Machine Automation Controller NJ-series

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

Digital Temprature Controller
(E5□D / E5□C / E5□C-T)

Network
Connection
Guide

always in control

P520-E1-02

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 guide are the trademarks or registered trademarks of their respective companies.

Table of Contents

1. Related Manuals

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

The table below lists the manuals pertaining to this guide.

Cat. No.	Model	Manual name
W500	NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ-series CPU Unit Hardware User's Manual
W501	NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ/NX-series CPU Unit Software User's Manual
W502	NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ/NX-series Instructions Reference Manual
W504	SYSMAC-SE2□□□	Sysmac Studio Version 1 Operation Manual
W494	CJ1W-SCU□2	CJ-series Serial Communications Units Operation Manual for NJ-series CPU Unit
H225	E5□D	Digital Temperature Controllers Communications Manual
H224	E5□D	Digital Temperature Controllers User’s Manual
H175	E5□C	Digital Temperature Controllers Communications Manual
H174	E5□C	Digital Temperature Controllers User’s Manual
H186	E5□C-T	Digital Temperature Controllers Programmable Type Communications Manual
H185	E5□C-T	Digital Temperature Controllers Programmable Type User’s Manual

2. Terms and Definitions

Term	Explanation and Definition
Serial Gateway mode	This is a function of PLC that performs serial communications by automatically converting a message (command data) to a specified protocol (CompoWay/F, Modbus-RTU, or Modbus-ASCII), depending on the type of message.

3. Precautions

(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 a safety circuit, in order to ensure safety and minimize the risk of abnormal occurrence.

(2) To ensure system safety, make sure to always read and follow the information provided in all Safety Precautions and Precautions for Safe Use in the manuals for each device which is 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 guide without the permission of OMRON Corporation.

(5) The information contained in this guide is current as of March 2018. It is subject to change for improvement without notice.

The following notations are used in this guide.

⚠ WARNING	Indicates a potentially hazardous situation which, if not avoided, may 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 property damage.

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 the text. This example shows a general precaution for something that you must do.

4. Overview

This guide describes procedures for connecting a Digital Temperature Controller (E5□D, E5□C or E5□C-T) to an NJ-series Machine Automation Controller + Serial Communications Unit (hereinafter referred to as the "Controller") via serial communications, both produced by OMRON Corporation (hereinafter referred to as "OMRON"), and for checking their communication status.
Refer to *Section 6. Serial Communications Settings* and *Section 7. Serial Communications Connection Procedure* to understand setting methods and key points to send or receive a message via serial communications.
The program in the prepared Sysmac Studio project file is used to check the serial connection by sending or receiving a message of "Read Controller Attributes" to/from the Digital Temperature Controller.

■The send/receive messages of "Read Controller Attributes"

Controller	Serial communications (RS-485)	Digital Temperature Controller
Sending command data	Command data →	Executing the command
Receiving response data and storing in memory	← Response data	Returning response data

Prepare the Sysmac Studio project file with a latest version beforehand.
To obtain the Sysmac Studio project file, contact your OMRON representative.

Name	File name	Version
Sysmac Studio project file (extension: csm2)	P520_NJ_CWF485_OMRON_E5CD_V100.csm2	Ver.1.00

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

⚠ Caution

This guide aims to explain wiring methods and communications settings necessary to connect corresponding devices and provides the setting procedures. The program used in this guide is not designed to be constantly used at a site but is designed to check if the connection is properly established. Both functionalities and performances are therefore not fully considered for the program.
When you actually construct a system, please use the wiring methods, communications settings and setting procedures described in this guide as a reference, and design a 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-□□□□ NJ101-□□□□
OMRON	Serial Communications Unit	CJ1W-SCU□2
OMRON	Digital Temperature Controller	E5CD-□□2□□M-002 E5CD-□□2□□M-004 E5ED-□□4□□M-004 E5ED-□□4□□M-008 E5ED-□□4□□M-022 E5CC-□□□□□M-002 E5CC-□□□□□M-003 E5CC-□□□□□M-004 E5AC-□□□□SM-004 E5AC-□□□□SM-008 E5AC-□□□□SM-009 E5AC-□□□□SM-012 E5AC-□□□□SM-014 E5EC-□□□□□M-004 E5EC-□□□□□M-008 E5EC-□□□□□M-009 E5EC-□□□□□M-012 E5EC-□□□□□M-014 E5DC-□□□□□M-002 E5DC-□□□□□M-015 E5GC-□□□□□M-015 E5CC-T□□3□SM-002 E5CC-T□□3□SM-003 E5CC-T□□3□SM-004 E5AC-T□□4□SM-004 E5AC-T□□4□SM-008 E5AC-T□□4□SM-020 E5AC-T□□4□SM-022 E5EC-T□□4□SM-004 E5EC-T□□4□SM-008 E5EC-T□□4□SM-020 E5EC-T□□4□SM-022

Precautions for Correct Use

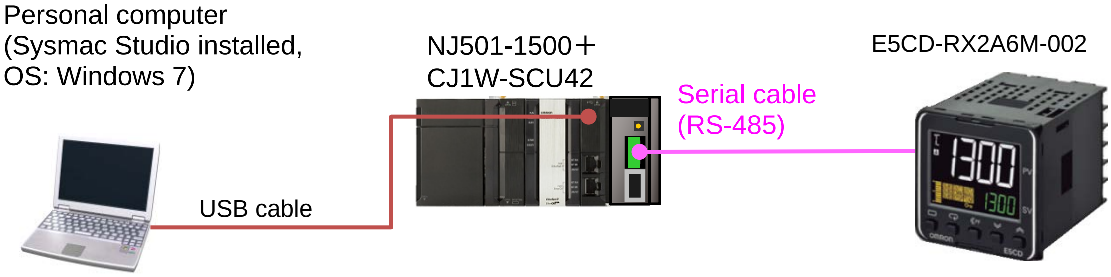
In this guide, the devices with models and versions listed in *5.2. Device Configuration* are used as examples of applicable devices to describe the procedures for connecting the devices and checking their connection.
You cannot use devices with versions lower than the versions listed in *5.2*.
To use the above devices with models not listed in *5.2.* or versions higher than those listed in *5.2.*, check the differences in the specifications by referring to the manuals before operating the devices.

Additional Information

This guide describes the procedures for establishing the network connection.
It does not provide information on operation, installation, wiring method, device functionality, or device operation, which is not related to the connection procedures.
Refer to the manuals or contact your OMRON representative.

5.2. Device Configuration

The hardware components to reproduce the connection procedures in this guide are as follows:



Manufacturer	Name	Model	Version
OMRON	Serial Communications Unit	CJ1W-SCU42	Ver.2.0
OMRON	NJ-series CPU Unit	NJ501-1500	Ver.1.16
OMRON	Power Supply Unit	NJ-PA3001	
OMRON	Sysmac Studio	SYSMAC-SE2□□□	Ver.1.21
OMRON	Sysmac Studio project file	P520_NJ_CWF485_OMRON_E5CD_V100.csm2	Ver.1.00
-	Personal computer (OS: Windows 7)	-	
-	USB cable (USB 2.0 type B connector)	-	
-	Serial cable (RS-485)	-	
OMRON	Digital Temperature Controller	E5CD-RX2A6M-002	

Precautions for Correct Use

Prepare the project file with a latest version beforehand.
To obtain the project file, contact your OMRON representative.

Precautions for Correct Use

Update Sysmac Studio to the version specified in this *Clause 5.2.* or to a higher version.
If you use a version higher than the one specified, the procedures and related screenshots described in *Section 7.* and the subsequent sections may not be applicable.
In that case, use the equivalent procedures described in this guide by referring to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504).

Precautions for Correct Use

Turn ON the terminating resistance switch on the Serial Communications Unit and connect 120 Ω (1/2 W) terminating resistance to the terminals of the Digital Temperature Controller at either end of the RS-422A/485 transmission path.

Additional Information

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

Additional Information

The system configuration in this guide uses USB for the connection between the personal computer and the Controller. For information on how to install the 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 parameters and cable wiring, which are set up in this guide.

6.1. Parameters

The following parameters are required to connect the Controller and the Digital Temperature Controller via serial communications.

Setting item	Controller (Serial Communications Unit)	Digital Temperature Controller
Device name	J01	-
Unit No.	0	-
Communications Unit No. (slave address)	-	1 (default)
Serial communications port (connection)	Port 1 (RS-422A/485)	-
Terminating resistance	Terminating resistance ON (TERM: ON）	-
2-wire or 4-wire	2-wire (WIRE: 2)	2-wire (fixed)
Serial communications mode	Serial Gateway	-
Data length (transmission character)	7 bits (default)	7 bits (default)
Stop bits	2 bits (default)	2 bits (default)
Parity (parity bit)	Even (default)	Even (default)
Baud rate	9,600 bps (default)	9,600 bps (default)
Communications method	-	CompoWay/F (default)
Send data wait time	-	20 ms (default)

Precautions for Correct Use

The connection procedure described in this guide assumes that the following Serial Communications Unit, port and setting values are used.

Model: CJ1W-SCU42
Serial communications port: Port 1
Unit No.: 0
Device name: J01

If you connect devices under different conditions, refer to *Section 9. Program* and create a program by changing the variable names and setting values.

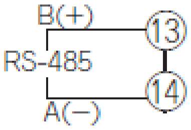
6.2. Cable Wiring

Refer to *Section 3 Installation and Wiring* of the *CJ-series Serial Communications Units Operation Manual for NJ-series CPU Unit* (Cat. No. W494) for details on cable wiring.
Check the connector configurations and pin assignments before wiring.

■Connector configuration and pin assignment

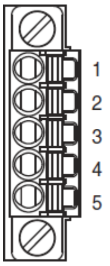
Digital Temperature Controller (E5CD) applicable connector: Terminal block

Pin No.	Signal name	Input/Output
1-12		
13	B(+)	Input/Output
14	A(-)	Input/Output
15-18		



Serial Communications Unit (CJ1W-SCU42) applicable connector: Terminal block

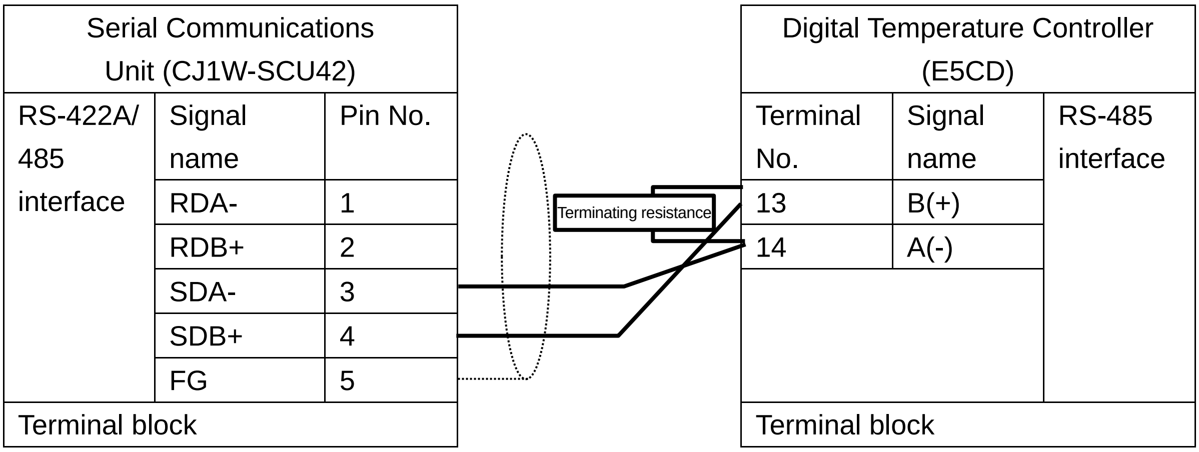
Pin No.	Symbol	Signal name	Input/Output
1 (See note 1.)	RDA	Receive data -	Input
2 (See note 1.)	RDB	Receive data +	Input
3 (See note 1.)	SDA	Send data -	Output
4 (See note 1.)	SDB	Send data +	Output
5 (See note 2.)	FG	Shield	-



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

2: Pin 5 (Shield) is connected to the GR terminal on the Power Supply Unit though the Serial Communications Unit. The cable shield can thus be grounded if you ground the GR terminal of the Power Supply Unit.

■Cable and pin assignment



Serial Communications Unit (CJ1W-SCU42)		
RS-422A/ 485 interface	Signal name	Pin No.
	RDA-	1
	RDB+	2
	SDA-	3
	SDB+	4
	FG	5
Terminal block		

Digital Temperature Controller (E5CD)		
Terminal No.	Signal name	RS-485 interface
13	B(+)	
14	A(-)	
Terminal block		

Connect 120 Ω (1/2 W) terminating resistance between B(+) and A(-) of the Digital Temperature Controller that is connected at the end of the network.

Additional Information

For information on the connector configurations and pin assignments of the other models, refer to their respective manuals.

7. Serial Communications Connection Procedure

This section describes the procedures for connecting the Controller to the Digital Temperature Controller via serial communications. The procedures for setting up the Controller and the Digital Temperature Controller in this guide are based on the factory default settings.
For the initialization, refer to *Section 8. Initialization Method*.

7.1. Work Flow

Take the following steps to connect the Controller and the Digital Temperature Controller via serial communications and to send or receive a message.

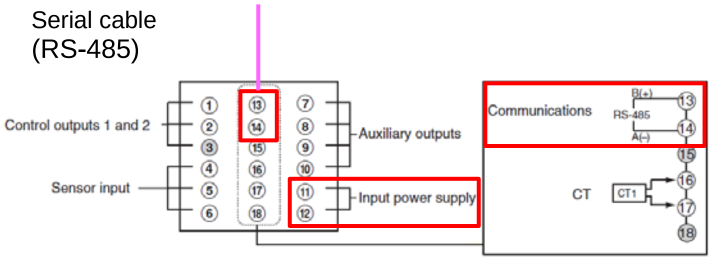
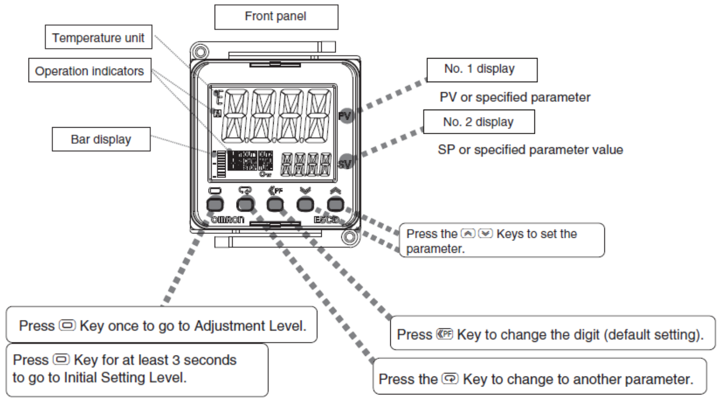
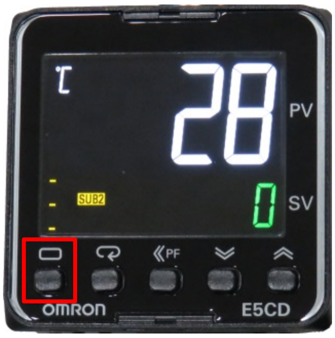
7.2. Digital Temperature Controller Setup	Set up the Digital Temperature Controller.
↓	
7.2.1. Parameter Settings	Set parameters for the Digital Temperature Controller.
↓	
7.3. Controller Setup	Set up the Controller.
↓	
7.3.1. Hardware Settings	Set the hardware switches on the Serial Communications Unit and connect the cables.
↓	
7.3.2. Starting Sysmac Studio and Importing the Project File	Start Sysmac Studio and import the project file.
↓	
7.3.3. Checking the Parameters and Building Programs	Check the parameters of the Serial Communications Unit, execute the program check on the project data, and build the project programs.
↓	
7.3.4. Going Online and Transferring the Project Data	Go online with Sysmac Studio and transfer the project data to the Controller.
↓	
7.3.5. Transferring the Unit Settings	Transfer the setting data of the Serial Communications Unit.
↓	
7.4. Serial Communication Status Check	Start the send/receive processing and confirm that serial communications performs normally.
↓	
7.4.1. Checking Received Data	Check that correct data is written to the variables of the Controller in Sysmac Studio.

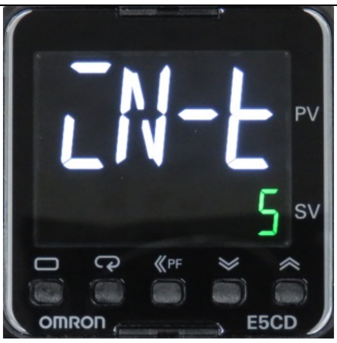
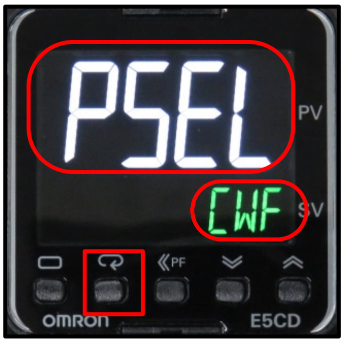
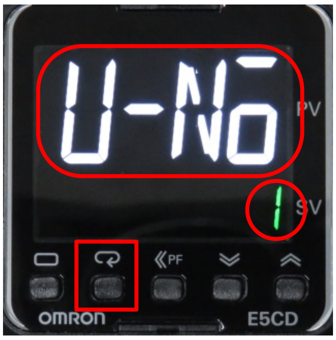
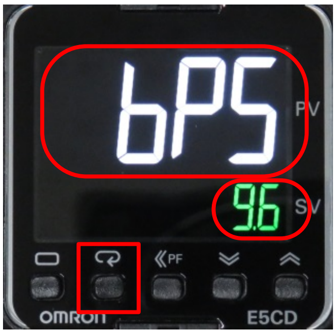
7.2. Digital Temperature Controller Setup

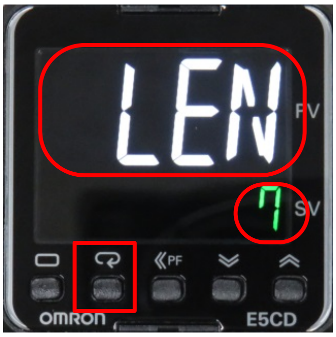
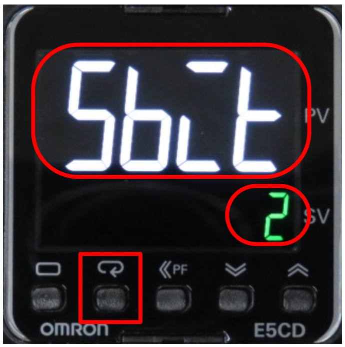
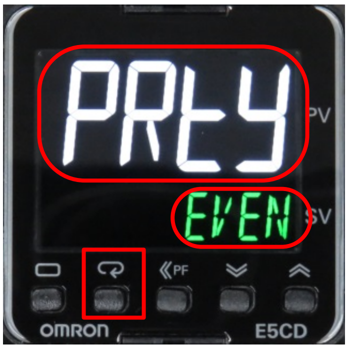
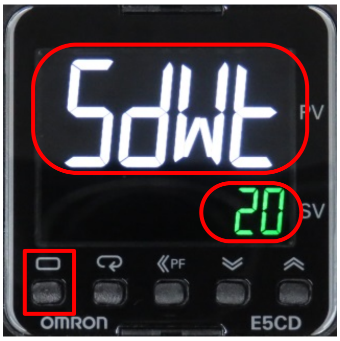
Set up the Digital Temperature Controller.

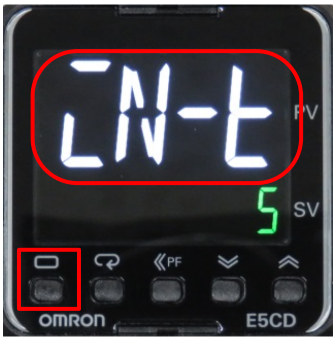
7.2.1. Parameter Settings

Set parameters for the Digital Temperature Controller.

1	Connect the power supply and a serial cable to the terminal block located on the back of Digital Temperature Controller. *Only the parameters in Communications Setting Level are described in this guide. If you use the parameters in Adjustment Level that is specific to each of the models, refer to their respective manuals.	Serial cable (RS-485)  Control outputs 1 and 2 Sensor input Auxiliary outputs Input power supply Communications B(+) RS-485 A(−) CT CT1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
2	Check the positions of each of the keys, No.1 and No. 2 displays and Operation indicators. In this guide, the keys are described as follows: □ (Level) Key ↺ (Mode) Key ▲ (Up) Key ▼ (Down) Key	 Front panel Temperature unit Operation indicators Bar display No. 1 display PV or specified parameter No. 2 display SP or specified parameter value PV SV Press the ▲ ▼ Keys to set the parameter. Press □ Key once to go to Adjustment Level. Press □ Key for at least 3 seconds to go to Initial Setting Level. Press PF Key to change the digit (default setting). Press the ↺ Key to change to another parameter. OMRON E5CD
3	Turn ON Digital Temperature Controller.	
4	The current temperature is displayed on No. 1 display once Digital Temperature Controller is turned ON. (Operation Level) Press the □ (Level) Key for at least 3 seconds.	 □ (Level) Key for at least 3 seconds

5	"IN-t" (Initial Setting Level) is displayed on No. 1 display. Press the □ (Level) Key again for less than 1 second.		□ (Level) Key
6	The display changes to Communications Setting Level. "PSEL" (Protocol Setting) and "CWF" (CompoWay/F) are displayed on No. 1 and No. 2 displays, respectively. *If the setting value is different, press the ︽ (Up) or ︾ (Down) Key to change the parameter. Press the ↩ (Mode) Key.		< Setting value> CWF / Mod (default: CWF) CWF: CompoWay/F Mod: Modbus-RTU ↩ (Mode) Key
7	"U-No" (Communications Unit No.) is displayed. Check that Communications Unit No. is 1. *If the setting value is different, change it in the same way as step 6. Press the ↩ (Mode) Key.		<Setting value> 0 to 99 (default: 1) ↩ (Mode) Key
8	"bPS" (Communications Baud Rate) is displayed. Check that Communications Baud Rate is 9.6 kbps. *If the setting value is different, change it in the same way as step 6. Press the ↩ (Mode) Key.		<Setting value> 9.6, 19.2, 38.4 or 57.6 kbps (default: 9.6) ↩ (Mode) Key

9	"LEN" (Communications Data Length) is displayed. Check that Communications Data Length is 7 bits. *If the setting value is different, change it in the same way as step 6. Press the ↩ (Mode) Key.		<Setting value> 7 or 8 bits (default: 7) ↩ (Mode) Key
10	"SbIt" (Communications Stop Bits) is displayed. Check that Communications Stop Bits is 2 bits. *If the setting value is different, change it in the same way as step 6. Press the ↩ (Mode) Key.		<Setting value> 1 or 2 bits (default: 2) ↩ (Mode) Key
11	"PRtY" (Communications Parity) is displayed. Check that Communications Parity is EVEN. *If the setting value is different, change it in the same way as step 6. Press the ↩ (Mode) Key.		<Setting value> NONE, EVEN or ODD (default: EVEN) ↩ (Mode) Key
12	"SdWt" (Send Data Wait Time) is displayed. Check that Send Data Wait Time is 20. *If the setting value is different, change it in the same way as step 6. Press the □ (Level) Key for less than 1 second.		<Setting value> 0 to 99 ms (default: 20) □ (Level) Key.

13	"ĪN-t" ((Initial Setting Level) is displayed. Press the □ (Level) Key for at least 1 second.	 □ (Level) Key for at least 1 second
14	The display returns to Operation Level as shown in step 4.	
15	Turn OFF Digital Temperature Controller.	

7.3. Controller Setup

Set up the Controller.

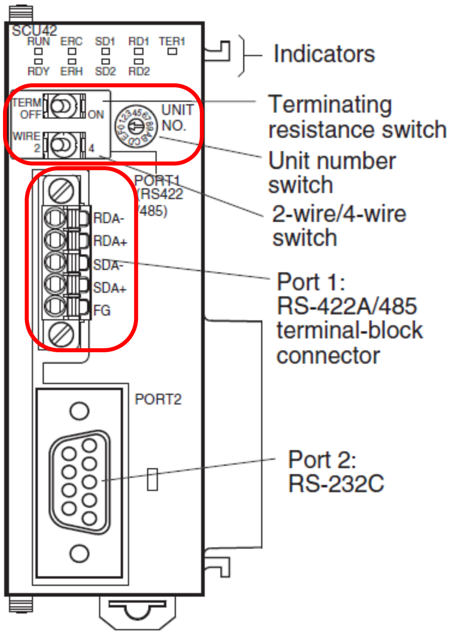
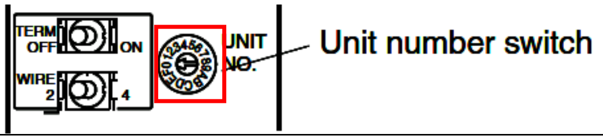
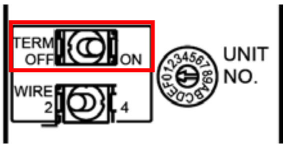
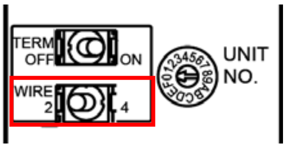
7.3.1. Hardware Settings

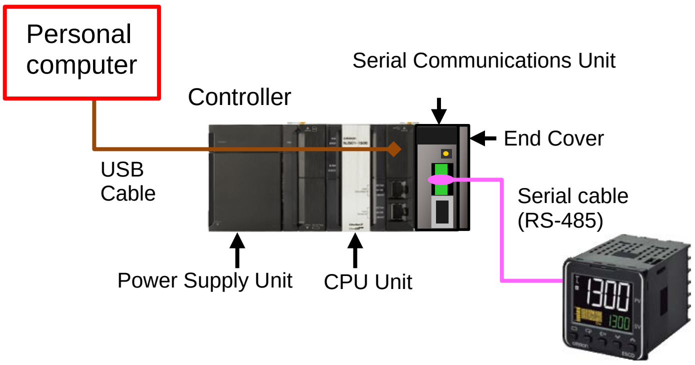
Set the hardware switches on the Serial Communications Unit and connect the cables.

Precautions for Correct Use

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

If it is ON, the settings described in the following steps and subsequent procedures may not be applicable.

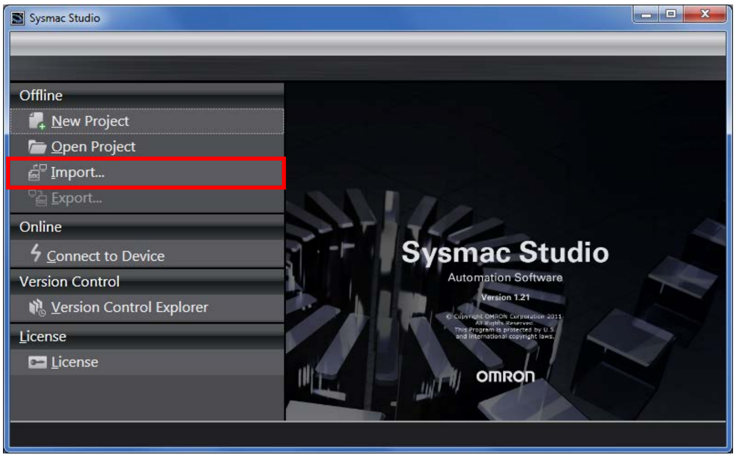
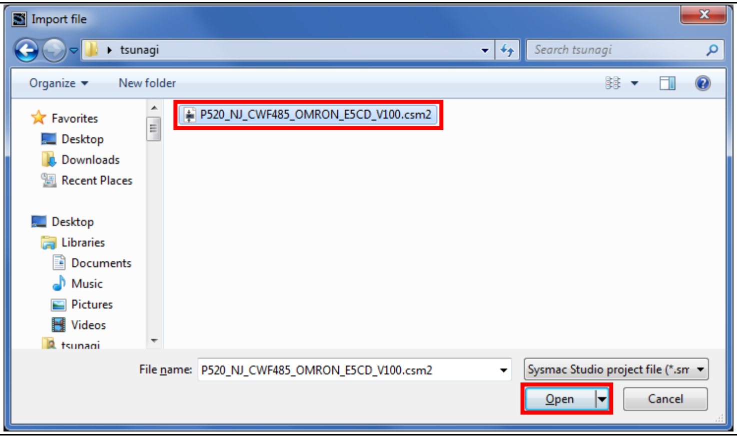
1	Make sure that Controller is powered OFF.	
2	Check the positions of the hardware switches and Port 1 on the front panel of Serial Communications Unit by referring to the figure on the right.	 SCU42 RUN ERC SD1 RD1 TER1 RDY ERH SD2 RD2 TERM OFF ON WIRE 2 4 UNIT NO. PORT1 (RS422 485) RDA- RDA+ SDA- SDA+ FG PORT2 Indicators Terminating resistance switch Unit number switch 2-wire/4-wire switch Port 1: RS-422A/485 terminal-block connector Port 2: RS-232C
3	Set Unit number switch to *0*. *The unit number is set to 0 as the factory default setting.	
4	Set Terminating resistance ON/OFF switch to *ON*. (Terminating resistance ON)	 ・TERM (Terminating resistance ON/OFF switch) OFF: Terminating resistance OFF ON: Terminating resistance ON
5	Set 2-wire or 4-wire switch to *2*. (2-wire)	 ・WIRE (2-wire or 4-wire switch) 2: 2-wire, 4: 4-wire

6	Connect Serial Communications Unit to Controller as shown on the right. Connect Digital Temperature Controller and Port 1 on Serial Communications Unit with the serial cable. Connect Personal computer and Controller with a USB cable.	 Personal computer Serial Communications Unit Controller End Cover USB Cable Serial cable (RS-485) Power Supply Unit CPU Unit

7.3.2. Starting Sysmac Studio and Importing the Project File

Start Sysmac Studio and import the project file.

Install Sysmac Studio and the USB driver on your personal computer beforehand.

1	Start Sysmac Studio. *If the User Account Control Dialog Box is displayed at start, make a selection to start Sysmac Studio.	
2	Click **Import**.	 Sysmac Studio Offline New Project Open Project Import... Export... Online Connect to Device Version Control Version Control Explorer License License Sysmac Studio Automation Software Version 1.21 omron
3	The Import file Dialog Box is displayed. Select *P520_NJ_CWF485_ OMRON_E5CD_V100.csm2* and click **Open**. *Obtain the project file from OMRON.	 Import file tsunagi Search tsunagi Organize New folder Favorites Desktop Downloads Recent Places Desktop Libraries Documents Music Pictures Videos tsunagi P520_NJ_CWF485_OMRON_E5CD_V100.csm2 Sysmac Studio project file (*.sm Open Cancel File name: P520_NJ_CWF485_OMRON_E5CD_V100.csm2

4 The P520_NJ_CWF485_OMRON_E5CD_V100 project is displayed.

The following panes are displayed in this window.

Left: Multiview Explorer

Top right: Toolbox

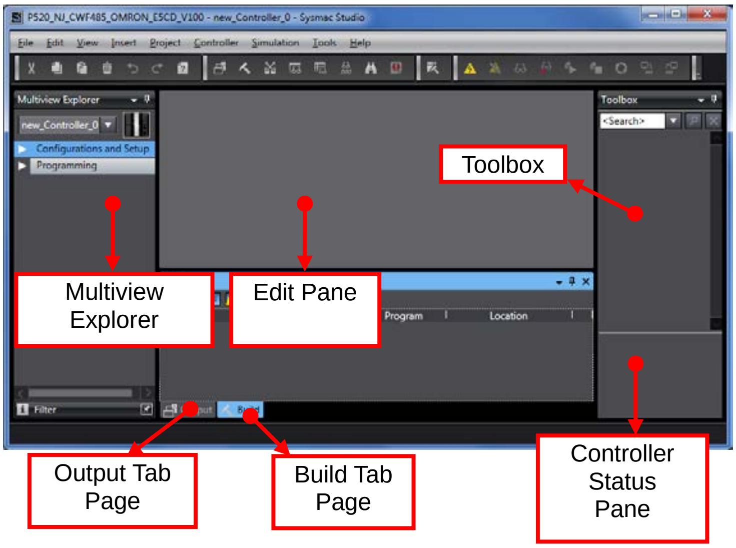
Bottom right: Controller Status Pane

Top middle: Edit Pane

The following tabs are displayed in the bottom middle of this window.

Output Tab Page

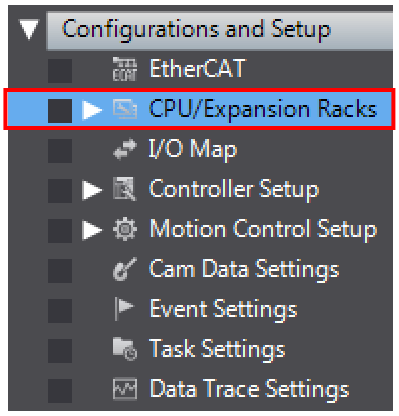
Build Tab Page



7.3.3. Checking the Parameters and Building Programs

Check the parameters of the Serial Communications Unit, execute the program check on the project data, and build the project programs.

1 Double-click **CPU/Expansion Racks** under **Configurations and Setup** in the Multiview Explorer.

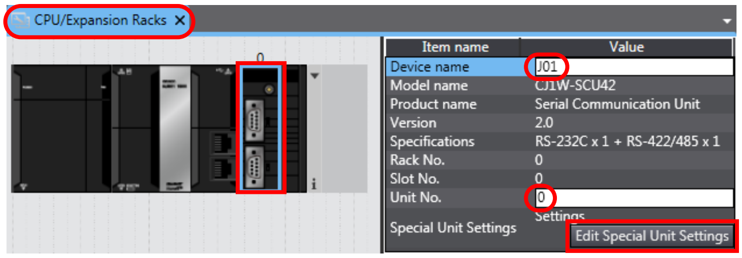


2 The CPU/Expansion Racks Tab Page is displayed in the Edit Pane.
Select Serial Communications Unit as shown on the right.

The setting values of CJ1W-SCU42 are displayed.
Check that the device name is J01 and that the unit No. is 0.

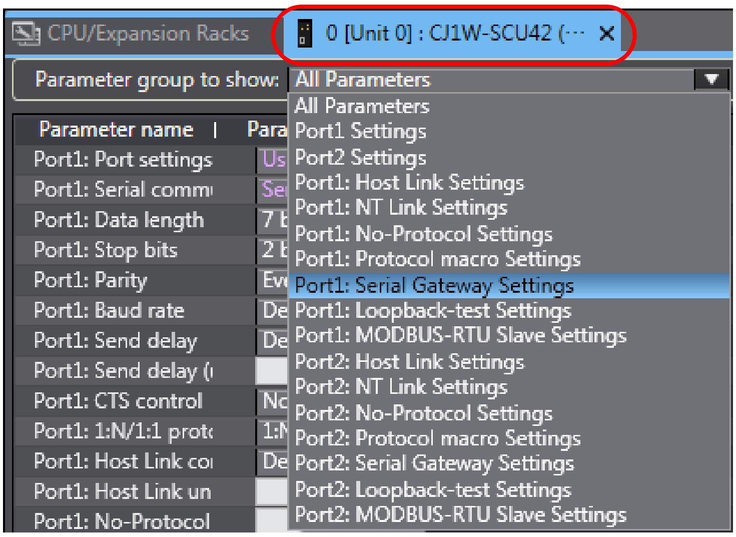
*If the setting values are
different from the above,
change the values.

Click **Edit Special Unit Settings**.



Item name	Value
Device name	J01
Model name	CJ1W-SCU42
Product name	Serial Communication Unit
Version	2.0
Specifications	RS-232C x 1 + RS-422/485 x 1
Rack No.	0
Slot No.	0
Unit No.	0
Special Unit Settings	Settings Edit Special Unit Settings

3 The 0 [Unit 0] : CJ1W-SCU42 (J01) Tab Page is displayed.
Select ***Port1: Serial Gateway Settings*** from the pull-down list of Parameter group to show.


CPU/Expansion Racks
0 [Unit 0] : CJ1W-SCU42 (···
All Parameters
Port1 Settings
Port2 Settings
Port1: Host Link Settings
Port1: NT Link Settings
Port1: No-Protocol Settings
Port1: Protocol macro Settings
Port1: Serial Gateway Settings
Port1: Loopback-test Settings
Port1: MODBUS-RTU Slave Settings
Port2: Host Link Settings
Port2: NT Link Settings
Port2: No-Protocol Settings
Port2: Protocol macro Settings
Port2: Serial Gateway Settings
Port2: Loopback-test Settings
Port2: MODBUS-RTU Slave Settings

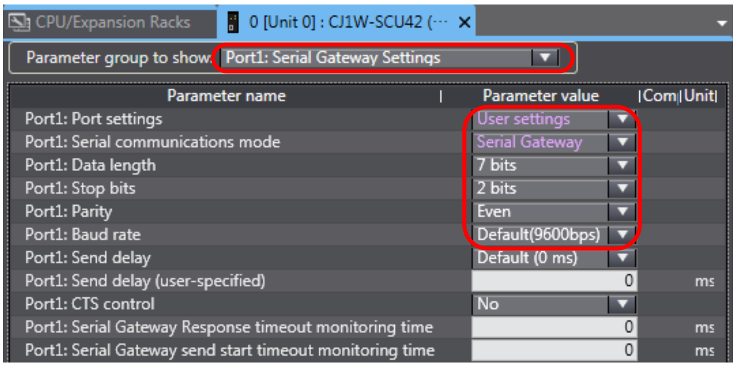
Parameter group to show: All Parameters

Parameter name	Para
Port1: Port settings	Us
Port1: Serial comm	Se
Port1: Data length	7 b
Port1: Stop bits	2 b
Port1: Parity	Ev
Port1: Baud rate	De
Port1: Send delay	De
Port1: Send delay (	
Port1: CTS control	No
Port1: 1:N/1:1 prot	1:N
Port1: Host Link co	De
Port1: Host Link un	
Port1: No-Protocol	

4 Parameter group to show is set to Port1: Serial Gateway Settings.
The setting items of Port1: Serial Gateway Settings are shown.

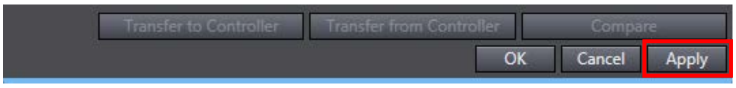
Check that Port1: Port settings is set to User settings and that the other settings are the same as those listed in *6.1. Parameters*.

*If the setting values are different, select the values from the pull-down list. After changing the values, click **Apply**.

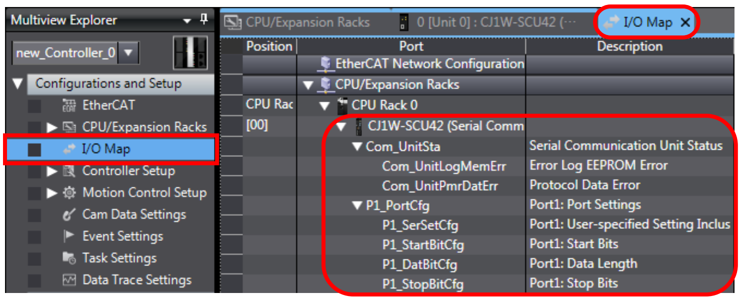


Parameter group to show: Port1: Serial Gateway Settings

Parameter name	Parameter value	Com|Unit|
Port1: Port settings	User settings	
Port1: Serial communications mode	Serial Gateway	
Port1: Data length	7 bits	
Port1: Stop bits	2 bits	
Port1: Parity	Even	
Port1: Baud rate	Default(9600bps)	
Port1: Send delay	Default (0 ms)	
Port1: Send delay (user-specified)	0	ms
Port1: CTS control	No	
Port1: Serial Gateway Response timeout monitoring time	0	ms
Port1: Serial Gateway send start timeout monitoring time	0	ms



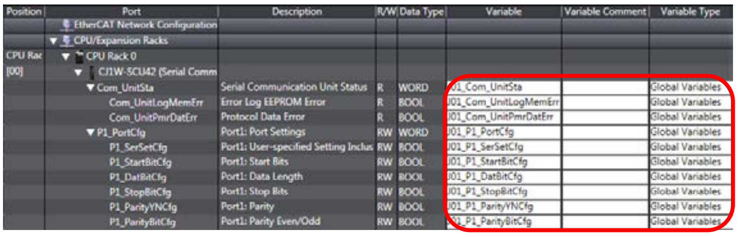
5 Double-click **I/O Map** under **Configurations and Setup** in the Multiview Explorer.
The I/O Map Tab Page is displayed, and then the parameters of Serial Communications Unit are listed.



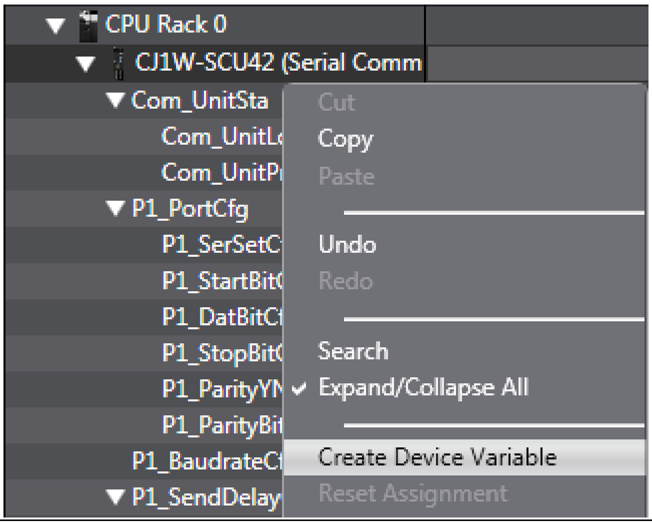
Position	Port	Description
	EtherCAT Network Configuration	
	CPU/Expansion Racks	
CPU Rac	CPU Rack 0	
[00]	CJ1W-SCU42 (Serial Comm	
	Com_UnitSta	Serial Communication Unit Status
	Com_UnitLogMemErr	Error Log EEPROM Error
	Com_UnitPmrDatErr	Protocol Data Error
	P1_PortCfg	Port1: Port Settings
	P1_SerSetCfg	Port1: User-specified Setting Inclus
	P1_StartBitCfg	Port1: Start Bits
	P1_DatBitCfg	Port1: Data Length
	P1_StopBitCfg	Port1: Stop Bits

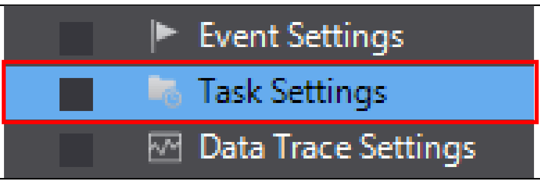
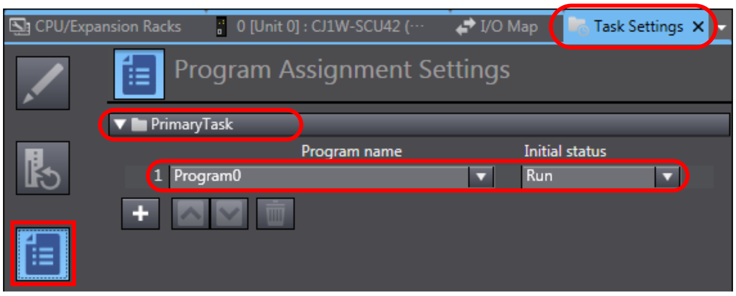
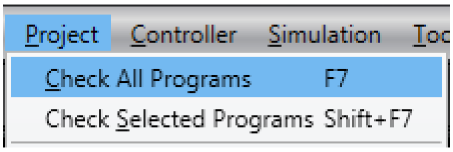
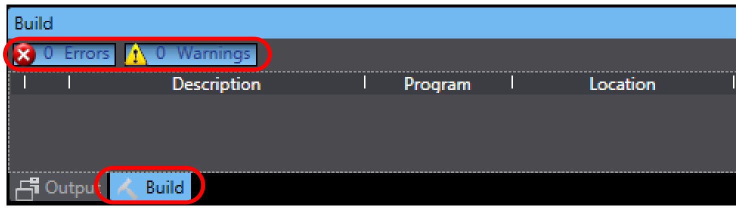
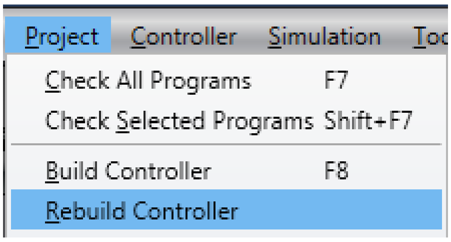
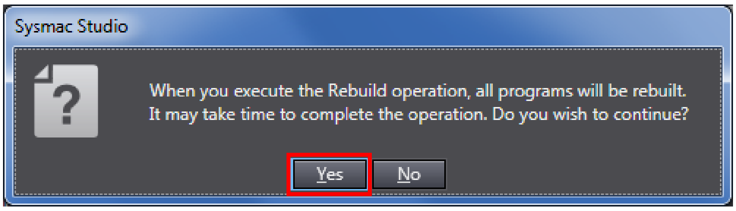
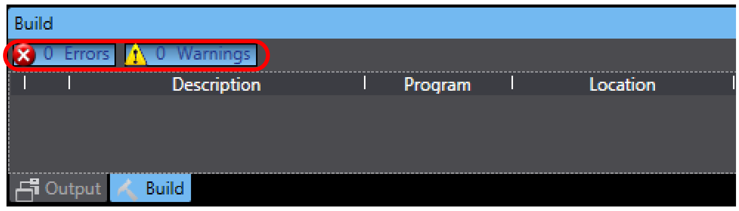
6 Check that the variable names in the *Variable* Column start with J01 and that "Global Variables" is set for each of them in the *Variable Type* Column.

*If the settings are different from the above, right-click **CJ1W-SCU42** and select ***Create Device Variable***.



Position	Port	Description	R/W	Data Type	Variable	Variable Comment	Variable Type
	EtherCAT Network Configuration						
	CPU/Expansion Racks						
CPU Rac	CPU Rack 0						
[00]	CJ1W-SCU42 (Serial Comm						
	Com_UnitSta	Serial Communication Unit Status	R	WORD	01_Com_UnitSta		Global Variables
	Com_UnitLogMemErr	Error Log EEPROM Error	R	BOOL	J01_Com_UnitLogMemErr		Global Variables
	Com_UnitPmrDatErr	Protocol Data Error	R	BOOL	J01_Com_UnitPmrDatErr		Global Variables
	P1_PortCfg	Port1: Port Settings	RW	WORD	J01_P1_PortCfg		Global Variables
	P1_SerSetCfg	Port1: User-specified Setting Inclus	RW	BOOL	J01_P1_SerSetCfg		Global Variables
	P1_StartBitCfg	Port1: Start Bits	RW	BOOL	J01_P1_StartBitCfg		Global Variables
	P1_DatBitCfg	Port1: Data Length	RW	BOOL	J01_P1_DatBitCfg		Global Variables
	P1_StopBitCfg	Port1: Stop Bits	RW	BOOL	J01_P1_StopBitCfg		Global Variables
	P1_ParityYNCfg	Port1: Parity	RW	BOOL	J01_P1_ParityYNCfg		Global Variables
	P1_ParityBitCfg	Port1: Parity Even/Odd	RW	BOOL	01_P1_ParityBitCfg		Global Variables



7	Double-click **Task Settings** under **Configurations and Setup** in the Multiview Explorer.	
8	The Task Settings Tab Page is displayed in the Edit Pane. Click the **Program Assignment Settings** Button. Check that Program0 is set in the *Primary Task* Field and that Run is set as the initial status.	 CPU/Expansion Racks 0 [Unit 0] : CJ1W-SCU42 (··· I/O Map Task Settings Program Assignment Settings PrimaryTask Program name: 1 Program0 Initial status: Run
9	Select ***Check All Programs*** from the Project Menu.	 Project Controller Simulation Toc Check All Programs F7 Check Selected Programs Shift+F7
10	The Build Tab Page is displayed. Check that "0 Errors" and "0 Warnings" are displayed.	 Build 0 Errors 0 Warnings Description Program Location Output Build
11	Select ***Rebuild Controller*** from the Project Menu.	 Project Controller Simulation Toc Check All Programs F7 Check Selected Programs Shift+F7 Build Controller F8 Rebuild Controller
12	A confirmation dialog box is displayed. Check the contents and click **Yes**.	 When you execute the Rebuild operation, all programs will be rebuilt. It may take time to complete the operation. Do you wish to continue?
13	Check that "0 Errors" and "0 Warnings" are displayed on the Build Tab Page.	 Build 0 Errors 0 Warnings Description Program Location Output Build

7.3.4. Going Online and Transferring the Project Data

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

⚠ WARNING

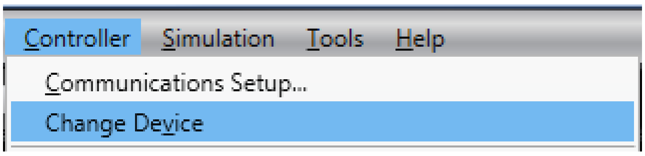
Always confirm safety at the destination node before you transfer a user program, configuration data, setup data, device variables, or values in memory used for CJ-series Units from Sysmac Studio. The devices or machines may perform unexpected operation regardless of the operating mode of the CPU Unit.

⚠ Caution

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

1 Turn ON Controller and Digital Temperature Controller.

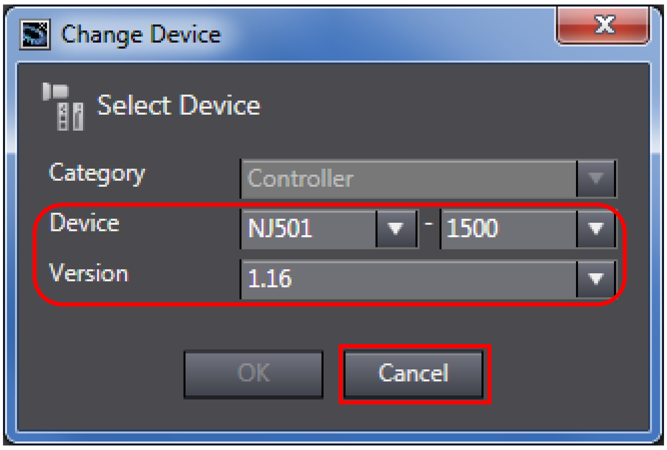
2 Select ***Change Device*** from the Controller Menu.



3 The Change Device Dialog Box is displayed.
Check that the *Device* and *Version* Fields are set as shown on the right.

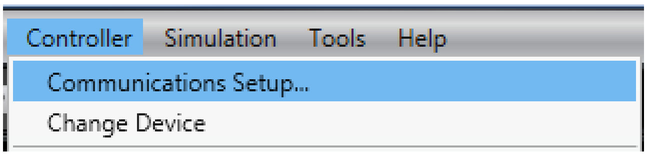
Click **Cancel**.

*If the settings are different, select the setting items from the pull-down list, and click **OK**.



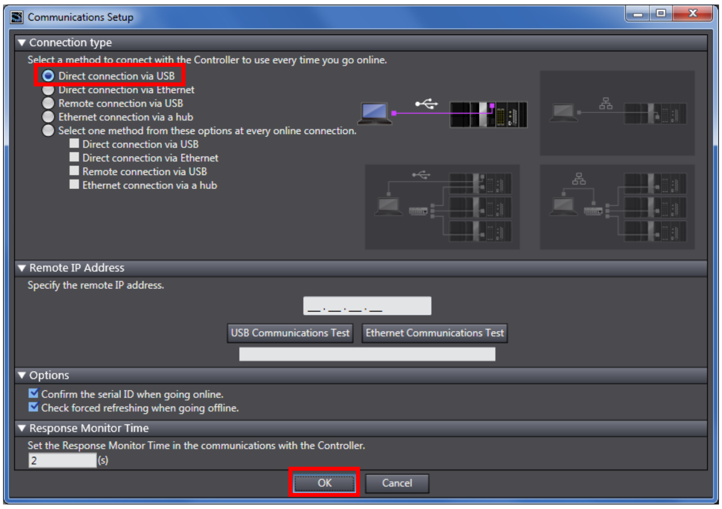
Category: Controller
Device: NJ501 - 1500
Version: 1.16

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



5 The Communications Setup Dialog Box is displayed. Select *Direct connection via USB* in the *Connection type* Field.

Click **OK**.



Communications Setup

▼ Connection type

Select a method to connect with the Controller to use every time you go online.

- ● Direct connection via USB
- ○ Direct connection via Ethernet
- ○ Remote connection via USB
- ○ Ethernet connection via a hub
- ○ Select one method from these options at every online connection.
  - □ Direct connection via USB
  - □ Direct connection via Ethernet
  - □ Remote connection via USB
  - □ Ethernet connection via a hub

▼ Remote IP Address

Specify the remote IP address.

__.__.__.__

USB Communications Test | Ethernet Communications Test

▼ Options

- ☑ Confirm the serial ID when going online.
- ☑ Check forced refreshing when going offline.

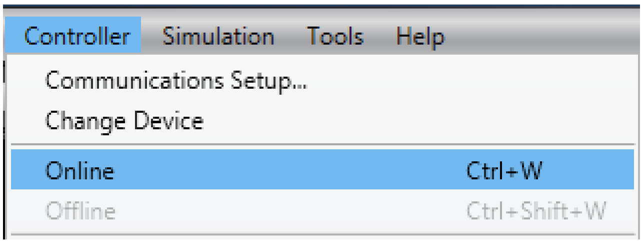
▼ Response Monitor Time

Set the Response Monitor Time in the communications with the Controller.

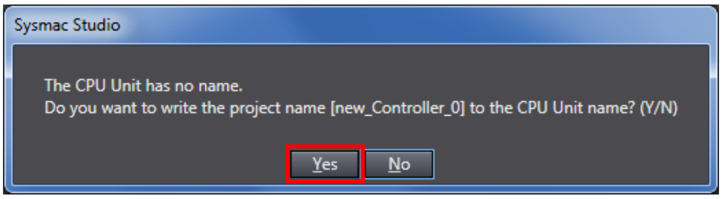
2 (s)

OK | Cancel

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



The dialog box on the right is displayed, check the contents and click **Yes**.



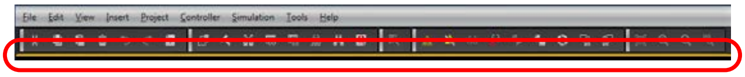
Sysmac Studio

The CPU Unit has no name.
Do you want to write the project name [new_Controller_0] to the CPU Unit name? (Y/N)

Yes | No

*The contents of the dialog box vary depending on the status of Controller. Check the contents and click on an appropriate button to proceed with the processing.

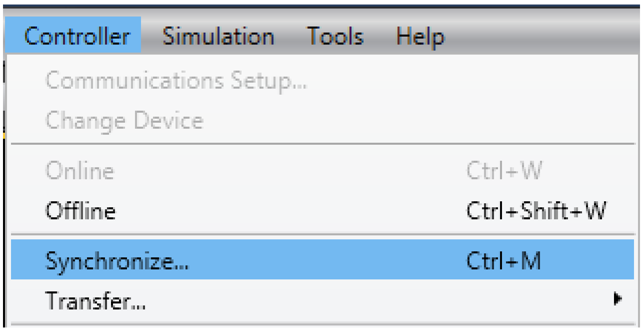
7 When an online connection is established, a yellow line is displayed under the toolbar.



Additional Information

For details on the online connections to the Controller, refer to *Section 6. Online Connections to a Controller* of the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504).

8 Select ***Synchronize*** from the Controller Menu.

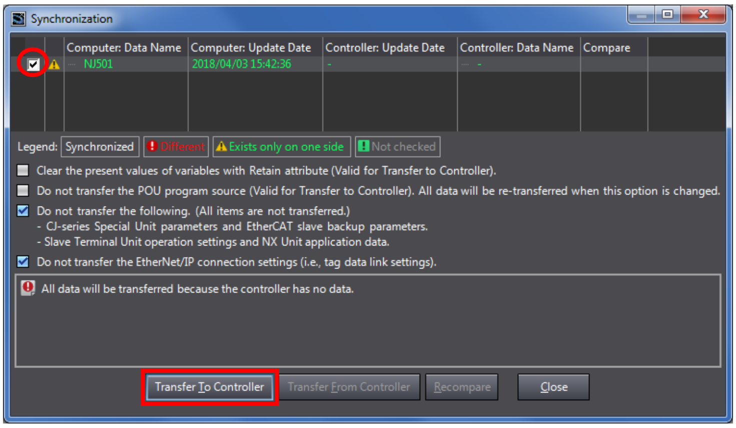


9 The Synchronization Dialog Box is displayed.

Check that the data to transfer (NJ501 shown on the right) is selected.

Click **Transfer To Controller**.

*After executing "Transfer To Controller", the Sysmac Studio data is transferred to Controller, and the data is synchronized.



Synchronization

		Computer: Data Name	Computer: Update Date	Controller: Update Date	Controller: Data Name	Compare
☑	⚠	NJ501	2018/04/03 15:42:36	-	-	

Legend: Synchronized | ! Different | ⚠ Exists only on one side | ! Not checked

☐ Clear the present values of variables with Retain attribute (Valid for Transfer to Controller).

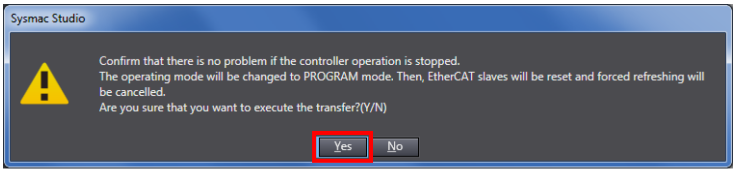
☐ Do not transfer the POU program source (Valid for Transfer to Controller). All data will be re-transferred when this option is changed.

☑ Do not transfer the following. (All items are not transferred.)
- CJ-series Special Unit parameters and EtherCAT slave backup parameters.
- Slave Terminal Unit operation settings and NX Unit application data.

☑ Do not transfer the EtherNet/IP connection settings (i.e., tag data link settings).

All data will be transferred because the controller has no data.

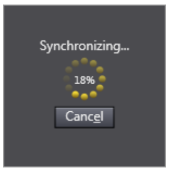
10 A confirmation dialog box is displayed. Confirm that there is no problem, and click **Yes**.



Sysmac Studio

Confirm that there is no problem if the controller operation is stopped.
The operating mode will be changed to PROGRAM mode. Then, EtherCAT slaves will be reset and forced refreshing will be cancelled.
Are you sure that you want to execute the transfer?(Y/N)

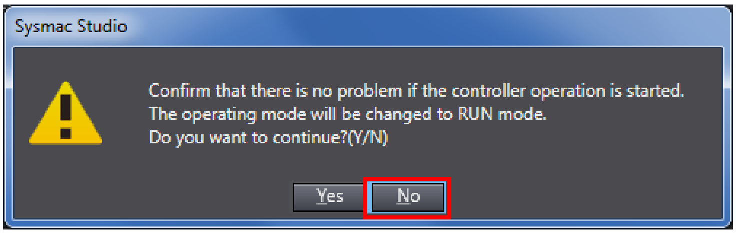
A screen is displayed stating "Synchronizing".



Synchronizing...

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

*Do not return to RUN mode.



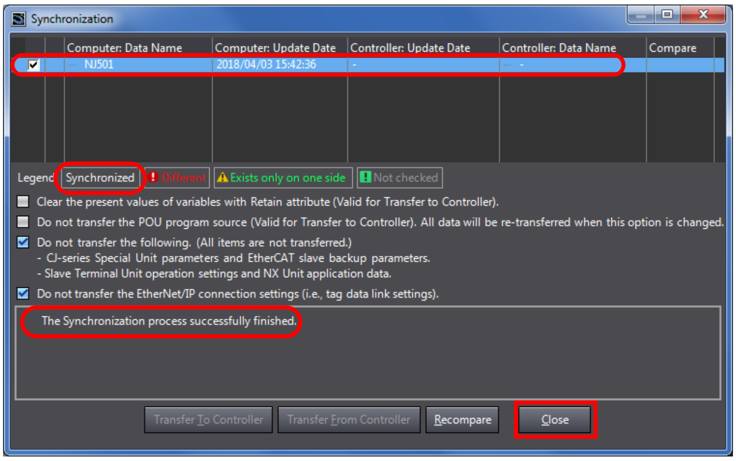
Sysmac Studio

Confirm that there is no problem if the controller operation is started.
The operating mode will be changed to RUN mode.
Do you want to continue?(Y/N)

11 As shown in the figure on the right, the font color that is used to display the synchronized data changes to the same color as the one used to specify "Synchronized". Check that a message is displayed stating "The Synchronization process successfully finished". Confirm that there is no problem, and click **Close**.

*When the Sysmac Studio project data coincides with the Controller data, a message is displayed stating "The synchronization process successfully finished".

*If the synchronization fails, check the wiring and repeat from step 1.



	Computer: Data Name	Computer: Update Date	Controller: Update Date	Controller: Data Name	Compare
☑	NJ501	2018/04/03 15:42:36	-	-	

□ Clear the present values of variables with Retain attribute (Valid for Transfer to Controller).

□ Do not transfer the POU program source (Valid for Transfer to Controller). All data will be re-transferred when this option is changed.

☑ Do not transfer the following. (All items are not transferred.)
- CJ-series Special Unit parameters and EtherCAT slave backup parameters.
- Slave Terminal Unit operation settings and NX Unit application data.

☑ Do not transfer the EtherNet/IP connection settings (i.e., tag data link settings).

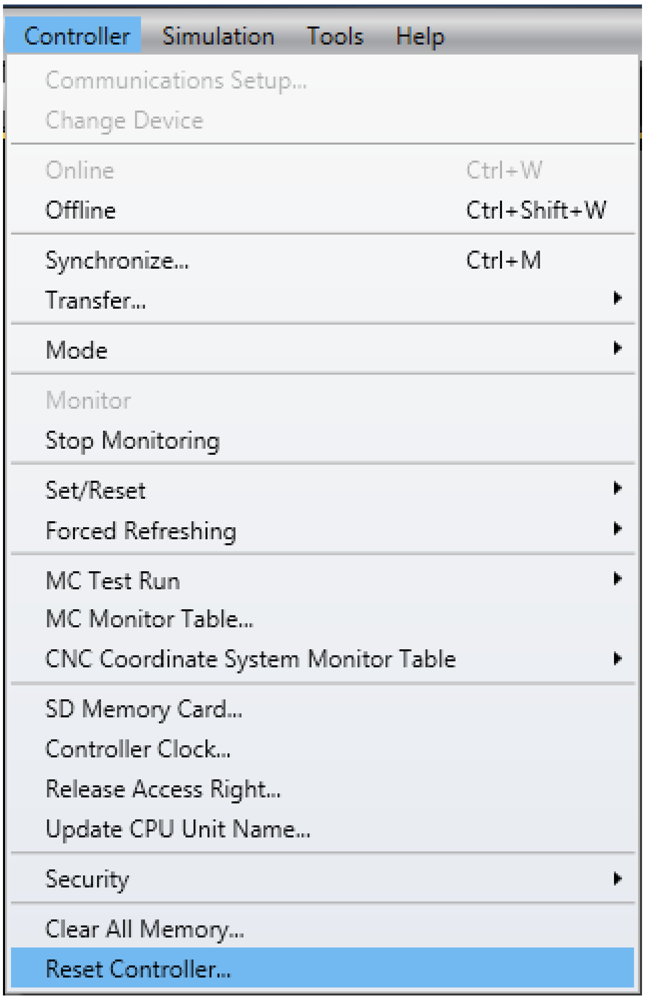
The Synchronization process successfully finished.

Precautions for Correct Use

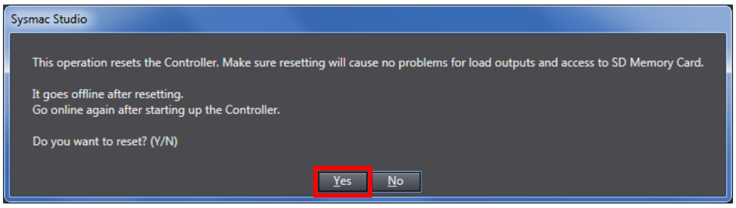
Always confirm safety before you reset the Controller or any components in step 13 and subsequent steps.

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

*"Reset Controller" cannot be selected when the operating mode of Controller is RUN Mode. In this case, select ***Mode*** - ***PROGRAM Mode*** from the Controller Menu to change to PROGRAM mode, and then perform this step.



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

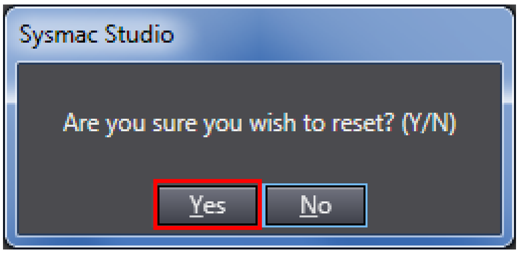

Sysmac Studio

This operation resets the Controller. Make sure resetting will cause no problems for load outputs and access to SD Memory Card.

It goes offline after resetting.
Go online again after starting up the Controller.

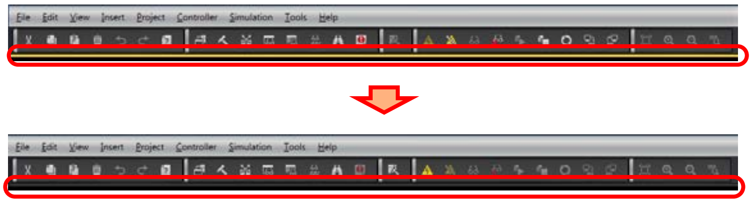
Do you want to reset? (Y/N)

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


Sysmac Studio

Are you sure you wish to reset? (Y/N)

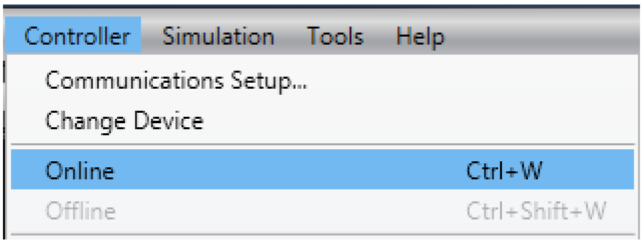
15 The Controller is reset, and Sysmac Studio goes offline. The yellow line under the toolbar disappears.


File Edit View Insert Project Controller Simulation Tools Help
File Edit View Insert Project Controller Simulation Tools Help

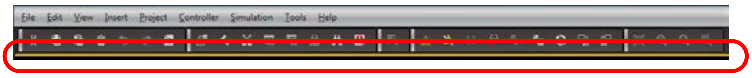
7.3.5. Transferring the Unit Settings

Transfer the setting data of the Serial Communications Unit.

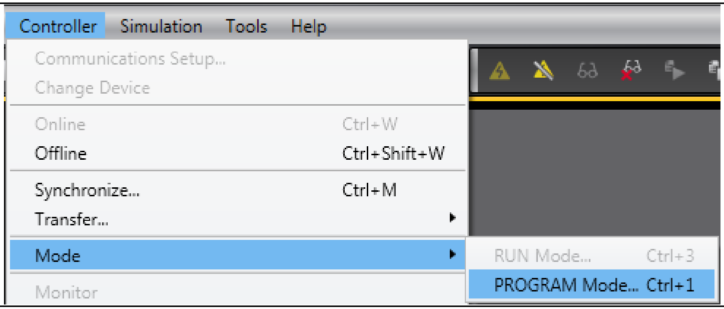
1 Select ***Online*** from the Controller Menu.



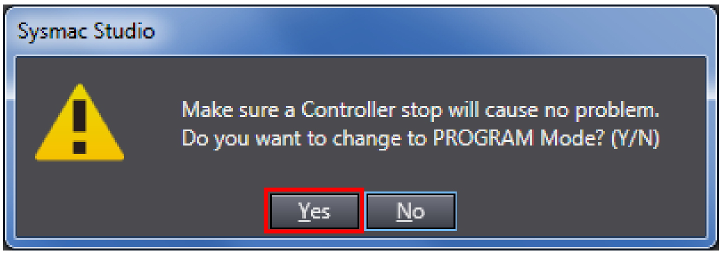
2 When an online connection is established, a yellow line is displayed under the toolbar.



3 Select ***Mode - PROGRAM Mode*** from the Controller Menu.

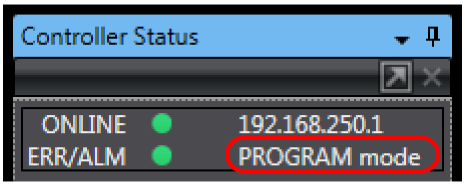


4 A confirmation dialog box is displayed. Confirm that there is no problem, and click **Yes**.



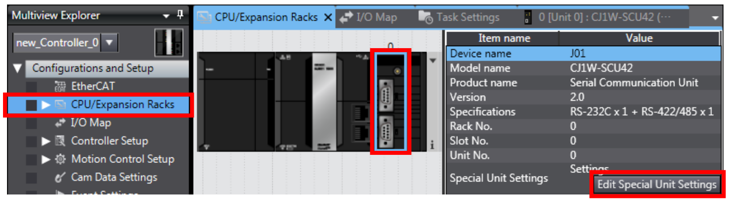
Make sure a Controller stop will cause no problem.
Do you want to change to PROGRAM Mode? (Y/N)

5 PROGRAM mode is displayed in the Controller Status Pane.



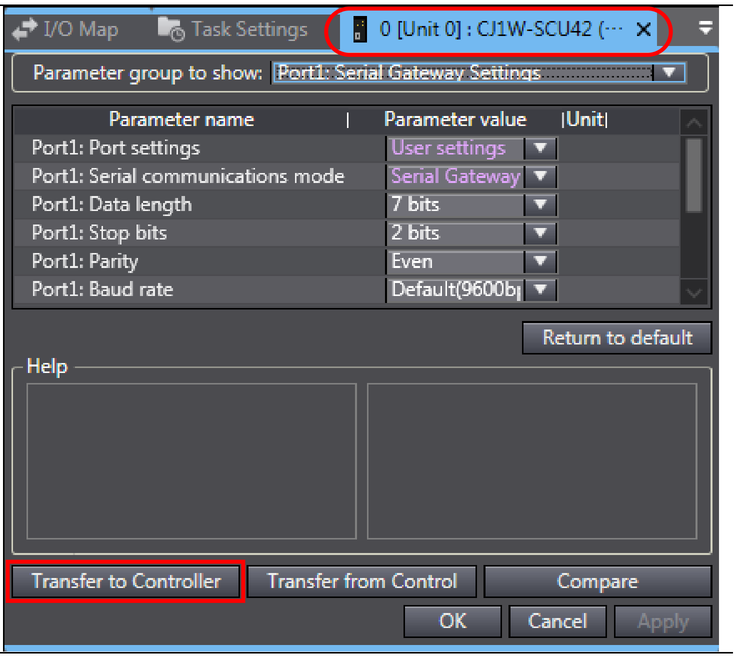
ONLINE ● 192.168.250.1
ERR/ALM ● PROGRAM mode

6 Double-click **CPU/Expansion Racks** under **Configurations and Setup** in the Multiview Explorer.
Select Serial Communications Unit displayed on the CPU Rack configuration.
Click **Edit Special Unit Settings**.



Item name	Value
Device name	J01
Model name	CJ1W-SCU42
Product name	Serial Communication Unit
Version	2.0
Specifications	RS-232C x 1 + RS-422/485 x 1
Rack No.	0
Slot No.	0
Unit No.	0
Special Unit Settings	Settings Edit Special Unit Settings

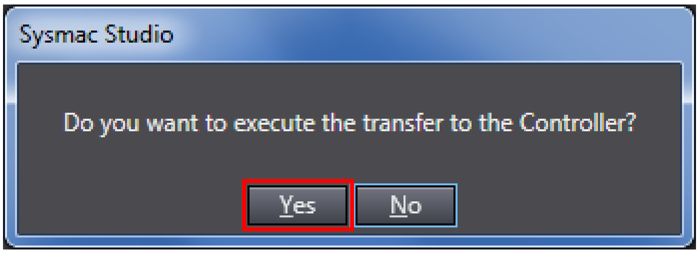
7 The 0 [Unit 0] : CJ1W-SCU42 (J01) Tab Page is displayed.
Click **Transfer to Controller**.



Parameter group to show: Port1: Serial Gateway Settings

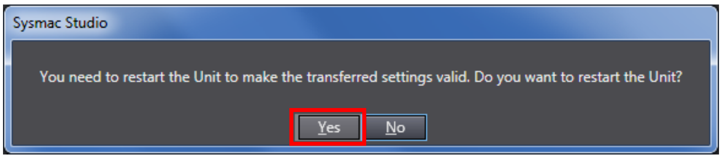
Parameter name	Parameter value	Unit
Port1: Port settings	User settings	
Port1: Serial communications mode	Serial Gateway	
Port1: Data length	7 bits	
Port1: Stop bits	2 bits	
Port1: Parity	Even	
Port1: Baud rate	Default(9600b|	

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



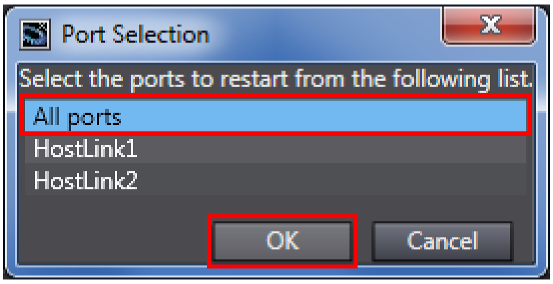
Do you want to execute the transfer to the Controller?

A dialog box is displayed indicating that transferring is being performed. After that, the dialog box on the right is displayed. Check the contents and click **Yes**.



You need to restart the Unit to make the transferred settings valid. Do you want to restart the Unit?

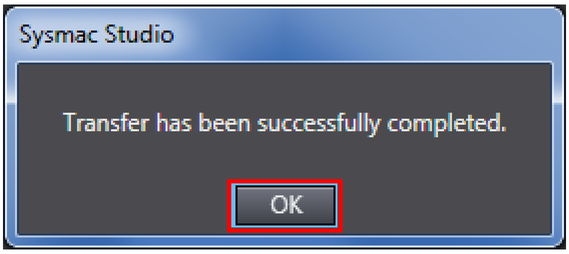
9 The Port Selection Dialog Box is displayed.
Select *All ports* and click **OK**.



Select the ports to restart from the following list.

- All ports
- HostLink1
- HostLink2

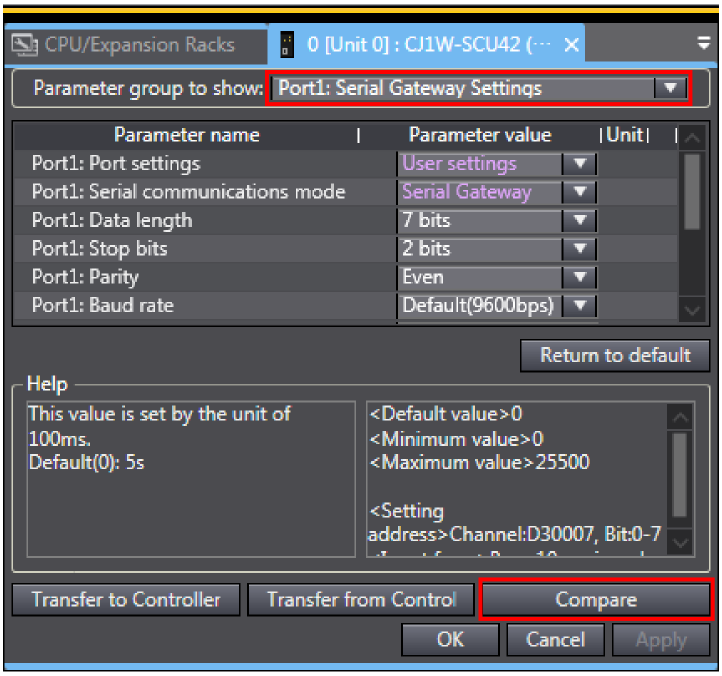
10 A confirmation dialog box is displayed. Check the contents and click **OK**.



Transfer has been successfully completed.

11 Select ***Port1: Serial Gateway Settings*** from the pull-down list of Parameter group to show.

Click **Compare**.



Parameter group to show: Port1: Serial Gateway Settings

Parameter name	Parameter value	Unit
Port1: Port settings	User settings	
Port1: Serial communications mode	Serial Gateway	
Port1: Data length	7 bits	
Port1: Stop bits	2 bits	
Port1: Parity	Even	
Port1: Baud rate	Default(9600bps)	

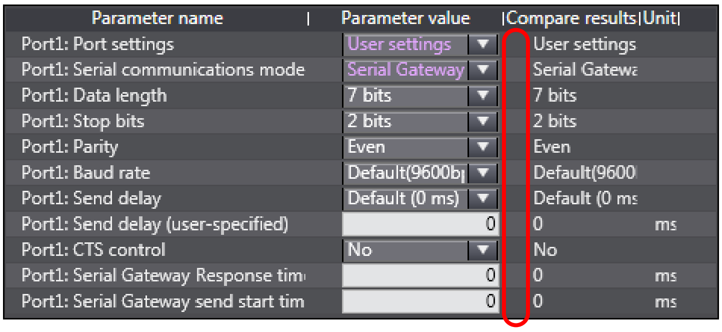
Help

This value is set by the unit of 100ms.
Default(0): 5s

<Default value>0
<Minimum value>0
<Maximum value>25500

<Setting address>Channel:D30007, Bit:0-7

12 Check that "≠" (mismatch) is not shown within the red oval in the figure on the right.



Parameter name	Parameter value		Compare results	Unit
Port1: Port settings	User settings		User settings	
Port1: Serial communications mode	Serial Gateway		Serial Gatewa	
Port1: Data length	7 bits		7 bits	
Port1: Stop bits	2 bits		2 bits	
Port1: Parity	Even		Even	
Port1: Baud rate	Default(9600b		Default(9600	
Port1: Send delay	Default (0 ms)		Default (0 ms	
Port1: Send delay (user-specified)	0		0	ms
Port1: CTS control	No		No	
Port1: Serial Gateway Response tim	0		0	ms
Port1: Serial Gateway send start tim	0		0	ms

7.4. Serial Communication Status Check

Start the send/receive processing and confirm that serial communications performs normally.

⚠ Caution

If you change the variable values on a Watch Tab Page when Sysmac Studio is online with the CPU Unit, the devices connected to the Controller may operate regardless of the operating mode of the CPU Unit.
Always ensure safety before you change the variable values on a Watch Tab Page when Sysmac Studio is online with the CPU Unit.

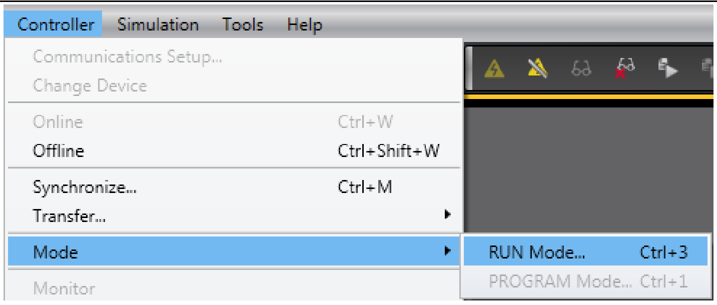
Precautions for Correct Use

Check that the serial cable is connected before performing the following procedure.
If not, turn OFF both devices, and then connect the serial cable.

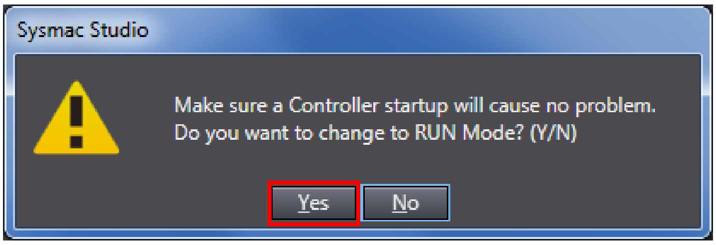
7.4.1. Checking Received Data

Check that correct data is written to the variables of the Controller in Sysmac Studio.

1 Select ***Mode*** - ***RUN Mode*** from the Controller Menu.

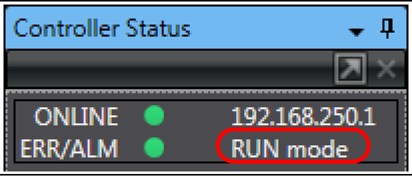


2 The dialog box on the right is displayed. Confirm that there is no problem, and click **Yes**.


Sysmac Studio

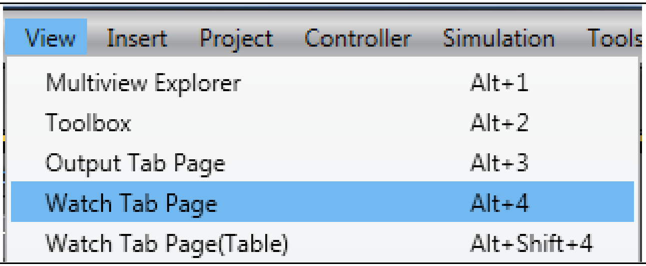
Make sure a Controller startup will cause no problem.
Do you want to change to RUN Mode? (Y/N)

3 RUN mode is displayed in the Controller Status Pane.

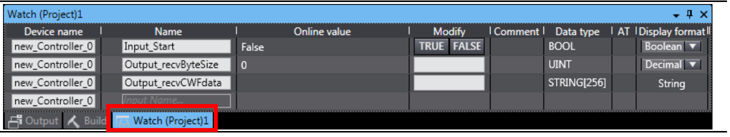

Controller Status

ONLINE ● 192.168.250.1
ERR/ALM ● RUN mode

4 Select ***Watch Tab Page*** from the View Menu.



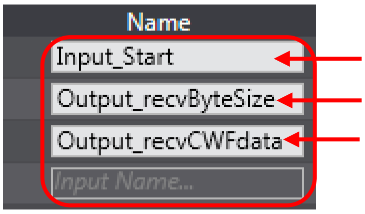
5 Select the **Watch (Project)1** Tab.


Watch (Project)1

Device name	Name	Online value	Modify	Comment	Data type	AT	Display format
new_Controller_0	Input_Start	False	TRUE FALSE		BOOL		Boolean ▼
new_Controller_0	Output_recvByteSize	0			UINT		Decimal ▼
new_Controller_0	Output_recvCWFdata				STRING[256]		String
new_Controller_0	Input Name...						

6 Check that the variables shown on the right are displayed in the *Name* Column.

*Click *Input Name* to add a variable.



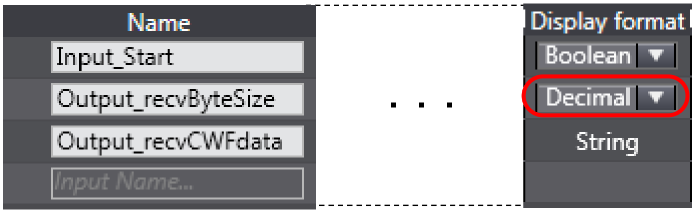
Communications start switch

The number of data bytes

String data

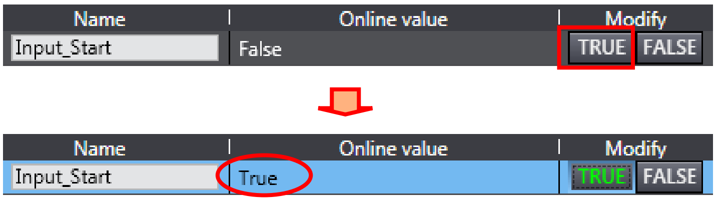
7 Check that the display format of *Output_recvByteSize* is Decimal.

*If not, select ***Decimal*** from the pull-down list.



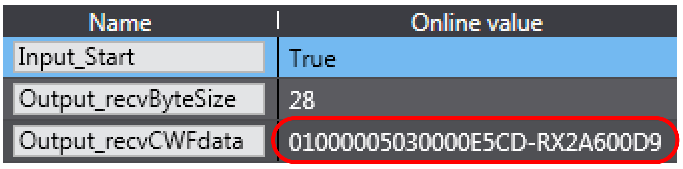
8 Click **TRUE** in the *Modify* Column for *Input_Start*.

The online value of *Input_Start* changes to True, and the communications processing starts.



9 Check received data on the Watch Tab Page of Sysmac Studio.

*The example on the right shows that the online values of *Output_recvByteSize* (the number of data bytes) and *Output_recvCWFdata* (string data) are 28 bytes and “01 00 00 0503 0000 E5CD-RX2A6 00D9”, respectively.
The response data varies depending on the device used.



Name	Online value
Input_Start	True
Output_recvByteSize	28
Output_recvCWFdata	0100000503000E5CD-RX2A600D9

Response data

01= Node No.

00= Subaddress

00= End code

0503= Command (MRC, SRC)

0000= Response (MRES, SRES)

E5CD-RX2A6 = Receive data (Controller Attributes)

00D9 = Buffer size

8. Initialization Method

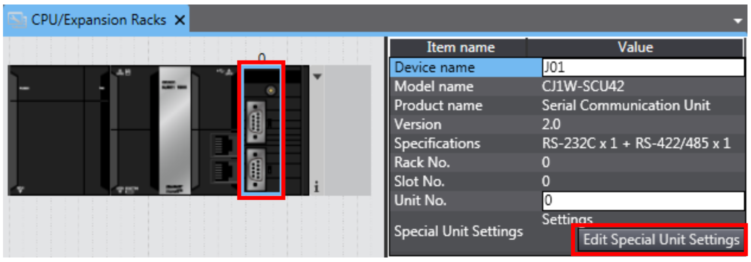
The setting procedures in this guide are based on the factory default settings.
Some settings may not be applicable unless you use the devices with the factory default settings.

8.1. Initializing a Controller

To initialize the settings of a Controller, it is necessary to initialize a Serial Communications Unit and a CPU Unit. Change the operating mode of the Controller to PROGRAM mode before the initialization.

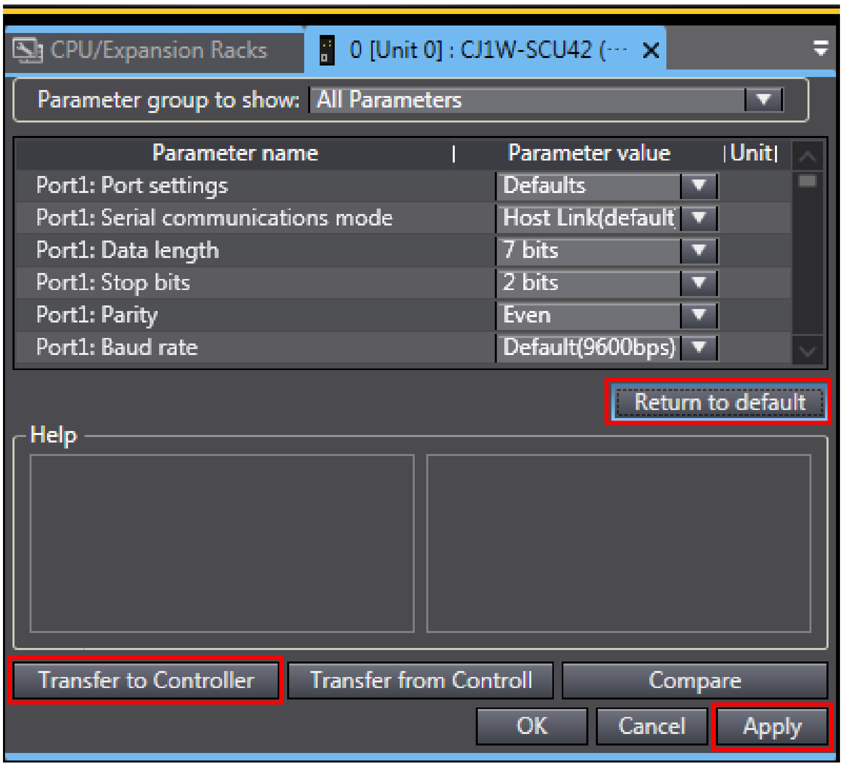
8.1.1. Serial Communications Unit

To initialize the settings of a Serial Communications Unit, select the displayed Serial Communications Unit (CJ1W-SCU42) and click **Edit Special Unit Settings** on the CPU/Expansion Racks Tab Page of Sysmac Studio.



Item name	Value
Device name	J01
Model name	CJ1W-SCU42
Product name	Serial Communication Unit
Version	2.0
Specifications	RS-232C x 1 + RS-422/485 x 1
Rack No.	0
Slot No.	0
Unit No.	0
Special Unit Settings	Settings Edit Special Unit Settings

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



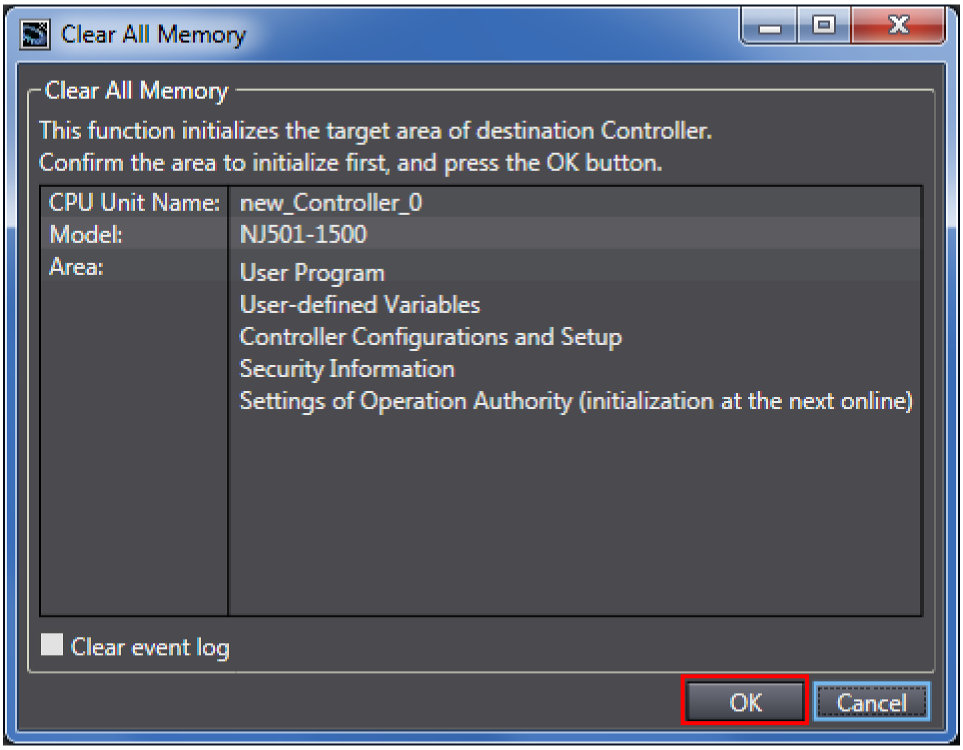
Parameter group to show: All Parameters

Parameter name	Parameter value	Unit
Port1: Port settings	Defaults	
Port1: Serial communications mode	Host Link(default	
Port1: Data length	7 bits	
Port1: Stop bits	2 bits	
Port1: Parity	Even	
Port1: Baud rate	Default(9600bps)	

8.1.2. CPU Unit

To initialize the settings of a CPU Unit, select ***Clear All Memory*** from the Controller Menu in Sysmac Studio.

The Clear All Memory Dialog Box is displayed. Check the contents and click **OK**.



Clear All Memory

This function initializes the target area of destination Controller.
Confirm the area to initialize first, and press the OK button.

CPU Unit Name:	new_Controller_0
Model:	NJ501-1500
Area:	User Program User-defined Variables Controller Configurations and Setup Security Information Settings of Operation Authority (initialization at the next online)

□ Clear event log

8.2. Initializing a Digital Temperature Controller

To initialize the settings of a Digital Temperature Controller, refer to *Parameter Initialization* in *6-8 Advanced Function Setting Level* of the *Digital Temperature Controllers User's Manual* (Cat. No. H224/H174/H185).

9. Program

This section describes the details on the program in the project file that is used in this guide.

9.1. Overview

The following explains the specifications and functions of the program that are used to check the connection between the Digital Temperature Controller (hereinafter referred to as the "Destination Device") and the Controller (Serial Communications Unit (hereinafter referred to as the "SCU")).

This program performs communications via CompoWay/F using the Serial Gateway function of the SCU, to send/receive the "Read Controller Attributes" command to/from the Destination Device and detect a normal end or an error end.
A normal end of the send/receive processing means a normal end of communications via CompoWay/F.
An error end means an error end of communications via CompoWay/F and an error of the Destination Device (identified in the response data from the Destination Device).

Here, the prefix "10#" (possible to omit) is added to decimal data and the prefix "16#" is added 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)
The prefix "<data type>#" is also added to decimal or hexadecimal data when specifying a specific data type. (e.g.,"WORD#16#03E8")

Additional Information

OMRON has confirmed that normal communications can be performed using this program under the conditions of *5.2. Device Configuration*. However, we do not guarantee the normal operation under disturbances such as electrical noise or device performance variation.

9.1.1. Outline of Processing

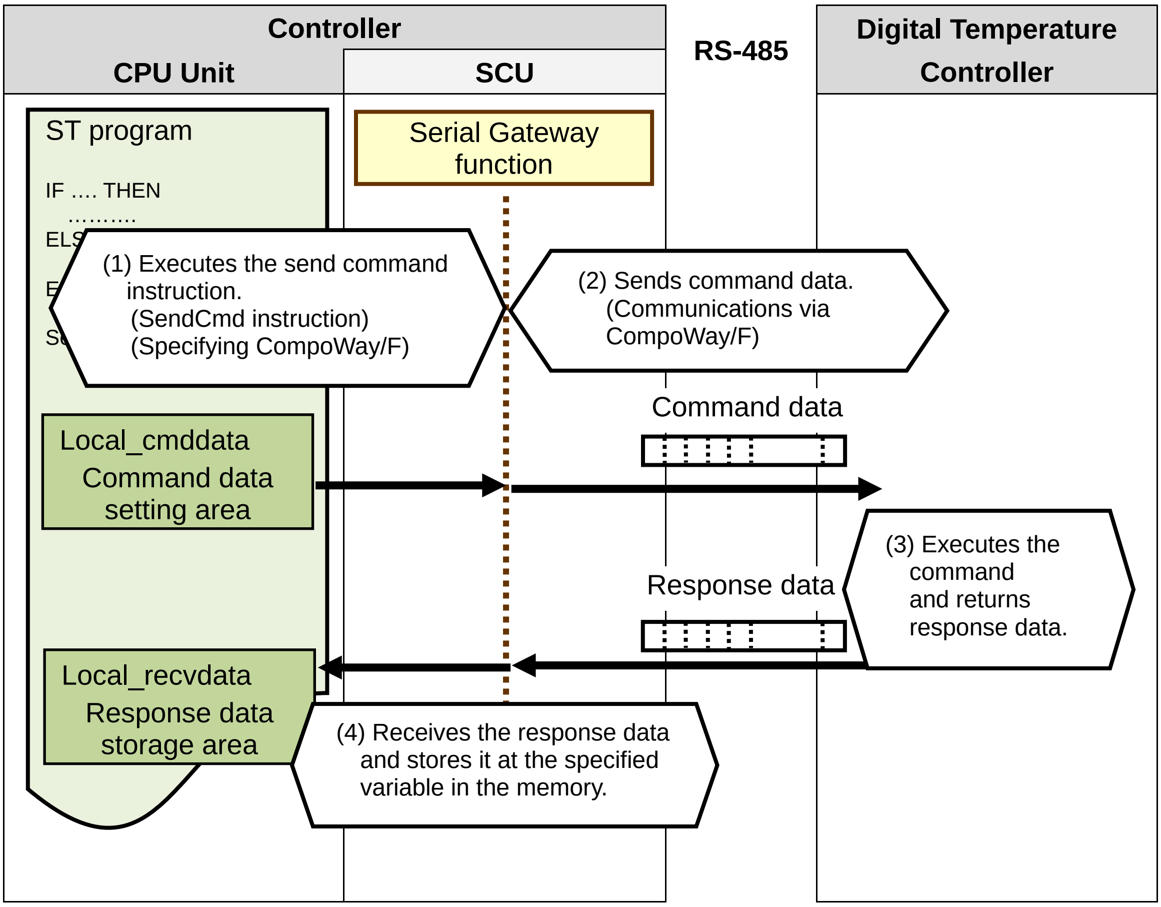
The following figure shows the data flow from when the Controller (SCU) sends CompoWay/F command to the Destination Device until the Controller receives response data from the Destination Device.

(1)The ST program executes the send command instruction (SendCmd instruction) for which CompoWay/F is specified.

(2)The Controller sends command data of "Read Controller Attributes" to the Digital Temperature Controller using communications via CompoWay/F.

(3)The Digital Temperature Controller executes the command by receiving the command data from the Controller, and returns response data to the Controller.

(4)The Controller receives the response data from the Digital Temperature Controller and stores in the specified variable.


Controller
CPU Unit
SCU
RS-485
Digital Temperature Controller
ST program
IF .... THEN
....
ELS
E
S
Serial Gateway function
(1) Executes the send command instruction. (SendCmd instruction) (Specifying CompoWay/F)
(2) Sends command data. (Communications via CompoWay/F)
Command data
Local_cmddata Command data setting area
(3) Executes the command and returns response data.
Response data
Local_recvdata Response data storage area
(4) Receives the response data and stores it at the specified variable in the memory.

9.1.2. SendCmd Instruction and Send/Receive Messages

The following describes the function block for sending a command (hereinafter referred to as "the SendCmd instruction") and the general operation of sending/receiving a message.

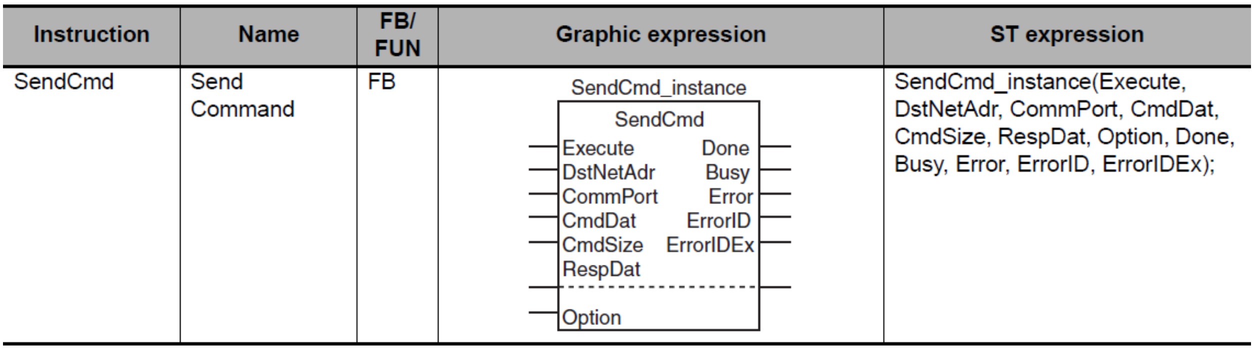
Additional Information

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

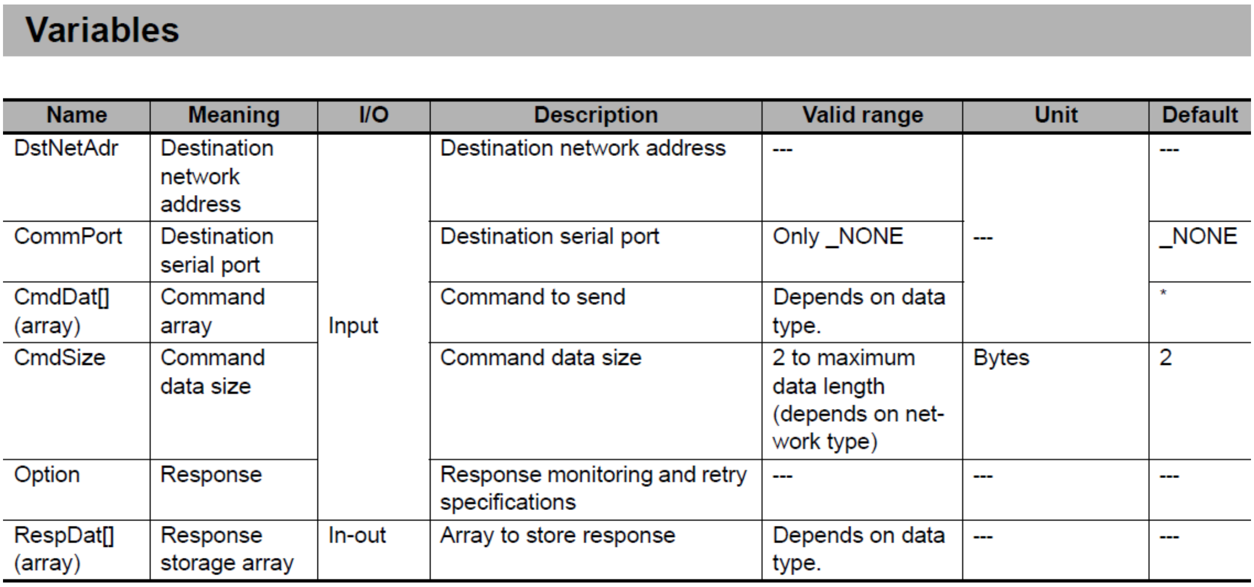
●SendCmd instruction

Name	Function block	Description
Send Command	SendCmd	The SendCmd instruction uses a Serial Gateway and sends a command to SCU.

●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	Input	Destination network address	---	---	---
CommPort	Destination serial port		Destination serial port	Only _NONE		_NONE
CmdDat[] (array)	Command array		Command to send	Depends on data type.		*
CmdSize	Command data size		Command data size	2 to maximum data length (depends on net-work 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.	---	---

[DstNetAdr: Destination network address]

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

Variable		Setting item	Data type	Description
DstNetAdr		Destination network address	_sDNET_ ADR	Destination network address
	NetNo	Network address	USINT	USINT#16#00 (fixed): Local network
	NodeNo	Node address	USINT	USINT#16#00 (fixed): Communications in local Controller
	UnitNo	Unit address	BYTE	Make the following setting. Serial port's unit address For SCU Port 1: 16#80 + 16#04 x [Unit number (hex)] Port 2: 16#81 + 16#04 x [Unit number (hex)] For example, when the unit number (No.10) port 2 is used, the unit address of serial port is as follows: = 16#81 + 16#04 x 16#0A = 16#81 + 16#28 = 16#A9

[CommPort: Destination serial port]

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

Variable	Setting item	Data type	Description
CommPort	Destination serial port	_ePORT	_NONE: The destination is not a serial port in Host Link Mode. Set _*NONE* to specify the serial port unit address as a destination unit address.

[CmdSize: Command data size]

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

Variable	Setting item	Data type	Description
CmdSize	Command data size	UINT	Sets the number of command data bytes. (UNIT#2 to maximum data length)

[Option: Response]

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

Variable		Setting item	Data type	Description
Option		Response	_sRESPO NSE	Response monitoring and retry specifications
	isNonResp	No response	BOOL	TRUE: Response is not required. FALSE: Response is required.
	TimeOut	Timeout time	UINT	Sets UINT#1 to 65535 (indicates 0.1 to 6553.5 seconds). (UINT#0: 2 seconds (default))
	Retry	Retry count	USINT	Sets USINT#0 to 15 (0 to 15 times).

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

With this program, the command data (send data), after being set with the STRING array variable (*Local_cmdCWFdata*), is converted and stored in the BYTE array variable (*Local_cmddata*), and then transferred to the command array (CmdDat[]).

Command array

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]······		
28	03	AA	BB	CC	DD	EE	FF	GG	HH	II	**	**	**
CompoWay/F command		Node No.		Subaddress		SID	Command				Text		
							MRC		SRC				

The CompoWay/F command (16#2803) is a command that can be executed when using CompoWay/F with the Serial Gateway function of the SCU.

The shaded areas of the command data (from AA onwards) are expressed in ASCII codes.

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

With this program, the response data (receive data), after being received in the response storage array (RespDat[]), is stored in the BYTE array variable (*Local_recvdata*).

The data is then converted into the STRING array variable (*Output_recvCWFdata*) and is checked.

Response storage array

[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	**
CompoWay/F command		Comm- and end code		Node No.		Sub address		End code		Command				Response				Text
										MRC		SRC		MRES		SRES		

The shaded areas of the response data (from AA onwards) are expressed in ASCII codes.

The response data that are not boxed with thick lines such as AA will be the same values as those of the command code.

The response data boxed with thick lines (cc, dd, ee to hh) will be stored in the following variables as Destination Device error codes when a Destination Device error occurs.

For details, refer to *9.7.2. Destination Device Error*.

Ouitput_CWFErrCode1: End code

Ouitput_CWFErrCode2: Response (MRES, SRES)

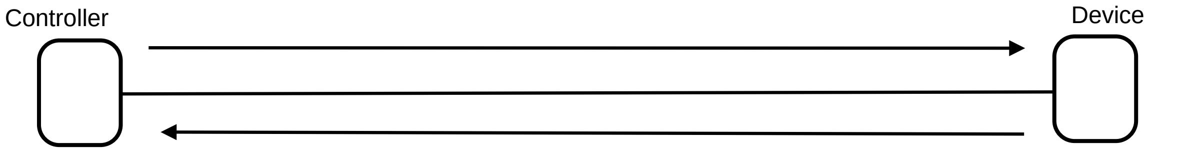
"aa" and "bb" command end codes are not used in this guide or this program.

●Send/Receive Messages

[Frames of send/receive messages]

Send message (command)

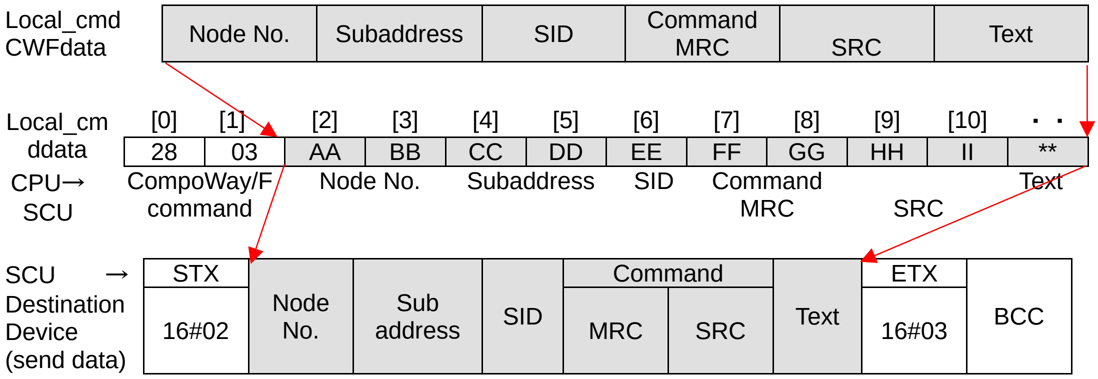
STX	Node No.	Sub address	SID	Command		Text	ETX	BCC
16#02				MRC	SRC		16#03	



Reception message (response)

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

[Relationship between send command (SendCmd instruction operand) and send message]



Local_cmd CWFdata

Node No.	Subaddress	SID	Command MRC	SRC	Text

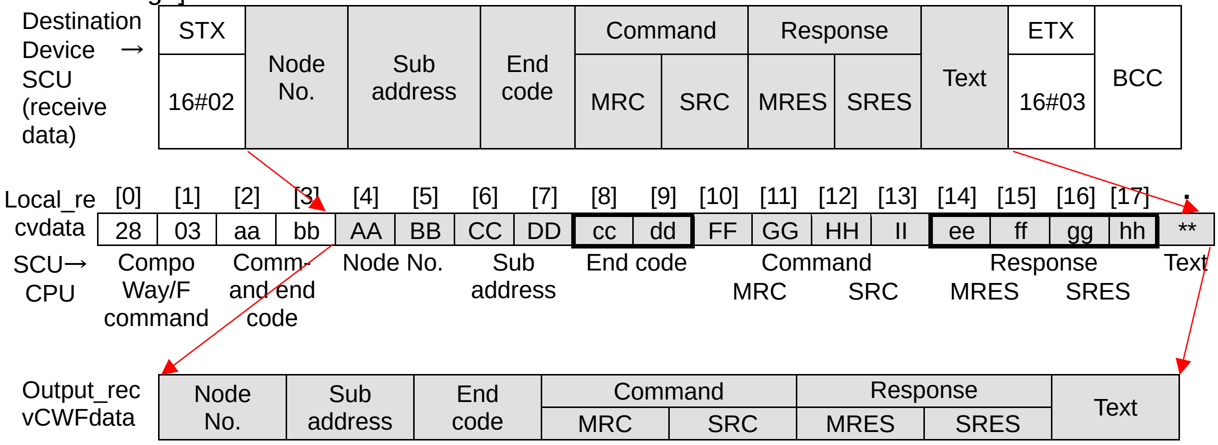
Local_cmddata CPU→SCU

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	· ·
28	03	AA	BB	CC	DD	EE	FF	GG	HH	II	**
CompoWay/F command		Node No.		Subaddress		SID	Command MRC		SRC		Text

SCU → Destination Device (send data)

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

[Relationship between receive response (SendCmd instruction operand) and receive message]



Destination Device → SCU (receive data)

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

Local_recvdata SCU→CPU

[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	**
Compo Way/F command		Command end code		Node No.		Sub address		End code		Command MRC		SRC		Response MRES		SRES		Text

Output_recvCWFdata

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

9.2. Destination Device Command

The following describes the Destination Device command used in this program.

9.2.1. Overview of the Command

This program reads the Destination Device information by using the "Read Controller Attributes" command.

Command name	Description
Read Controller Attributes	Reads the attributes of the Destination Device.

9.2.2. Detailed Description of the Function

The following describes the "Read Controller Attributes" command.

●SendCmd instruction send data

[DstNetAdr: Destination network address]

Variable		Setting item	Data type	Setting value	
DstNetAdr		Destination network address	_sDNET_ADR	-	
	NetNo	Network address	USINT	16#00	Fixed: Local network
	NodeNo	Node address	USINT	16#00	Fixed: Within local controller
	UnitNo	Unit address	BYTE	16#80	Unit number 0 + Port number 1

[CommPort: Destination serial port]

Variable	Setting item	Data type	Setting value	
CommPort	Destination serial port	_ePORT	_NONE	Fixed

[CmdSize: Command data size]

Variable	Setting item	Data type	Setting value	
CmdSize	Command data size	UINT	11	11 bytes

[Option: Response]

Variable		Setting item	Data type	Setting value	
Option		Response	_sRESPONSE	-	
	isNonResp	Response is not required.	BOOL	FALSE	Response is required.
	TimeOut	Timeout time	UINT	0	Default: 2 sec
	Retry	Retry count	USINT	3	3 times

[CmdDat[ ]: Command array]

Variable		Setting item		Data type	Setting value	
CmdDat	Element number	Command array		ARRAY[0..255] OF BYTE	-	
	0	CompoWay/F command		BYTE	16#28	Fixed
	1			BYTE	16#03	
	2	Node No.		BYTE	16#30	"01": Unit No. of Destination Device
	3			BYTE	16#31	
	4	Subaddress		BYTE	16#30	"00": Fixed (Not used)
	5			BYTE	16#30	
	6	SID		BYTE	16#30	"0": Fixed (Not used)
	7	Command	MRC	BYTE	16#30	"0503": "Read Controller Attributes" command
	8			BYTE	16#35	
	9		SRC	BYTE	16#30	
	10			BYTE	16#33	
	:	Text		BYTE···	16#00	" ": The setting is not required for this command.

After the data boxed in red above is set with the following STRING array variable, it is transferred to the command array (CmsDat[]).

Variable	Data type	Data
Local_cmdCWFdata	STRING[256]	CONCAT(NodeNo, SubAddress,SID, MRCSRC, SendText);

●SendCmd instruction receive data

[RespDat[ ]: Response storage array]

Variable		Storing item		Data type	Storing value	
RespDat	Element number	Response storage array		ARRAY[0..255] OF BYTE	-	
	0	CompoWay/F command		BYTE	16#28	Fixed: Same as CmdDat[0]
	1			BYTE	16#03	Fixed: Same as CmdDat[1]
	2	Command end code		BYTE	16#**	End code of CompoWay/F command
	3			BYTE	16#**	
	4	Node No.		BYTE	16#30	"01": Unit No. of Destination Device
	5			BYTE	16#31	
	6	Subaddress		BYTE	16#30	"00": Fixed (Not used)
	7			BYTE	16#30	
	8	End code		BYTE	16#**	"xx": End code ("00": Normal end)
	9			BYTE	16#**	
	10	Command	MRC	BYTE	16#30	"0503": "Read Controller Attributes" command
	11			BYTE	16#35	
	12		SRC	BYTE	16#30	
	13			BYTE	16#33	
	14	Response	MRES	BYTE	16#**	"xxxx" ("0000": Normal end)
	15			BYTE	16#**	
	16		SRES	BYTE	16#**	
	17			BYTE	16#**	
	:	Text		BYTE···	16#**··	"xxxx···"

●Send message

This is the command frame 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, ASCII codes are sent.
- STX, ETX and BCC are automatically added to the send message by the SCU.
- Data not used is moved forward.

02	30	31	30	30	30	30	35	30	33		03	xx
STX	Node No.		Subaddress		SID	Command MRC		Command SRC		Text (Not used)	ETX	BCC

●Receive message (at normal operation)

This is the response frame of the normal message received by the Controller from the Destination Device according to the setting of the "Read Controller Attributes" command.

- Except for STX, ETX and BCC, ASCII codes are received.
- STX, ETX and BCC are automatically removed from the receive message by the SCU.
- Data not used is moved forward.

02	30	31	30	30	30	30	30	35	30	33	30	30	30	30
STX	Node No.		Subaddress		End code		Command MRC		Command SRC		Response MRES		Response SRES	
	**	**	**	**	**	**	**	**	**	**	**	**	**	**
	Model number										Buffer size			
													03	**
													ETX	BCC

●Receive message (at error operation)

This is the response frame of the error message received by the Controller from the Destination Device.

- Except for STX, ETX and BCC, ASCII codes are received.
- STX, ETX and BCC are automatically removed from the receive message by the SCU.

02	30	31	30	30	**	**	30	35	30	33	**	**	**	**
STX	Node No.		Subaddress		End code		Command MRC		Command SRC		Response MRES		Response SRES	
													03	**
													ETX	BCC

For details on the end codes and response codes, refer to *9.7.2 Destination Device Error*.

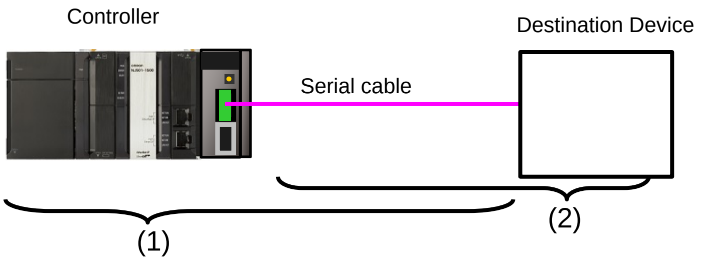
Additional Information

For details on the Destination Device command and data format, refer to *Chapter 3 Communications Data for CompoWay/F* of the *Digital Temperature Controllers Communications Manual* (Cat. No. H225/H175/H186).

9.3. Error Detection Processing

With this program, the error detection processing is performed according to the following descriptions (1) and (2).

For information on error codes, refer to *9.7. Error Processing*.



(1)Errors at the execution of the SendCmd instruction (SendCmd instruction errors)

Errors such as a Unit error, a command format error and a parameter error are detected as SendCmd instruction errors when executing the SendCmd instruction. If an error occurs, the error code *ErrorID* or *ErrorIDEx* of the SendCmd instruction will be generated to identify the error. If an error in communications with the Destination Device 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) that is the allocated variable area of the SCU will be stored in the output variable to show the communication error status.

(2)Errors in the Destination Device (Destination Device errors)

Destination Device errors include a command error, a parameter error and an execution failure in the Destination Device. An error is identified in the response data (receive data) that is returned from the Destination Device. If an error occurs in the Destination Device, the corresponding error code will be stored in the end code or response of the receive data.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	･
Local _recv data	28	03	aa	bb	AA	BB	CC	DD	cc	dd	FF	GG	HH	II	ee	ff	gg	hh	**
	CompoWay/F command		Command end code		Node No.		Subaddress		End code		Command				Response				Text

9.4. Variables

The variables used in this program are listed below.

9.4.1. Lists of Variables

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

●External variables

[User-defined global variables]

Variable name	Data type	Description
Input_Start	BOOL	Communications start switch (The program is started when this switch changes from FALSE to TRUE.)
Input_NodeNo	STRING[3]	Destination node address
Input_MRCSRC	STRING[5]	Command execution code Read Controller Attributes: "0503"
Input_SendText	STRING[128]	Command parameter Read Controller Attributes: " " (no parameter)
Output_recvByteSize	UINT	The number of receive data bytes
Output_recvCWFdata	STRING[256]	An area that stores receive data. (STRING type: 256 characters)
Output_CmdErrorID	WORD	An area that stores an error code of SendCmd instruction. Normal end: 16#0000
Output_CmdErrorIDEx	DWORD	An area that stores an expansion error code of SendCmd instruction. Normal end: 16#00000000
Output_TransErrCode	WORD	Transmission error status when a communication error occurs. Storage area of J01_P1_TransErrSta Normal end: 16#0000
Output_CWFErrCode1	WORD	An area that stores the Destination Device end code when a Destination Device error occurs. Normal end: 16#0000
Output_CWFErrCode2	WORD	An area that stores the response code when a Destination Device error occurs. Normal end: 16#0000

[Device variables for CJ-series Unit] (SCU)

Variable name	Data type	Description
J01_P1_TransErr	BOOL	Transmission error
J01_P1_TransErrSta	BOOL	Transmission error status

Additional Information

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

[System-defined variable]

Variable name	Data type	Description
_Port_isAvailable	BOOL	Network Communications Instruction Enabled Flag TRUE: A port is available. FALSE: A port is not available.

Additional Information

For information on the system-defined variables for the SendCmd instruction, refer to *Serial Communications Instructions* in *Section 2. Instruction Descriptions* of the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502).

●Internal variables (instance variables)

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

[Instance for the SendCmd instruction]

Variable name	Data type	Description
SendCmd_instance	SendCmd	This function block sends a command to the SCU using the Serial Gateway function.

Additional Information

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

●Internal variables

Variable name	Data type	Description
Local_Busy	BOOL	Communications processing status flag (Communications processing in progress)
Local_Done	BOOL	Communications processing status flag (Communications processing normal end)
Local_Error	BOOL	Communications processing status flag (Communications processing error end)
Local_State	DINT	Processing number
Local_ExecFlgs	BOOL	Communications instruction execution 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_SubAddress	STRING[3]	Subaddress "00" (fixed)
Local_SID	STRING[2]	SID "0" (fixed)
Local_cmddata	ARRAY[0..255] OF BYTE	Command array for SendCmd instruction setting (256 bytes)
Local_recvdata	ARRAY[0..255] OF BYTE	Response storage array for SendCmd instruction setting (256 bytes)
Local_cmdCWFdata	STRING[256]	An area that stores send data (STRING type: 256 characters)

9.5. ST Program

9.5.1. Functional Components of the Program

The program used in this guide 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. Communications processing in progress status	The communications processing is started.
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 SendCmd instruction control data 2.5. Setting the send variables 2.6. Initializing the receive data storage areas 2.7. Initialization setting end processing	The receive data storage areas and the error code storage areas are initialized. The parameters and send data are set for the SendCmd instruction. (CompoWay/F)
3.CompoWay/F communications processing	3.1. Determining the communications processing status and setting the execution flag 3.2. Executing the communications instruction	The SendCmd instruction (CompoWay/F) is executed. A normal end or an error end of the execution is detected.
4.Processing number error process	-	The error processing is performed when a non-existent number is detected.

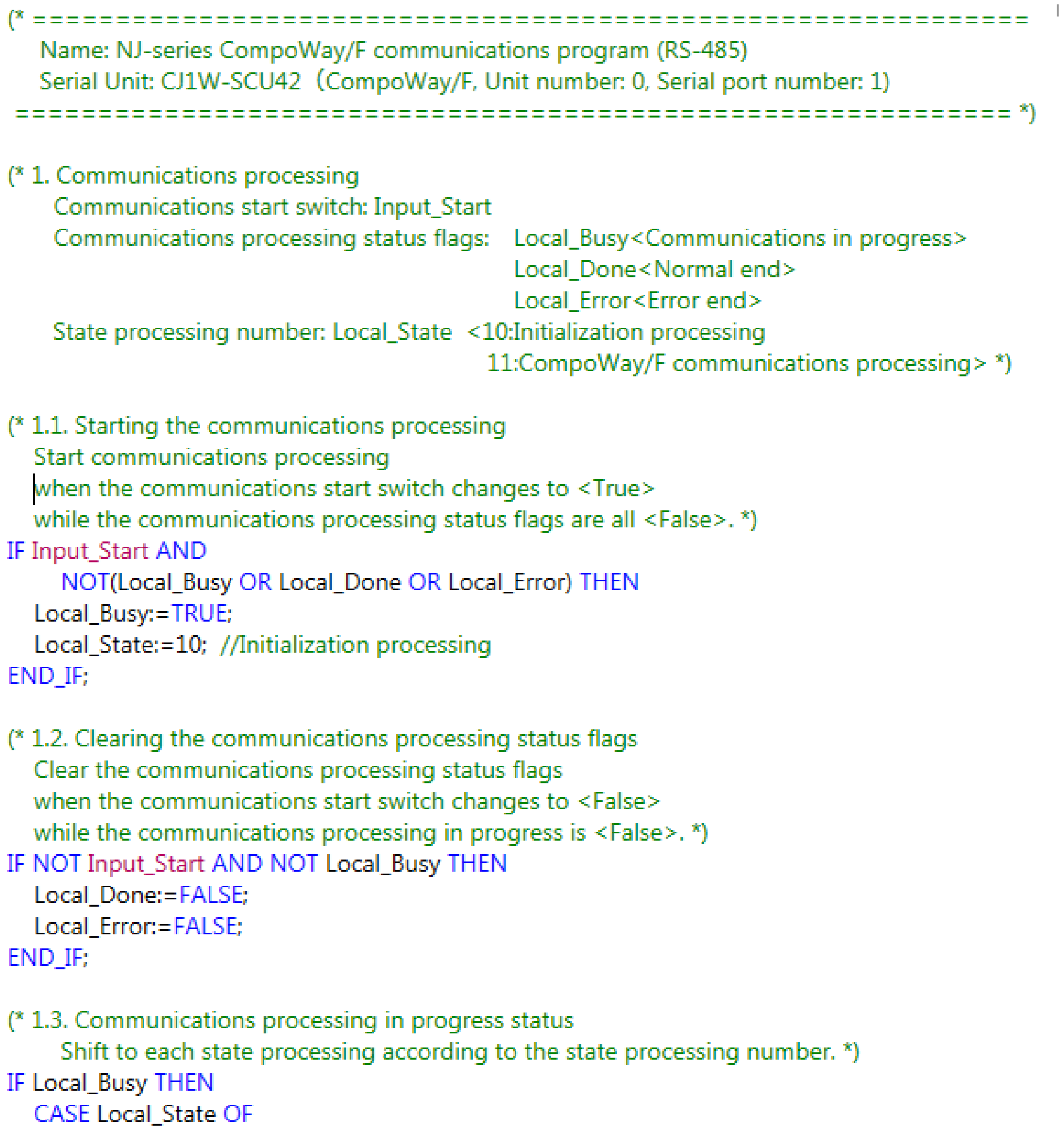
9.5.2. Program list

The program used in this guide is shown below.

The send data (command data) setting to the Destination Device is boxed in red in this program list.

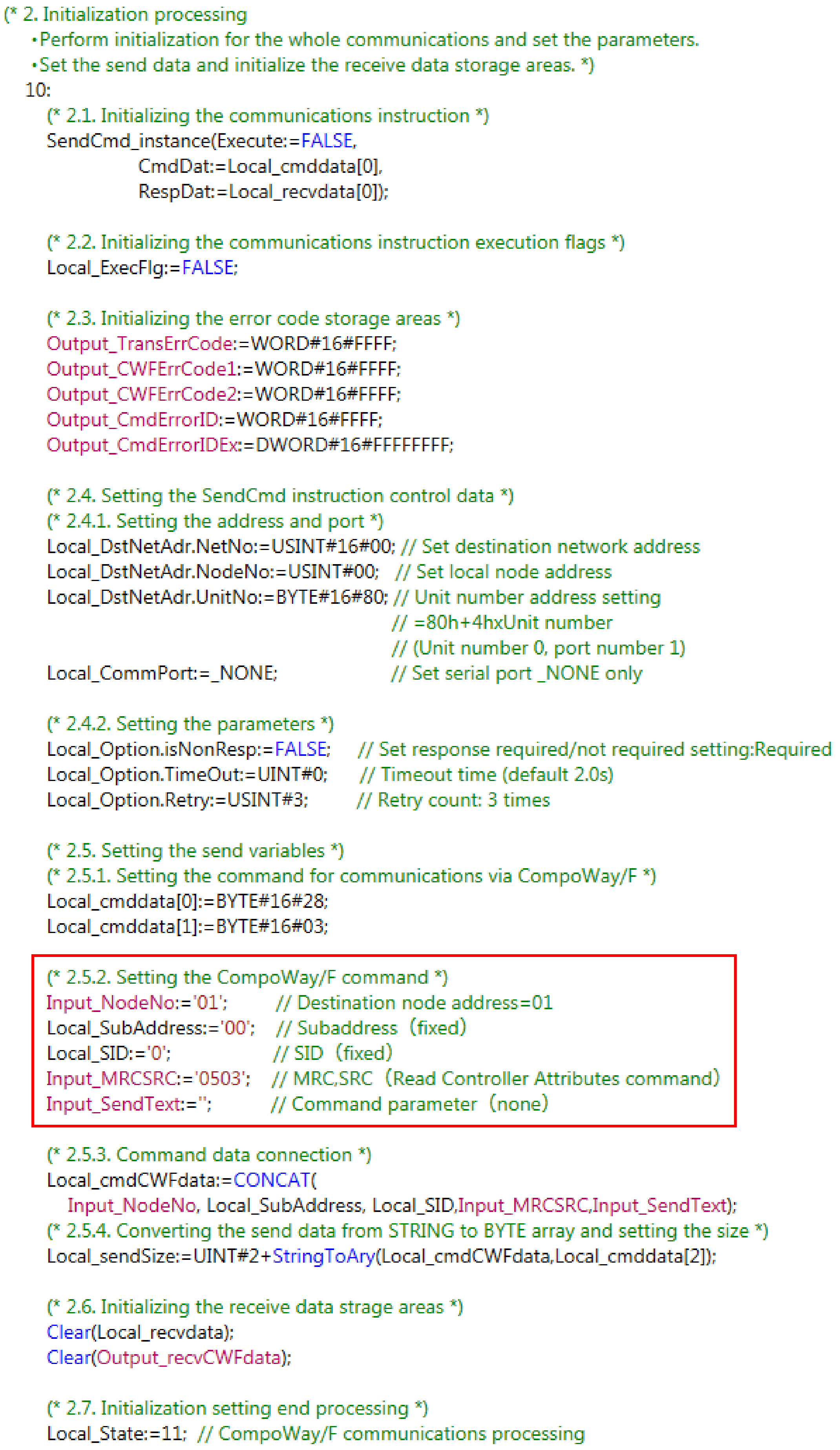
●Program: Program0 (General-purpose serial communications connection check program)

1. Communications processing



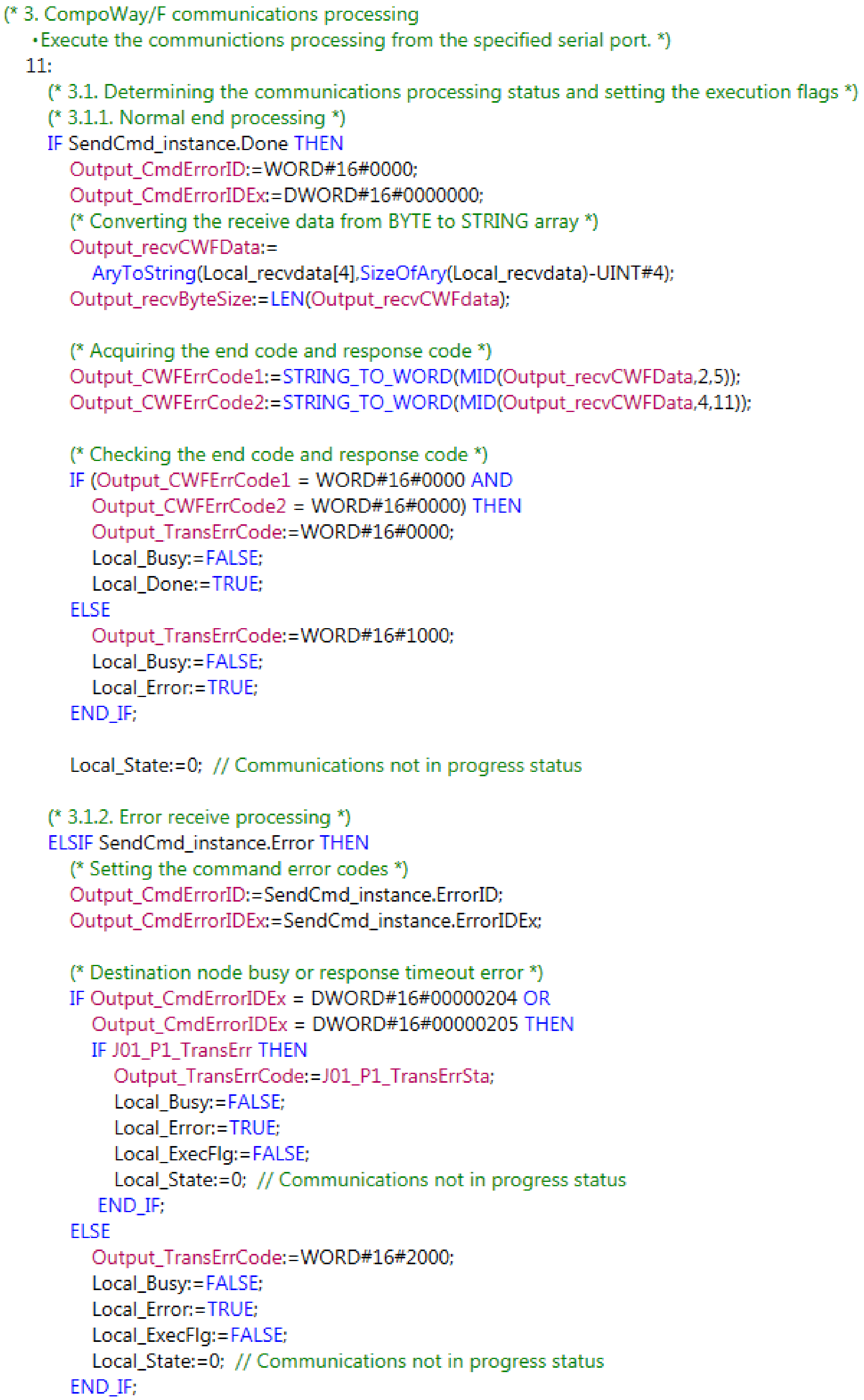
```
(* ============================================================
    Name: NJ-series CompoWay/F communications program (RS-485)
    Serial Unit: CJ1W-SCU42 (CompoWay/F, Unit number: 0, Serial port number: 1)
 ============================================================ *)

(* 1. Communications processing
      Communications start switch: Input_Start
      Communications processing status flags:   Local_Busy<Communications in progress>
                                                Local_Done<Normal end>
                                                Local_Error<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 <True>
    while the communications processing status flags are all <False>. *)
IF Input_Start AND
      NOT(Local_Busy OR Local_Done OR Local_Error) THEN
    Local_Busy:=TRUE;
    Local_State:=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 <False>
    while the communications processing in progress is <False>. *)
IF NOT Input_Start AND NOT Local_Busy THEN
    Local_Done:=FALSE;
    Local_Error:=FALSE;
END_IF;

(* 1.3. Communications processing in progress status
      Shift to each state processing according to the state processing number. *)
IF Local_Busy THEN
    CASE Local_State OF
```

2. Initialization processing

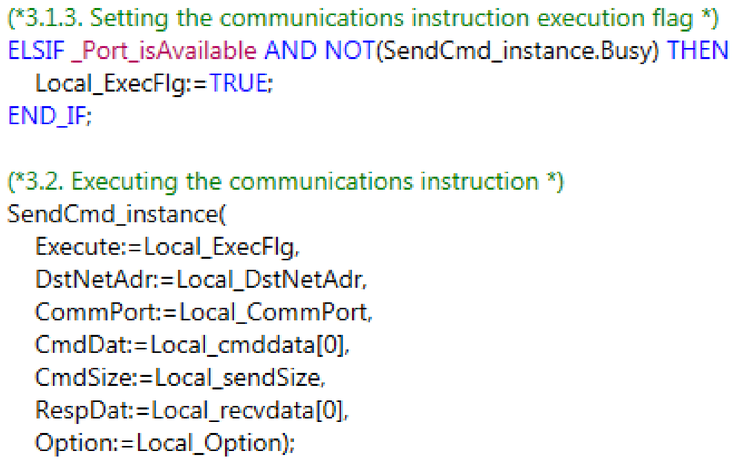


```
(* 2. Initialization processing
  ・Perform initialization for the whole communications and set the parameters.
  ・Set the send data and initialize the receive data storage areas. *)
  10:
    (* 2.1. Initializing the communications instruction *)
    SendCmd_instance(Execute:=FALSE,
                CmdDat:=Local_cmddata[0],
                RespDat:=Local_recvdata[0]);

    (* 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 the SendCmd instruction control data *)
    (* 2.4.1. Setting the address and port *)
    Local_DstNetAdr.NetNo:=USINT#16#00; // Set destination network address
    Local_DstNetAdr.NodeNo:=USINT#00;   // Set local node address
    Local_DstNetAdr.UnitNo:=BYTE#16#80; // Unit number address setting
                                        // =80h+4hxUnit number
                                        // (Unit number 0, port number 1)
    Local_CommPort:=_NONE;              // Set serial port _NONE only

    (* 2.4.2. Setting the parameters *)
    Local_Option.isNonResp:=FALSE;    // Set response required/not required setting:Required
    Local_Option.TimeOut:=UINT#0;     // Timeout time (default 2.0s)
    Local_Option.Retry:=USINT#3;      // Retry count: 3 times

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

    (* 2.5.2. Setting the CompoWay/F command *)
    Input_NodeNo:='01';        // Destination node address=01
    Local_SubAddress:='00';    // Subaddress（fixed）
    Local_SID:='0';            // SID（fixed）
    Input_MRCSRC:='0503';      // MRC,SRC（Read Controller Attributes command）
    Input_SendText:='';        // Command parameter（none）

    (* 2.5.3. Command data connection *)
    Local_cmdCWFdata:=CONCAT(
      Input_NodeNo, Local_SubAddress, Local_SID,Input_MRCSRC,Input_SendText);
    (* 2.5.4. Converting the send data from STRING to BYTE array and setting the size *)
    Local_sendSize:=UINT#2+StringToAry(Local_cmdCWFdata,Local_cmddata[2]);

    (* 2.6. Initializing the receive data strage areas *)
    Clear(Local_recvdata);
    Clear(Output_recvCWFdata);

    (* 2.7. Initialization setting end processing *)
    Local_State:=11;  // CompoWay/F communications processing
```

3. CompoWay/F communications processing

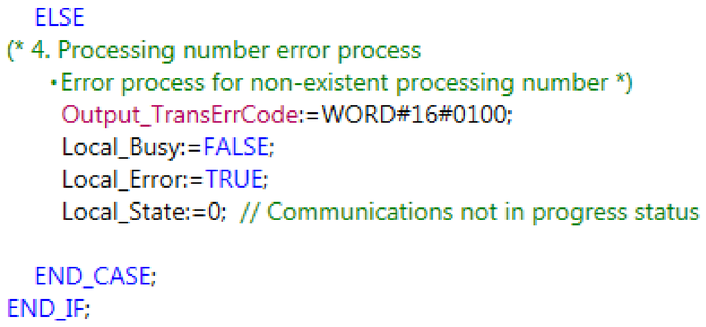


```
(* 3. CompoWay/F communications processing
  •Execute the communictions processing from the specified serial port. *)
  11:
    (* 3.1. Determining the communications processing status and setting the execution flags *)
    (* 3.1.1. Normal end processing *)
    IF SendCmd_instance.Done THEN
      Output_CmdErrorID:=WORD#16#0000;
      Output_CmdErrorIDEx:=DWORD#16#0000000;
      (* Converting the receive data from BYTE to STRING array *)
      Output_recvCWFData:=
        AryToString(Local_recvdata[4],SizeOfAry(Local_recvdata)-UINT#4);
      Output_recvByteSize:=LEN(Output_recvCWFdata);

      (* 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));

      (* 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_Busy:=FALSE;
        Local_Done:=TRUE;
      ELSE
        Output_TransErrCode:=WORD#16#1000;
        Local_Busy:=FALSE;
        Local_Error:=TRUE;
      END_IF;

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

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

      (* 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_Busy:=FALSE;
          Local_Error:=TRUE;
          Local_ExecFlg:=FALSE;
          Local_State:=0;  // Communications not in progress status
         END_IF;
      ELSE
        Output_TransErrCode:=WORD#16#2000;
        Local_Busy:=FALSE;
        Local_Error:=TRUE;
        Local_ExecFlg:=FALSE;
        Local_State:=0;  // Communications not in progress status
      END_IF;
```



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

(*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);
```

4. Processing number error process

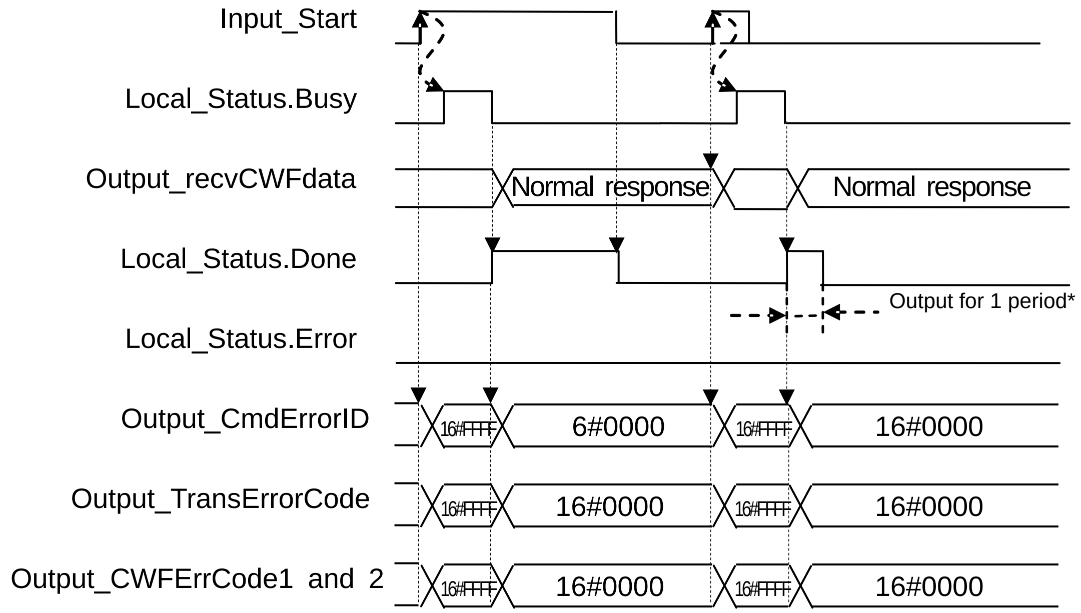


```
    ELSE
(* 4. Processing number error process
      ･Error process for non-existent processing number *)
        Output_TransErrCode:=WORD#16#0100;
        Local_Busy:=FALSE;
        Local_Error:=TRUE;
        Local_State:=0;  // Communications not in progress status

    END_CASE;
END_IF;
```

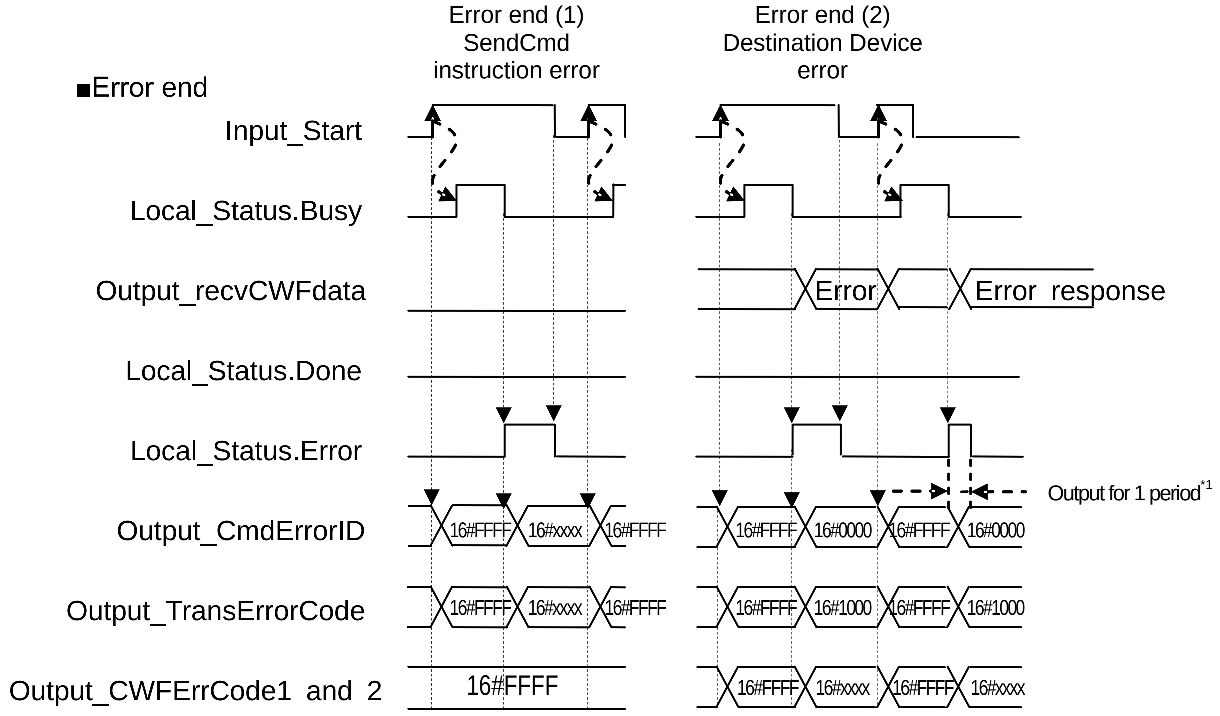
9.6. Timing Charts

The timing charts are shown below.

■Normal end



■Error end



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

9.7. Error Processing

The errors that may occur during the program execution are shown below.

9.7.1. SendCmd Instruction Error

These error codes are generated when the SendCmd instruction ends in error.

●SendCmd instruction error code [*Output_CmdErrorID*, *Output_CmdErrorIDE*x]

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 description
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 FINS command was sent or received. Detailed information is stored in *Output_CmdErrorIDEx*.
16#0801	The port is being used.

Additional Information

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

For information on 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]

For information on error codes to be stored, refer to the list of expansion error code *ErrorIDEx* described in *SendCmd* in *Section 2. Instruction Descriptions* of the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502).

●Transmission error status [Output_TransErrCode]

The data of J01_P1_TransErrSta transmission error status is stored in *Output_TransErrCode*.

When a SendCmd instruction error, a destination device error or a processing number error occurs, bits 8, 12 and 13 will change to TRUE, respectively.

[Status of each bit at a transmission error]

Bit	Description	
15	1: Transmission error	0: No transmission error
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 status.

Value	Description
16#0000	Normal end
16#FFFF	Initialized

9.7.2. Destination Device Error

These error codes are used for errors in the Destination Device.

●Destination Device error code [*Output_CWFErrCode1*, *Output_CWFErrCode2*]

[Output_CWFErrCode1] (End code)

Bit	15 … 8	7 … 0
	#00 Fixed	"**" End code

End code	Name	Description	Priority
00	Normal completion	The command ended normally without error.	None
0F	FINS command error	The specified FINS command could not be executed. The FINS response code should indicate why the command could not be executed.	8
10	Parity error	The sum total of bits whose received data is "1" does not match the set value of the "communications parity" bit.	2
11	Framing error	Stop bit is "0".	1
12	Overrun error	An attempt was made to transfer new data when the reception data buffer was already full.	3
13	BCC error	The calculated BCC value is different from the received BCC value.	5
14	Format error	- The command text contains characters other than 0 to 9, and A to F. - There was no SID and command text, or there was no command text. - "MRC/SRC" not included in command text.	7
16	Sub-address error	- Illegal (unsupported) subaddress - There was no subaddress, SID and command text. - Subaddress was less than two characters, and there was no SID and command text	6
18	Frame length error	The received frame exceeds the specified (supported) number of bytes.	4

[Output_CWFErrCode2] (Response code)

Response code	Error name	Priority
0000	Normal completion	None
0401	Unsupported command	1
1001	Command too long	2
1002	Command too short	3
1101	Area type error	4
1103	Start 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 on the Destination Device errors, refer to the *Digital Temperature Controllers User's Manual* (Cat. No. H224/H174/H185) and the *Digital Temperature Controllers Communications Manual* (Cat. No. H225/H175/H186).

10. Revision History

Revision code	Date of revision	Description of revision
01	January 2013	First edition
02	June 2018	Added the model numbers of the Digital Temperature Controller. Revision and changes associated with upgrades of the Controller and Sysmac Studio.

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

OMRON (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 2013-2018 All Rights Reserved.
In the interest of product improvement,
specifications are subject to change without notice.