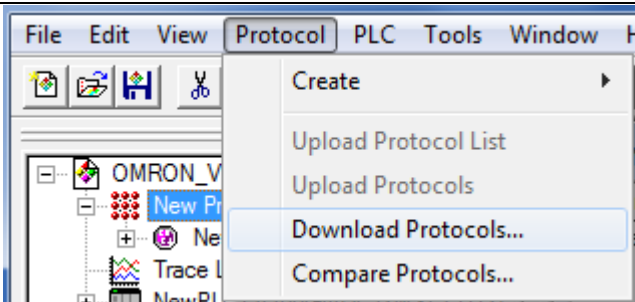
The project workspace shows a tree view with 'New Protocol List' (highlighted in blue), 'New Protocol', 'Trace List', and 'NewPLC1 [Program] [NJ501 - 1500]'. The 'New Protocol List' icon is highlighted.

11 The Project Window on the right is displayed. Confirm that SCU[0] is entered in the Target Column.

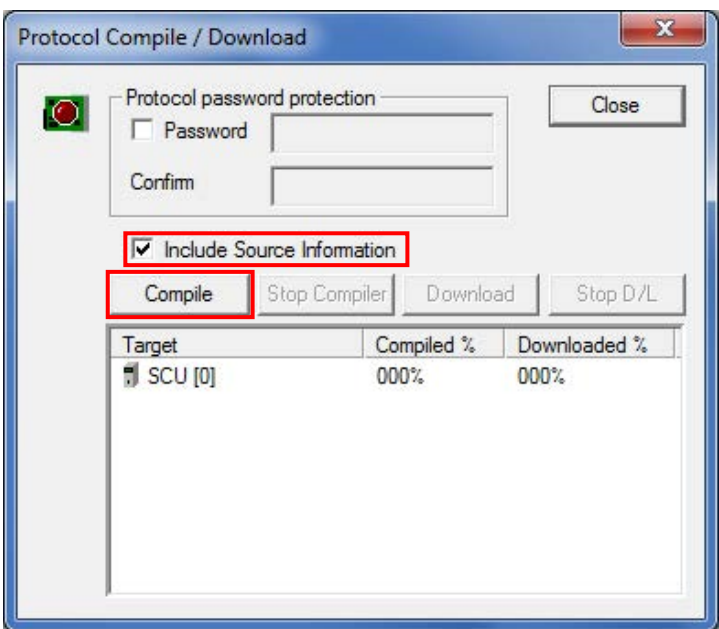
*If SCU[0] is not entered, select *SCU[0]* as shown on the right figure.



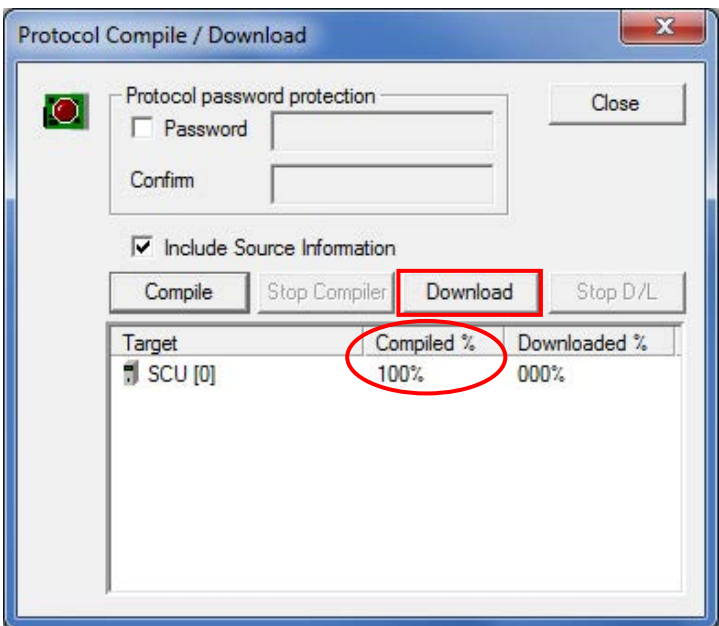
12 Select *New Protocol List* and select **Download Protocols** from the Protocol Menu.



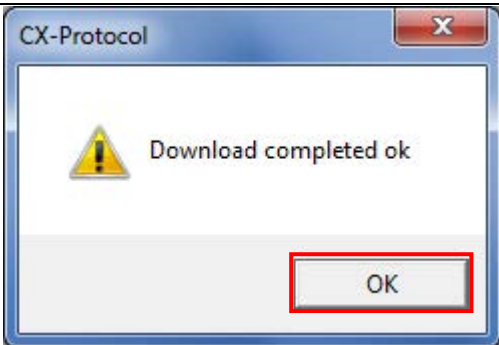
13 The dialog box on the right is displayed. Select the *Include Source Information* Check Box and click the **Compile** Button.



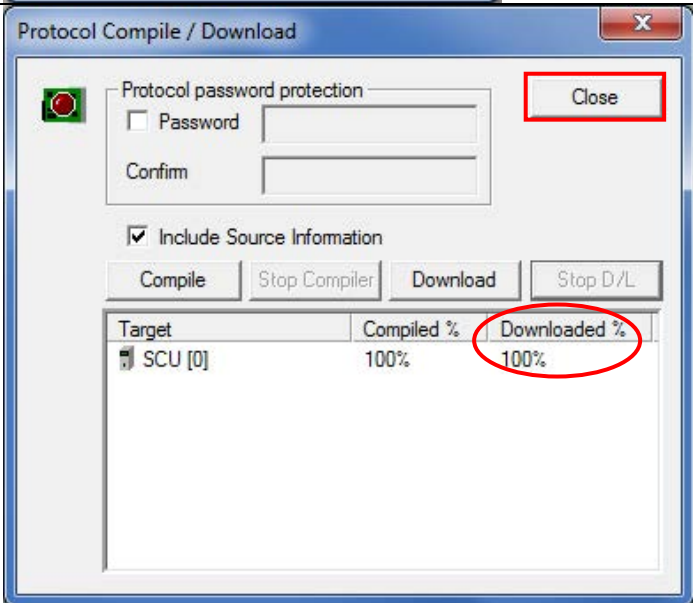
14 When 100% is displayed in the Compiled % Column, the compiling operation is completed. After confirming that the compiling operation is completed, click the **Download** Button.



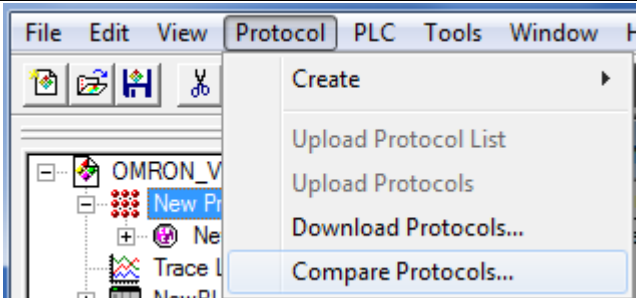
15 The dialog box on the right is displayed. Confirm that there is no problem and click the **OK** Button.



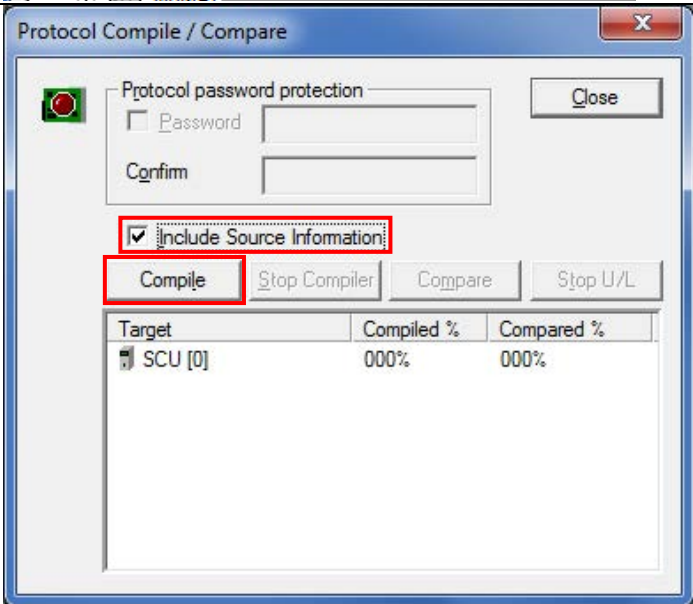
16 Check that 100% is displayed in the Downloaded % Column in the right figure, and click the **Close** Button.



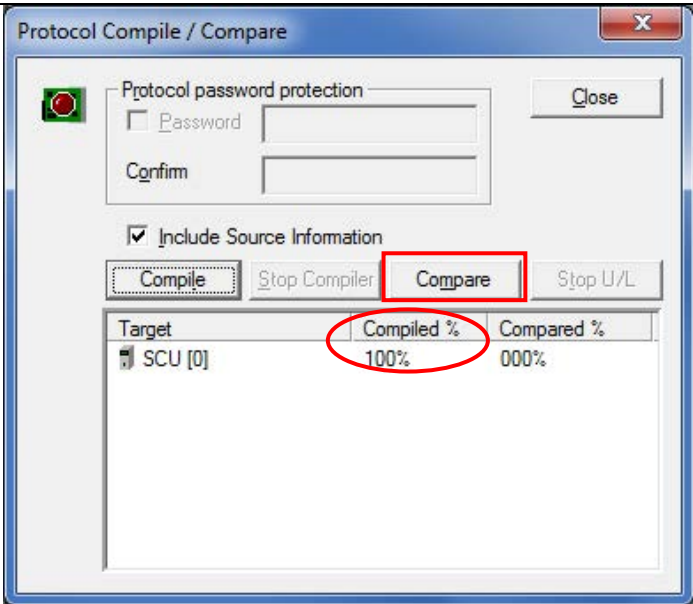
17 Select *New Protocol List* and select **Compare Protocols** from the Protocol Menu.



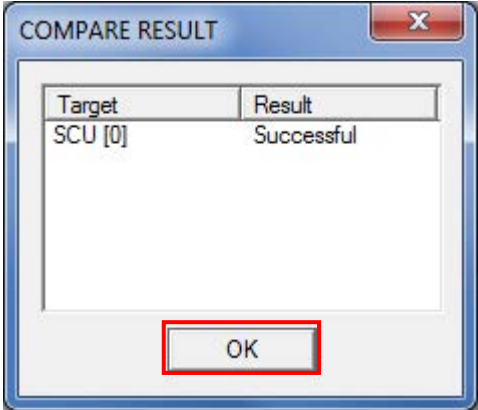
18 The dialog box on the right is displayed. Select the *Include Source Information* Check Box and click the **Compile** Button.



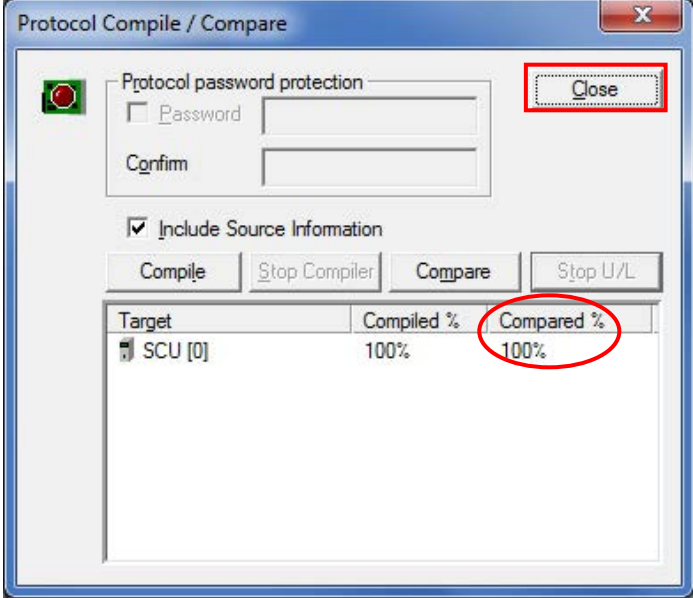
19 When 100% is displayed in the Complied % Column, the compiling operation is completed. Confirming that the compiling operation is completed, and click the **Compare** Button.



20 The dialog box on the right is displayed. Confirm that Successful is displayed in the Result Column and click the **OK** Button.



21 Confirm that 100% is displayed in the Compared % Column in the right figure, and click the **Close** Button.



7.4. Checking the Serial Communications

Execute the program and confirm that serial communications are normally performed.

⚠ Caution

Sufficiently confirm safety before you change the values of variables on a Watch Tab Page when the Sysmac Studio is online with the CPU Unit. Incorrect operation may cause the devices that are connected to Output Units to operate regardless of the operating mode of the Controller.



Precautions for Correct Use

This document explains the procedures for setting up the Code Reader from the factory default setting. If the Code Reader was changed from the factory default setting, make sure to initialize by referring to Section 8.2. Initializing the Code Reader.

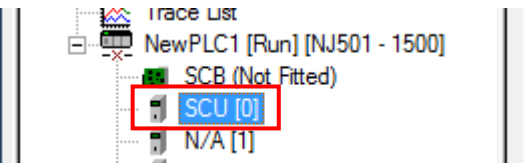
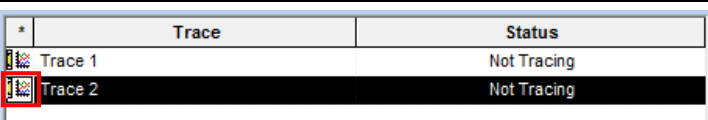
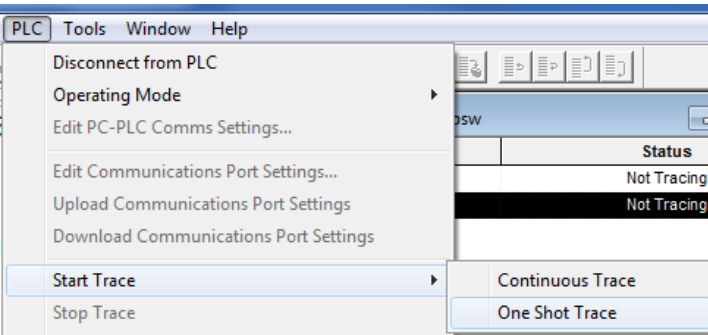
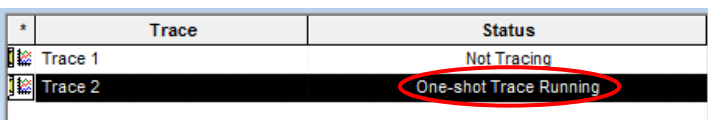
Precautions for Correct Use

Confirm that the serial cable is connected before proceeding to the following procedure. If it is not connected, turn OFF the power supply to each device, and then connect the serial cable.

7.4.1. Starting Tracing

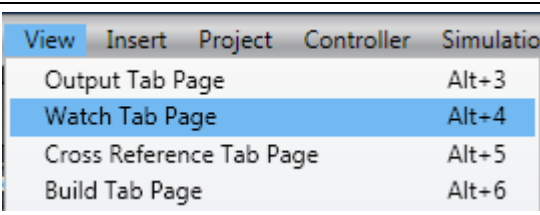
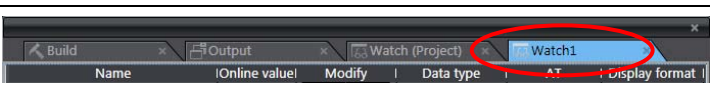



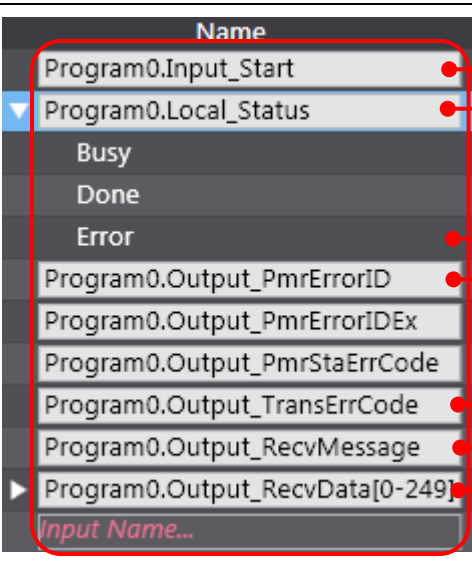

Start tracing with the CX-Protocol.

1	Select Operating Mode - Run from the PLC Menu of the CX-Protocol.	
2	The dialog box on the right is displayed. Confirm that there is no problem and click the Yes Button.	
3	Confirm that the operating mode was changed to the Run mode, and double-click NewPLC1 .	

<p>4 The tree under NewPLC1 expands. Select the Serial Communications Unit (SCU[0] is selected in the right figure).</p>	 <p>The screenshot shows a tree structure under 'NewPLC1 [Run] [NJ501 - 1500]'. Underneath, there is a sub-tree containing 'SCB (Not Fitted)' and 'SCU [0]'. The 'SCU [0]' item is highlighted with a blue selection box.</p>						
<p>5 Select the Trace 2 Icon on the Project Window. Confirm that Trace 2 is highlighted as shown in the right figure.</p> <p>*Trace 2 corresponds to port 2 of the Serial Communications Unit.</p>	 <table border="1"> <thead> <tr> <th>Trace</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>Trace 1</td> <td>Not Tracing</td> </tr> <tr> <td>Trace 2</td> <td>Not Tracing</td> </tr> </tbody> </table>	Trace	Status	Trace 1	Not Tracing	Trace 2	Not Tracing
Trace	Status						
Trace 1	Not Tracing						
Trace 2	Not Tracing						
<p>6 Select Start Trace - One Shot Trace from the PLC Menu.</p>	 <p>The screenshot shows the 'PLC' menu open. The 'Start Trace' option is selected, and its sub-menu is visible, showing 'Continuous Trace' and 'One Shot Trace'. The 'One Shot Trace' option is highlighted.</p>						
<p>7 Confirm that the status of Trace 2 in the Project Window was changed to One-shot Trace Running.</p>	 <table border="1"> <thead> <tr> <th>Trace</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>Trace 1</td> <td>Not Tracing</td> </tr> <tr> <td>Trace 2</td> <td>One-shot Trace Running</td> </tr> </tbody> </table>	Trace	Status	Trace 1	Not Tracing	Trace 2	One-shot Trace Running
Trace	Status						
Trace 1	Not Tracing						
Trace 2	One-shot Trace Running						

7.4.2. Executing the Program

Execute the program with the Sysmac Studio.

1	Select Watch Tab Page from the View Menu of the Sysmac Studio.	
2	The Watch1 Tab Page is displayed in the lower section of the Edit Pane.	
3	<p>Confirm that the variables shown on the right are displayed in the Name Columns.</p> <p>*To add a variable, click <i>Input Name</i>.</p> <p>*If  is displayed to the left of the variable instead of , click  and open the configuration of the structure variable.</p> <p>*Program0 of the Name is omitted from the following descriptions.</p>	 <p>Start input</p> <p>Program execution status</p> <p>Error codes</p> <p>Receive data after converted to a string</p> <p>Receive data storage area</p>
4	Read the bar code shown on the right as an example. Position the Code Reader so that it can read the bar code shown on the right.	

5 Click **TRUE** on the *Modify* Column of *Input_Start*.

The online value of *Input_Start* changes to True.

The program operates, serial communications are performed with the Code Reader, and then the bar code is read.

*If Read confirmation LED that is on the top of the Code Reader is lit green and a buzzer sounds, then the read operation succeeds.

*If the bar code shown in the document cannot be read, read a bar code shown in *Preliminary Examination* in *Section 2 Introduction Procedure* of the *Ultra Small Multi-code Reader V400-R2 Series User's Manual* (Cat. No. Z333). When reading, change *Input_Start* to False and then change *Input_Start* to True again by using the same procedure as this step.

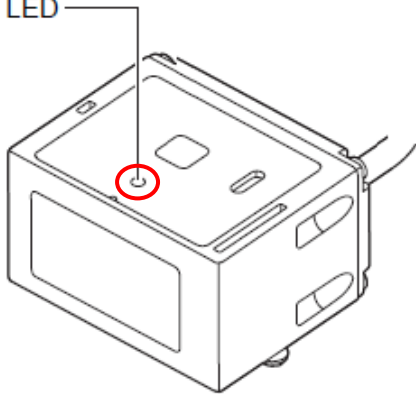
Name	Online value	Modify
Program0.Input_Start	False	TRUE FALSE



Name	Online value	Modify
Program0.Input_Start	True	TRUE FALSE

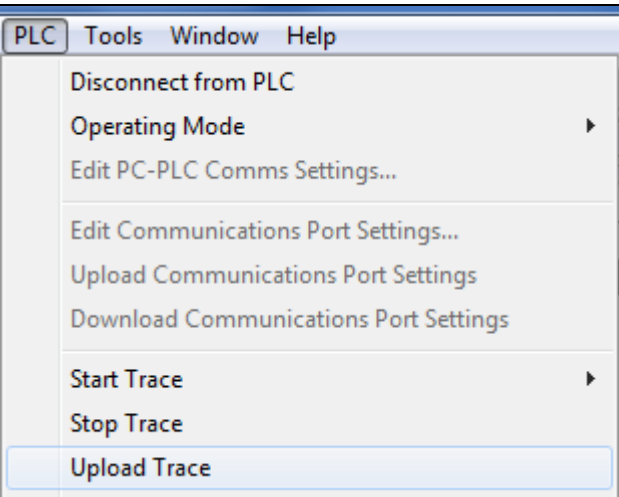
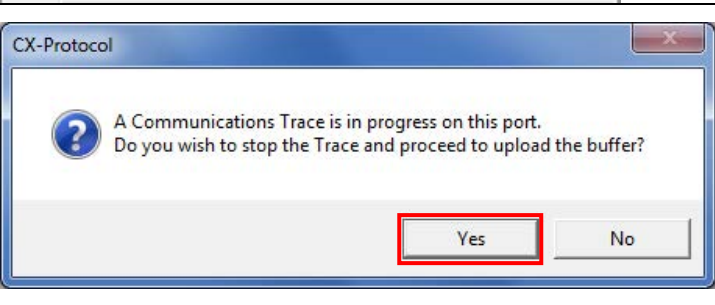
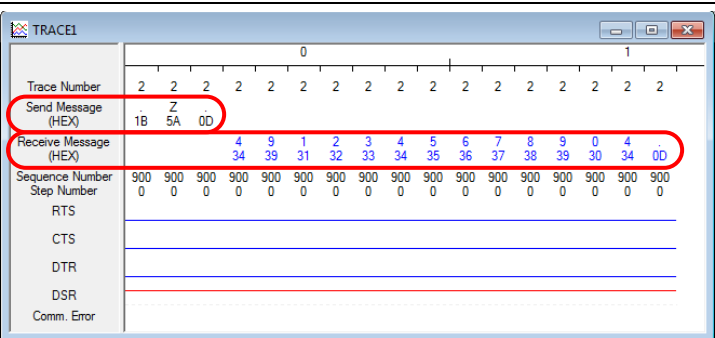
A buzzer sounds if the read operation succeeds.

Read confirmation LED



7.4.3. Checking the Trace Data

Confirm that the correct data is sent and received by checking the trace data of the CX-Protocol.

<p>1 Select Upload Trace from the PLC Menu from the CX-Protocol.</p>																																																																																																																																																																																																																																																																																																																																			
<p>2 A dialog box on the right is displayed. Check the contents and click the Yes Button.</p>																																																																																																																																																																																																																																																																																																																																			
<p>3 Check the send/receive message on the trace data file shown in the right figure.</p> <p>*In the example on the right, the Reading Trigger command is sent from the Controller and the bar code data that are shown in step 4 of Section 7.4.2. are received.</p>	 <table border="1" data-bbox="710 1160 1428 1496"> <thead> <tr> <th>Trace Number</th> <th colspan="16">0</th> <th colspan="16">1</th> </tr> </thead> <tbody> <tr> <td>Send Message (HEX)</td> <td colspan="16">Z</td> <td colspan="16"></td> </tr> <tr> <td>Receive Message (HEX)</td> <td>1B</td><td>5A</td><td>0D</td> <td>4</td><td>9</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>4</td> <td>34</td><td>39</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>30</td><td>34</td><td>0D</td> </tr> <tr> <td>Sequence Number</td> <td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td> <td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td><td>900</td> </tr> <tr> <td>Step Number</td> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>RTS</td> <td colspan="32"></td> </tr> <tr> <td>CTS</td> <td colspan="32"></td> </tr> <tr> <td>DTR</td> <td colspan="32"></td> </tr> <tr> <td>DSR</td> <td colspan="32"></td> </tr> <tr> <td>Comm. Error</td> <td colspan="32"></td> </tr> </tbody> </table> <ul style="list-style-type: none"> ■ Send message <ul style="list-style-type: none"> Hex :1B 5A 0D String :[ESC]"Z"[CR] ■ Receive message <ul style="list-style-type: none"> Hex :34 39 31 32 33 34 35 36 37 38 39 30 34 0D String :“4912345678904”[CR] 	Trace Number	0																1																Send Message (HEX)	Z																																Receive Message (HEX)	1B	5A	0D	4	9	1	2	3	4	5	6	7	8	9	0	4	34	39	31	32	33	34	35	36	37	38	39	30	34	0D	Sequence Number	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	Step Number	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RTS																																	CTS																																	DTR																																	DSR																																	Comm. Error																																
Trace Number	0																1																																																																																																																																																																																																																																																																																																																		
Send Message (HEX)	Z																																																																																																																																																																																																																																																																																																																																		
Receive Message (HEX)	1B	5A	0D	4	9	1	2	3	4	5	6	7	8	9	0	4	34	39	31	32	33	34	35	36	37	38	39	30	34	0D																																																																																																																																																																																																																																																																																																					
Sequence Number	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900																																																																																																																																																																																																																																																																																																						
Step Number	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																						
RTS																																																																																																																																																																																																																																																																																																																																			
CTS																																																																																																																																																																																																																																																																																																																																			
DTR																																																																																																																																																																																																																																																																																																																																			
DSR																																																																																																																																																																																																																																																																																																																																			
Comm. Error																																																																																																																																																																																																																																																																																																																																			

7.4.4. Checking the Receive Data

Confirm that the correct data are written to the variables of the Controller with the Sysmac Studio.

<p>1</p>	<p>Confirm that the online value of <i>Local_Status.Done</i>, which indicates the program execution status, was changed to True.</p> <p>*This means the program ended normally.</p> <p>*In the case of an error end, <i>Local_Status.Error</i> changes to True. Troubleshoot the error by referring to step 2.</p>	<p>[Normal end]</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Online value</th> <th>Modify</th> </tr> </thead> <tbody> <tr> <td>Program0.Input_Start</td> <td>True</td> <td>TRUE FALSE</td> </tr> <tr> <td>▼ Program0.Local_Status</td> <td></td> <td></td> </tr> <tr> <td> Busy</td> <td>False</td> <td>TRUE FALSE</td> </tr> <tr> <td> Done</td> <td>True</td> <td>TRUE FALSE</td> </tr> <tr> <td> Error</td> <td>False</td> <td>TRUE FALSE</td> </tr> </tbody> </table> <p>[Error end]</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Online value</th> <th>Modify</th> </tr> </thead> <tbody> <tr> <td>Program0.Input_Start</td> <td>True</td> <td>TRUE FALSE</td> </tr> <tr> <td>▼ Program0.Local_Status</td> <td></td> <td></td> </tr> <tr> <td> Busy</td> <td>False</td> <td>TRUE FALSE</td> </tr> <tr> <td> Done</td> <td>False</td> <td>TRUE FALSE</td> </tr> <tr> <td> Error</td> <td>True</td> <td>TRUE FALSE</td> </tr> </tbody> </table>	Name	Online value	Modify	Program0.Input_Start	True	TRUE FALSE	▼ Program0.Local_Status			Busy	False	TRUE FALSE	Done	True	TRUE FALSE	Error	False	TRUE FALSE	Name	Online value	Modify	Program0.Input_Start	True	TRUE FALSE	▼ Program0.Local_Status			Busy	False	TRUE FALSE	Done	False	TRUE FALSE	Error	True	TRUE FALSE
Name	Online value	Modify																																				
Program0.Input_Start	True	TRUE FALSE																																				
▼ Program0.Local_Status																																						
Busy	False	TRUE FALSE																																				
Done	True	TRUE FALSE																																				
Error	False	TRUE FALSE																																				
Name	Online value	Modify																																				
Program0.Input_Start	True	TRUE FALSE																																				
▼ Program0.Local_Status																																						
Busy	False	TRUE FALSE																																				
Done	False	TRUE FALSE																																				
Error	True	TRUE FALSE																																				
<p>2</p>	<p>If the program ended normally, confirm that each error code is 0.</p> <p>*If the program ended in an error, the error code corresponding to the error is stored. Troubleshoot the error by referring to 9.8. Error Process.</p>	<p>[Normal end]</p> <table border="1"> <tbody> <tr> <td>Program0.Output_PmrErrorID</td> <td>0000</td> <td></td> </tr> <tr> <td>Program0.Output_PmrErrorIDEx</td> <td>0000 0000</td> <td></td> </tr> <tr> <td>Program0.Output_PmrStaErrCode</td> <td>0000</td> <td></td> </tr> <tr> <td>Program0.Output_TransErrCode</td> <td>0000</td> <td></td> </tr> </tbody> </table> <p>[Error end]</p> <table border="1"> <tbody> <tr> <td>Program0.Output_PmrErrorID</td> <td>0000</td> <td></td> </tr> <tr> <td>Program0.Output_PmrErrorIDEx</td> <td>0000 0000</td> <td></td> </tr> <tr> <td>Program0.Output_PmrStaErrCode</td> <td>0000</td> <td></td> </tr> <tr> <td>Program0.Output_TransErrCode</td> <td>9320</td> <td></td> </tr> </tbody> </table>	Program0.Output_PmrErrorID	0000		Program0.Output_PmrErrorIDEx	0000 0000		Program0.Output_PmrStaErrCode	0000		Program0.Output_TransErrCode	0000		Program0.Output_PmrErrorID	0000		Program0.Output_PmrErrorIDEx	0000 0000		Program0.Output_PmrStaErrCode	0000		Program0.Output_TransErrCode	9320													
Program0.Output_PmrErrorID	0000																																					
Program0.Output_PmrErrorIDEx	0000 0000																																					
Program0.Output_PmrStaErrCode	0000																																					
Program0.Output_TransErrCode	0000																																					
Program0.Output_PmrErrorID	0000																																					
Program0.Output_PmrErrorIDEx	0000 0000																																					
Program0.Output_PmrStaErrCode	0000																																					
Program0.Output_TransErrCode	9320																																					
<p>3</p>	<p>Check the received data (code that was read) in the Watch Window of the Sysmac Studio.</p> <p>*In the example on the right, the data stored in <i>Output_RecvMessage</i> is 4912345678904. It is the same as the trace data in step 3 of Section 7.4.3.</p> <p>*The number of words used to store the receive data (0008 in hexadecimal) is stored in <i>Output_RecvData[0]</i>. The bar code data that were read are stored from <i>Output_RecvData[1]</i> to [7]. The number of used words is 8 including <i>Output_RecvData[0]</i>.</p>	<table border="1"> <tbody> <tr> <td>Program0.Output_RecvMessage</td> <td>4912345678904</td> <td></td> </tr> <tr> <td>▼ Program0.Output_RecvData[0-249]</td> <td></td> <td></td> </tr> <tr> <td> Output_RecvData[0]</td> <td>0008</td> <td></td> </tr> <tr> <td> Output_RecvData[1]</td> <td>3439</td> <td></td> </tr> <tr> <td> Output_RecvData[2]</td> <td>3132</td> <td></td> </tr> <tr> <td> Output_RecvData[3]</td> <td>3334</td> <td></td> </tr> <tr> <td> Output_RecvData[4]</td> <td>3536</td> <td></td> </tr> <tr> <td> Output_RecvData[5]</td> <td>3738</td> <td></td> </tr> <tr> <td> Output_RecvData[6]</td> <td>3930</td> <td></td> </tr> <tr> <td> Output_RecvData[7]</td> <td>3400</td> <td></td> </tr> <tr> <td> Output_RecvData[8]</td> <td>0000</td> <td></td> </tr> <tr> <td> Output_RecvData[9]</td> <td>0000</td> <td></td> </tr> </tbody> </table> <p>Receive data (Hex)</p> <ul style="list-style-type: none"> • Output_RecvData[0]:"0008" • Output_RecvData[1]:"3439" • Output_RecvData[2]:"3132" • Output_RecvData[3]:"3334" • Output_RecvData[4]:"3536" • Output_RecvData[5]:"3738" • Output_RecvData[6]:"3930" • Output_RecvData[7]:"3400" <p>Number of words used to store the receive data Eight words from 0 to 7</p> <p>Values of Bar Code (String) Output_RecvMessage "4912345678904"</p>	Program0.Output_RecvMessage	4912345678904		▼ Program0.Output_RecvData[0-249]			Output_RecvData[0]	0008		Output_RecvData[1]	3439		Output_RecvData[2]	3132		Output_RecvData[3]	3334		Output_RecvData[4]	3536		Output_RecvData[5]	3738		Output_RecvData[6]	3930		Output_RecvData[7]	3400		Output_RecvData[8]	0000		Output_RecvData[9]	0000	
Program0.Output_RecvMessage	4912345678904																																					
▼ Program0.Output_RecvData[0-249]																																						
Output_RecvData[0]	0008																																					
Output_RecvData[1]	3439																																					
Output_RecvData[2]	3132																																					
Output_RecvData[3]	3334																																					
Output_RecvData[4]	3536																																					
Output_RecvData[5]	3738																																					
Output_RecvData[6]	3930																																					
Output_RecvData[7]	3400																																					
Output_RecvData[8]	0000																																					
Output_RecvData[9]	0000																																					

8. Initialization Method

This document explains the setting procedure from the factory default setting.

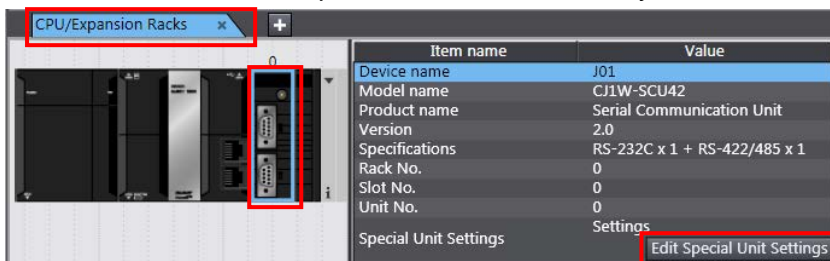
Some settings may not be applicable as described in this document unless you use the devices with the factory default setting.

8.1. Initializing the Controller

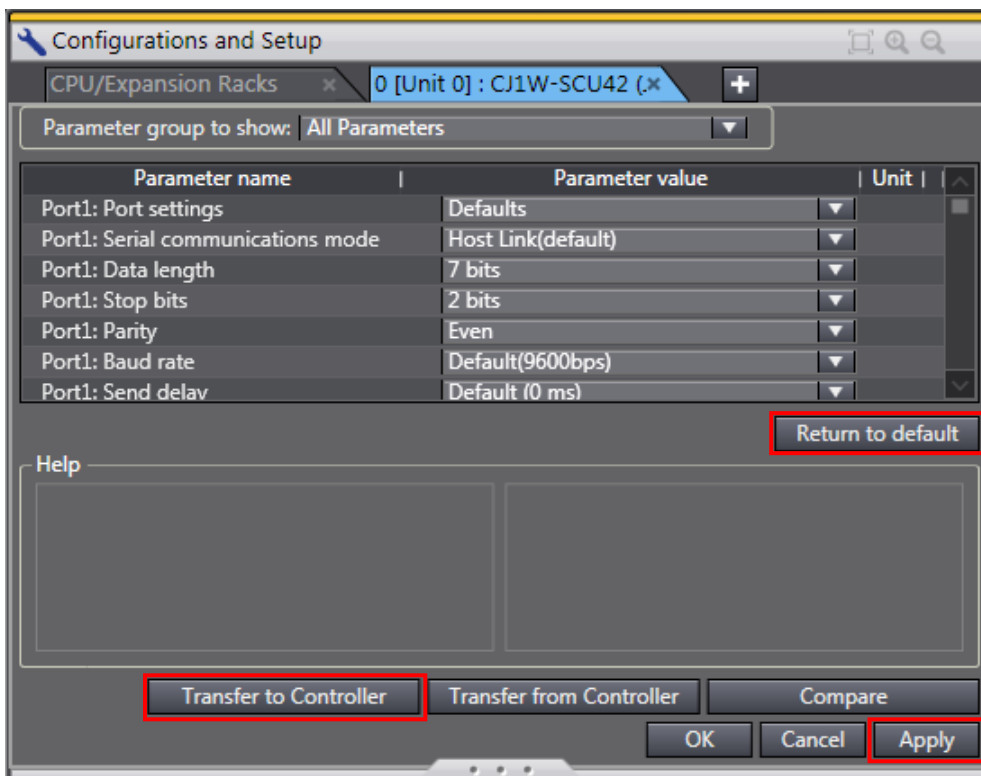
To initialize the Controller, it is necessary to initialize the Serial Communications Unit and the CPU Unit. Place in PROGRAM Mode before the initialization.

8.1.1. Serial Communications Unit

To initialize the settings of the Serial Communications Unit, click **Edit Special Unit Settings** of CJ1W-SCU42 in CPU/Expansion Racks from the Sysmac Studio.

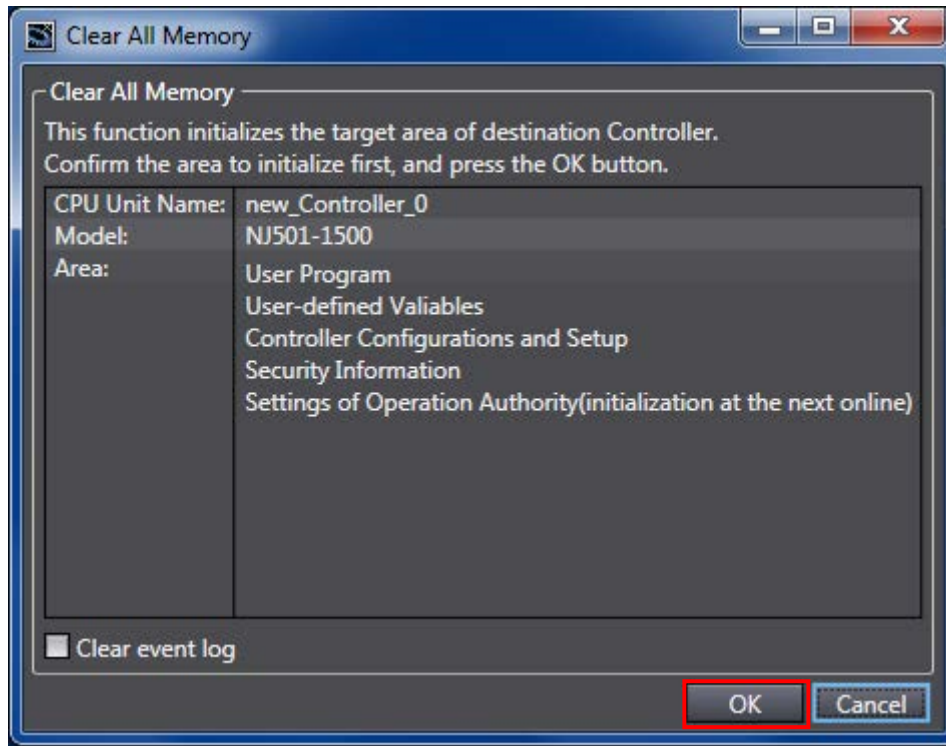


Click the **Return to default** Button and click the **Apply** Button. Then, click the **Transfer To Controller** Button.



8.1.2. CPU Unit

To initialize the settings of the Controller, select **Clear All Memory** from the Controller Menu of the Sysmac Studio. The Clear All Memory Dialog Box is displayed. Click the **OK** Button.



8.2. Initializing the Code Reader

For the initialization of the Code Reader, refer to *Return to the factory default setting* in *Menu Sheet/Command List* in *Section 4 Setting Method* of the *Ultra Small Multi-code Reader V400-R2 Series User's Manual* (Cat. No. Z333).

9. Program

This section describes the details on the program and the protocol macro data used in this document.

9.1. Overview

This section explains the specifications and functions of the program and the protocol macro data that are used to check the connection between the Code Reader (hereinafter referred to as the destination device) and the Controller (Serial Communications Unit) (hereinafter referred to as the Serial Communications Unit).

This program and protocol macro data use the protocol macro function of the Serial Communications Unit to send/receive the Reading Trigger command to/from the destination device and to detect whether the operation ends normally or ends in an error.

A normal end of this program means a normal end of the communications sequence of the protocol macro.

An error end means an error end of the communications sequence of the protocol macro.

In this section, the prefix "10#" (possible to omit) is added to decimal data and the prefix "16#" to hexadecimal data when it is necessary to distinguish between decimal and hexadecimal data. (e.g., "1000" or "10#1000" for decimal data and "16#03E8" for hexadecimal data, etc.) Also, to specify a specific data type, the prefix "<data type>#" is added. (e.g., "WORD#16#03E8")

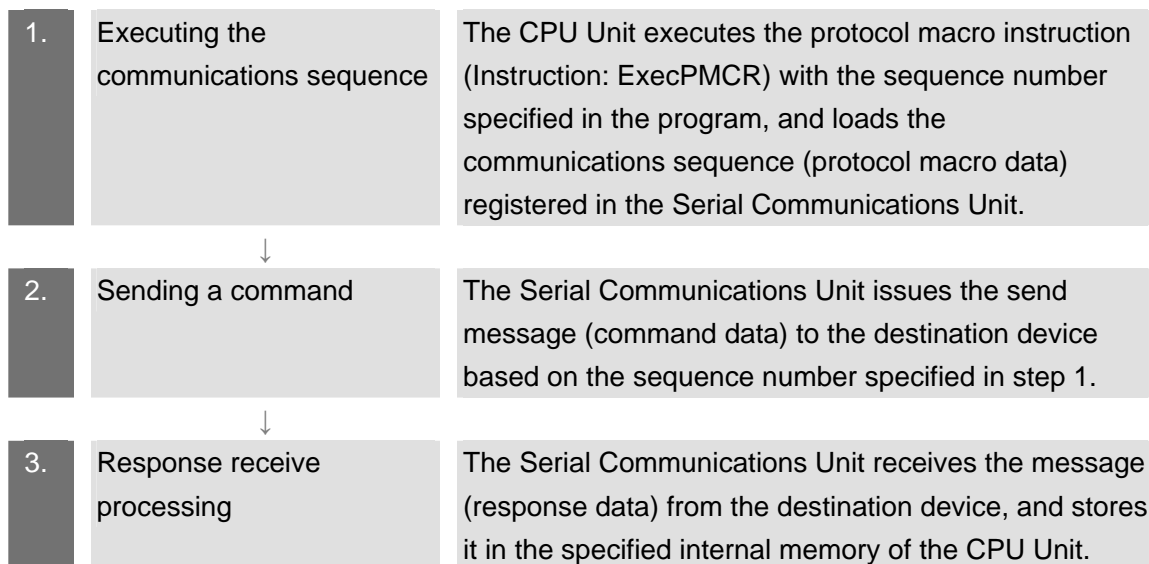


Additional Information

OMRON has confirmed that normal communications can be performed using this program and protocol macro data under the OMRON evaluation conditions including the test system configuration, version and product lot number of each device which was used for evaluation. OMRON does not guarantee the normal operation under the disturbance such as electrical noise or the performance variation of the device.

9.1.1. Communications Data Flow

The following figure shows the data flow from when the Controller (Serial Communications Unit) issues command data via serial communications to the destination device until when it receives the response data from the destination device.



9.1.2. Function Block for Executing Protocol Macro and Send/Receive Messages

This section describes the function block for executing the protocol macro (hereinafter referred to as ExecPMCR instruction) and outlines the general operation of the send/receive messages.



Additional Information

Refer to *Serial Communications Instructions (ExecPMCR)* in Section 2 *Instruction Descriptions of the NJ-series Instructions Reference Manual (Cat. No. W502)* for details.

•ExecPMCR instruction

This program uses the following standard instruction to perform serial communications.

Name	Function blocks	Explanation
Protocol macro	ExecPMCR	Requests execution of a communications sequence (protocol data) registered in a Serial Communications Unit.

•ExecPMCR instruction argument data

Instruction	Name	FB/ FUN	Graphic expression	ST expression
ExecPMCR	Protocol Macro	FB	<pre> ExecPMCR_instance ExecPMCR Execute Done Port Busy SeqNo Error SrcDat ErrorID DstDat ErrorID ErrorIDEx </pre>	ExecPMCR_instance(Execute, Port, SeqNo, SrcDat, DstDat, Done, Busy, Error, ErrorID, ErrorIDEx);

Variables

Name	Meaning	I/O	Description	Valid range	Unit	Default
Port	Destination port	Input	Destination port	---	---	---
SeqNo	Communications sequence number		Communications sequence number	0 to 999		0
SrcDat[] (array)	Send data array		Send data array	Depends on data type.		*
DstDat[] (array)	Receive data array	In-out	Receive data array	Depends on data type.	---	---

* If you omit the input parameter, the default value is not applied. A building error will occur.

•Data type (_sPORT) of destination port Port

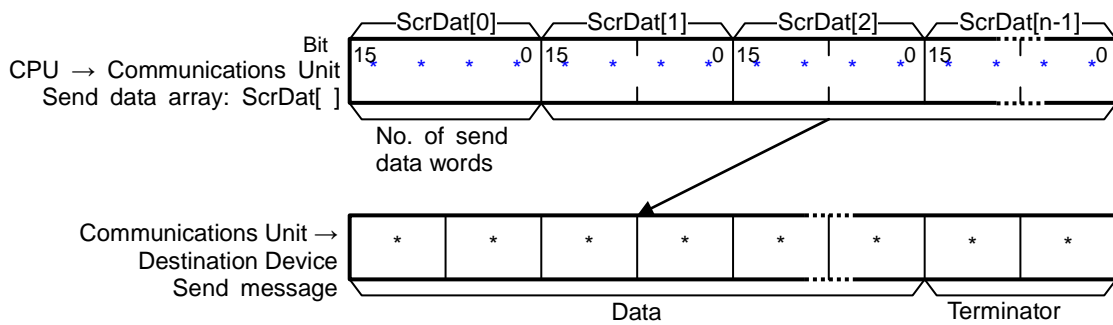
Name	Meaning	Description	Data type	Valid range	Unit	Default
Port	Destination port	Destination port	_sPORT	---	---	---
UnitNo	Unit number	Unit number of Serial Communications Unit	_eUnitNo	_CBU_No00 to _CBU_No15	---	_CBU_No00
PhysicPortNo	Serial port number	Serial port number on Serial Communications Unit	USINT	1 or 2		1

•Send/Receive messages

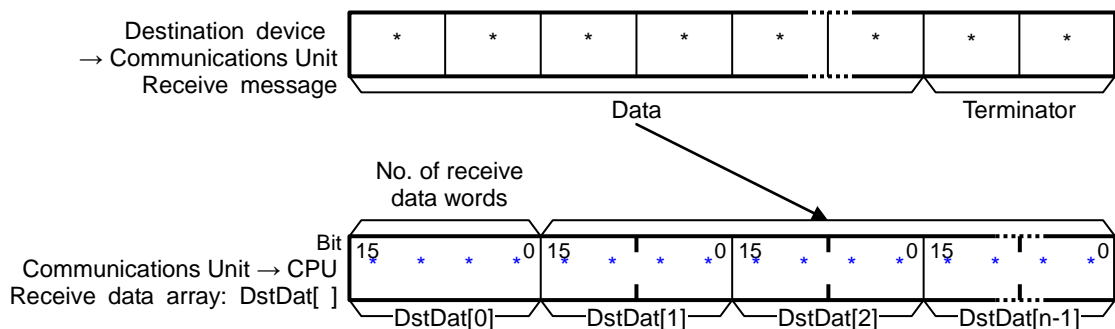
[Overview of send/receive messages]



[Relationship between the SrcDat[] send data array and the send message]



[Relationship between the DstDat[] receive data array and the receive message]



9.2. Communications sequence

This section explains the communications sequence (protocol macro data) that can be used for the ExecPMCR instruction of this program.

9.2.1. Communications sequence No.

A communications sequence (protocol macro data) that is registered in the Serial Communications Unit is identified by a communications sequence number. The Controller executes the corresponding command on the destination device by specifying a communications sequence number in the ExecPMCR instruction.

This protocol macro data includes the following communications sequence.

No.	Command name	Description
900	Reading Trigger	Issues the reading trigger for a code.

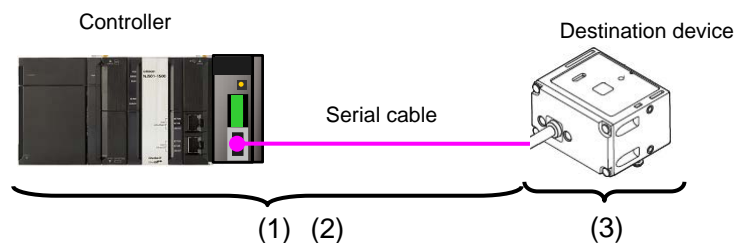
9.2.2. ExecPMCR instruction input variables

The ExecPMCR instruction input variables for communication sequence No.900 Reading Trigger are set as follows.

Variable	Name (data format)	Data (explanation)
Port	Destination port specification (_sPROT)	-
UnitNo	Unit number (_eUnitNo)	_eUnitNo#_CBU_No00 (Specifies the unit number for the Serial Communications Unit.)
PhysicPortNo	Serial port number (USINT)	USINT#10#2 (Use serial port No.2.)
SeqNo	Communications sequence number (UINT)	UINT#10#900 (Specifies Reading Trigger.)
SrcDat[] (array)	Send data array (WORD)	WORD#16#0000 (The number of words of send data is 0.: Specifies that no parameter is required for the command.)
DstDat[] (array)	Receive data array (WORD)	(Receive data. Setting is unnecessary)

9.3. Error Detection Processing

This program detects and handles the errors (1) to (3) below. For information on error codes, refer to 9.8 *Error Process*.



(1) Errors when executing the ExecPMCR instruction (ExecPMCR instruction error)

Errors occurred during an execution of the ExecPMCR instruction such as a unit error and communications error are detected as ExecPMCR instruction errors. An error is detected with the error code (ErrorID) and the expansion error code (ErrorIDEx) of the ExecPMCR instruction.

(2) Errors at an execution of the protocol macro (protocol macro error)

When an operation ends in an error during an execution of a protocol macro for some reason, it is detected as a protocol macro error. An error is detected with the Sequence Abort Completion Flag (J01_P2_PmrSeqAbtSta).

(3) Errors in the destination device (Destination device errors)

Destination device errors include a command error, a parameter error, data error, and an execution failure in the destination device. An error is detected with the response data which is returned from the destination device. With this program, the destination device does not send a response when an error occurs. Therefore, this error is detected as a timeout error described in (2).

The causes of errors (1) and (2) may be related to each other. Therefore, all of the following error codes are stored when either error occurs.

- Error code of ExecPMCR instruction (ErrorID)
- Expansion error code of ExecPMCR instruction (ErrorIDEx)
- Protocol macro error code (lower four bits of J01_P2_PmrSta)
- Transmission error status (J01_P2_TransErrSta)

9.4. Variables

The variables used in this program are listed below.

9.4.1. List of user-defined variables

The following tables list the data type, external variables (user-defined global variables/device variable for CJ-series Unit/system-defined variable) and internal variables that are used in this program.

•Data type (Structure)

[Communications processing status flags]

Name	Data type	Explanation
sStatus	STRUCT	Structure of communications processing status flags
Busy	BOOL	Communications processing in progress flag TRUE: Processing is in progress. / FALSE: Processing is not in progress
Done	BOOL	Communications processing normal end flag TRUE: Normal end / FALSE: Other than normal end
Error	BOOL	Communications processing error end flag TRUE: Error end / FALSE: Other than error end

•External variables

[User-defined global variables]

Variable name	Data type	Explanation
Input_Start	BOOL	Communication start switch The program starts when this flag changes from FALSE to TRUE
Input_SendData	ARRAY[0..249] OF WORD	Send data storage area (500 bytes (WORD))
Output_RecvData	ARRAY[0..249] OF WORD	Receive data storage area (500 bytes (WORD)) An area that stores DstDat[] of the ExecPMCR instruction
Output_RecvMessage	STRING[498]	An area that stores the receive data after converted into a string. (498 characters)
Output_PmrErrorID	WORD	An area that stores an error code of the ExecPMCR instruction Normal end: 16#0000
Output_PmrErrorIDEx	DWORD	An area that stores an expansion error code of the ExecPMCR instruction Normal end: 16#00000000
Output_PmrStaErrCode	WORD	An area that stores a protocol macro error code Normal end: 16#0000
Output_TransErrCode	WORD	An area that stores the transmission error status flag Storage area of J01_P2_TransErrSta Normal end: 16#0000

*You need to use an AT specification for the area that stores DstDat[] of the ExecPMCR instruction to specify an address in the memory used for CJ-series Units.

[Device variables for CJ-series Unit] (Serial Communications Unit)

Variable name	Data type	Explanation
J01_P2_PmrSta	WORD	Protocol macro operation status Bits 03 to 00: Protocol macro error code
J01_P2_PmrExecSta	BOOL	Protocol Macro Execution Flag
J01_P2_PmrSeqEndSta	BOOL	Sequence End Completion Flag
J01_P2_PmrSeqAbtSta	BOOL	Sequence Abort Completion Flag
J01_P2_TransErrSta	WORD	Transmission error status

**Additional Information**

For details on the variables of the Serial Communications Unit, refer to *2-3 Device Variable for CJ-series Unit* in the *CJ-series Serial Communications Units Operation Manual for NJ-series CPU Unit* (Cat.No. W494).

[System-defined variable]

Variable name	Data type	Explanation
_Port_isAvailable	BOOL	Communications Port Enabled Flag TRUE: A port is available FALSE: A port is not available

**Additional Information**

For information on the system-defined variables when the ExecPMCR instruction is used, refer to *Related System-defined Variables in Serial Communications Instructions (ExecPMCR)* in *Section 2 Instruction Descriptions* of the *NJ-series Instructions Reference Manual* (Cat. No. W502).

- Internal variables (Instance variables)

The following tables list the internal variables used to execute the function blocks in the program. An internal variable is called an "instance". The name of the function block to use is specified as the data type of the variable.

Variable name	Data type	Explanation
ExecPMCR_instance	ExecPMCR	Requests execution of a communications sequence (protocol data) registered in a Serial Communications Unit.
F_TRIG_instance	F_TRIG	Outputs TRUE for one task period only when the input signal changes to FALSE.

**Additional Information**

For information on the ExecPMCR instruction, refer to *Serial Communications Instructions (ExecPMCR)* in *Section 2 Instruction Descriptions* of the *NJ-series Instructions Reference Manual* (Cat. No. W502).

**Additional Information**

For information on the F_TRIG instruction, refer to *Sequence Input Instructions (F_TRIG)* in *Section 2 Instruction Descriptions* of the *NJ-series Instructions Reference Manual* (Cat. No. W502).

● Internal variables

Variable name	Data type	Explanation
Local_Status	sStatus	Communications processing status flags This variable is defined as sStatus structure
Local_State	DINT	Status processing number
Local_ExecFlgs	BOOL	Communications instruction execution flag
Local_EndExecPMCR	BOOL	Protocol macro execution end detection flag
Local_InPort	_sPort	Used port
Local_SeqNo	UINT	Communications sequence number
Local_PmrStaCode	WORD	Protocol macro error code edit area
Local_RecvWordData	ARRAY[0..248] OF WORD	An area that extracts and stores the data to be converted into a string from the receive data (498 bytes)
Local_RecvWordSize	UINT	A size of data to be converted into a string (unit: WORD)
Local_RecvByteData	ARRAY[0..497] OF BYTE	An area that stores data in a byte array, which will be converted into a string (498 bytes)
Local_RecvByteSize	UINT	A size of data to be converted into a string (unit: BYTE)

9.5. ST Program

9.5.1. Functional Components of the Program

This program is written in the ST language. The functional components are as follows:

Major classification	Minor classification	Description
1.Communications processing	1.1. Starting the communications processing 1.2. Clearing the communications processing status flags 1.3. Entering in the communications in progress status and shifting to each state processing	The communications processing starts.
2.Initialization processing	2.1. Initializing the communications instruction 2.2. Initializing the communications instruction execution flag 2.3. Initializing the error code storage areas 2.4. Setting the input variable of the ExecPMCR instruction 2.5. Initializing the receive data storage areas 2.6. Initialization setting end processing	The parameter setting of the communications instruction (ExecPMCR instruction) and the receive data storage areas are cleared.
3.PMCR communications processing	3.1. Determining the communications processing status and setting the execution flag 3.2. Executing the communications instruction	The ExecPMCR instruction is executed. A normal end or an error end of the execution is detected.
4.Processing number error process	-	The error processing is executed when a non-existent status processing number is detected.

9.5.2. Program List

The program is shown below.

- Program: Program0 (General-purpose serial communications connection check program)

1.Communications processing

```
(* =====
Name: NJ-series general-purpose serial (RS-232C) communications connection
      check program (Protocol macro)
Applicable device:
  Manufacturer: OMRON Corporation
  Device: Ultra Small Multi-code Reader
  Series/Model: V400-R2 Series
Serial Unit: CJ1W-SCU42 (Protocol macro, Unit number: 0, Serial port number: 2)
Version information: V1.00 February 8, 2013 New release
(C)Copyright OMRON Corporation 2013 All Rights Reserved.
===== *)
```

```
(* 1. Communications processing
  Communications start switch: Input_Start
  Communications processing status flags: Local_Status<STRUCT>
    .Busy: Communications in progress
    .Done: Communications normal end
    .Error: Communications error end
  State processing number: Local_State
    10: Initialization processing
    11: CompoWay/F communications processing *)
```

```
(* 1.1. Starting the communications processing
  Start communications processing when the communications start switch changes to ON
  when communications processing status flags have been cleared. *)
```

```
IF Input_Start AND
  NOT (Local_Status.Busy OR Local_Status.Done OR Local_Status.Error) THEN
  Local_Status.Busy:=TRUE;
  Local_State:=10; //10: Initialization processing
END_IF;
```

```
(* 1.2. Clearing the communications processing status flags
  Clear the communications processing status flags when the communications start switch
  changes to OFF while communications processing is not in progress. *)
```

```
IF NOT Input_Start AND NOT Local_Status.Busy THEN
  Local_Status.Done:=FALSE;
  Local_Status.Error:=FALSE;
END_IF;
```

```
(* 1.3. Entering the communications in progress status and shifting to each state processing *)
```

```
IF Local_Status.Busy THEN
  CASE Local_State OF
```

2.Initialization processing

To change the input values such as the unit number of the Serial Communications Unit, the port number to be used, protocol macro sequence number and send data size, change the code indicated by a red frame.

(* 2. Initialization processing

- Perform initialization for the whole communications and set the parameters.
- Set the send data and initialize the receive data storage areas. *)

10:

(* 2.1. Initializing the communications instruction *)

```
ExecPMCR_instance(
    Execute:=FALSE,
    SrcDat:=Input_SendData[0],
    DstDat:=Output_RecvData[0]);
```

(* 2.2. Initializing the communications instruction execution flag *)

```
Local_ExecFlg:=FALSE;
```

(* 2.3. Initializing the error code storage areas *)

```
Clear(Output_PmrErrorID);
Clear(Output_PmrErrorIDEx);
Clear(Output_PmrStaErrCode);
Clear(Output_TransErrCode);
```

(* 2.4. Setting the ExecPMCR instruction control data *)

```
Local_InPort.UnitNo:=_eUnitNo#_CBU_No00;
Local_InPort.PhysicPortNo:=USINT#2;
Local_SeqNo:=UINT#900;
Input_SendData[0]:=WORD#0;
```

(* 2.5. Initializing the receive data storage areas *)

```
Clear(Output_RecvData);
Clear(Local_RecvWordData);
Clear(Local_RecvByteData);
Clear(Output_RecvMessage);
```

(* 2.6. Initialization setting end processing *)

```
Local_State:=11; //11: PMCR communications processing
```


3. PMCR communications processing

(* 3. PMCR communications processing

-Execute the communications processing from the specified serial port. *)

11:

(* 3.1. Determining the communications processing status and setting the execution flag *)

(* 3.1.1. Normal/Error detection processing when protocol macro execution ends *)

F_TRIG_instance(J01_P2_PmrExecSta,Local_EndExecPMCR);

IF Local_EndExecPMCR THEN

 IF NOT J01_P2_PmrSeqEndSta THEN //Error end of protocol macro

 Output_TransErrCode:=J01_P2_TransErrSta;

 Local_Status.Error:=TRUE;

 ELSE

 //Normal end processing

 Output_TransErrCode:=WORD#16#0000;

 Local_Status.Done:=TRUE;

 //Convert the receive data from BYTE array to STRING

 Local_RecvWordSize:=WORD_TO_UINT(Output_RecvData[0])-1;

 MemCopy(Output_RecvData[1],Local_RecvWordData[0],Local_RecvWordSize);

 ToAryByte(Local_RecvWordData,_eBYTE_ORDER#_HIGH_LOW,Local_RecvByteData[0]);

 Local_RecvByteSize:=Local_RecvWordSize*2;

 Output_RecvMessage:=AryToString(Local_RecvByteData[0],Local_RecvByteSize);

 END_IF;

 Local_EndExecPMCR:=FALSE;

 Local_Status.Busy:=FALSE;

 Local_ExecFlg:=FALSE;

 Local_State:=0; //To 0: Communications not in progress status

(* 3.1.2. Error detection processing during protocol macro execution *)

ELSIF ExecPMCR_instance.Error THEN //ExecPMCR instruction error

 Output_PmrErrorID:=ExecPMCR_instance.ErrorID;

 Output_PmrErrorIDEx:=ExecPMCR_instance.ErrorIDEx;

 MoveDigit(J01_P2_PmrSta,USINT#0,Local_PmrStaCode,USINT#0,USINT#1);

 Output_PmrStaErrCode:=Local_PmrStaCode;

 Output_TransErrCode:=J01_P2_TransErrSta;

 Local_Status.Busy:=FALSE;

 Local_Status.Error:=TRUE;

 Local_ExecFlg:=FALSE;

 Local_State:=0; //To 0: Communications not in progress status

(* 3.1.3. Setting the communications instruction execution flag *)

ELSIF _Port_isAvailable AND

 NOT J01_P2_PmrExecSta AND

 NOT ExecPMCR_instance.Busy THEN

 Local_ExecFlg:=TRUE;

END_IF;

(* 3.2. Executing the communications instruction *)

ExecPMCR_instance(

 Execute:=Local_ExecFlg,

 Port:=Local_InPort,

 SeqNo:=Local_SeqNo,

 SrcDat:=Input_SendData[0],

 DstDat:=Output_RecvData[0]);

4.Processing number error process

```
(* 4. Processing number error process
-Error process for nonexistent processing number *)
99:
  Output_TransErrCode:=WORD#16#0010;
  Local_Status.Busy:=FALSE;
  Local_Status.Error:=TRUE;
  Local_State:=0; //0: Communications not in progress status

ELSE
  Local_State:=99; //To 99: Processing number error process

END_CASE;

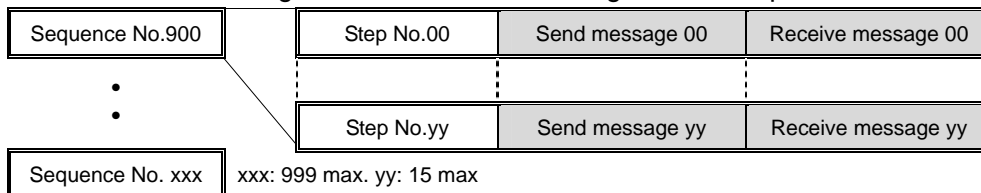
END_IF;
```

9.6. Protocol macro data

Protocol macro data consists of sequence, step, send/receive message, and matrix. Its composition is described as follows.

- When there is only one receive message format for a step (send/receive once)

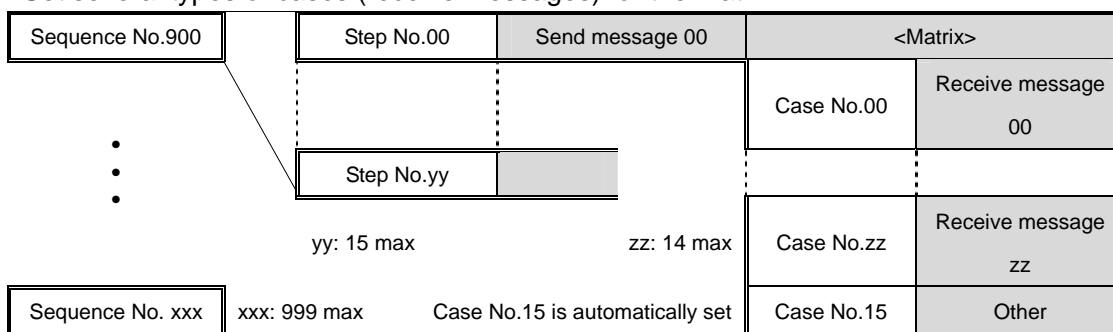
- Set one send message and one receive message for the step



- When there are several types of receive message formats for a step (send/receive once)

- Set the send message and matrix for the step

- Set several types of cases (receive messages) for the matrix



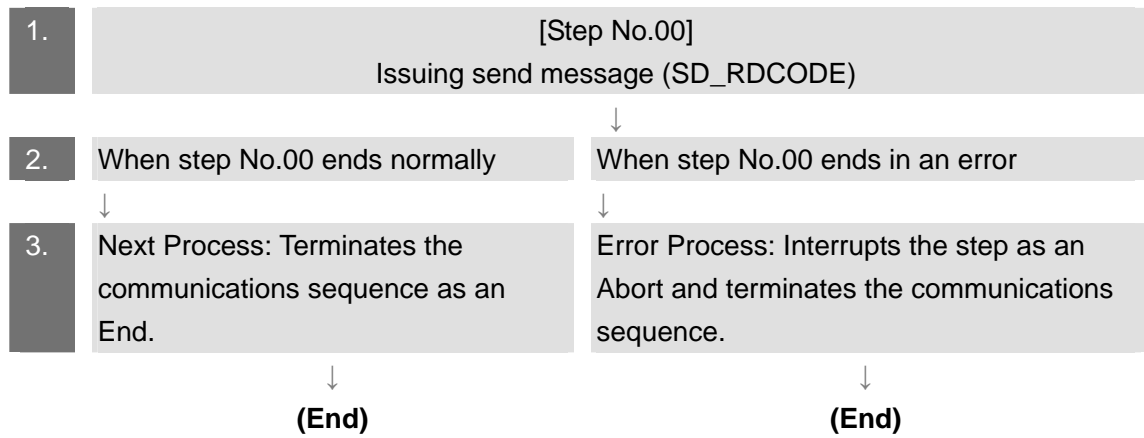
9.6.1. Composition of Protocol Macro Data

In this protocol macro data, there is only 1 type of receive message (SD_RDCODE) for the send message (SD_RDCODE). Therefore, the following composition is used without using the matrix. Refer to 9.6.6. *Receive Message Settings* for details.



9.6.2. Protocol Macro Processing Procedure

This section describes the processing procedure of the protocol macro.



9.6.3. Sequence Settings

This protocol macro data performs Reading Trigger (read code) by using communications sequence No. 900. Set the timeout periods for the communications sequence.



Additional Information

Refer to *3-2 Sequence Attributes (Common to All Steps) of the CX-Protocol Operation Manual* (Cat. No. W344) for details on sequence settings.

- Timeout period setting

The following is the settings of the timeout periods (Timer Tr, Tfr, and Tfs) which are set for the sequence.

[Communications sequence setting screen]

#	Communication Sequence	Link Word	Control	Response	Timer Tr	Timer Tfr	Timer Tfs
900	New Sequence	---	Set	Scan	3 sec	3 sec	3 sec

<Settings>

Item	Description	Explanation
Timer Tr	Receive wait monitoring time	Monitors the time from the receive wait status to the reception of the first data (header) in the step of the sequence. This timer is set to 3 seconds in this protocol macro data.
Timer Tfr	Receive finished monitoring time	Monitors the time from the reception of the first data to the completion of the reception in the step of the sequence. This timer is set to 3 seconds in this protocol macro data.
Timer Tfs	Send finished monitoring time	Monitors the time from the sending of the header to the sending of the last data. This timer is set to 3 seconds in this protocol macro data.



Additional Information

Refer to *Section 4-5 Calculation Method of Monitoring Time of the CX-Protocol Operation Manual* (Cat. No. W344) for the calculation method of monitoring time.

9.6.4. Step Settings

This section describes the step settings for communications sequence No. 900. The settings include retry count, send/receive messages (message names), next process, and error process. The sequence of this protocol macro data includes Step No.00 only.



Additional Information

Refer to 3-3 Step Attributes of the *CX-Protocol Operation Manual* (Cat. No. W344) for details on step settings.

•Retry count setting

This section describes the retry count setting for the step. The step is retried for the specified number of times (0 to 9 times) when an error occurs. If an error occurs after retries, the step moves to the error process.

The retry count is enabled for the Send&Receive command only.

<Step setting screen>

* Step	Repeat	Command	Retry	Send Wait	Send Message	Recv Message	Response	Next	Error
00	RSET/001	Send & Receive	3	---	SD_RDCODE	RV_RDCODE	YES	End	Abort

<Settings>

Step No.	Retry count
00	3

•Send/Receive message (message name) settings

This section describes the settings for the send/receive messages of the step. Here, a registered send message name and message name are selected.

<Step setting screen>

* Step	Repeat	Command	Retry	Send Wait	Send Message	Recv Message	Response	Next	Error
00	RSET/001	Send & Receive	3	---	SD_RDCODE	RV_RDCODE	YES	End	Abort

<Settings>

Step No.	Send message	Receive message
00	SD_RDCODE	RV_RDCODE

- Next process and error process settings

This section describes the settings for the next process and error process of the step. The process set in the Next Column is executed when the step execution ends normally. If a communications error occurs, the process set in the Error Column is executed.

<Step setting screen>

* Step	Repeat	Command	Retry	Send Wait	Send Message	Recv Message	Response	Next	Error
00	RSET/001	Send & Receive	3	---	SD_RDCODE	RV_RDCODE	YES	End	Abort

<Settings>

Step No.	Next process	Error process
00	End	Abort

<Process list>

Process	Description
End	Ends the communications sequence
Next	Moves to the next step No
Abort	Interrupts the step and ends the communications sequence
Goto	Moves to the specified step number
Matrix	Uses the settings of the matrix

9.6.5. Send Message Settings

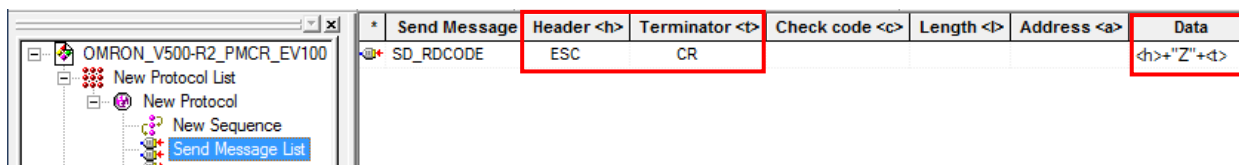
This section explains the settings of the send message.



Additional Information

Refer to 3-4 Communication Message Attributes of the CX-Protocol Operation Manual (Cat. No. W344) for details on send message settings.

<Send message setting screen>



•Settings of SD_RDCODE send message

<Settings>

<h>+\"Z\"+<t>
(1) (2) (3)

No.	Code	Description
(1)	<h >(Header)	Type: Code, Data: ESC
(2)	"Z"	Constant ASCII
(3)	<t>(Terminator *)	Type: Code, Data: CR

*The footer for the destination device is the terminator for the protocol macro data. Hereinafter, the term "terminator" is used.

<Send message command format>

This is the command format of the message that is sent from the Serial Communications Unit to the destination device according to the settings of SD_RDCODE.

<ESC>	"Z"	<CR>
-------	-----	------

Command	Number of bytes	Remarks
Header	1	Fixed: <ESC> (16#1B)
"Z"	1	Fixed: "Z" (16#5A) (Reading Trigger (destination device command))
Terminator *	1	Fixed: <CR> (16#0D)

9.6.6. Receive Message Settings

This section describes the settings of the receive message.



Additional Information

Refer to 3-4 *Communication Message Attributes* of the *CX-Protocol Operation Manual* (Cat. No. W344) for details on receive message settings.

[Receive message setting screen]



•Setting of receive message RV_RDCODE

<Settings>

(W(1),*)+<t>

(1) (2)

No.	Code	Description
(1)	(W(1),*)	Message data: Variable All variable length data are read and stored from array variable [n], which is specified with <i>DstDat</i> of the ExecPMCR instruction, + [1]. (With this program, data is stored from <i>Output_RecvData[1]</i> .)
(2)	<t>(Terminator)	Message data: Constant hex, Type: Code, Data: CR

*The header <h> is set to None.

<Response format of receive message>

This is the response format of the receive message which is received by the Serial Communications Unit from the destination device according to the settings of RV_RDCODE.

Read code value	<CR>
-----------------	------

Command	Number of bytes	Remarks
Data	Variable length	Variable: Read code value
Terminator	1	Fixed: <CR> (16#0D)

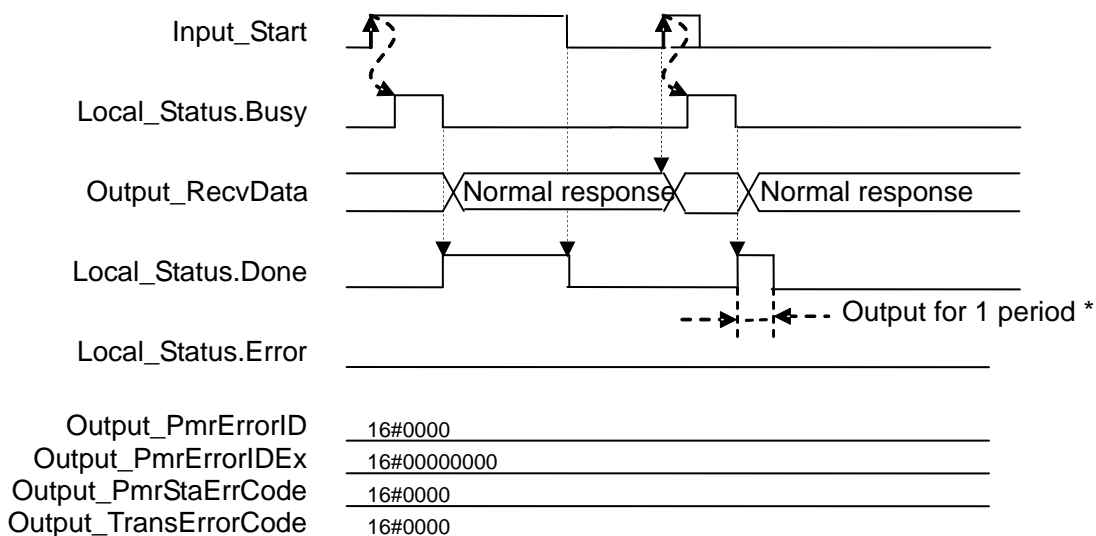
9.7. Timing Charts

The timing charts of the program are shown below.

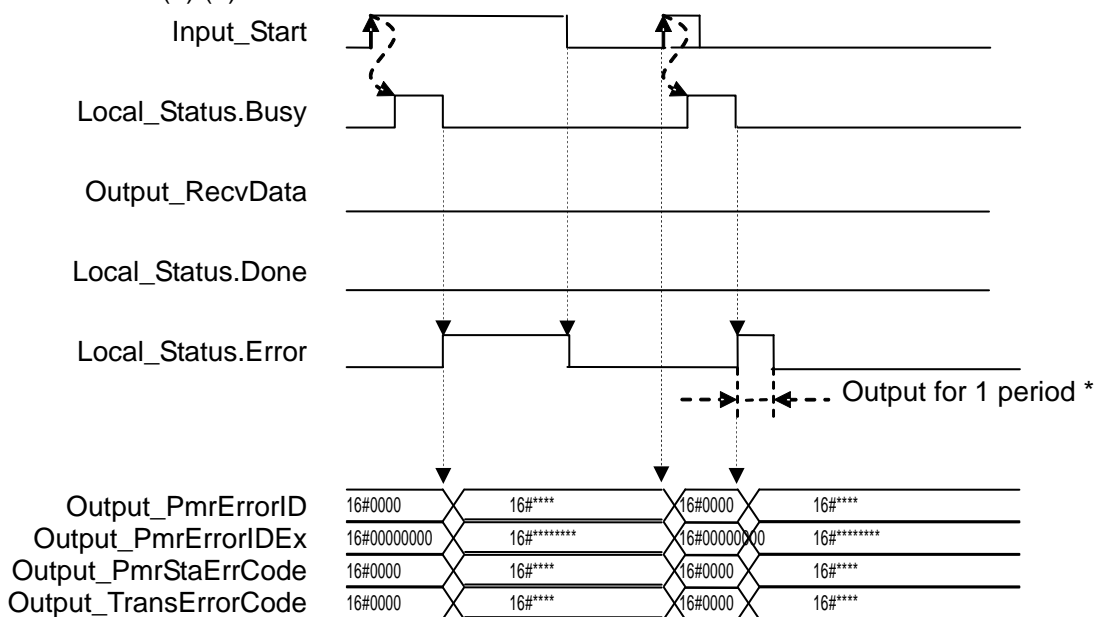
The definitions of the timing chart patterns are as follows.

Pattern	Normal end	Error end (1) ExecPMCR instruction error	Error end (2) Protocol macro error
Command	Normal	Error	Error
Destination device	Normal	Normal or error	Normal or error
Response	Yes	None	None

●Normal end



●Error end (1) (2)



*If *Input_Start* changes from TRUE to FALSE during an execution (Busy=TRUE), a normal end or an error end is output for one period after the processing is completed (Busy=FALSE).

9.8. Error processing

The following tables list the errors that are generated by executing this program.

9.8.1. ExecPMCR Instruction Errors

The error codes generated when the ExecPMCR instruction ends in an error are given below.

•Output_PmrErrorID

After the ExecPMCR is executed, the content of *ExecPMCR_instance.ErrorID* is set.

Code	Description
16#0000	Normal end
16#0400	An input parameter for an instruction exceeded the valid range for an input variable.
16#0406	The memory address or data size of the data specified in the instruction is incorrect.
16#0407	The results of instruction processing exceeded the data area range of the output parameter.
16#0800	An error occurred when a FINS command was sent or received.
16#0801	The FINS port is being used.



Additional Information

For details on ErrorID and error codes not listed in the table above, refer to *A-2 Error Code Descriptions* and *A-3 Error Code Details* in *Appendices of the NJ-series Instructions Reference Manual* (Cat. No. W502).

•Output_PmrErrorIDEx

The content of *ExecPMCR_instance.ErrorIDEx* is set when the value of *ExecPMCR_instance.ErrorID* is 16#0800.

Code	Description
16#00000000	Normal end
16#00001106	A communications sequence number that is not registered was specified.
16#00002201	The operation was not possible because a protocol macro is already in execution.
16#00002202	The operating mode is wrong.
16#00002401	A checksum error occurred in the protocol macro data or the data transfer is not yet completed.



Additional Information

For details on ErrorIDEx and error codes not listed in the table above, refer to *Serial Communications instructions (ExecPMCR)* in *Section 2 Instruction Descriptions* of the *NJ-series Instructions Reference Manual* (Cat. No. W502).

9.8.2. Protocol Macro Error Codes

The protocol macro error codes generated when a protocol macro related error occurs are given below.

These codes are set in *Output_PmrStaErrCode*.

- *Output_PmrStaErrCode*

The contents of the lower 4-bit of *J01_P2_PmrSta* are set.

Code	Description
16#0000	There are no errors
16#0002	Sequence No. error
16#0003	Symbol specification area exceeded error
16#0004	Protocol macro syntax error



Additional Information

For details on the protocol macro error codes (*PmrStaErrCode*), refer to 4-2-6 *Protocol Status* of the *CJ-series Serial Communications Units Operation Manual for NJ-series CPU Unit* (Cat. No. W494).

9.8.3. Transmission error status

The error flags used when a transmission error occurs are shown below.

They are set in *Output_TransErrCode*.

- *Output_TransErrCode*

The content of *J01_P2_TransErrSta* is set.

Bit	Description
15	1:Transmission error 0:No transmission error
14	1:Send Finished Monitoring Time Exceeded 0:Normal
13	1:Receive Finished Monitoring Time Exceeded 0:Normal
12	1:Receive Wait Monitoring Time Exceeded 0:Normal
8 to 11	Retry count
7	1:FCS Check Error 0:Normal
6	1:Command Error 0:Normal
5	1:Timeout 0:Normal
4	1:Overrun error 0:Normal
3	1:Framing error 0:Normal
2	1:Parity error 0:Normal
0 and 1	(Not used)



Additional Information

For details on the transmission error status (*TransErrCode*), refer to 4-2-5 *Device Variables for CJ-series Unit for Status* of the *CJ-series Serial Communications Units Operation Manual for NJ-series CPU Unit* (Cat.No. W494).

10. Revision History

Revision code	Date of revision	Revision reason and revision page
01	Sep. 3, 2013	First edition

OMRON Corporation Industrial Automation Company

Tokyo, 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

One Commerce Drive Schaumburg,
IL 60173-5302 U.S.A.

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

OMRON ASIA PACIFIC PTE. LTD.

No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967

Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON (CHINA) CO., LTD.

Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China

Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:

© OMRON Corporation 2013 All Rights Reserved.
In the interest of product improvement,
specifications are subject to change without notice.

Cat. No. P567-E1-01

0911(-)