OMRON

Machine Automation Controller NJ-series

General-purpose Serial Connection Guide (RS-485 CompoWay/F) OMRON Corporation

Digital Controller
(E5CC/E5EC/E5AC)

Network

Connection

Guide

sysmac
always in control

About Intellectual Property Right 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.

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

Table of Contents

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-[][][][]	NJ-series CPU Unit Hardware User's Manual
W501	NJ501-[][][][]	NJ-series CPU Unit Software User's Manual
W494	CJ1W-SCU[]2	CJ-series Serial Communications Units Operation Manual for NJ-series CPU Unit
W502	NJ501-[][][][]	NJ-series Instructions Reference Manual
W504	SYSMAC-SE2[][][]	Sysmac Studio Version 1 Operation Manual
H175	E5CC/E5EC/E5AC	Digital Temperature Controllers Communications Manual
H174	E5CC/E5EC/E5AC	Digital Temperature Controllers User's Manual

2. Terms and Definition

Terms	Explanation and Definition
Serial gateway mode	The received message is automatically converted to CompoWay/F, Modbus-RTU, or Modbus-ASCII by the SCU Unit, depending on the type of message.

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 for 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 users are 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 of or whole part of this document without the permission of OMRON Corporation.

(5) This document provides the latest information as of February 2012. The information contained in this document is subject to change for improvement without notice.

The following notation is used in this document.

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

Precautions for Safe Use

Indicates precautions on what to do and what not to do to ensure using the product safely.

Precautions for Correct Use

Indicates precautions on what to do and what not to do to ensure proper operation and performance.

Additional Information

Provides useful information.
Additional information to increase understanding or make operation easier.

Symbols

The circle and slash symbol indicates operations that you must not do.
The specific operation is shown in the circle and explained in text.
This example indicates prohibiting disassembly.

The triangle symbol indicates precautions (including warnings).
The specific operation is shown in the triangle and explained in text.
This example indicates a precaution for electric shock.

The triangle symbol indicates precautions (including warnings).
The specific operation is shown in the triangle and explained in text.
This example indicates a general precaution.

The filled circle symbol indicates operations that you must do.
The specific operation is shown in the circle and explained in text.
This example shows a general precaution for something that you must do.

4. Overview

This document describes the procedure for connecting the Digital Controller (E5CC/E5EC/E5AC) of OMRON Corporation (hereinafter referred to as OMRON) with the NJ-series Machine Automation Controller (hereinafter referred to as Controller) via serial communications, and the procedure for checking their connection.

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

This project file is used to check a serial connection by sending the CompoWay/F command to the destination device.

Obtain the latest "Sysmac Studio project file" from OMRON beforehand.

Name	File name	Version
Sysmac Studio project file (extension: SMC)	OMRON_CompoWayF485_EV100.SMC	Ver.1.00

5. Applicable Devices and Support Software

5.1. Applicable Devices

The following devices can be connected.

Manufacturer	Meaning	Model	Version
OMRON	NJ-series CPU Unit	NJ501-[][][][]	Versions
OMRON	Serial Communications Unit	CJ1W-SCU[]2	listed in
OMRON	Digital Controller	E5CC-[]-003 E5CC/EC/AC-[]-004 E5EC/AC-[]-009 E5EC/AC-[]-014	Section 5.2 and higher versions

📝 **Additional Information**

As applicable devices above, the devices listed in Section 5.2. are actually used in this document to check the connection. When using devices not listed in Section 5.2, check the connection by referring to the procedure in this document.

Precautions for Correct Use

You can connect devices with the versions listed in Section 5.2 or higher versions.

For devices whose versions are not listed in Section 5.2, versions are not managed or there is no version restriction.

To connect a device whose model number is not listed in Section 5.2, use the same version of the device that is listed.
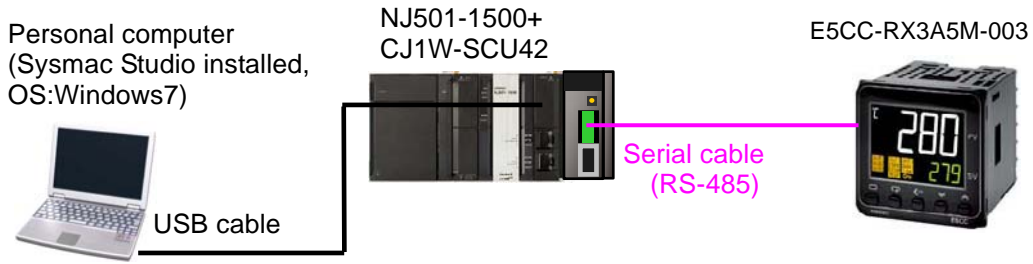
Additional Information

This document describes the procedure to establish the network connection. It does not provide information about operation, installation nor wiring method of each device.

For details on above products (other than communication connection procedures), refer to the manuals for the corresponding products or contact your OMRON representative.

5.2. Device Configuration

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

Personal computer (Sysmac Studio installed, OS:Windows7)

NJ501-1500+
CJ1W-SCU42

E5CC-RX3A5M-003

Serial cable
(RS-485)

USB cable

Manufacturer	Name	Model	Version
OMRON	Serial Communications Unit	CJ1W-SCU42	Ver.2.0
OMRON	NJ-series CPU Unit	NJ501-1500	
OMRON	Power Supply Unit	NJ-PA3001	
OMRON	Sysmac Studio	SYSMAC-SE2[][][]	Ver.1.00
OMRON	Sysmac Studio project file	OMRON_CompoWayF485_EV100.SMC	Ver.1.00
-	Personal computer (OS:Windows7)	-	
-	USB cable (USB 2.0 type B connector)	-	
-	Serial cable (RS-485)	-	
OMRON	Digital Controller	E5CC-RX3A5M-003	

Precautions for Correct Use

Obtain the latest Sysmac Studio project file from OMRON in advance.
(To obtain the files, contact your OMRON representative.)

Additional Information

It may not be possible to reproduce the same operation with different devices and versions. Check the configuration, models and versions. If your configuration differs from the ones above, contact your OMRON representative.

Additional Information

For information on the serial cable (RS-485), refer to *3-3 RS-232C and RS-422A/485 Wiring* in the *CJ-series Serial Communications Units Operation Manual for NJ-series CPU Unit* (Cat.No. W494).

Additional Information

Update the Sysmac Studio to the version specified in this section or higher version 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).

Additional Information

In this document, a USB is used to connect with 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 provides specifications such as communications parameters and cable wiring that are set in this document.

📓 **Additional Information**

This document and project file can be used to perform operations using the settings and command described in this section. Modifications are necessary to perform communications using different settings.

6.1. Serial Communications Settings

The table below lists the settings for serial communications.

	CJ1W-SCU42	E5CC-RX3A5M-003
Unit number	0	-
Communications Unit No. (Slave address)	-	1 (default value)
Communications (connection) port	Port 1 (RS-422/485)	-
TERM (Terminating resistance ON/OFF switch)	ON (Terminating resistance ON)	-
WIRE (2-wire/4-wire switch)	2 (2-wire)	2-wire (fixed)
Serial communications mode	Serial gateway	-
Data length (transmission character)	7 bits (default value)	7 bits (default value)
Stop bit	2 bits (default value)	2 bits (default value)
Parity (parity bit)	Even (default value)	Even (default value)
Baud rate	9,600 bps (default value)	9,600 bps (default value)
Communications method	-	CompoWay/F (default value)
Send data wait time	-	20 ms (default)

☝ **Precautions for Correct Use**

This manual describes the procedure for setting the CJ1W-SCU42 Serial Communications Unit when the unit number 0, communications port 1 and device name J01 are used. To connect devices under different conditions, refer to *9. Project File* and create an ST program by changing the variable names and setting values.

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 for wiring.

■Connector configuration and pin assignment
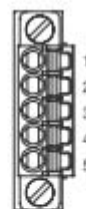
<E5CC/E5EC/E5AC> Applicable connector: Terminal block

Pin No.	Signal name	I/O
1 to 12		
13	B(+)	I/O
14	A(-)	I/O
15-18		
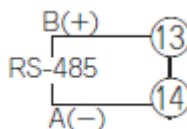
<OMRON CJ1W-SCU42> Applicable connector: Terminal block

Pin No.	Symbol	Signal name	I/O
1 (See note 1.)	RDA	Receive data –	Input
2 (See note 1.)	RDB	Receive data +	Input
3 (See note 1.)	SDA	Send data –	Output
4 (See note 1.)	SDB	Send data +	Output
5 (See note 2.)	FG	Shield	---

Note   1.   For 2-wire connections, use either pins 1 and 2 or pins 3 and 4.

2.   Pin 5 (the shield) is connected to the GR terminal on the Power Supply Unit though the Serial Communications Unit. The cable shield can thus be grounded by grounding the GR terminal of the Power Supply Unit.

■Cable/Pin arrangement

Serial Communications Unit (CJ1W-SCU42)			Digital Controller (E5CC/E5EC/E5AC)		
RS-422A /485 interface	Signal name	Pin No.	Terminal number	Signal name	RS-485 interface
	RDA-	1	13	B(+)	
	RDB+	2	14	A(-)	
	SDA-	3			
	SDB+	4			
	FG	5			
Terminal block connector			Terminal block		
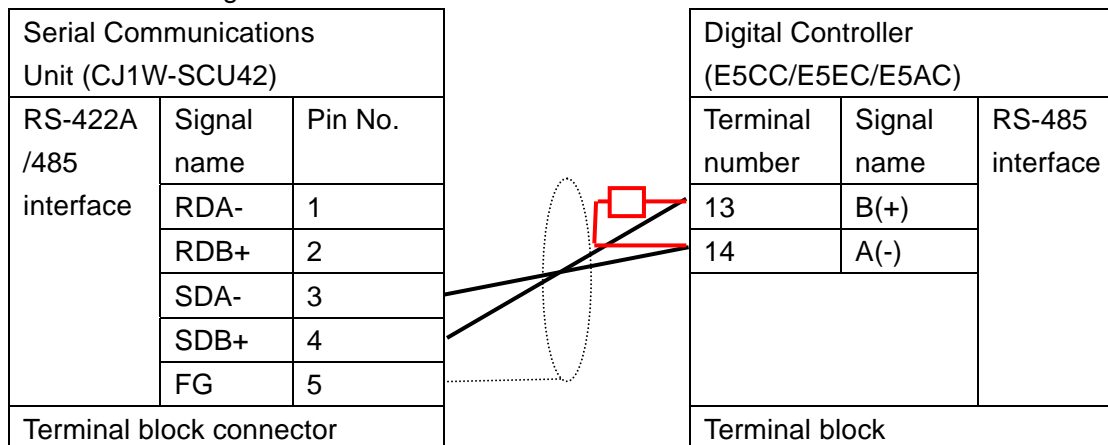
Precautions for Correct Use

Turn ON the terminating resistance switch of the Serial Communications Unit, and connect the terminating resistance of 120Ω (1/2W) to the Digital Controller connected at the end of the network by referring to the pin arrangement above.

6.3. Example of Checking Connection

This document shows an example of an ST (structured text) program in which the controller sends/receives CompoWay/F messages to/from Digital Controller.

The Controller and Digital Controller send and receive the "Unit Properties Read" messages.

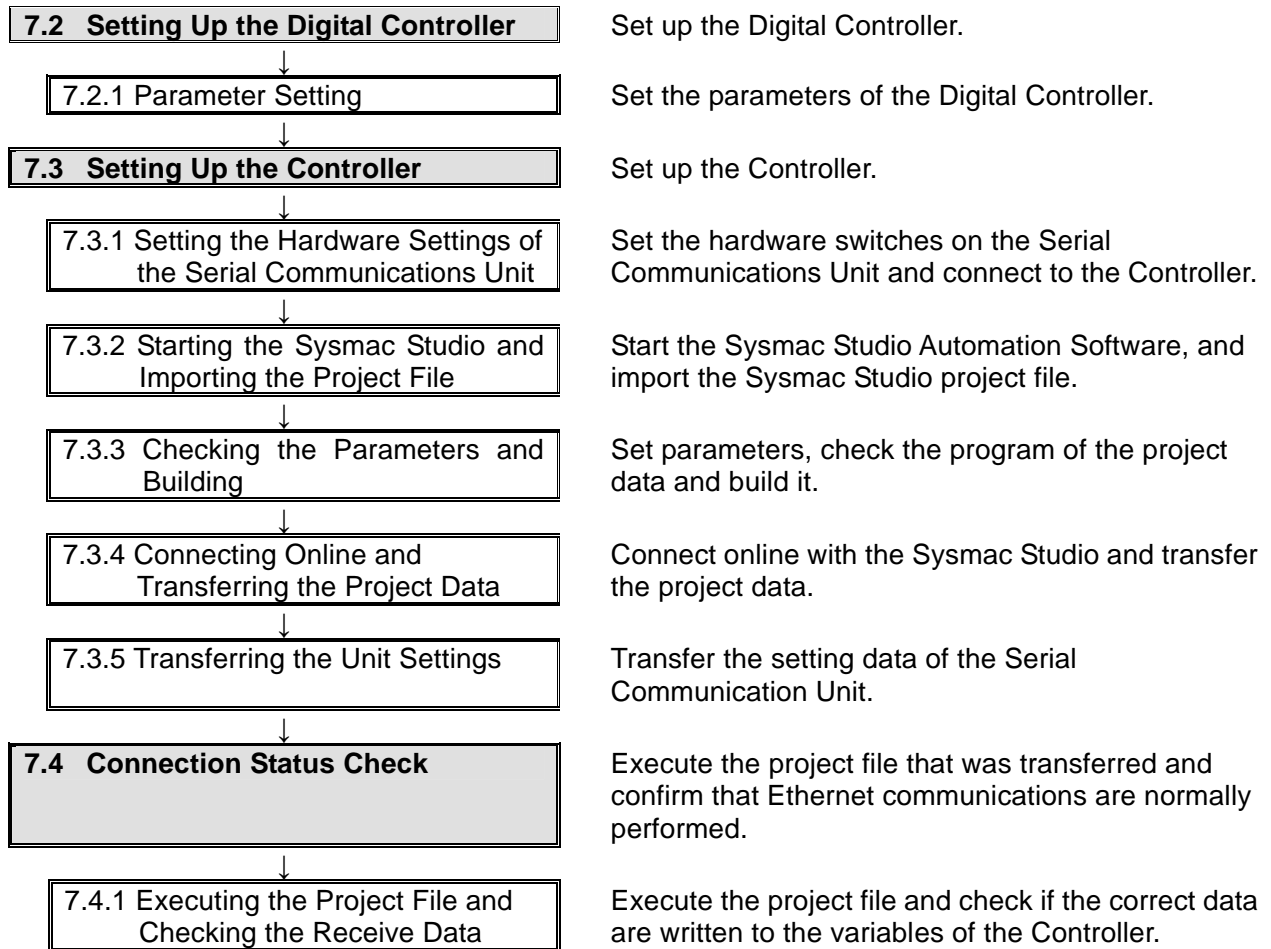The following figure shows the outline of the operation.
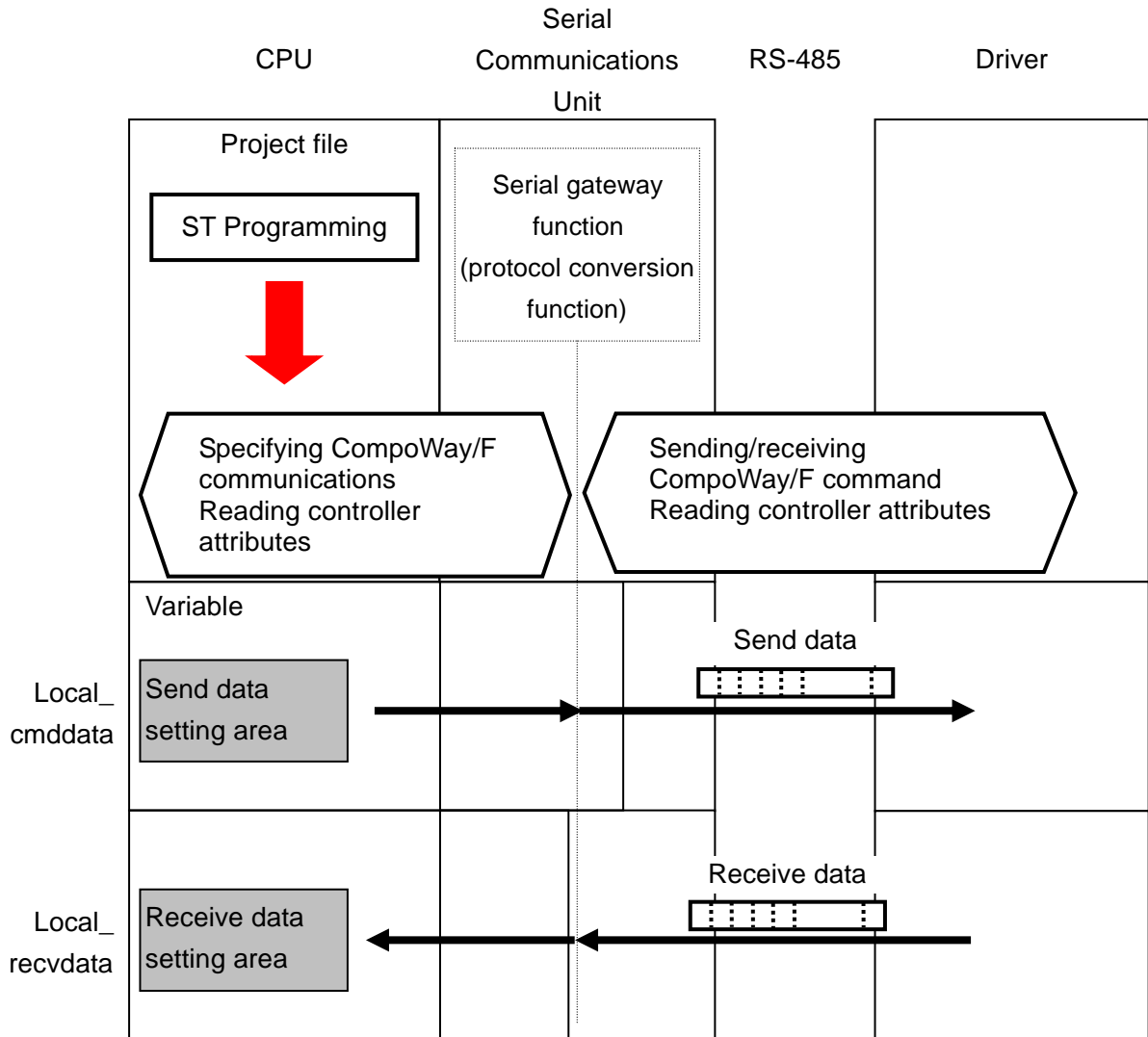
7. Connection Procedure

This section describes how to connect the Controller via serial communications.
This document explains the procedures for setting up the Controller and Digital Controller 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 Controller via serial communications.

7.2 Setting Up the Digital Controller	Set up the Digital Controller.
↓	
7.2.1 Parameter Setting	Set the parameters of the Digital Controller.
↓	
7.3 Setting Up the Controller	Set up the Controller.
↓	
7.3.1 Setting the Hardware Settings of the Serial Communications Unit	Set the hardware switches on the Serial Communications Unit and connect to the Controller.
↓	
7.3.2 Starting the Sysmac Studio and Importing the Project File	Start the Sysmac Studio Automation Software, and import the Sysmac Studio project file.
↓	
7.3.3 Checking the Parameters and Building	Set parameters, check the program of the project data and build it.
↓	
7.3.4 Connecting Online and Transferring the Project Data	Connect online with the Sysmac Studio and transfer the project data.
↓	
7.3.5 Transferring the Unit Settings	Transfer the setting data of the Serial Communication Unit.
↓	
7.4 Connection Status Check	Execute the project file that was transferred and confirm that Ethernet communications are normally performed.
↓	
7.4.1 Executing the Project File and Checking the Receive Data	Execute the project file and check if the correct data are written to the variables of the Controller.

Precautions for Correct Use

Obtain the latest Sysmac Studio project file from OMRON in advance.
(To obtain the files, contact your OMRON representative.)
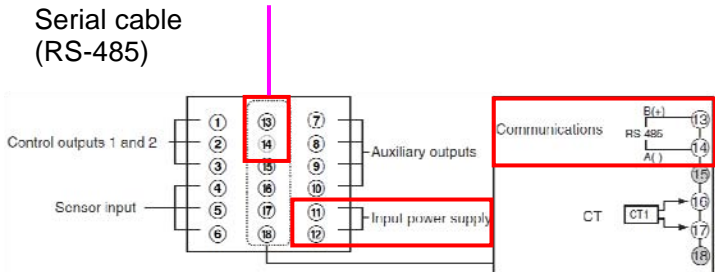
7.2. Setting Up the Digital Controller

Set up the Digital Controller.

7.2.1. Parameter Setting

Set the parameters for the Digital Controller.

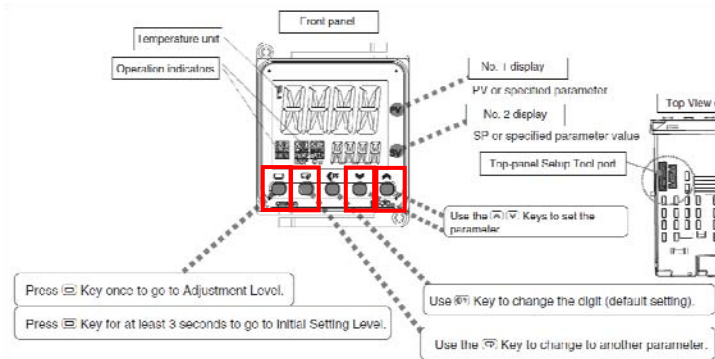|		
1	Connect the power supply and serial cable to the terminal block that is on the back surface of the Digital Controller. *This document describes the communications functions only. When using the control functions of each device, refer to the manuals for each unit.	Serial cable (RS-485)
2	Refer to the right figure and check the keys, No. 1 and No. 2 displays and operation indicators of the Digital Controller. In this document, the following symbols are used to indicate the keys. □: Level key ▭: Mode key ⌃: Up key ⌄: Down key Turn ON the power to the Digital Controller.	
3	When the power to the Digital Controller is turned ON, the No.1 display shows the present temperature.(Operation level) Press the □ (Level) Key for at least 3 seconds.	 Press the □ (Level) Key for at least 3 seconds.

4	The No. 1 display shows *ĪN-t* (Initial setting level). Press the ☐ (Level) Key again.		
			☐ (Level) Key
5	The Communication Setting Level is entered. The No. 1 display shows *PSEL* (protocol setting) and the No. 2 display shows *CWF* (CompoWay/F). *If the setting values are different, change the parameters using the ⌃ (Up) key and ⌄ (Down) key. Press the ⎆ (Mode) Key.	<Setting value> CWF/Mod (Default value: CWF) CWF:CompoWay/F Mod:Modbus-RTU	
			⎆ (Mode) Key
6	*U-Nō* (Communications Unit No.) is displayed. Check that the communications Unit No. is *1*. *If the setting value is different, change the value using the same procedure as step 5. Press the ⎆ (Mode) Key.	<Setting value> 0 to 99 (default value: 1)	
			⎆ (Mode) Key
7	*bPS* (Baud rate) is displayed. Check that the communications baud rate is *9.6* kbps. *If the setting value is different, change the value using the same procedure as step 5. Press the ⎆ (Mode) Key.	<Setting value> 9.6/19.2/38.4/57.6 kbps (default value: 9.6)	
			⎆ (Mode) Key

15

8	*LEN* (Communications data length) is displayed. Check that the communications data length is 7 bits. *If the setting value is different, change the value using the same procedure as step 5. Press the ⌨ (Mode) Key.		\<Setting value\> 7/8 bits (default value: 7)
		⌨ (Mode) Key	
9	*Sbit* (Communications stop bits) is displayed. Check that the communications stop bits are *2* bits. *If the setting value is different, change the value using the same procedure as step 5. Press the ⌨ (Mode) Key.		\<Setting value\> 1/2 bits (default value: 2)
		⌨ (Mode) Key	
10	*PRtY* (Communications parity) is displayed. Check that the communications parity is *EVEN*. *If the setting value is different, change the value using the same procedure as step 5. Press the ⌨ (Mode) Key.		\<Setting value\> None/Even/Odd (Default value: Even)
		⌨ (Mode) Key	
11	*Sdwt* (send data wait time) is displayed. Check that the send data wait time is *20*. *If the setting value is different, change the value using the same procedure as step 5. Press the ☐ (Level) Key.		\<Setting value\> 0 to 99 ms (default value: 20)
		☐ (Level) Key	

16

12	*ᴉN-ᵗ* (initial setting level) is displayed. Press the ☐ (Level) Key for at least 1 second.	
		Press the ☐ (Level) Key for at least 1 second.
13	The display is returned to the status of step 3 (Operation level). *If you changed the setting values, cycle the power supply so that the settings values take effect.	

7.3. Setting Up the Controller

Set up the Controller.

7.3.1. Setting the Hardware Settings of the Serial Communications Unit

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 settings.
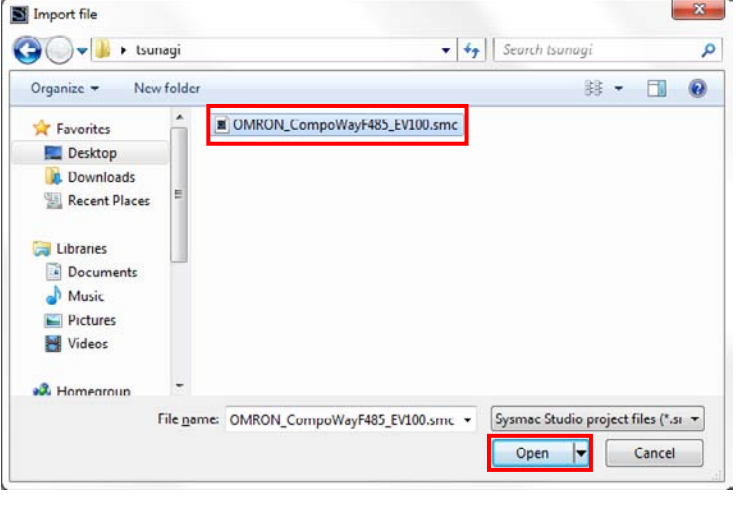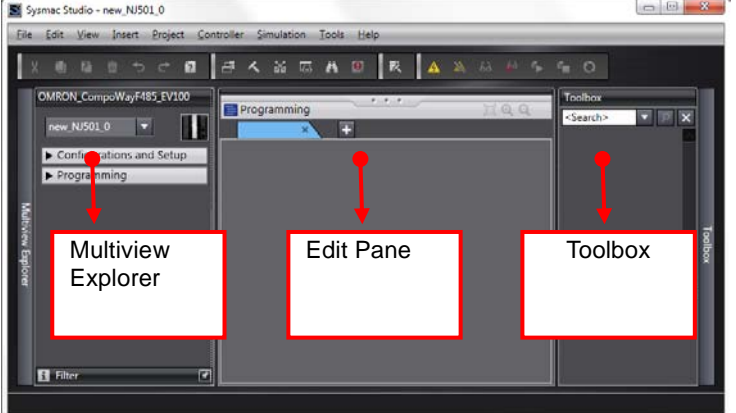
1	Make sure that the power supply to the Controller is OFF when you perform settings.	*If the power supply is turned ON, the following procedure may not be applicable.
2	Connect the serial cable (RS-485) to Port 1 terminal block connector. *This setting is required to use the Port 1 of Serial Communications Unit.	
3	Set the Unit No. Switch to 0.	
4	Set the terminating resistance ON/OFF switch for port 1 to ON (terminating resistance ON).	TERMTerminating resistance ON/OFF switch OFF:Terminating resistance OFF [ON]:Terminating resistance ON
5	Set the 2-wire/4-wire selector switch for port 1 to 2 (2-wire).	WIRE:2-wire/4-wire switch 2:2-wire;4:4-wire

18

| 6 | Connect the Serial Communications Unit and the End Cover to the Controller. Connect the personal computer, Digital Controller and Controller using the serial cable and USB cable as shown in 5.2 Device Configuration.
Turn ON the power supply to the Controller. | NJ501-1500   CJ1W-SCU42
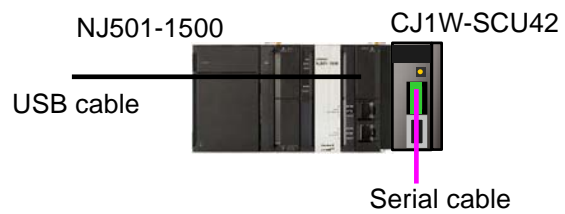
USB cable

Serial cable |

7.3.2. **Starting the Sysmac Studio and Importing the Project File**
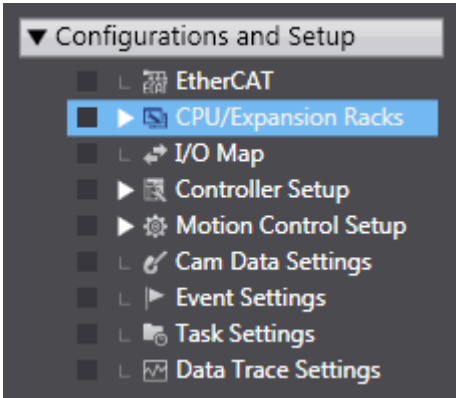
Start the Sysmac Studio Automation Software, and import the Sysmac Studio project file.

Install the software and USB driver beforehand.

1	Start the Sysmac Studio. Click the **Import** Button. *If a dialog box is displayed at start confirming the access right, select an option to start.	
2	The Import file Dialog Box is displayed. Select OMRON_CompoWayF485_EV 100.SMC (Sysmac Studio project file) and click the **Open** Button. *Obtain the Sysmac Studio project file from OMRON.	
3	The OMRON_CompoWayF485_EV 100 project screen is displayed. The left pane is called Multiview Explorer, the right pane is called Toolbox and the middle pane is called Edit Pane.	

20
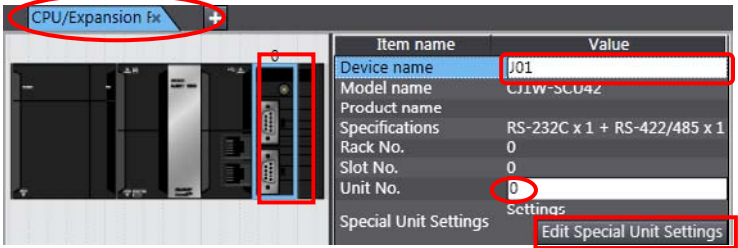
7.3.3. Setting the Parameters and Building

Set the parameters, check the program of the project data and build it.
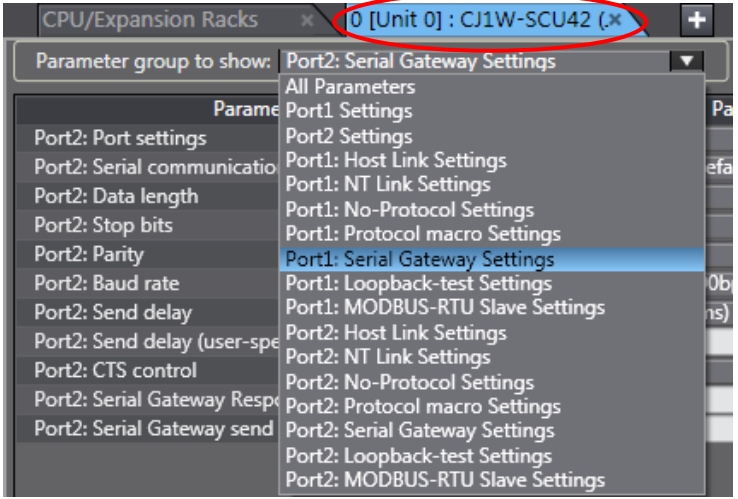
1	Double-click **CPU/Expansion Racks** under Configurations and Setup in the Multiview Explorer.	
2	The CPU/Expansion Racks Tab is displayed on the Edit Pane. Select the Serial Communications Unit figure as shown on the right. Check that CJ1W-SCU42 is displayed, the device name is J01, and the unit number is 0. *If the setting is different, change the value. Click **Edit Special Unit Settings**.	
3	The 0 [Unit 0]: Tab is displayed. Open the pull-down menu of Parameter group to show and select Port1: Serial Gateway Settings.	

4	Parameter group to show is set to Port1: Serial Gateway Settings. The setting items for Port 1: Serial Gateway Settings are shown. Check that the Port1: Port settings are set to User settings and Port 1: Serial communications mode is set to Serial Gateway. *If the settings are different from the above, change the values from the pull-down menu.	
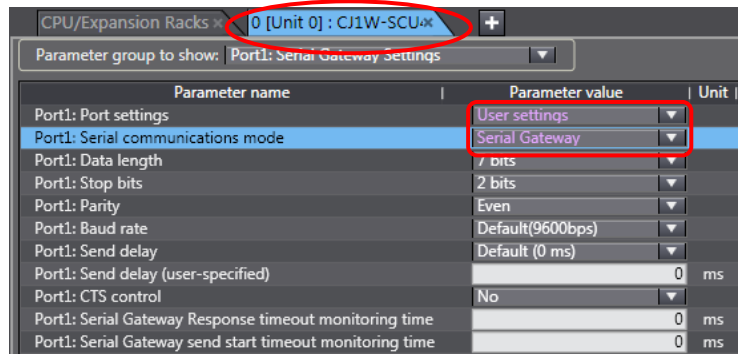
5	Change the other items to the following settings so that they are the same as for 6.1.
	
	    Data length: 7 bits
	    Stop bits: 2 bits
	    Parity: Even
	    Baud rate: Default value (9600 bps)
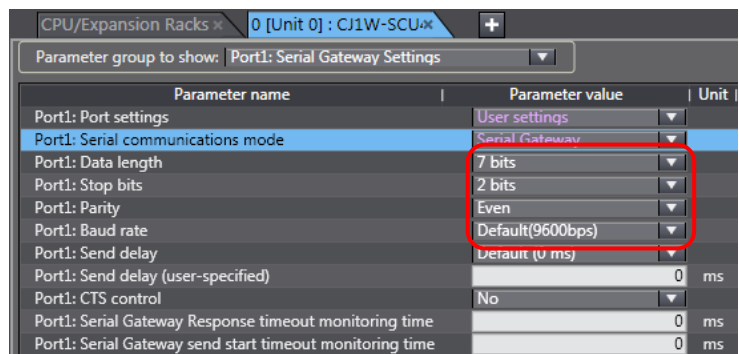	
	Click the **Apply** Button after changing values.	
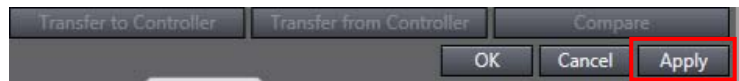
6	Double-click **I/O Map** under Configurations and Setup on the Multiview Explorer. The I/O Map Tab is displayed and then the parameters for the unit are listed.	

22
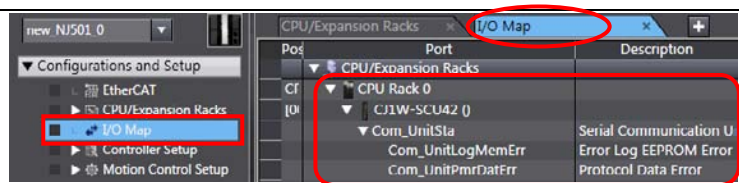
7	Check that data in the Variable Columns start with J01 and the Global Variable is set in each Variable Type Column.	
	*If the settings are different from the above, right-click on CJ1W-SCU42 and select ***Create Device Variable***.	
8	Double-click the **Task Settings** under Configurations and Setup in the Multiview Explorer.	
9	The Task Settings Tab Page is displayed in the Edit Pane. Select the program assignment settings button and check that Program0 is set under PrimaryTask.	
10	Select ***Check All Programs*** from the Project Menu.	
11	The Build Tab Page is displayed in the Edit Pane. Check that both error and warning are 0.	

12	Select *Rebuild Controller* from the Project Menu.	
	A screen is displayed indicating the conversion is being performed.	
13	Check that both error and warning are 0 in the Build Tab Page.	
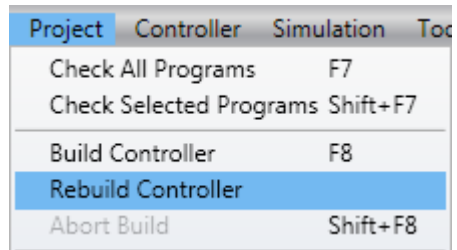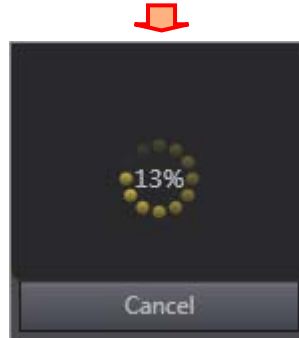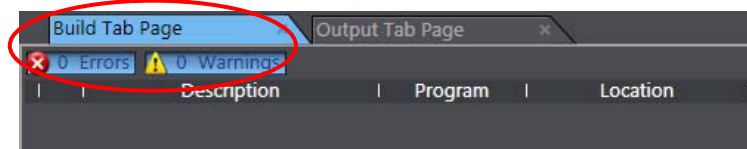
7.3.4. Going Online and Transferring the Project Data

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

⚠ **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.
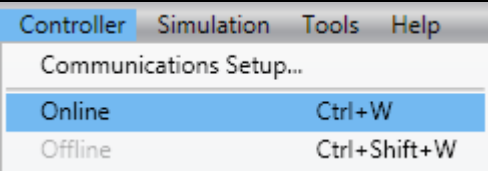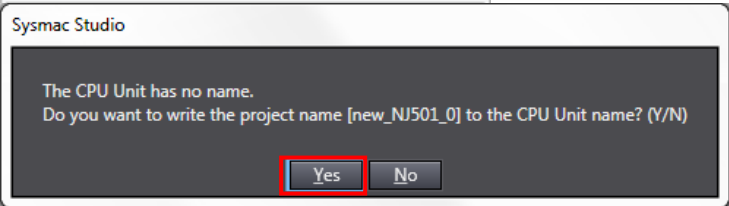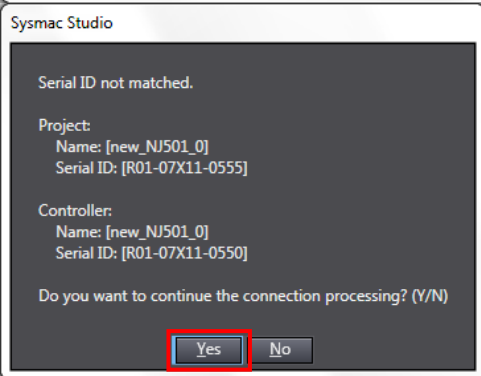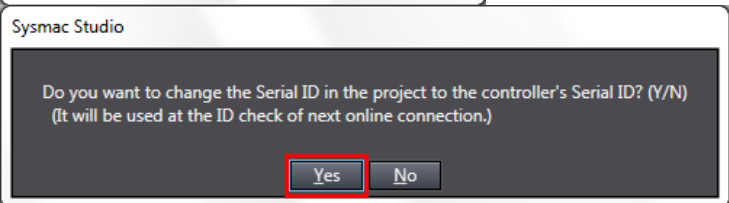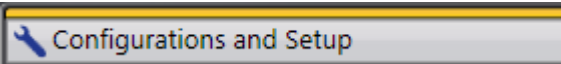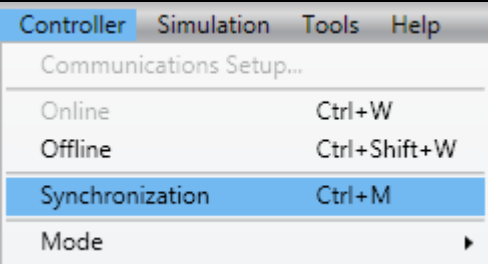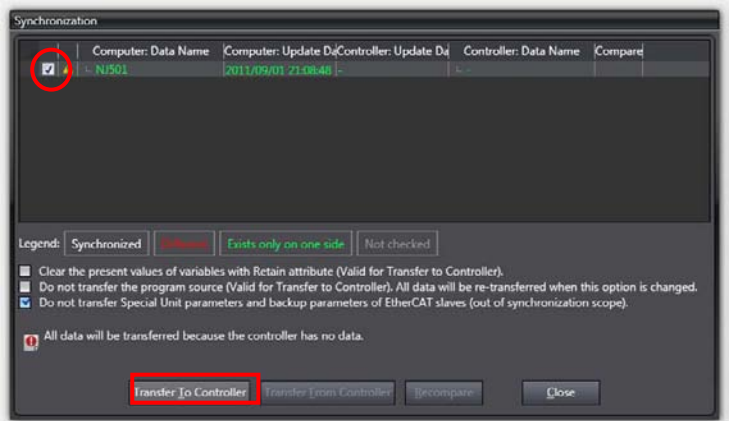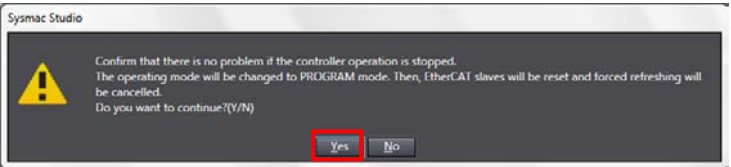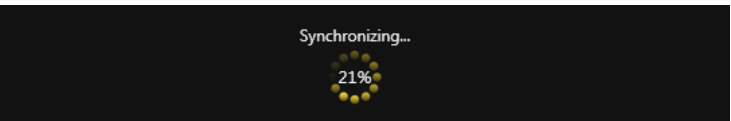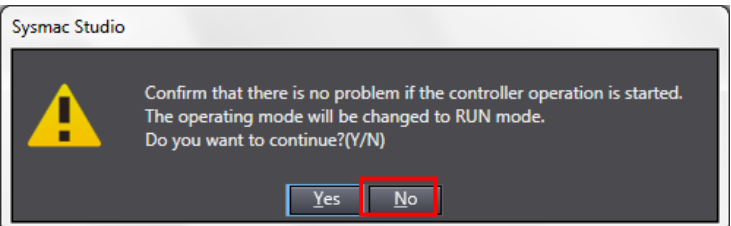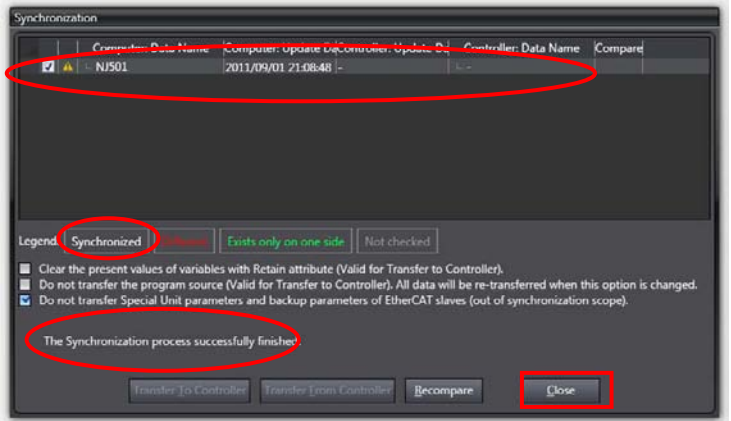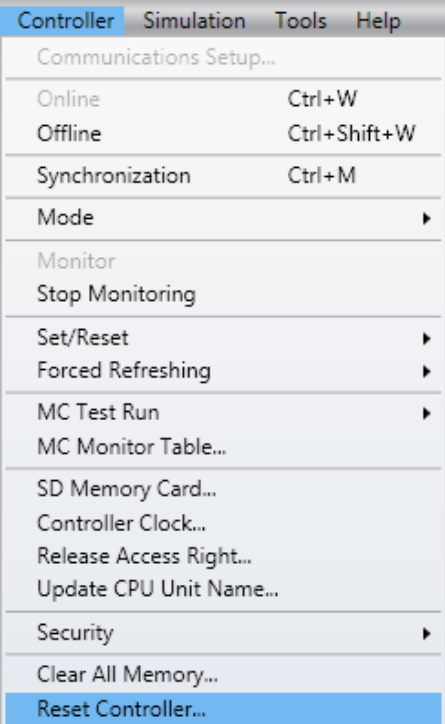
1	Select **Communications Setup** from the Controller Menu.	
2	The Communications Setup Dialog Box is displayed. Select the *Direct connection via USB* Option in the Connection Type Field. Click the **OK** Button.	

📝 **Additional Information**

For details on the online connections to a Controller, refer to *Section 5 Going Online with a Controller* in the *Sysmac Studio Version 1.0 Operation Manual* (Cat. No. W504).

3	Select *Online* from the Controller Menu.	Controller · Simulation · Tools · Help
		Communications Setup...
		Online · Ctrl+W
		Offline · Ctrl+Shift+W
	A confirmation dialog is displayed. Click the **Yes** Button.	Sysmac Studio · The CPU Unit has no name. · Do you want to write the project name [new_NJ501_0] to the CPU Unit name? (Y/N) · Yes · No
	*A displayed dialog depends on the status of the Controller used. Select the **Yes** Button or other button to proceed with the processing. · *Displayed serial ID differs depending on the device.	Sysmac Studio · Serial ID not matched. · Project: · Name: [new_NJ501_0] · Serial ID: [R01-07X11-0555] · Controller: · Name: [new_NJ501_0] · Serial ID: [R01-07X11-0550] · Do you want to continue the connection processing? (Y/N) · Yes · No
		Sysmac Studio · Do you want to change the Serial ID in the project to the controller's Serial ID? (Y/N) · (It will be used at the ID check of next online connection.) · Yes · No
4	When an online connection is established, a yellow bar is displayed on the top of the Edit Pane.	🔧 Configurations and Setup
5	Select *Synchronization* from the Controller Menu.	Controller · Simulation · Tools · Help · Communications Setup... · Online · Ctrl+W · Offline · Ctrl+Shift+W · Synchronization · Ctrl+M · Mode ▶
6	The Synchronization Dialog Box is displayed. Check that the data to transfer (NJ501 in the right figure) is selected. Then, click the **Transfer to Controller** Button.	Synchronization · Computer: Data Name / Computer: Update Da / Controller: Update Da / Controller: Data Name / Compare · ☑ NJ501 · 2011/09/01 21:08:48 · Legend: Synchronized / Different / Exists only on one side / Not checked · ☐ Clear the present values of variables with Retain attribute (Valid for Transfer to Controller). · ☐ Do not transfer the program source (Valid for Transfer to Controller). All data will be re-transferred when this option is changed. · ☑ Do not transfer Special Unit parameters and backup parameters of EtherCAT slaves (out of synchronization scope). · All data will be transferred because the controller has no data. · Transfer To Controller · Transfer From Controller · Recompare · Close

7	A confirmation dialog is displayed. Click the **Yes** Button.	
	A screen stating "Synchronizing" is displayed.	
	A confirmation dialog box is displayed. Click the **No** Button.	
8	Check 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. *If the synchronization fails, check the wiring and repeat the procedure described in this section.	
9	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.	

| 10 | A confirmation dialog box is displayed several times. Click the **Yes** Button. | |
| 11 | The controller is reset, and Sysmac Studio goes offline. The yellow bar on the top of the Edit Pane disappears. Use steps 1 to 4 to go online. | |
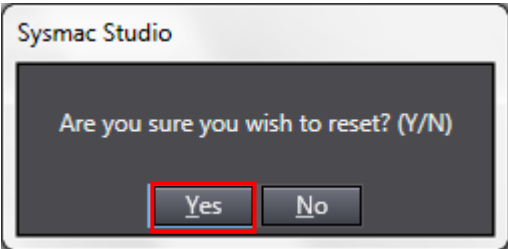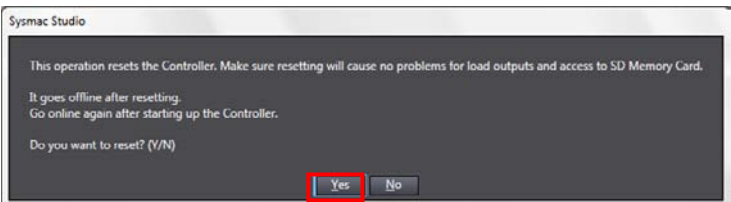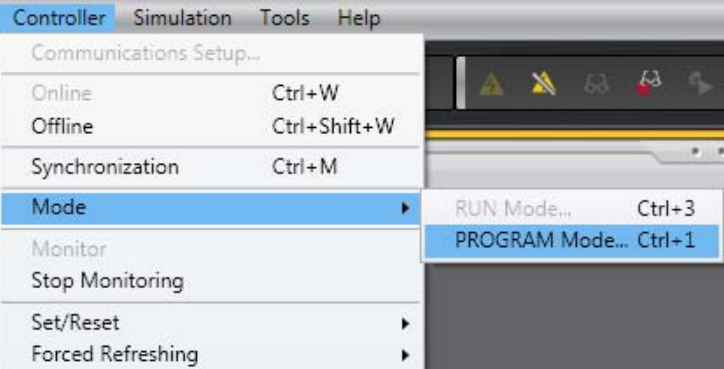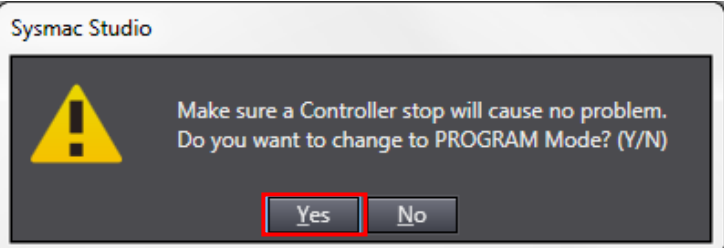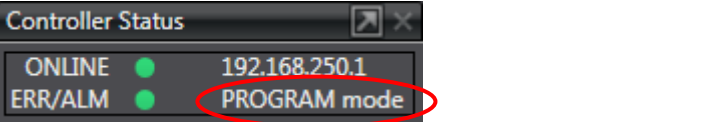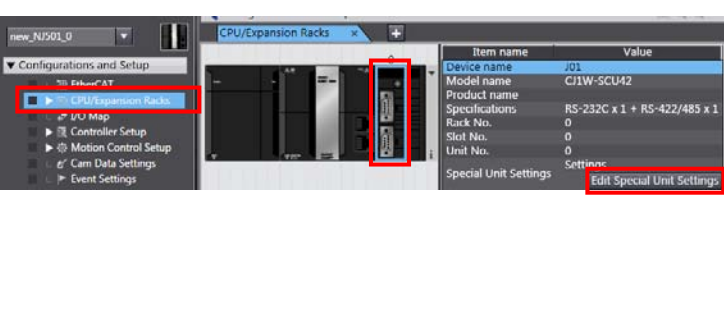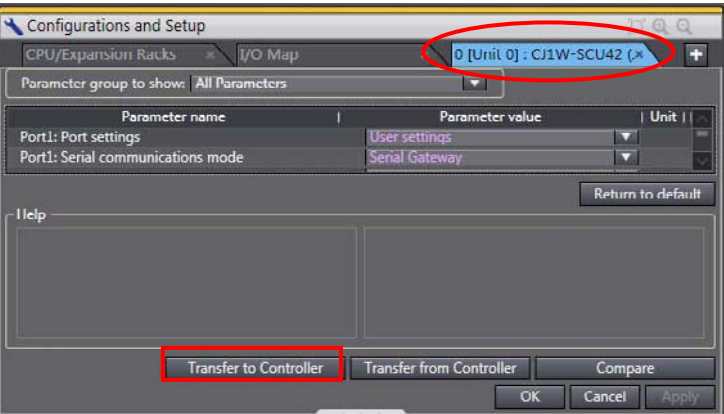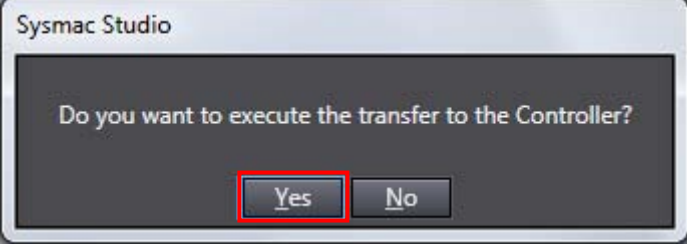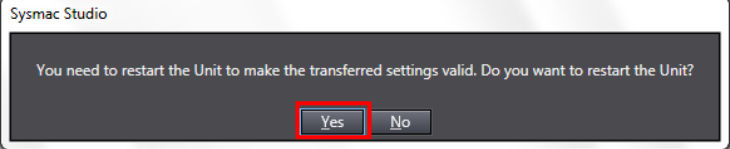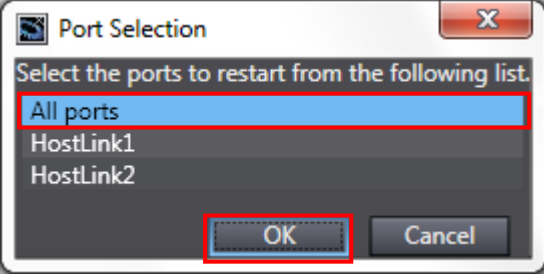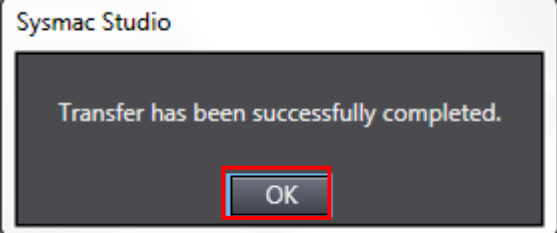
7.3.5. Transferring the Unit Settings

Transfer the setting data of the Serial Communication Unit.

1	Select *Mode* - *PROGRAM Mode* from the Controller Menu.	
2	A confirmation dialog box is displayed. Click the **Yes** Button.	
3	PROGRAM mode is displayed on the Controller Status Pane.	
4	Double-click **CPU/Expansion Racks** under Configurations and Setup in the Multiview Explorer. Select the Serial Communications Unit figure. Click **Edit Special Unit Settings**.	
5	The 0 [Unit 0]: Tab is displayed. Click the **Transfer to Controller** Button.	

6	A confirmation dialog box is displayed. Click the **Yes** Button. A dialog box is displayed indicating transferring is being performed. A confirmation dialog box is displayed. Click the **Yes** Button.	Sysmac Studio Do you want to execute the transfer to the Controller? Yes No Sysmac Studio You need to restart the Unit to make the transferred settings valid. Do you want to restart the Unit? Yes No
7	The Port Selection Dialog Box is displayed. Select **All ports** and click the **OK** Button. *You can select **HostLink1** instead of All ports.	Port Selection Select the ports to restart from the following list. All ports HostLink1 HostLink2 OK Cancel
8	A confirmation dialog box is displayed. Click the **OK** Button.	Sysmac Studio Transfer has been successfully completed. OK

7.4. Connection Status Check

Execute the project file that was transferred and confirm that serial communications are performed normally.

▶ **Precautions for Correct Use**

Please confirm that the serial cable has been connected before proceeding to the following steps.

If it is not connected, turn OFF the power of the devices, and then connect the serial cable.

7.4.1. Executing the Project File and Checking the Receive Data

Execute the project file and check if the correct data are written to the variables of the Controller.

⚠ **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.

1	Select *Mode* - *RUN Mode* from the Controller Menu.	
	A confirmation dialog box is displayed. Click the **Yes** Button.	
2	RUN mode is displayed on the Controller Status Pane.	
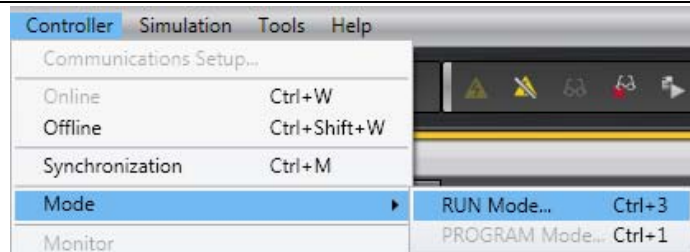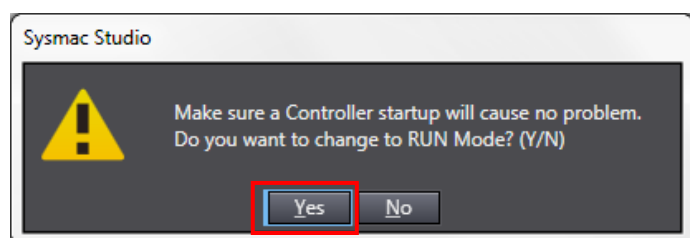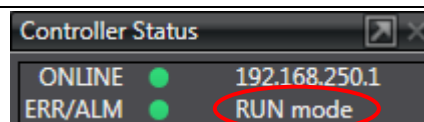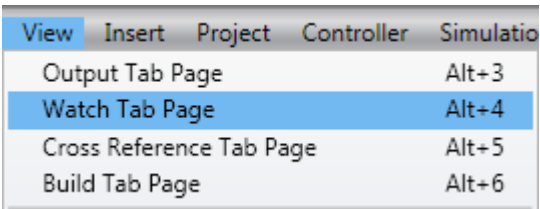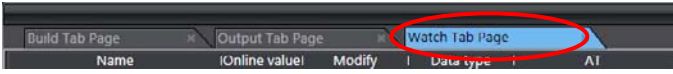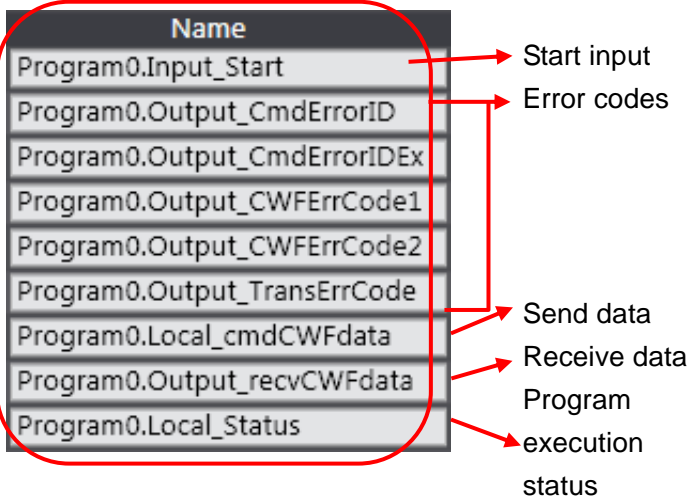
3	Select *Watch Tab Page* from the View Menu.	View Insert Project Controller Simulatio Output Tab Page — Alt+3 Watch Tab Page — Alt+4 Cross Reference Tab Page — Alt+5 Build Tab Page — Alt+6
4	The Watch Tab Page is displayed in the lower section of the Edit Pane.	Build Tab Page × Output Tab Page × Watch Tab Page Name IOnline valuel Modify Data type Al
5	Check that the variables shown on the right are displayed in the Name Columns. *If necessary variables are not displayed, click **Input Name...** to add. *Program0 of the Name is omitted from the following descriptions.	**Name** Program0.Input_Start → Start input Program0.Output_CmdErrorID → Error codes Program0.Output_CmdErrorIDEx Program0.Output_CWFErrCode1 Program0.Output_CWFErrCode2 Program0.Output_TransErrCode Program0.Local_cmdCWFdata → Send data Program0.Output_recvCWFdata → Receive data Program0.Local_Status → Program execution status
6	Click **TRUE** on the Modify Column of *Input_Start*. The Online value of *Input_Start* changes to True. The program is operated and CompoWay/F communications are performed with the destination device.	Name IOnline valuel Modify Program0.Input_Start False TRUE FALSE Name IOnline valuel Modify Program0.Input_Start True TRUE FALSE
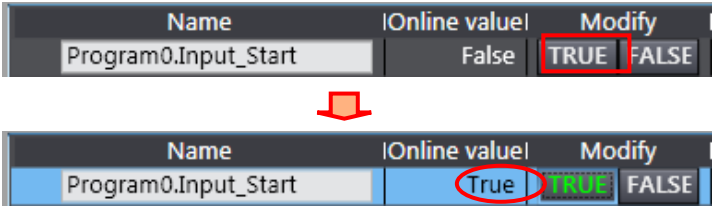
7	If the communications end normally, each error code changes to 0. *In the case of error end, the error code for an error is stored. For details on error codes, refer to *9.7 Error Processing*. The Online value of *Local_Status.Done*, which indicates the program execution status, changes to True. In the case of error end, *Local_Status.Error* changes to True. *When *Input_Start* changes to FALSE, each *Local_Status* variable also changes to False. For details, refer to *9.6 Timing Charts*.	

Name	Online value	Modify
Program0.Input_Start	True	TRUE FALSE
Program0.Output_CmdErrorID	0000	
Program0.Output_CmdErrorIDEx	0000 0000	
Program0.Output_CWFErrCode1	0000	
Program0.Output_CWFErrCode2	0000	
Program0.Output_TransErrCode	0000	

Name	Online value	Modify
Program0.Local_Status		
Busy	False	TRUE FALSE
Done	True	TRUE FALSE
Error	False	TRUE FALSE

8	The response data received from the destination device is stored in *Output_recvCWFdata* (*Local_cmdCWFdata* is a send command.) *The receive data depends on the Digital Controller used. *Refer to *9.2.2 Detailed Description of the Command* for details of the command.	

Name
Program0.Local_cmdCWFdata
Program0.Output_recvCWFdata

Online value
010000503
01000005030000E5CC-RX3A500D9

Receive data

01= Node No.

00=Subaddress

00=End code

0503= Command (MRC,SRC)

0000= Response (MRES,SRES)

E5CC-RX3A5=Receive data (Controller attribute)

00D9=Buffer size

33

8. Initialization Method

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

If the device settings have been changed from the factory default setting, some settings may not be applicable as described in this procedure.

8.1. Controller

To initialize the Controller, it is necessary to initialize the CPU Unit and Serial Communications Unit.

8.1.1. CPU Unit

To initialize the settings of the Controller, select ***Clear All Memory*** from the Controller Menu of the Sysmac Studio.
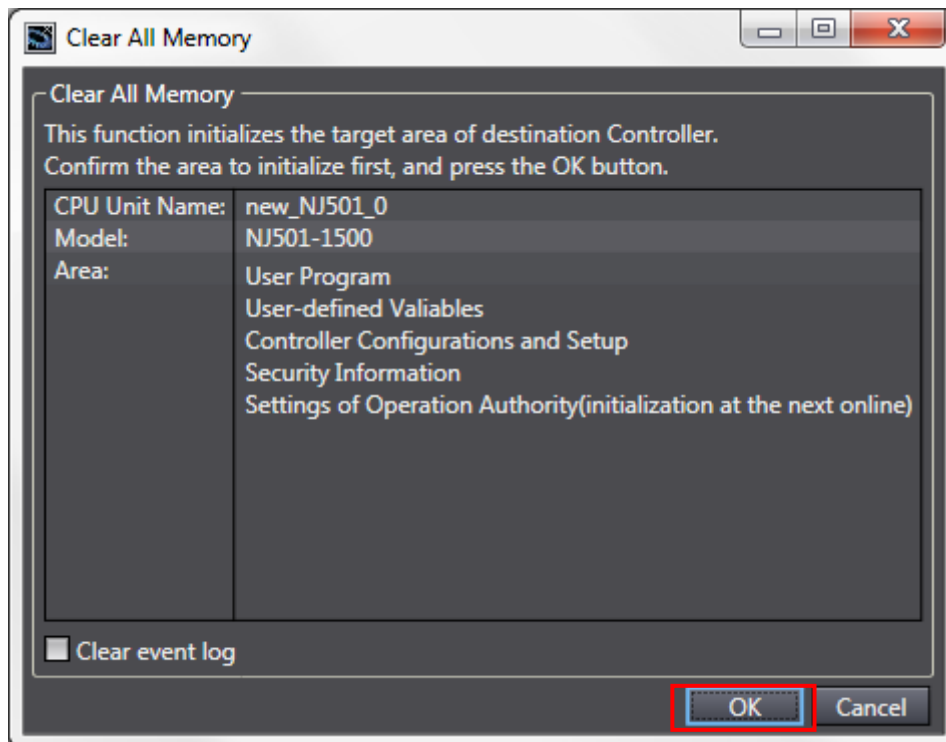
8.1.2. Serial Communications Unit

To initialize the settings of the Serial Communications Unit, select **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.
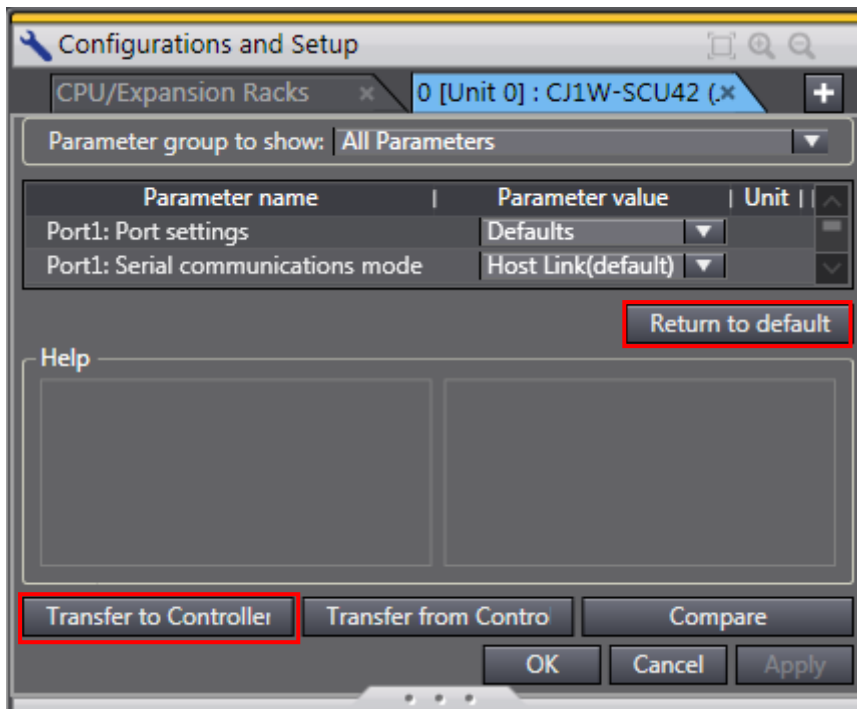
8.2. Digital Controller

For information on how to initialize the Digital Controller, refer to *Parameter Initialization* of *6-8 Advanced Function Setting Level* in the *Digital Temperature Controllers User's Manual* (Cat.No. H175).

9. Project file

This section describes the details of the project file used in this document.

9.1. Overview

This section explains the specifications and functions of the project file used to connect the Digital Controller (E5CC/E5EC/E5AC) (hereinafter referred to as a destination device) to the Controller (Serial Communications Unit) (hereinafter referred to as an SCU Unit).

The project file is a Sysmac Studio project file.
The following data has already been set in this project file.
•SCU Unit communications settings and program task settings
•A program and function block to perform CompoWay/F communications
•Variable tables and data type definitions of the variables used in ST programs

This project file uses CompoWay/F communications (serial gateway function) of the SCU Unit to read the controller attributes of the destination device and to detect whether the operation ends normally or abnormally.
The normal end of this project file indicates the normal end of the CompoWay/F communications.
The error end indicates the error end of the CompoWay/F communications and destination device error (judged on the response data from the destination device).

📑 **Additional Information**

OMRON has confirmed that normal communications can be performed using this project file under the OMRON evaluation conditions including the test system configuration, version of each product, and product Lot, No. of each device which was used for evaluation.
OMRON does not guarantee the normal operation under the disturbance such as electrical noise and the performance variation of the device.

📑 **Additional Information**

With Sysmac Studio, a data type + "#" are prefixed to decimal data and a data type + "#" + "16" + "#" are prefixed to hexadecimal data when it is necessary to distinguish between decimal and hexadecimal data. (e.g., INT#1000 decimal -> INT#16#03E8 hexadecimal. For DINT, a data type + "#" are unnecessary.)

9.1.1. Communications Data Flow

The following shows the data flow from issuing a CompoWay/F command from the Controller (SCU Unit) to the destination device to receiving the response data from the destination device.

1.	Sending a command	The send message set with the ST program is sent from the SCU Unit to the destination device.
	↓	
2.	Receiving a response	The response data, which was received by the SCU Unit from the destination device, is stored in the specified internal variable of the CPU Unit.
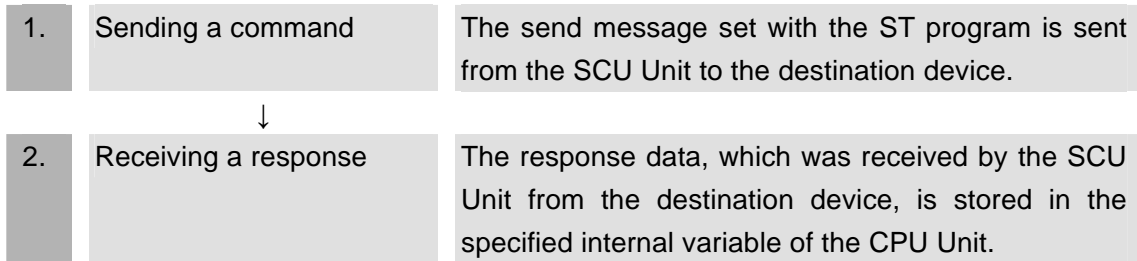
9.1.2. SendCmd Instruction and Send/Receive Message

This section outlines the send command instruction (SendCmd, hereinafter referred to as "SendCmd instruction") and send/receive process of message.

📖 **Additional Information**

For details, refer to *2 Instruction Descriptions - Communications Instructions* in the *NJ-series Instructions Reference Manual* (Cat. No. W502).

●SendCmd instruction argument data

Instruction	Name	FB/FUN	Graphic expression	ST expression
SendCmd	Send Command	FB	SendCmd_instance SendCmd —Execute Done— —DstNetAdr Busy— —CommPort Error— —CmdDat ErrorID— —CmdSize ErrorIDEx— —RespDat - - - - - - - - - - —Option	SendCmd_instance(Execute, DstNetAdr, CommPort, CmdDat, CmdSize, RespDat, Option, Done, Busy, Error, ErrorID, ErrorIDEx);

Variables

Name	Meaning	I/O	Description	Valid range	Unit	Default
DstNetAdr	Destination network address		Destination network address	---		---
CommPort	Destination serial port		Destination serial port	Only _NONE	---	_NONE
CmdDat[] (array)	Command array	Input	Command to send	Depends on data type.		*
CmdSize	Command data size		Command data size	2 to maximum data length (depends on network type)	Bytes	2
Option	Response		Response monitoring and retry specifications	---	---	---
RespDat[] (array)	Response storage array	In-out	Array to store response	Depends on data type.	---	---

[CmdDat[]array: Command array (Send data)]

The command (send data) is converted from the *Local_cmdCWFdata* string variable to *Local_cmddata* BYTE array variable before setting.

Variable of BYTE array: Local_cmddata,SendCmd_instance.CmdDat

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	· ·
Command	28	03	AA	BB	CC	DD	EE	FF	GG	HH	II	**
	CompoWay/F communications command		Node number		Subaddress		SID	Command MRC		SRC		Text

*#16#2803 CompoWay/F communications command is a serial gateway function of the SCU Unit and is a command used for the CompoWay/F.

*The command data after AA in the shaded areas are expressed in ASCII.

[RespDat[]array: Response storage array (receive data)]

The response (receive data) is received in a BYTE array. After receiving, the BYTE array (Local_recvdata) is converted into a character string variable (Output_recvCWFdata) and checked.

BYTE array variable: Local_recvdata,SendCmd_instance.RespDat

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	· ·
Response	28	03	aa	bb	AA	BB	CC	DD	cc	dd	FF	GG	HH	II	ee	ff	gg	hh	**
	CompoWay/F comm. command	Comma-nd end code	Node number		Subaddress		End code				Command MRC		SRC		Response MRES		SRES		Text

*The response data after AA in the shaded areas are expressed in ASCII.

*Response data not in thick frames such as AA will be the same as those of the command code.

*Response data in thick frames will be stored in the variables below as the destination device error code when a destination device error occurs. "aa" and "bb" command end codes are not used in this document or this project file. For details on error codes, refer to *9.7.2 Destination Device Errors*.

Ouitput_CWFErrCode1: End code

Ouitput_CWFErrCode2: Response (MRES,SRES)

[DstNetAdr: Destination network address]

The following table shows variables that store a destination network address.

Variable		Items	Data type	Description	Set value	Default
DstNetAdr		Destination network address	_sDNET_ADR	Destination network address		
	NetNo	Network address	USINT	USINT#16#00 (fixed): Local network	USINT# 16#00	USINT# 16#00
	NodeNo	Node address	USINT	USINT#16#00 (fixed): Communications in the local controller	USINT# 16#00	USINT# 16#00
	UnitNo	Unit address	BYTE	Configure the settings as follows. Unit address of serial port For Serial Communication Unit Port 1: #16#80 + #16#04 x [Unit number (Hexadecimal)] Port 2: #16#81 + #16#04 x [Unit number (Hexadecimal)] (Example of unit number (No.10) port 2: serial port unit address= #16#81+#16#04x#16#0A=#16# 81+#16#28= #16#A9)	BYTE #16#80	BYTE #16#00

[CommPort: Destination serial port]

The following table shows a variable that stores a destination serial port number.

Variable	Items	Data type	Description	Set value	Default
CommPort	Destination serial port	_ePORT	_NONE: The destination is not a serial port in Host Link Mode. *To specify the serial port unit address as the Destination unit address, set _NONE.	_NONE	_NONE

[CmdSize: Command data size]

The following table shows the variable that stores the number of bytes of the command (send data).

Variable	Items	Data type	Description	Set value	Default
CmdSize	Command data size	UINT	Sets the number of command data bytes. (#0002 to maximum data length)	UINT#11	UINT#0

[Option: Response]

The following table shows the variables that store settings to receive a response.

Variable		Items	Data type	Description	Set value	Default
Option		Response	_sRESPONSE	Response monitoring and retry specifications		
	isNonResp	No response	BOOL	TRUE: Response is not required FALSE: Response is required	FALSE	FALSE
	TimeOut	Timeout time	UINT	Sets UINT#1 to 65535 (indicates 0.1 to 6553.5 seconds). (UINT#0: 2 seconds (Default))	#0	UINT#0
	Retry	No. of resends	USINT	Sets USINT#0 to #15 (0 to 15 times).	#3	USINT #0

●Send/receive message

[Overview of send/receive message]

Send message (Command)

STX 16#02	Node No.	Sub address	SID	Command		Text	ETX 16#03	BCC
				MRC	SRC			

Controller → Destination device

STX 16#02	Node No.	Sub address	End code	Command		Response		Text	ETX 16#03	BCC
				MRC	SRC	MRES	SRES			

Receive message (Response)
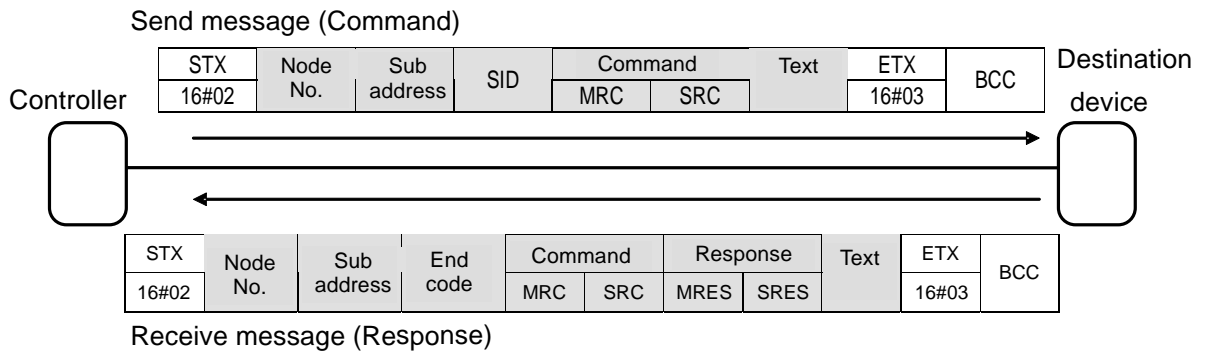
[Relation between send command (SendCmd instruction operand) and send message]

Local_cmdCWFdata

Node No.	Subaddress	SID	Command MRC	SRC	Text

Local_cmddata

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	・・
28	03	AA	BB	CC	DD	EE	FF	GG	HH	II	**

CPU→SCU: CompoWay/F comm. command | Node No. | Subaddress | SID | Command MRC | SRC | Text

SCU→Destination device (Send data)

STX 16#02	Node No.	Sub address	SID	Command		Text	ETX 16#03	BCC
				MRC	SRC			
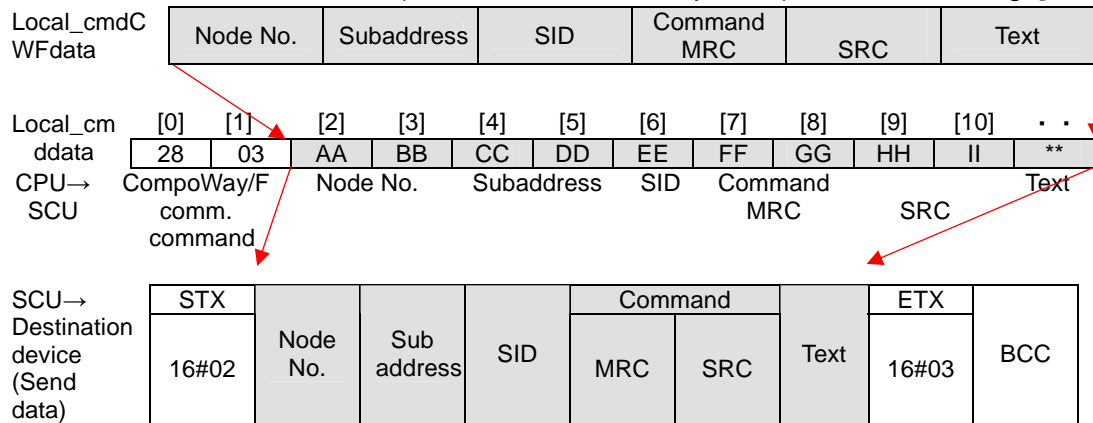
[Relation between receive response (SendCmd instruction operand) and receive message]

Destination device → SCU (Receive data)

STX 16#02	Node No.	Subaddress	End code	Command		Response		Text	ETX 16#03	BCC
				MRC	SRC	MRES	SRES			

Local_recv data

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	・
28	03	aa	bb	AA	BB	CC	DD	cc	dd	FF	GG	HH	II	ee	ff	gg	hh	**

SCU→CPU: CompoWay/F comm. command | Command end code | Node No. | Subaddress | End code | Command MRC / SRC | Response MRES / SRES | Text

Output_recvCWFdata

Node No.	Subaddress	End code	Command		Response		Text
			MRC	SRC	MRES	SRES	

9.2. Destination Device Command

This section explains the destination device command used in this project file.

9.2.1. Overview of the Command

This project file uses the Read Controller Attributes command to read information from the destination device.

Command name	Description
Read Controller Attributes	Reads the attributes of the destination device.

9.2.2. Detailed Description of the Command

This section explains the Read Controller Attributes command.

●Send message command format

This is the command format of the message that is sent by the Controller to the destination device according to the setting of the Read Controller Attributes command.

•Except for STX, ETX and BCC, data are sent in ASCII.

•STX, ETX and BCC are automatically added to the send message by the SCU Unit.

•Data not used are moved forward.

Data	No of bytes	Remarks	Variable name and setting value of the CWFCmdsSet instruction		Data type
STX	1	Fixed (#16#02)			
Node number	2	Sets the communications unit No. of the destination device.	NodeNo	"01"	STRING[3]
Subaddress	2	Not used (Fixed value)	SubAddress	"00"	STRING[3]
SID	1	Not used (Fixed value)	SID	"0"	STRING[2]
Command (MRC+SRC)	4	Sets the Read Controller Attributes command code.	MRCSRC	"0503"	STRING[5]
Text	*	Not used for the Read Controller Attributes command.	SendText	""	STRING [128]
ETX	1	Fixed (#16#03)			
BCC	1	BCC			

•Data are set and joined in CWFCmdsSet_instance.

Variable	Description (data type)	Data	Description
Local_cmd CWFdata	Send message (STRING[256])	CONCAT(NodeNo, SubAddress, SID, MRCSRC, SendText)	After converting from a string variable to a BYTE array variable, this variable is used as send data of the SendCmd instruction (SendCmd_instance).

●Response format of the receive message

This is the response format of the normal message received by the Controller from the destination device according to the settings of the Read Controller Attributes command.

•Except for STX, ETX and BCC, data are received in ASCII.

•STX, ETX and BCC are automatically removed from the receive message by the SCU Unit.

•Data not used are moved forward.

Data	Number of bytes	Receive data	Remarks
STX	1		Fixed (#16#02)
Node number	2	"01"	Same as command data
Subaddress	2	"00"	Same as command data
End code	2	"00"	Normal end
Command (MRC+SRC)	4	"0503"	Same as command data
Response (MRES+SRES)	4	"0000"	Normal end
Text	*		The data corresponding to the command is stored. For Read Controller Attributes command, the following data are stored.
Model	10	**	Reads the model of the destination device.
BufferSize	4	**	Reads the communication buffer size.
ETX	1		Fixed (#16#03)
BCC	1		BCC

●Response format of the receive message (error)

The response format for an error message received by the Controller from the destination device.

•Except for STX, ETX and BCC, data are received in ASCII.

•STX, ETX and BCC are automatically removed from the receive message by the SCU Unit.

Data	Number of bytes	Receive data	Remarks
STX	1		Fixed (#16#02)
Node number	2	"01"	Same as command data
Subaddress	2	"00"	Same as command data
End code	2	"XX"	End code For details on error codes, refer to *9.7.2 Destination Device Error*.
Command (MRC+SRC)	4	"0503"	Same as command data
Response (MRES+SRES)	4	"yyyy"	Response codes For details on error codes, refer to *9.7.2 Destination Device Error*.
ETX	1		Fixed (#16#03)
BCC	1		BCC

Additional Information

To use a destination device command other than the command used in this document, you may need to set up the destination device according to the command used. For details on the destination device command and send/receive messages, refer to *Chapter 2 CompoWay/F Communications Procedures* and *Chapter 3 Communications Data for CompoWay/F* in the *Digital Temperature Controllers Communications Manual* (Cat. No. H175).

9.3. Error Detection Processing

This section explains the error detection processing of this project file.

9.3.1. Error Detection in the Project File

This project file detects and handles errors of the following items (1) to (2). For error codes, refer to *9.7 Error Processing*.
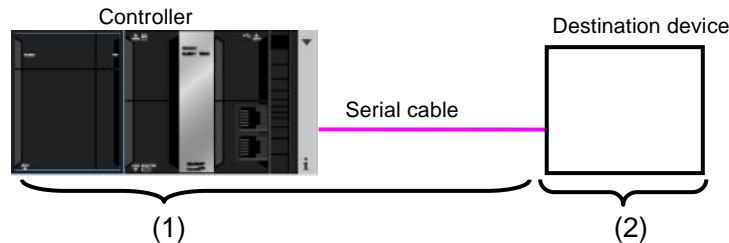
(1)Errors at execution of SendCmd instruction (SendCmd instruction error)

The errors in the Unit, command format, or parameters at the execution of the SendCmd instruction are detected as "SendCmd instruction errors". An error is detected according to the error codes *ErrorID* and *ErrorIDEx* of the SendCmd instruction.

If a communications error is caused by a transmission error due to, for example, a character corruption or unmatched baud rate setting, the transmission error status (J01_P1_TransErrSta) allocated variable area of the SCU is stored in the output variable to show the communication error status.

(2)Errors in the destination device (Destination device error)

Errors that occur in the destination device are detected as the destination device errors such as a command error, parameter error and an error that prevents execution. An error is detected according to the response data which is sent from the destination device. When an error occurred in the destination device, the error codes are set in the end code and response of the receive data.

9.4. Variables

The following lists the variables used in this project file.

9.4.1. List of Variables

Following table lists variables necessary to execute this project file.

●Input Variables

The following table lists variables used to operate this project file.

Name	Data type	Description
Input_Start	BOOL	This project file is started by turning OFF (FALSE) and then ON (TRUE). After checking the normal end output or error end output, change this to TRUE and then FALSE.

●Output Variables

The following table lists variables in which execution results of this project file are reflected.

Name	Data type	Description
Output_recvCWFdata	STRING[256]	The received data is stored in a string array.
Output_CmdErrorID	WORD	Stores an error code when a SendCmd instruction error occurs.
Output_CmdErrorIDEx	DWORD	Stores an expansion error code when a SendCmd instruction error occurs.
Output_TransErrCode	WORD	Stores the data of *J01_P1_TransErrSta* transmission error status when a communications error occurs.
Output_CWFErrCode1	WORD	Stores an error end code received from a destination device when an error occurs in a destination device.
Output_CWFErrCode2	WORD	Stores an error response code received from a destination device when an error occurs in a destination device.

●Internal variables

The following table lists variables used only for operations of this project file.

Name		Data type	Description
Local_Status		sStatus (STRUCT)	Program execution status
	Busy	BOOL	TRUE while executing this project file. FALSE while not executing this project file.
	Done	BOOL	TRUE for a normal end of this project file. FALSE when *Input_Start* changes to FALSE.
	Error	BOOL	TRUE for an error end of this project file. FALSE when *Input_Start* changes to FALSE.
Local_State		DINT	Status processing number
Local_ExecFlgs		BOOL	Communications instruction execution flag
Local_InitialSettingOK		BOOL	Normal initialization setting check flag
Local_DstNetAdr		_sDNET_ADR	Destination address for SendCmd instruction setting
Local_CommPort		_ePORT	Destination serial port specification for SendCmd instruction setting
Local_sendSize		UINT	Send command data size for SendCmd instruction setting
Local_Option		_sRESPONSE	Response setting for SendCmd instruction setting
Local_cmddata		ARRAY[0..255] OF BYTE	Command array for SendCmd instruction setting
Local_recvdata		ARRAY[0..255] OF BYTE	Response storage array for SendCmd instruction setting
Local_cmdCWFdata		STRING[256]	Send data character string type

9.4.2. List of Variables Used in Function Block/Function

The following tables list the user-defined function block used in the program to execute this project file.

For variables used for the function block, refer to *9.5.3 Detailed Description of Function Block*.

Name	Data type	Description
CWFCmdsSet_instance	CWFCmdsSet	Sets a byte size of the send/receive data and a send message.

9.4.3. List of System Variables

The following table lists variables necessary to execute this project file.

The following allocations are determined according to the unit number that is set for the Serial Communications Unit and cannot be changed.

●Allocated variables

Name	Data type	Description
J01_P1_TransErr	BOOL	Transmission error flag (Unit 0, Port 1)
J01_P1_TransErrSta	WORD	Transmission error status (Unit 0, Port 1)

📓 **Additional Information**

For information on variables of the Serial Communications Unit, refer to *5-2 Device Variables for CJ-series Unit and System-defined Variables (During Serial Gateway Mode)* in the *CJ-series Serial Communications Units Operation Manual for NJ-series CPU Unit* (Cat.No. W494).

●System Variables

Name	Data type	Description
_Port_isAvailable	BOOL	Communications Port Enabled Flag

📓 **Additional Information**

For details, refer to *2 Instruction Descriptions - Communications Instruction (SendCmd)* in the *NJ-series Instructions Reference Manual* (Cat. No. W502).

9.5. Program (ST language)

9.5.1. Functional Components of ST Program

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

Major classification	Minor classification	Description
1. Communications processing	1.1. Starting communications processing 1.2. Clearing the communications processing status flags 1.3. Checking the communications processing in progress	Starts communications processing.
2. Initialization processing	2.1. Initializing the communications instruction 2.2. Initializing the communications instruction execution flags 2.3. Initializing the error code storage areas 2.4. Setting SendCmd instruction control data 2.5. Setting the send variable 2.6. Initialization setting end processing	Initializes the receive data storage area and error code storage area. Sets the parameters for the SendCmd instruction (CompoWay/F communications) and send data.
3. CompoWay/F communication processing	3.1 Normal receive processing 3.2 Error receive processing 3.3. Executing the communications instruction	Executes the SendCmd instruction (CompoWay/F communications). Judges whether the operation ended normally or abnormally based on the related flags and receive data.
4. Processing number error processing		Performs the error processing when a non-existent status processing number is detected during the execution of the program.

9.5.2. Detailed Description of Main Program

The following shows the project file.

Communications setting and send data (command), which need to be changed depending on the destination device, are set in the (CWFCmdsSet) function block. For information on how to change these values, refer to *9.5.3 Detailed Description of Function Block.*

[Main program:Program0]

1. Communications processing

```
(* =================================================
 Name: NJ Series CompoWay/F communications program (RS-485)
 Function: CompoWay/F communications main program
 Configuration: CJ1W-SCU42 Serial Unit
        (Serial gateway, Unit number: 0, Serial port number: 1)
 Remarks:
 Version information: January 17, 2012 V1.00 New release
 (C)Copyright OMRON Corporation 2012 All Rights Reserved.
 ================================================= *)

(* 1. Communications processing
  Variable description: Communications processing control
  Input start flag: Input_Start
  Communications processing status flag list:Local_Status<STRUCT>
  |-Communications processing in progress flag (Busy) :Local_Status.Busy
  |-Communications processing normal end flag  (Done) :Local_Status.Done
  |-Communications processing error end flag   (Error) :Local_Status.Error
  Status processing number :Local_State
        10:Initialization processing
        11:CompoWay/F communications processing
        Other:Processing number error processing *)

(* 1.1. Starting communications processing
   Start communications processing by turning ON the input start flag
   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:To initialization processing
END_IF;

(* 1.2. Clearing the communications processing status flags
   Clear the communications processing status flags by turning OFF
   the input start flag while communications processing is not in progress. *)
IF NOT(Local_Status.Busy) AND NOT(Input_Start) THEN
   Local_Status.Done:=FALSE;
   Local_Status.Error:=FALSE;
END_IF;

(* 1.3. Checking the communications in progress
     Execute communications and shift to each processing *)
IF Local_Status.Busy THEN
   CASE Local_State OF
```

2. Initialization processing

```
(* 2. Initialization processing
     -Initialize each processing for communications and set parameters
     -Set send data and initialize the receive data storage areas *)
```

```
10:
 (* 2.1. Initializing the communications instruction *)
    (* 2.1.1. Initializing the addresses *)
 Local_DstNetAdr.NetNo:=USINT#16#00; // Destination network address setting
 Local_DstNetAdr.NodeNo:=USINT#00;   // Local node address setting
 Local_DstNetAdr.UnitNo:=BYTE#16#00;  // Unit address setting

    (* 2.1.2. Initializing the command data
          and receive data storage area *)
 Clear(Local_cmddata);
 Clear(Local_recvdata);
 Clear(Output_recvCWFdata);

    (* 2.1.3. Initializing the parameters *)
 Local_Option.isNonResp:=FALSE;      // Response is not required
 Local_Option.TimeOut:=UINT#0;       // Timeout time
 Local_Option.Retry:=USINT#0;        // No. of resends

    (* 2.1.4. Initializing the SendCmd instruction *)
 SendCmd_instance(
    Execute:=FALSE,
    DstNetAdr:=Local_DstNetAdr,
    CommPort:=_NONE,
    CmdDat:=Local_cmddata[0],
    CmdSize:=UINT#0,
    RespDat:=Local_recvdata[0],
    Option:=Local_Option);

 (* 2.2. Initializing the communications instruction execution flags *)
 Local_ExecFlg:=FALSE;

 (* 2.3. Initializing the error code storage areas *)
 Output_TransErrCode:=WORD#16#FFFF;
 Output_CWFErrCode1:=WORD#16#FFFF;
 Output_CWFErrCode2:=WORD#16#FFFF;
 Output_CmdErrorID:=WORD#16#FFFF;
 Output_CmdErrorIDEx:=DWORD#16#FFFFFFFF;

 (* 2.4. Setting SendCmd instruction control data *)
    (* 2.4.1. Setting the address and port *)
 Local_DstNetAdr.NetNo:=USINT#16#00;
              // Destination network address setting  Fixed at 0
 Local_DstNetAdr.NodeNo:=USINT#00;
              // Local node address setting  Fixed at 0
 Local_DstNetAdr.UnitNo:=BYTE#16#80;
              // Unit address setting=80h+4hxUnit number
              // (Unit number 0, Port number 1)
 Local_CommPort:=_NONE;          // Serial port setting _NONE only

    (* 2.4.2. Setting the parameters *)
 Local_Option.isNonResp:=FALSE;      // Response is required
 Local_Option.TimeOut:=UINT#0;       // Timeout time (Default: 2.0 s)
 Local_Option.Retry:=USINT#3;        // No. of resends: 3 times

 (* 2.5. Setting the send variable *)
    (* 2.5.1. Setting the CompoWay/F communications command *)
 Local_cmddata[0]:=BYTE#16#28;
 Local_cmddata[1]:=BYTE#16#03;

    (* 2.5.2.Setting the CompoWay/F command *)
 CWFCmdsSet_instance(
    Execute:=TRUE,
    CWFdata:=Local_cmdCWFdata,
    Done=>Local_InitialSettingOK);

    (*2.5.3. Converting the send data from a string to BYTE array
          Setting the size *)
 Local_sendSize:=UINT#2+StringToAry(Local_cmdCWFdata,Local_cmddata[2]);

 (* 2.6. Initialization setting end processing *)
 IF Local_InitialSettingOK THEN
    Local_State:=11;      //To 11:CompoWay/F communications processing
 ELSE
    Local_Status.Busy:=FALSE;
    Local_Status.Error:=TRUE;
    Local_State:=0;       //To 0: Communication non-execution status
 END_IF;
```

3. CompoWay/F communications processing

```
(* 3. CompoWay/F communications processing
   -Execute communications processing from the specified port. *)
  11:
    (*3.1. Normal receive processing *)
    IF SendCmd_instance.Done THEN
       (*3.1.1. Clearing the command error code *)
       Output_CmdErrorID:=WORD#16#0000;
       Output_CmdErrorIDEx:=DWORD#16#0000000;

       (*3.1.2. Converting the receive data from BYTE array to string *)
       Output_recvCWFData:=
          AryToString(Local_RecvData[4],SizeOfAry(Local_recvdata)-UINT#4);

       (*3.1.3. Acquiring the end code and response code *)
       Output_CWFErrCode1:=STRING_TO_WORD(MID(Output_recvCWFData,2,5));
       Output_CWFErrCode2:=STRING_TO_WORD(MID(Output_recvCWFData,4,11));

       (*3.1.4. Checking the end code and response code *)
       IF (Output_CWFErrCode1 = WORD#16#0000 AND
          Output_CWFErrCode2 = WORD#16#0000) THEN
          Output_TransErrCode:=WORD#16#0000;
          Local_Status.Busy:=FALSE;
          Local_Status.Done:=TRUE;
       ELSE
          Output_TransErrCode:=WORD#16#1000;
          Local_Status.Busy:=FALSE;
          Local_Status.Error:=TRUE;
       END_IF;

       Local_State:=0;        //To 0: Communication non-execution status

    (*3.2. Error receive processing *)
    ELSIF SendCmd_instance.Error THEN
       (*3.2.1. Setting the command error code  *)
       Output_CmdErrorID:=SendCmd_instance.ErrorID;
       Output_CmdErrorIDEx:=SendCmd_instance.ErrorIDEx;

       (*3.2.2. Destination node busy or response timeout error *)
       IF Output_CmdErrorIDEx = DWORD#16#00000204 OR
          Output_CmdErrorIDEx = DWORD#16#00000205 THEN
          IF J01_P1_TransErr THEN
             Output_TransErrCode:=J01_P1_TransErrSta;
             Local_Status.Busy:=FALSE;
             Local_Status.Error:=TRUE;
             Local_ExecFlg:=FALSE;
             Local_State:=0;   //To 0: Communication non-execution status
          END_IF;
       ELSE
          Output_TransErrCode:=WORD#16#2000;
          Local_Status.Busy:=FALSE;
          Local_Status.Error:=TRUE;
          Local_ExecFlg:=FALSE;
          Local_State:=0;      //To 0: Communication non-execution status
       END_IF;

    (*3.3. Starting the communications instruction *)
       (*3.3.1. Setting the communications instruction execution flag *)
    ELSIF _Port_isAvailable AND NOT(SendCmd_instance.Busy) THEN
       Local_ExecFlg:=TRUE;
    END_IF;

       (*3.3.2. Executing the communications instruction *)
    SendCmd_instance(
       Execute:=Local_ExecFlg,
       DstNetAdr:=Local_DstNetAdr,
       CommPort:=Local_CommPort,
       CmdDat:=Local_cmddata[0],
       CmdSize:=Local_sendSize,
       RespDat:=Local_recvdata[0],
       Option:=Local_Option);
```

50

4. Processing number error processing

```
   ELSE
(* 4. Processing number error processing
     -Error processing for non-existent processing number *)
     Output_TransErrCode:=WORD#16#0100;
     Local_Status.Busy:=FALSE;
     Local_Status.Error:=TRUE;
     Local_State:=0;          //To 0: Communication non-execution status)

   END_CASE;

END_IF;
```

9.5.3. Detailed Description of Function Block

The following shows the function block of this project file.

●Description of CWFCmdsSet function block

Instruction	Meaning	FB/FUN	Graphic expression	ST expression
CWFCmdsSet	CompoWay/F communications send data setting	FB	None	CWFCmdsSet_instance(Execute, CWFdata, Done);

•In-out variable table (arguments)

•Input

Name	Data type	Meaning	Description	Valid range	Unit	Default
Execute	BOOL	Execute	The function block is executed when this parameter changes from OFF (FALSE) to ON (TRUE). (Always: TRUE)	Depends on data type	-	-

•In-out

Name	Data type	Meaning	Description	Valid range	Unit	Default
CWFdata	STRING [256]	Send data	Sets the CompoWay/F data sent to the destination device.	Depends on data type	-	-

•Output

Name	Data type	Meaning	Description	Valid range	Unit	Default
Done	BOOL	Normal end	TRUE after execution of the function block	Depends on data type	-	-
Busy	BOOL	Busy				
Error	BOOL	Error end	Not used (Not used in this project.)	-	-	-
ErrorID	WORD	Error information				
ErrorIDEx	DWORD	Error information				

•Internal variable table

Name	Data type	Meaning	Description	Valid range	Unit	Default
NodeNo	STRING[3]	Node number.	Sets the communications unit No. of the destination device.	Depends on data type	-	-
SubAddress	STRING[3]	Subaddress	Not used (optional)	Depends on data type	-	-
SID	STRING[2]	SID	Not used (optional)	Depends on data type	-	-
MRCSRC	STRING[5]	Command (MRC+SRC)	Sets the command code to use.	Depends on data type	-	-
SendText	STRING [128]	Text	Set the data corresponding to the command code.	Depends on data type	-	-

•Program

To change the destination device command, edit the code specified by the red frames on the function block below.

```
(* =====================================================
  Name: NJ Series CompoWay/F communications send data setting function block
  Function: Send data setting
  Applicable device:
   Manufacturer: OMRON Corporation
   Device: CompoWay/F corresponding device
  Remarks:
  Version information: January 17, 2012 V1.00 New release
  (C)Copyright OMRON Corporation 2012 All Rights Reserved.
  ===================================================== *)

(* Variable description:
 Argument:
  Name (variable name): Data type: Description
   -Input:
     Execute: BOOL: Execute flag
   -Output:
     Busy: BOOL: Not used
     Done: BOOL: Normal end
     Error:BOOL: Not used
     ErrorID: WORD: Not used
     ErrorIDEx: DWORD: Not used
   -In-out:
     CWFdata   STRING[256]   Send data
Return value: None
*)

IF Execute THEN
   (* Destination node address=01
     MRC=05,SRC=03: Read controller attributes    *)
   NodeNo:='01';          // Address=01
   SubAddress:='00';      // Subaddress
   SID:='0';              // SID
   MRCSRC:='0503';        // MRC,SRC
   SendText:='';          // No send data

   (* SendCmd CompoWay/F format send data setting
     Node No.+Subaddress+SID+Command (MRC,SRC)+Send data*)
   CWFdata:=CONCAT(NodeNo,SubAddress,SID,MRCSRC,SendText);

   Done:=TRUE;
END_IF;
RETURN;
```

9.6. Timing Charts

They are the timing charts of the ST program.

The following is the definition of the timing chart pattern.

Pattern	Normal end	Error end (1) SendCmd Instruction Error	Error end (2) Destination Device Error
Command	Normal	Error	Normal
Destination device	Normal	Normal or error	Error
Response	Yes	None	Yes

■Normal end

Output for one period

Output_CmdErrorID	#FFFF	#0000	#FFFF	#0000
Output_TransErrorCode	#FFFF	#0000	#FFFF	#0000
Output_CWFErrCode1 to 2	#FFFF	#0000	#FFFF	#0000

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

■Error end

	Error end (1) SendCmd Instruction Error			Error end (2) Destination Device Error			
Output_CmdErrorID	#FFFF	#xxxx	#FFFF	#FFFF	#0000	#FFF	#0000
Output_TransErrorCode	#FFFF	#xxxx	#FFFF	#FFFF	#1000	#FFF	#1000
Output_CWFErrCode1 to 2	#FFFF	#FFFF	#FFFF	#FFFF	#xxxx	#FFF	#xxxx

9.7. Error Processing

Error descriptions for this ST program are shown below.

9.7.1. SendCmd Instruction Error

These error codes are used at error ends of the SendCmd instruction.

●SendCmd instruction error code [Output_CmdErrorID, Output_CmdErrorIDEx]

An error code of *ErrorID* is stored in *Output_CmdsErrorID* and an error code of *ErrorIDEx* is stored in *Output_CmdsErrorIDEx.*

[Output_CmdErrorID]

Value	Error details
#16#0400	An input parameter for an instruction exceeded the valid range for an input variable.
#16#0406	The data position specified for an instruction exceeded the data area range.
#16#0407	The results of instruction processing exceeded the data area range of the output parameter.
#16#0800	An error occurred when a command was sent or received.
#16#0801	The port is being used.

📝 **Additional Information**

For details on errors, refer to *A-3 Error Code Details* in the *NJ-series Instructions Reference Manual* (Cat. No. W502).

For details on the troubleshooting, refer to *9-3 Troubleshooting* of the *CJ-series Serial Communications Units Operation Manual for NJ-series CPU Unit* (Cat.No. W494).

[Output_CmdErrorIDEx]

Value	Error	Correction
16#00000101	The local node is not part of the network.	Make the local node part of the network.
16#00000105	The IP address of the local node is out of range.	Set the rotary switches on the Serial Communications Unit correctly.
16#00000106	The IP address of the local node is also used by another node in the network.	Change one of the node addresses that are duplicated.
16#00000202	A Unit with the specified unit address does not exist at the destination.	Correctly set the unit address for the destination network address.
16#00000301	A Communications Controller Error occurred.	Refer to the operation manual for the relevant Unit and make suitable corrections.
16#00000304	The unit number setting is not correct.	Set the rotary switches on the Serial Communications Unit correctly.
16#00000401	The command that was sent is not supported.	Set the command string correctly.
16#00000402	The Unit model or version is not supported.	Check the Unit model and version.
16#00001001	The command is too long.	Set the command string correctly.
16#00001002	The command is too short.	Set the command string correctly.
16#00001003	The number of write elements that is specified in the command does not agree with the number of write data.	Specify the same number of write elements and write data.
16#00001004	The command format is incorrect.	Set the command string correctly.
16#0000110B	The response is too long.	Set the number of elements in the command string correctly.
16#0000110C	This is another parameter error.	Set the command string correctly.
16#00002202	The operating mode is wrong.	Check the operating mode.
16#00002502	There is an error in the part of memory for processing.	Transfer the correct data to memory.
16#00002503	The registered I/O Unit configuration does not agree with the physical Unit configuration.	Check the I/O Unit configuration.
16#00002504	There are too many local or remote I/O points.	Set the number of local and remote I/O points correctly.
16#00002505	An error occurred in a data transmission between the CPU Unit and a CPU Bus Unit.	Check the Unit and the Connecting Cable. After removing the error, execute a command to reset the error.
16#00002506	The same rack number, unit number, or I/O address is set more than once.	Correct the settings so that each number is unique.
16#00002507	An error occurred in a data transmission between the CPU Unit and an I/O Unit.	Check the Unit and the Connecting Cable. After removing the error, execute a command to reset the error.
16#00002509	There is an error in SYSMAC BUS/2 data transmission.	Check the Unit and the Connecting Cable. After removing the error, execute a command to reset the error.
16#0000250A	An error occurred in a CPU Bus Unit data transmission.	Check the Unit and the Connecting Cable. After removing the error, execute a command to reset the error.
16#0000250D	The same word setting is used more than once.	Set the I/O words correctly.
16#00002510	The end station setting is wrong.	Set the end station correctly.

●Transmission error status [Output_TransErrCode]

Data of *J01_P1_TransErrSta* transmission error status is set in *Output_TransErrCode*.

Bits 8, 12 and 13 are set when a SendCmd instruction error, destination device error or processing number error occurs.

[Bit status at a transmission error]

Bit	Description	
15	1:Transmission error	0:No transmission error
13 and 14	(Not used)	
13	1:SendCmd instruction error	0:Normal
12	1:Destination device error	0:Normal
9 to 11	(Not used)	
8	1:Processing number error	0:Normal
7	1:FCS check error	0:FCS check normal
6	(Not used)	
5	1:Timeout error	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)	

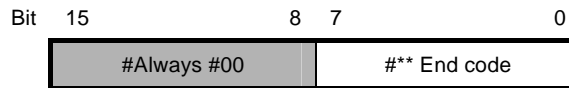#16#0000 and #16#FFFF indicate the following:

Value	Description
#16#0000	Normal end
#16#FFFF	After initialization processing of the ST program, the program is not operated yet.

9.7.2. Destination Device Error

These error codes are used for errors in the destination device.

●Output_CWFErrCode1,Output_CWFErrCode2 destination device error code

Output_CWFErrCode1 (End code)

Bit	15	8	7	0

#Always #00	#** End code

End code	Name	Description
"00"	Normal end	The command ended normally without error.
"0F"	Command error	The specified command could not be executed. The response code should indicate why the command could not be executed.
"10"	Parity error	A parity error occurred in a character when receiving.
"11"	Framing error	A framing error occurred in a character when receiving.
"12"	Overrun error	An overrun error occurred in a character when receiving.
"13"	BCC error	The received BCC value is incorrect.
"14"	Format error	•The command text contains characters other than 0 to 9, and A to F. This error does not apply to Echoback Tests. •There was no SID and command text. There was no command text. •"MRC/SRC" not included in command text.
"16"	Sub-address error	•Illegal (unsupported) sub-address of the receive frame •There was no sub-address, SID, and command text. •Sub-address was less than two characters, and there was no SID and command text
"18"	Frame length error	The received frame exceeds the specified (supported) number of bytes.

Output_CWFErrCode2 (Response Codes)

Response codes	Error name	Priority
0000	Normal end	None
0401	Unsupported command	1
1001	Command too long	2
1002	Command too short	3
1101	Area type error	4
1103	First address out-of-range error	5
1104	End address out-of-range error	6
1003	Number of elements/data mismatch	7
110B	Response too long	8
1100	Parameter error	9
3003	Read-only error	10
2203	Operation error	11

Additional Information

For details and troubleshooting the destination device errors, refer to the *Digital Temperature Controllers User's Manual* (Cat.No. H175) and the *Digital Temperature Controllers Communications Manual* (Cat. No. H174).

10. Revision History

Revision code	Date of revision	Revision reason and revision page
01	Jan. 31, 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:

 Cat. No. P520-E1-01

0213(-)