OMRON

Machine Automation Controller
NX-series

Heater Burnout Detection Unit Startup Guide

NX-HB□□□□

SYSMAC
always in control

P105-E1-01

NOTE

Introduction

The *NX-series Heater Burnout Detection Unit Startup Guide* (hereinafter, may be referred to as "this Guide") describes the startup procedures for temperature control using the NJ-series CPU Unit, NX-series Temperature Input Unit, and NX-series Heater Burnout Detection Unit in combination. A simple temperature control system is used for the discussion.

This Guide does not contain safety information and other details that are required for actual use. Thoroughly read and understand the manuals for all of the devices that are used in this Guide to ensure that the system is used safely. Review the entire contents of these materials, including all safety precautions, precautions for safe use, and precautions for correct use.

Intended Audience

This Guide is intended for the following personnel.

- Personnel in charge of introducing FA systems
- Personnel in charge of designing FA systems

The personnel must also have the following knowledge.

- Knowledge of electrical systems (an electrical engineer or the equivalent)
- Knowledge of NJ-series CPU Units
- Knowledge of operation procedure of Sysmac Studio
- Knowledge of NA-series Programmable Terminal
- Knowledge of temperature control

Applicable Products

This Guide covers the following products.

- Heater Burnout Detection Units of NX-series Machine Automation Controllers
- Temperature Input Units of NX-series Machine Automation Controllers
- CPU Units of NJ-series Machine Automation Controllers
- Sysmac Studio Automation Software
- NA-series Programmable Terminals

Special Information

The icons that are used in this Guide are described below.

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.

Terms and Conditions Agreement

Machine Automation Controller NJ-Series CPU Units

Warranties

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

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

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

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

Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

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

Suitability of Use

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

system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.
NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

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

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

Change in Specifications
Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

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

Sysmac Studio Automation Software

1. WARRANTY

(1) The warranty period for the Software is one year from the date of purchase, unless otherwise specifically agreed.

(2) If the User discovers defect of the Software (substantial non-conformity with the manual), and return it to OMRON within the above warranty period, OMRON will replace the Software without charge by offering media or download from OMRON's website. And if the User discovers defect of media which is attributable to OMRON and return it to OMRON within the above warranty period, OMRON will replace defective media without charge. If OMRON is unable to replace defective media or correct the Software, the liability of OMRON and the User's remedy shall be limited to the refund of the license fee paid to OMRON for the Software.

2. LIMITATION OF LIABILITY

(1) THE ABOVE WARRANTY SHALL CONSTITUTE THE USER'S SOLE AND EXCLUSIVE REMEDIES AGAINST OMRON AND THERE ARE NO OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTY OF MERCHANTABILITY OR FITNESS FOR PARTICULAR PURPOSE. IN NO EVENT, OMRON WILL BE LIABLE FOR ANY LOST PROFITS OR OTHER INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF USE OF THE SOFTWARE.

(2)OMRON SHALL HAVE NO LIABILITY FOR DEFECT OF THE SOFTWARE BASED ON MODIFICATION OR ALTERNATION TO THE SOFTWARE BY THE USER OR ANY THIRD PARTY.

(3)OMRON SHALL HAVE NO LIABILITY FOR SOFTWARE DEVELOPED BY THE USER OR ANY THIRD PARTY BASED ON THE SOFTWARE OR ANY CONSEQUENCE THEREOF.

3. APPLICABLE CONDITIONS

USER SHALL NOT USE THE SOFTWARE FOR THE PURPOSE THAT IS NOT PROVIDED IN THE ATTACHED USER MANUAL.

4. CHANGE IN SPECIFICATION

The software specifications and accessories may be changed at any time based on improvements and other reasons.

5. EXTENT OF SERVICE

The license fee of the Software does not include service costs, such as dispatching technical staff.

6. ERRORS AND OMISSIONS

The information in this manual has been carefully checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical, or proofreading errors, or omissions.

Precautions

- When building a system, check the specifications for all devices and equipment that will make up the system and make sure that the OMRON products are used well within their rated specifications and performances. Safety measures, such as safety circuits, must be implemented in order to minimize the risks in the event of a malfunction.
- Thoroughly read and understand the manuals for all devices and equipment that will make up the system to ensure that the system is used safely. Review the entire contents of these manuals, including all safety precautions, precautions for safe use, and precautions for correct use.
- Confirm all regulations, standards, and restrictions that the system must adhere to.
- Check the user program for proper execution before you use them for actual operation.

Trademarks

- Sysmac and SYSMAC are trademarks or registered trademarks of OMRON Corporation in Japan and other countries for OMRON factory automation products.
- EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- Microsoft product screen shot(s) reprinted with permission from Microsoft Corporation.

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

Software Licenses and Copyrights

The NJ-series CPU Units and Sysmac Studio incorporate certain third party software. The license and copyright information associated with this software is available at http://www.fa.omron.co.jp/nj_info_e/.

Related Manuals

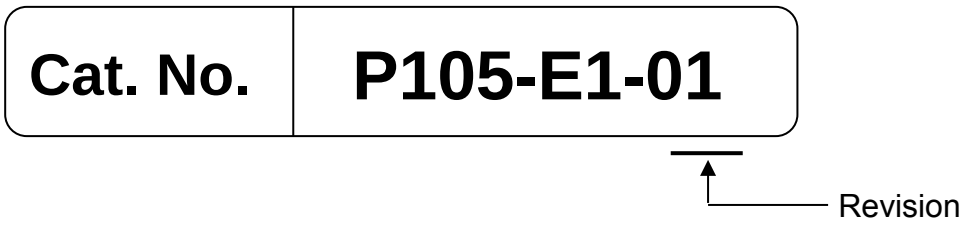
The following manuals are related to this Guide. Use these manuals for reference.

Manual name	Cat. No.	Model numbers	Application	Description
NX-series Heater Burnout Detection Unit Startup Guide (this guide)	P105-E1-01	NX-TS□□□□ NX-HB□□□□	Learning the startup procedures for a temperature control system using the NX-series Heater Burnout Detection Unit and NX-series Temperature Input Unit in combination.	The startup procedures for a temperature control system using the NX-series Heater Burnout Detection Unit and NX-series Temperature Input Unit.
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC-SE2□□□	Learning about the operating procedures and functions of the Sysmac Studio.	The operating procedures of the Sysmac Studio are described.
NX-series Analog I/O Units User's Manual for Temperature Input Units and Heater Burnout Detection Units	W566	NX-TS□□□□ NX-HB□□□□	Learning how to use NX-series Temperature Input Units and Heater Burnout Detection Units	The hardware, setup methods, and functions of the NX-series Temperature Input Units and Heater Burnout Detection Units are described.
NJ-series CPU Unit Hardware User's Manual	W500	NJ501-□□□□ NJ301-□□□□	Learning the basic specifications of the NJ-series CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NJ-series system is provided along with the following information on a Controller built with an NJ501 CPU Unit. ・Features and system configuration ・Introduction ・Part names and functions ・General specifications ・Installation and wiring ・Maintenance and inspection Use this manual together with the *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).
NJ-series CPU Unit Software User's Manual	W501	NJ501-□□□□ NJ301-□□□□	Learning how to program and set up an NJ-series CPU Unit. Mainly software information is provided.	The following information is provided on a Controller built with an NJ-series CPU Unit. ・CPU Unit operation ・CPU Unit features ・Initial settings ・Programming based on IEC 61131-3 language specifications Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No.W500).
NJ-series Troubleshooting Manual	W503	NJ501-□□□□ NJ301-□□□□	Learning about the errors that may be detected in an NJ-series Controller.	Concepts on managing errors that may be detected in an NJ-series Controller and information on individual errors are described. Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No.W500) and *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).

Manual name	Cat. No.	Model numbers	Application	Description
NX-series Data Reference Manual	W525	NX-□□□□□□	Referencing lists of the data that is required to configure systems with NX-series Units	Lists of the power consumptions, weights, and other NX Unit data that is required to configure systems with NX-series Units are provided.
NX-series EtherCAT Coupler Unit User's Manual	W519	NX-ECC201 NX-ECC202 NX-ECC203	Leaning how to use an NX-series EtherCAT Coupler Unit and EtherCAT Slave Terminals	The following items are described: the overall system and configuration methods of an EtherCAT Slave Terminal (which consists of an NX-series EtherCAT Coupler Unit and NX Units), and information on hardware, setup, and functions to set up, control, and monitor NX Units through EtherCAT.
NA-series Programmable Terminal Hardware User's Manual	V117	NA5-□W□□□□ NA5-□□W□□□□	Learning the specifications and settings required to install an NA-series PT and connect peripheral devices.	Information is provided on NA-series PT specifications, part names, installation procedures, and procedures to connect an NA Unit to peripheral devices. Information is also provided on maintenance after operation and troubleshooting.
NA-series Programmable Terminal Software User's Manual	V118	NA5-□W□□□□ NA5-□□W□□□□	Learning about NA-series PT pages and object functions.	NA-series PT pages and object functions are described.
NA-series Programmable Terminal Device Connection User's Manual	V119	NA5-□W□□□□ NA5-□□W□□□□	Learning the specifications required to connect devices to an NA-series PT.	Information is provided on connection procedures and setting procedures to connect an NA-series PT to a Controller or other device.
NA-series Programmable Terminal Startup Guide	V120	NA5-□W□□□□ NA5-□□W□□□□	Learning in concrete terms information required to install and start the operation of an NA-series PT.	The part names and installation procedures are described followed by page creation and transfer procedures with the Sysmac Studio. Also operation, maintenance, and inspection procedures after the project is transferred are described. Sample screen captures are provided as examples.
NJ/NX-series Instructions Reference Manual	W502	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning detailed specifications on the basic instructions of an NJ/NX-series CPU Unit.	The instructions in the instructions set (IEC 61131-3 specifications) are described. When programming, use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) or *NX-series CPU Unit Hardware User's Manual* (Cat. No. W535) and with the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501).

Revision History

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



Revision code	Date	Revised content
01	September 2016	Original production

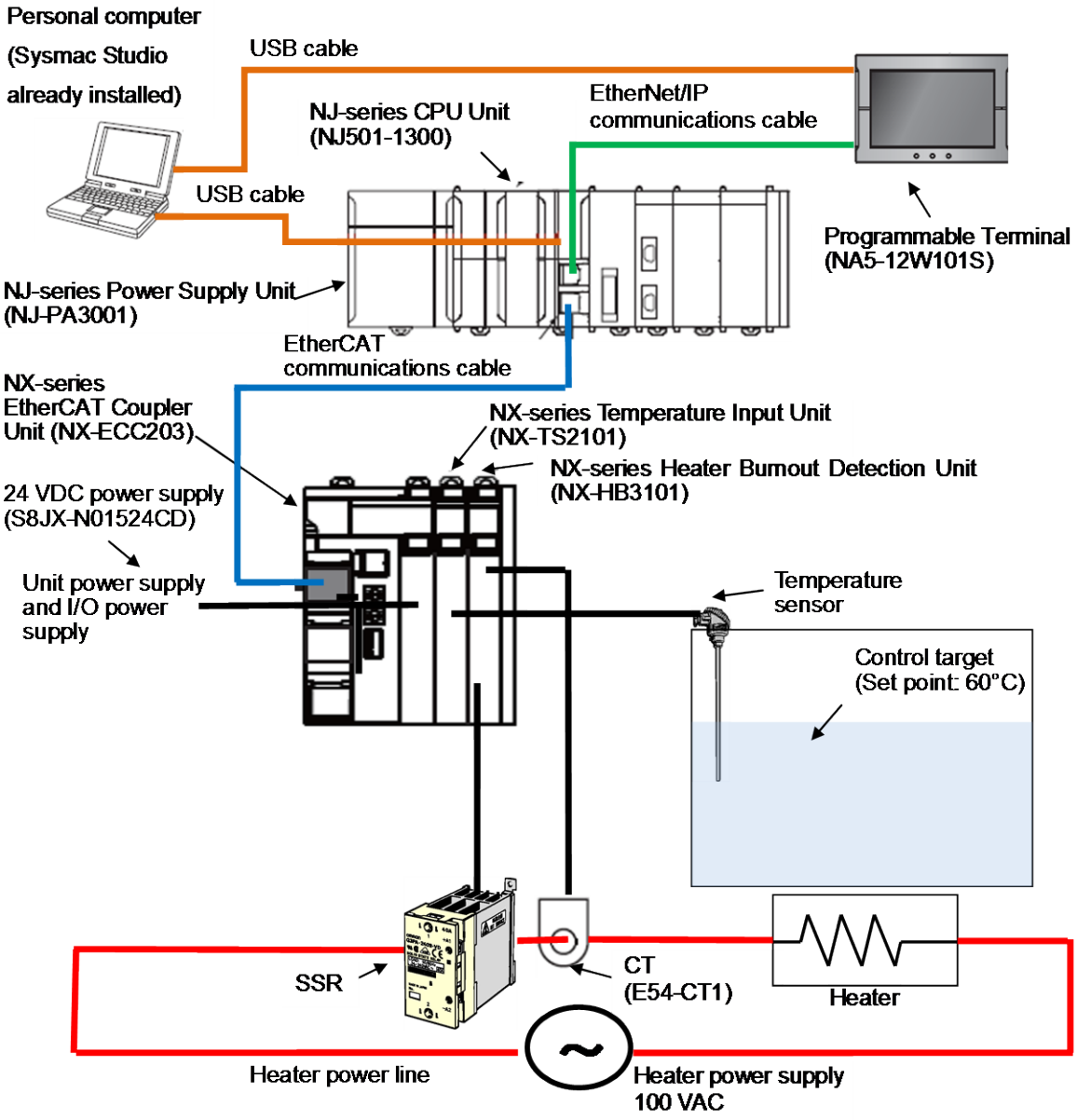
CONTENTS

Introduction 3
Intended Audience 3
Applicable Products. 3
Special Information 3
Terms and Conditions Agreement 4
Machine Automation Controller NJ-Series CPU Units 4
Sysmac Studio Automation Software 6
Precautions 7
Trademarks 7
Software Licenses and Copyrights 7
Related Manuals 8
Revision History 10
1. System Configuration and Configuration Devices 13
2. Before You Begin 17
2.1. Wiring the Devices and Installing the Software 17
3. Starting Up the Temperature Control System with the Function of Heater Burnout Alarm 18
3.1. Steps to Start Up the Temperature Control System with the Function of Heater Burnout Alarm 18
3.2. Unit Registration and I/O Allocation Settings 19
3.3. Unit Operation Settings for the Temperature Input Unit 21
3.3.1. Setting Parameters 21
3.3.2. Setting Procedure 22
3.4. Unit Operation Settings for the Heater Burnout Detection Unit. 23
3.4.1. Alarm Current Calculation 23
3.4.2. Setting Parameters 23
3.4.3. Setting Procedure 24
3.5. Generating Device Variables 25
3.6. Creating the User Program. 27
3.6.1. Ladder Programming 27
3.6.2. ST Programming 28
3.7. Creating the Window for the Programmable Terminal 29
3.8. Wiring the Units 31
3.8.1. Wiring of the Temperature Input Unit 31
3.8.2. Wiring of the Heater Burnout Detection Unit 32
3.9. Transfer to the CPU Unit and Slave Terminal 32
3.10. Transfer of the Window Data to the Programmable Terminal 33
3.11. Test Operation 33
3.11.1. Check on the Temperature Input Unit. 33
3.11.2. Check on the Heater Burnout Detection Unit 33
3.11.3. Autotuning. 35
4. Programming Example 39
4.1. Specifications 39
4.2. Settings for Unit 39
4.3. I/O Allocation Settings. 39
4.4. I/O Map 40
4.5. Sample programs 41
4.5.1. Variables Used in Programming 41

4.5.2. Ladder Programming 45
4.5.3. ST 47

1. System Configuration and Configuration Devices

This section describes the system configuration and configuration devices used in this Guide.
The system described in this Guide is a PID control system for one channel with the function of heater burnout alarm. Heating control is performed on a control target so that the temperature of the control target reaches the set temperature, and the presence or absence of an error in the heater serving for heating control is monitored. The system configuration is shown below.
The temperature value of the control target, as inputted to the Temperature Input Unit, is sent to the CPU Unit, which in turn executes the user program for system temperature control. The temperature control program makes use of the PIDAT_HeatCool instruction that the CPU Unit has. Based on the operation results of the CPU Unit, the Heater Burnout Detection Unit controls the control output for heater control. In addition, the Unit reads the current flowing through the heater power line with the use of a current transformer (CT) for each ON/OFF timing of the control output, monitoring the presence or absence of a heater burnout and an SSR failure.
As an example, this Guide uses the program described in *7-8-4 Programming Example* in the *NX-series Analog I/O Units User's Manual for Temperature Input Units and Heater Burnout Detection Units* (Cat. No. W566).



This example assumes that the heater in use has the following specifications.

Heater type	Heater power	Current
For single phase 100 VAC	1.4 kW	14.0 A

The table below shows the models of the devices used for explanation in this Guide. When selecting devices for actual use, refer to their respective manuals.

Device name	Model	Manual name
NX-series Temperature Input Unit	NX-TS2101 (version 1.1)	*NX-series Analog I/O Units User's Manual for Temperature Input Units and Heater Burnout Detection Units* (Cat. No. W566)
NX-series Heater Burnout Detection Unit	NX-HB3101 (version 1.0)	
NJ-series CPU Unit	NJ501-1300 (version 1.05 or	*NJ-series CPU Unit*

	later)	*Hardware User's Manual* (Cat. No. W500)
NJ-series Power Supply Unit	NJ-PA3001	
EtherCAT communications cables EtherNet/IP communications cables	XS5W-T421-CMD-K	
NX-series EtherCAT Coupler Unit	NX-ECC203 (version 1.0 or later)	*NX-series EtherCAT Coupler Unit User's Manual* (Cat. No. W519)
Ferrule	H0.25/12 (used for other than ground terminals) AI2,5-10 (used for ground terminals)	
Programmable Terminal	NA5-12W101S (version 1.01)	*NA-series Programmable Terminal Hardware User's Manual* (Cat. No. V117)
Switching power supply (24 VDC power supply)	S8JX-N01524CD	[Instruction Manual] *S8JX Switching Power Supply* (Cat. No. 1141546-0)
CT	E54-CT1	[Instruction Manual] *E54-CT1/CT3 Current Transformer* (Cat. No. 0648688-0C)
SSR	G3PE-225B DC12-24	[Instruction Manual] *G3PE Solid State Relay* (Cat. No. 2170307-3A)
Temperature sensor	E52-CA50C-N	―
USB cable	Commercially available USB cable*1	―
Heater and heater power line	Commercially available heater and heater power line	―
Valve	Commercially available valve	―

*1. Use a USB2.0 (or 1.1) cable (A to B connector) with a maximum length of 5.0 meters.

The table below shows the software, used for explanation in this Guide, to install on the personal computer (OS: Windows 7 64-bit edition).

Manufacturer	Name	Version
OMRON Corporation	Sysmac Studio	Version 1.16 or later

The following table lists the CTs that you can use with the NX-series Heater Burnout Detection Unit.

Manufacturer	Model	Max. continuous current	No. of turns
OMRON Corporation	E54-CT1	50 A	400±2 turns
	E54-CT3	120 A*1	

*1. With the NX-series Heater Burnout Detection Unit, the maximum continuous current that can flow to the heater is 50 A. Therefore, set the current that flows in the heater to 50 A or less.

2. Before You Begin

2.1. Wiring the Devices and Installing the Software

Wire the devices and install the Sysmac Studio to the personal computer as described in *1. System Configuration and Configuration Devices*.

Additional Information

- Refer to *3.8. Wiring the Unit* for the wiring of the Temperature Input Unit and Heater Burnout Detection Unit to the devices to connect. For the wiring of other devices, refer to their respective manuals.
- Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for how to install the Sysmac Studio.

3. Starting Up the Temperature Control System with the Function of Heater Burnout Alarm

3.1. Steps to Start Up the Temperature Control System with the Function of Heater Burnout Alarm

The steps to start up the temperature control system with the function of heater burnout alarm are as follows.
For the configuration, use the Sysmac Studio.

3.2	Unit registration and I/O allocation settings	Create a new project in the Sysmac Studio. Register the Heater Burnout Detection Unit and Temperature Input Unit offline. Set the I/O allocations.
▼		
3.3	Unit operation settings for the Temperature Input Unit	Create the Unit operation settings for the Temperature Input Unit according to the Unit functions that you will use.
▼		
3.4	Unit operation settings for the Heater Burnout Detection Unit	Create the Unit operation settings for the Heater Burnout Detection Unit according to the Unit functions that you will use.
▼		
3.5	Generating device variables	Generate the device variable for each port of the Units.
▼		
3.6	Creating the user program	Create the user program in the Sysmac Studio.
▼		
3.7	Creating the window for the Programmable Terminal	Create the window for the Programmable Terminal in the Sysmac Studio.
▼		
3.8	Wiring the Unit	Wire the Heater Burnout Detection Unit and Temperature Input Unit.
▼		
3.9	Transfer to the CPU Unit and Slave Terminal	Download the Unit settings and user program that you created on the Sysmac Studio to the Heater Burnout Detection Unit and Temperature Input Unit.
▼		
3.10	Transfer of the window to the Programmable Terminal	Transfer the window data that you created to the actual device, or the Programmable Terminal.

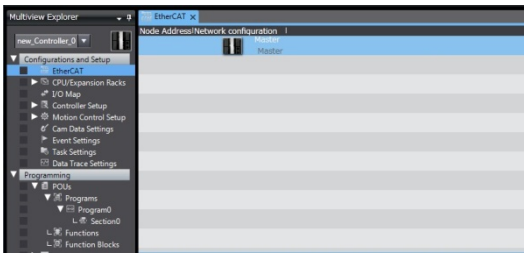
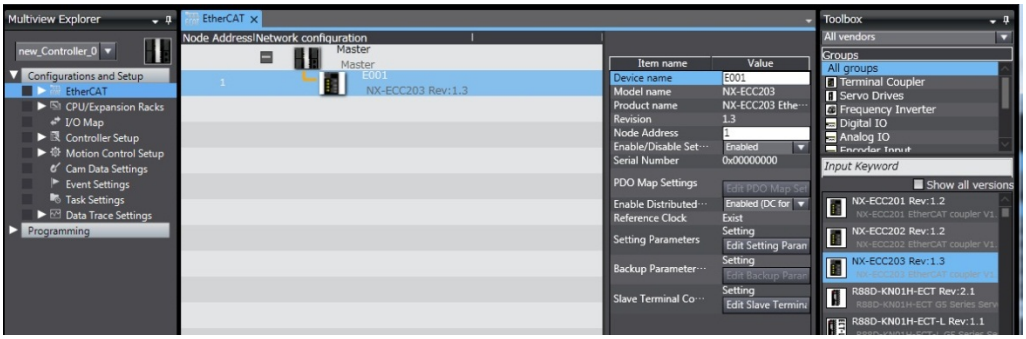
▼

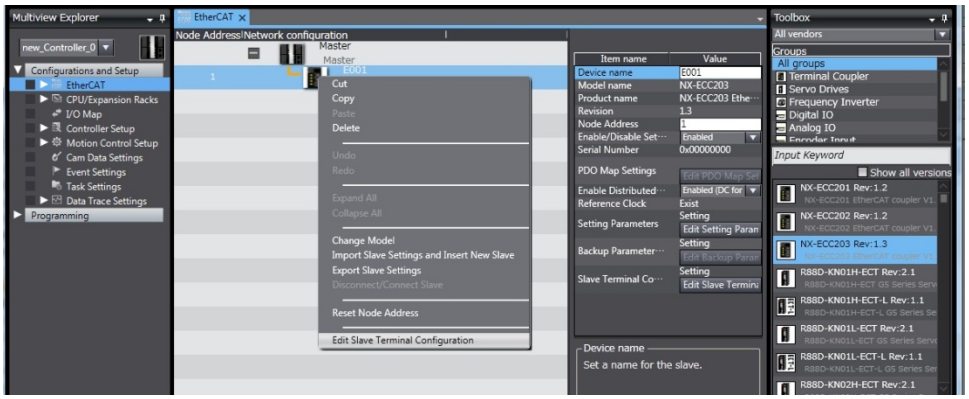
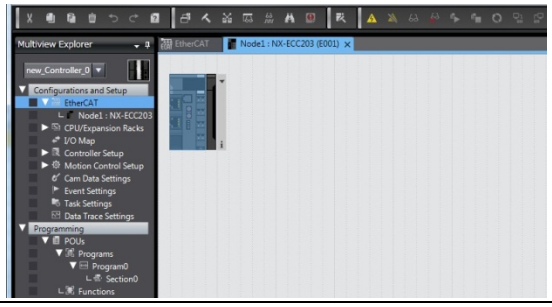
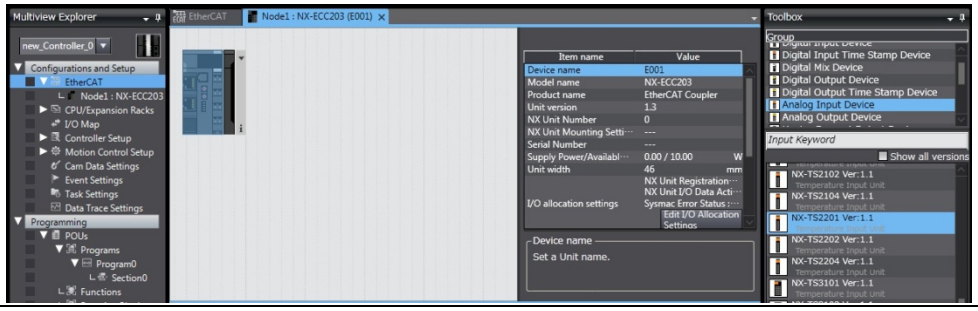
3.11	Test operation	Check the wiring and verify the measurement value. Then, perform autotuning to determine the optimal PID constants for temperature control.

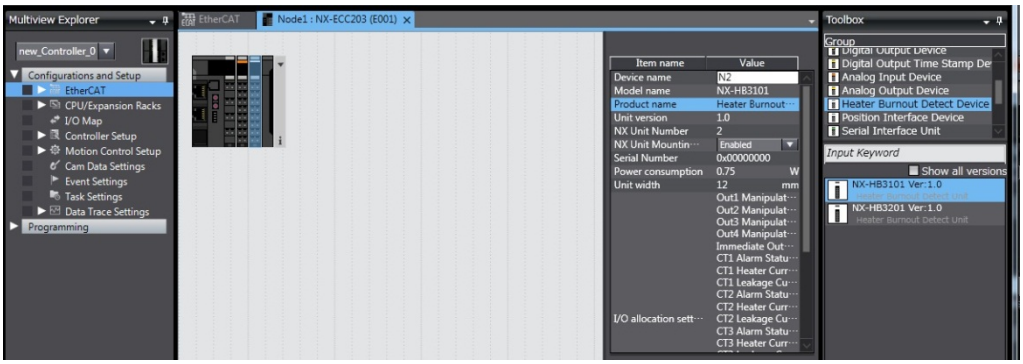
For the construction of a temperature control system using the NX-series Temperature Input Unit and Heater Burnout Detection Unit, this Guide describes the startup procedures for these two Units. For wiring and usage of the NJ-series CPU Unit, NX-series EtherCAT Coupler Unit, and Programmable Terminal, refer to their respective manuals. For manual information, see *Related Manuals*.

3.2. Unit Registration and I/O Allocation Settings

Follow these steps to set up the Unit configuration. Then, set the I/O allocation data for the EtherCAT Slave Terminal.

1	Double-click **EtherCAT** under **Configurations and Setup** in the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select *Edit*. The following window is displayed. 
2	From the **Toolbox** window, double-click **NX-ECC203**. Or, right-click **NX-ECC203** to select *Insert*. The NX-ECC203 is added under the Master Unit. 

3	Right-click the NX-ECC203 that you added in the EtherCAT window to select *Edit Slave Terminal Configuration*. Or, double-click **NX-ECC203** in the Multiview Explorer. 
4	The **Edit Slave Terminal Configuration** window is displayed. 
5	From the **Toolbox** window, click Analog Input Device. Then, double-click **NX-TS2101**. Or, right-click **NX-TS2101** to select *Insert*. The NX-TS2101 is added. 

6	From the **Toolbox** window, left-click **Heater Burnout Detect Device**, then double-click **NX-HB3101**. The NX-HB3101 is added. 

3.3. Unit Operation Settings for the Temperature Input Unit

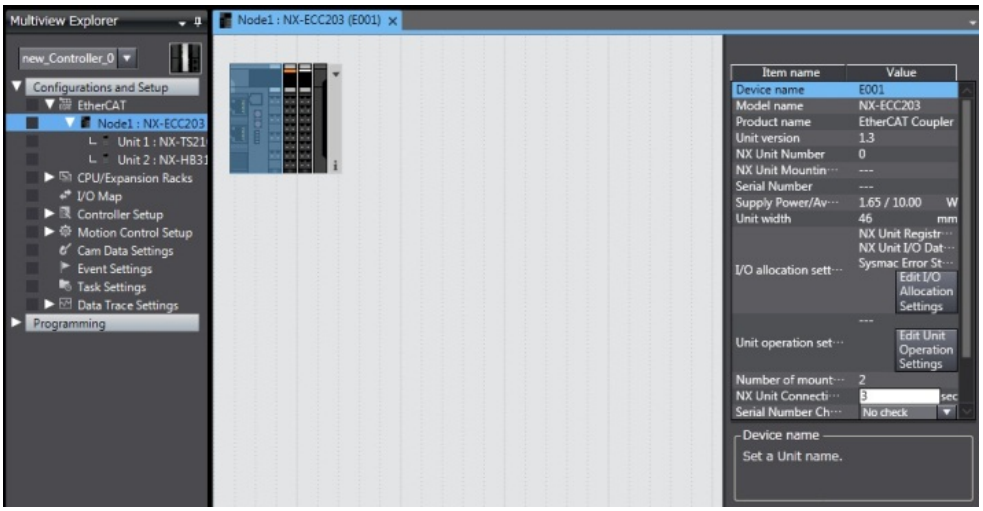
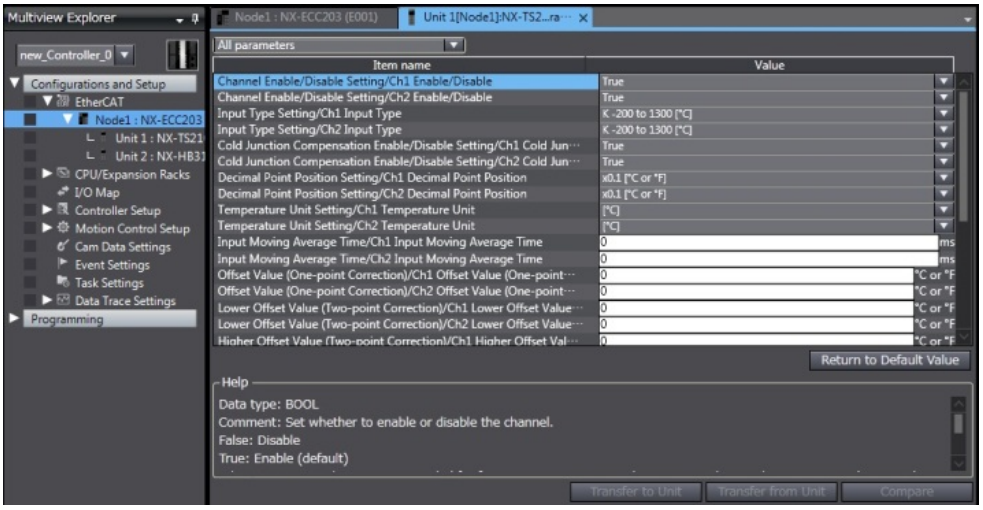
3.3.1. Setting Parameters

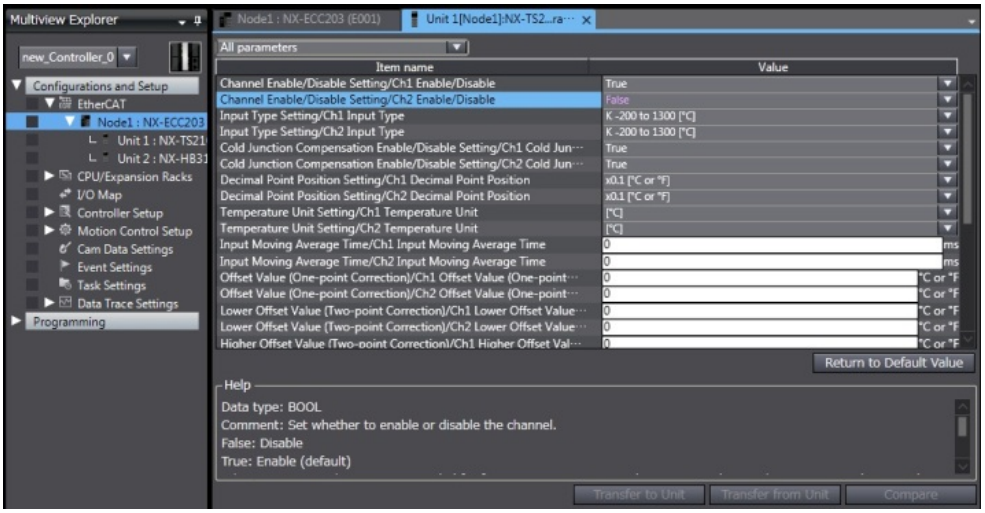
The table below shows the minimum set of setting parameters for the Temperature Input Unit required to provide the system described in *1. System Configuration and Configuration Devices*. For the parameters that are not listed in the table below, leave them at the default settings.

Item	Default	Value to set
Ch1 Enable/Disable	TRUE	TRUE
Ch2 Enable/Disable	TRUE	FALSE
Ch1 Input Type	K -200 to 1300°C	K -200 to 1300°C
Ch1 Decimal Point Position	0.1°C or 0.1°F	0.1°C or 0.1°F
Ch1 Temperature Unit (°C/°F)	°C	°C

3.3.2. **Setting Procedure**

Follow these steps to set up the Unit operation settings for the Temperature Input Unit.

1	In the Multiview Explorer, double-click the Communications Coupler Unit to which your Temperature Input Unit is connected to open the Edit Slave Terminal Configuration Tab Page. The following window is displayed. 
2	In the Edit Slave Terminal Configuration Tab Page, double-click your Temperature Input Unit. Or, right-click the Temperature Input Unit to select *Edit Unit Operation Settings*. The Edit Unit Operation Settings Tab Page is displayed. 

3	Set up the parameters according to *3.3.1. Setting Parameters*. For the parameters that are not listed in *3.3.1. Setting Parameters*, use their default settings. 

3.4. Unit Operation Settings for the Heater Burnout Detection Unit

3.4.1. Alarm Current Calculation

Calculate the alarm currents according to what type of heater is used and how the heater is wired. For details, refer to *7-7 CT Installation and Alarm Current Calculation* described in the *NX-series Analog I/O Units User's Manual for Temperature Input Units and Heater Burnout Detection Units*. The heater shown in This Guide is a single-phase heater of 1.4 kW at 100 VAC, so the currents are given by:

- Heater burnout detection current

$$\text{(Heater burnout detection current)} = \frac{\text{(Normal current)} + \text{(Current when heater burnout occurs)}}{2}$$

$$= \frac{14 + 0}{2} = 7\ [\text{A}]$$

- SSR failure detection current

$$\text{(SSR failure detection current)} = \frac{\text{(Leakage current)} + \text{(Current when SSR failure occurs)}}{2}$$

$$= \frac{0 + 14}{2} = 7\ [\text{A}]$$

3.4.2. Setting Parameters

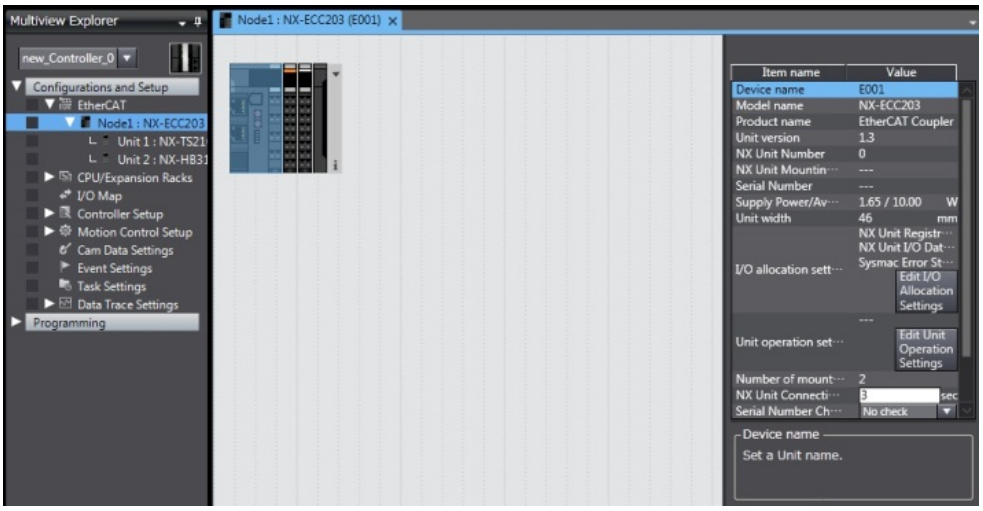
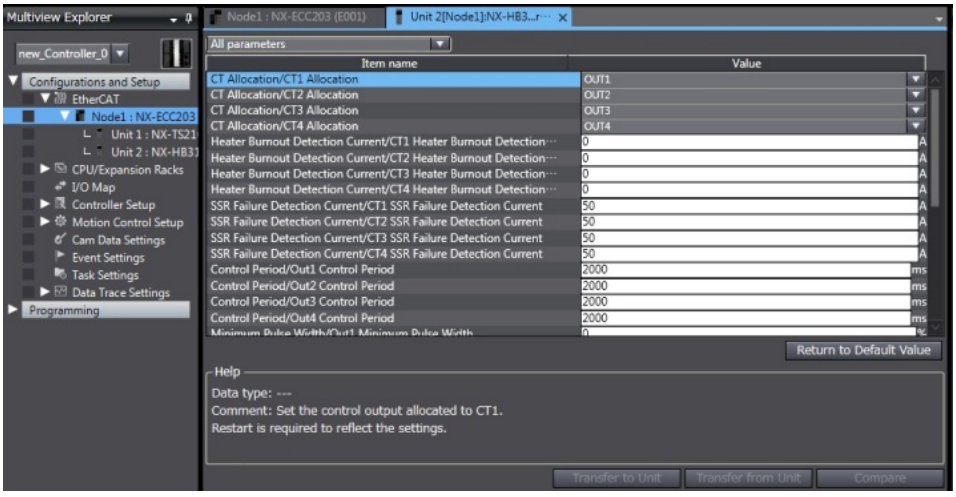
The table below shows the minimum set of setting parameters for the Heater Burnout Detection Unit required to provide the system described in *1. System Configuration and Configuration Devices*. For the parameters that are not listed in the table below, leave them at the default settings.

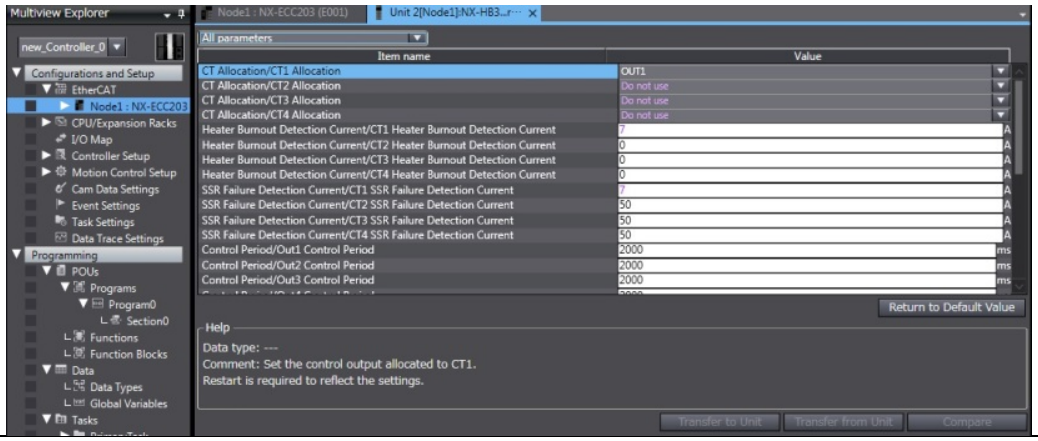
Item	Default	Value to set
CT1 Allocation	OUT1	OUT1
CT2 Allocation	OUT2	Do not use
CT3 Allocation	OUT3	Do not use
CT4 Allocation	OUT4	Do not use
CT1 Heater Burnout Detection Current	0	7
CT1 SSR Failure Detection Current	50	7

Out1 Control Period	2000	2000

3.4.3. Setting Procedure

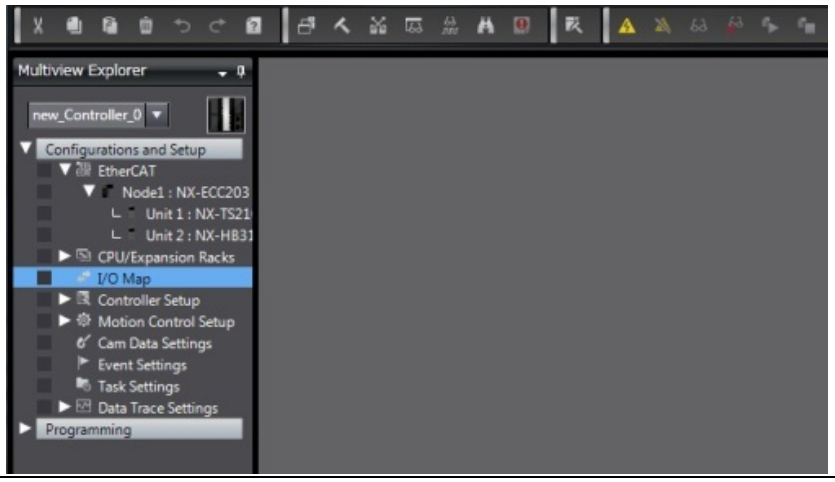
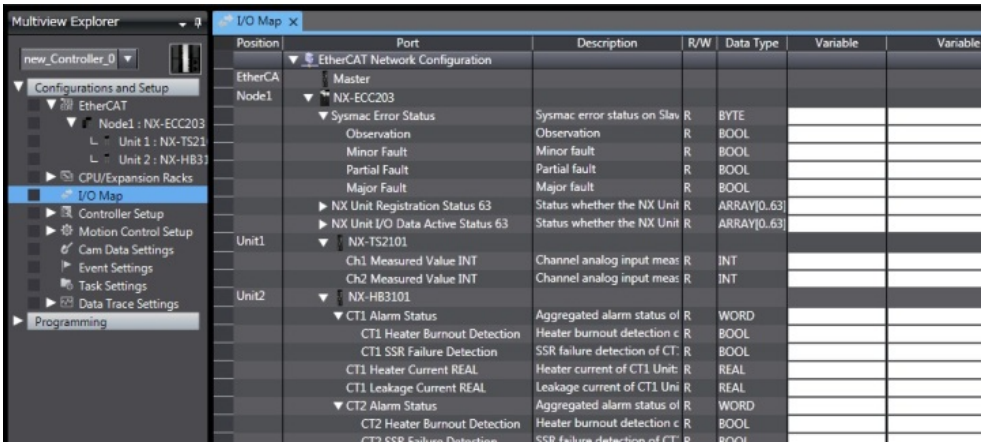
Follow these steps to set up the Unit operation settings for the Heater Burnout Detection Unit.

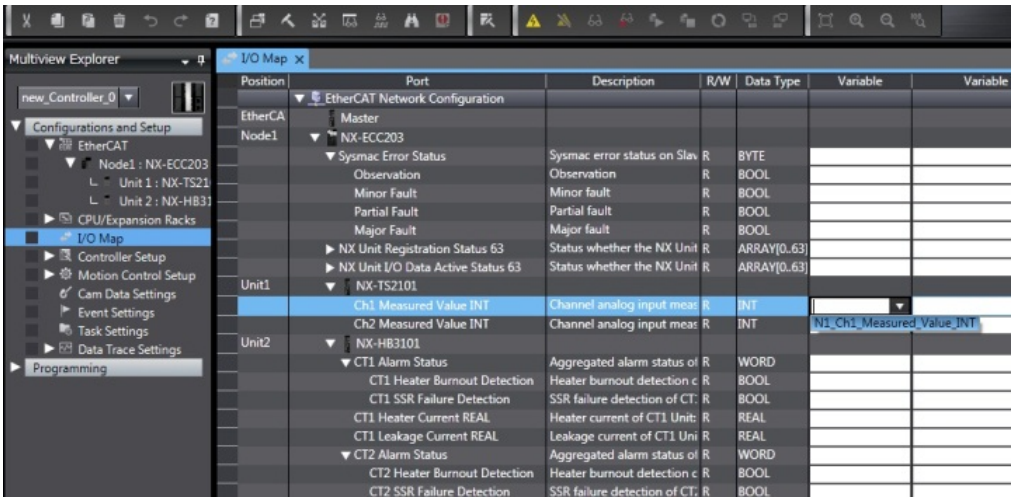
1	In the Multiview Explorer, double-click the Communications Coupler Unit to which your Heater Burnout Detection Unit is connected to open the Edit Slave Terminal Configuration Tab Page. The following window is displayed. 
2	In the Edit Slave Terminal Configuration Tab Page, double-click your Heater Burnout Detection Unit. Or, right-click the Heater Burnout Detection Unit to select *Edit Unit Operation Settings*. The Edit Unit Operation Settings Tab Page is displayed. 

3	Set up the parameters according to *3.4.2. Setting Parameters*. For the parameters that are not listed in *3.4.2. Setting Parameters*, use their default settings. 

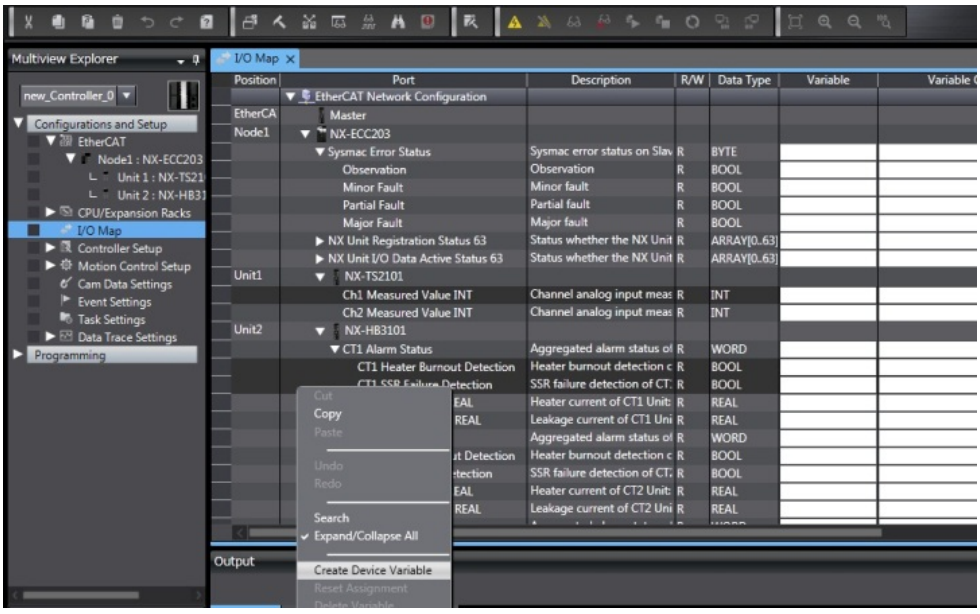
3.5. Generating Device Variables

Follow these steps to generate the device variables.

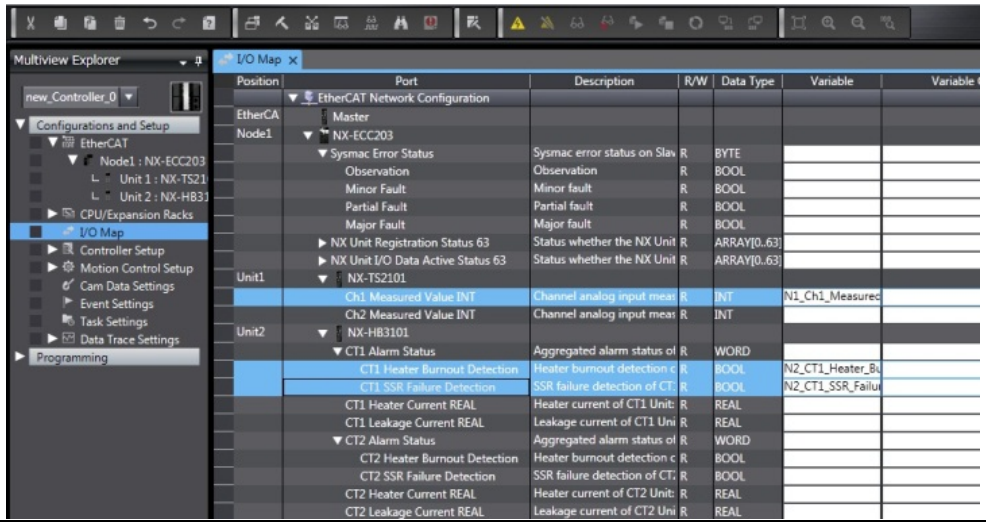
1	In the Multiview Explorer, double-click **I/O Map Tab Page**. Or, right-click it to select *Edit*. 
2	The **I/O Map Tab Page** is displayed. 
3	Select the I/O port for which you want to generate the device variable, then enter the variable in the **Variable** columun. (The variable name is arbitrary.)



4 To automatically generate device variable names, with one or more than one I/O port selected in **I/O Map Tab Page**, right click, then select *Create Device Variable*.



5 The **Variable** name and **Variable Type** are automatically set.



Additional Information

When you select *Create Device Variable* to generate the **Variable** name automatically, the resulting **Variable** name will consist of **Device name** and **Port name**. The default value of **Device name**, for the case of the NX Unit, is given as N + Sequentially assigned number from 1.

3.6. Creating the User Program

Create the user program offline using the SysmacStudio.

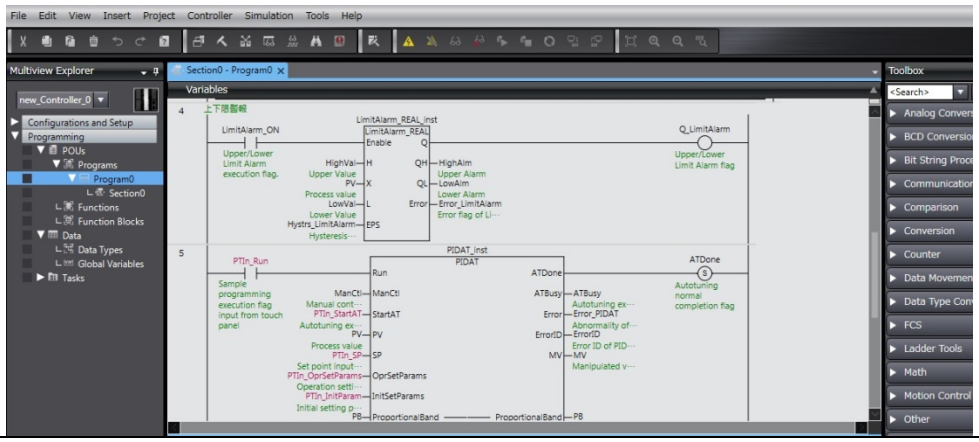
As an example, this Guide uses the program described in *4 Programming Example*.

This sample program processes the following tasks:

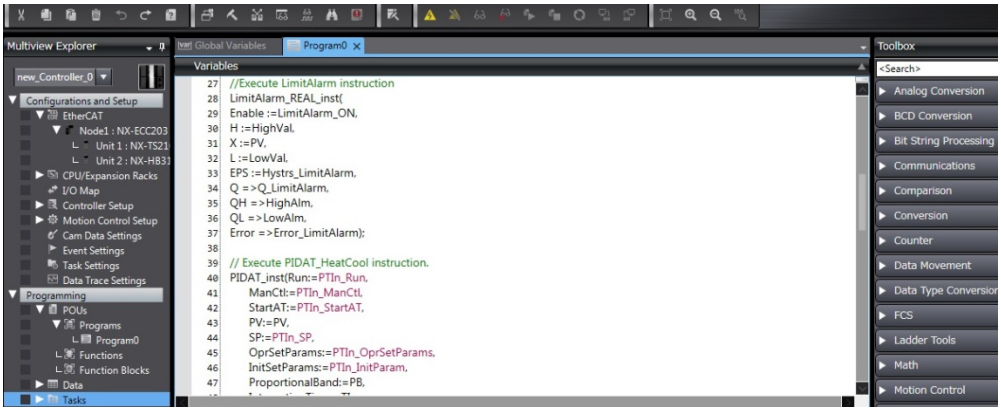
- Heating control is performed on the control target by the PID control. At the start of operation, autotuning is performed to find the optimal PID constants (executed from the PIDAT instruction).
- If a heater burnout or an SSR failure is detected, the control is stopped.
- When detecting Upper/Lower Limit Alarm, a warning signal is output Programmable Terminal.
- The data required for the above sequence is inputted from the Programmable Terminal.
- The execution state of the above sequence is outputted to the Programmable Terminal.

This sample program does not include a program to perform alarm output processing other than for a heater burnout and an SSR failure. Other alarm output processing such as the upper and lower limit alarm can be implemented by using analog instructions including the LimitAlarm_**Instruction (of the upper and limit alarm group) that the NJ/NX-series Controller has. For details on the instructions, refer to *NJ/NX-series Instructions Reference Manual* (Cat. No. W502).

3.6.1. Ladder Programming

1	In the Multiview Explorer, right-click **Programs** under **Programming**, then click *Add —Ladder*. The Ladder Editor is displayed.
2	Create the user program. Unregistered variable names should be registered to the variable table in entering data into the Ladder Editor. For the variables that need to be set as a global variable, go to the Global Variables. For details, refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504). 

3.6.2. ST Programming

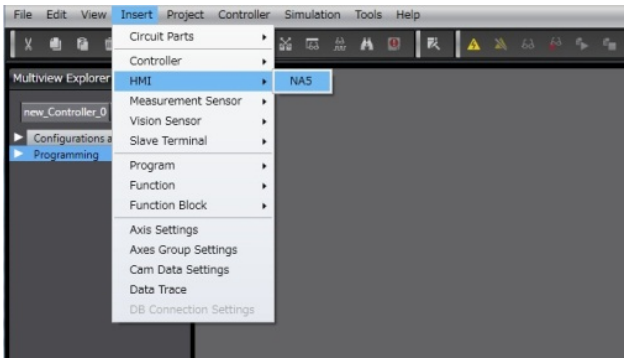
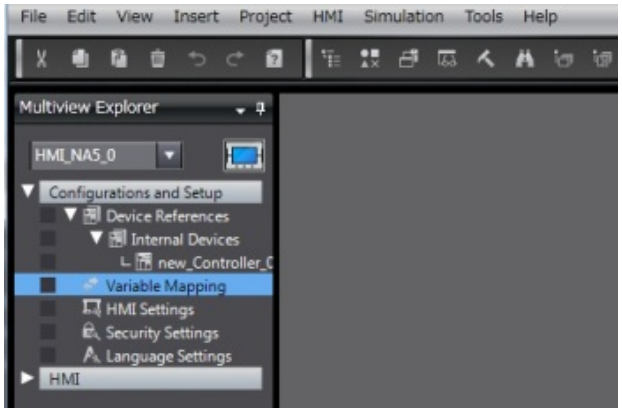
1	In the Multiview Explorer, right-click **Programs** under **Programming**, then click *Add*—*ST*. The ST Editor is displayed.
2	Create the user program. Unregistered variable names should be registered to the variable table in entering data into the ST Editor. For the variables that need to be set as a global variable, go to the Global Variables. For details, refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504). 

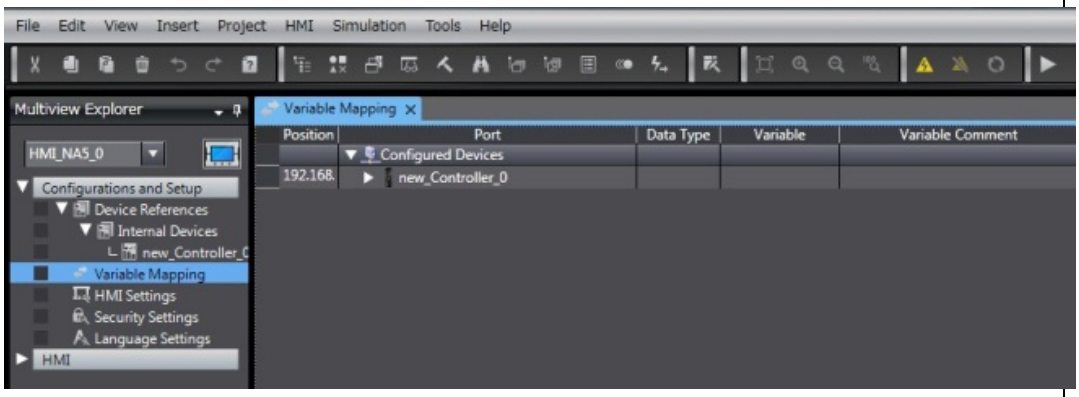
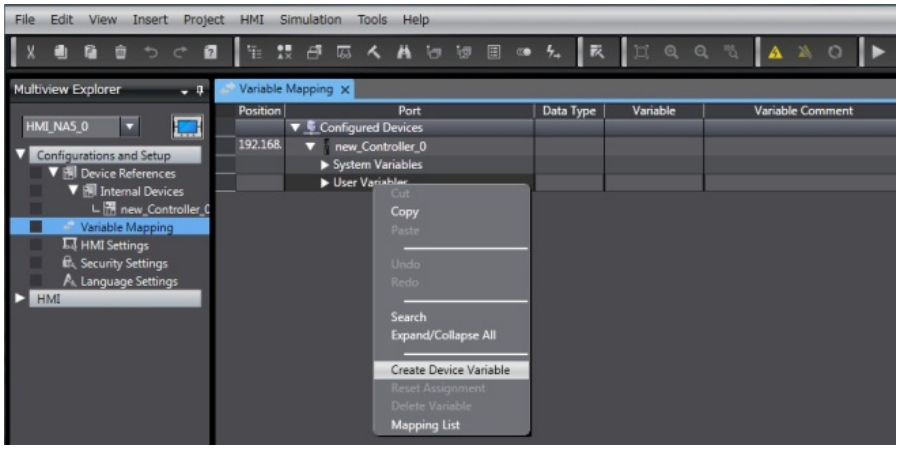
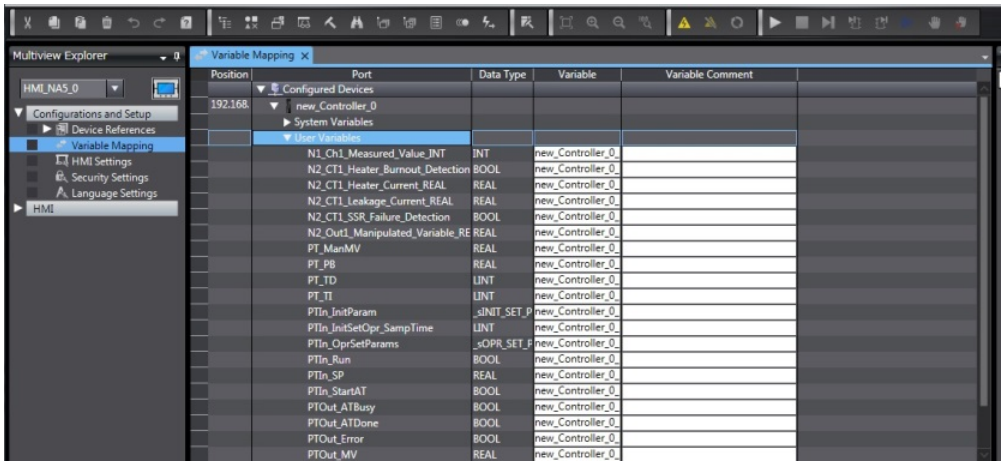
Precautions for Correct Use

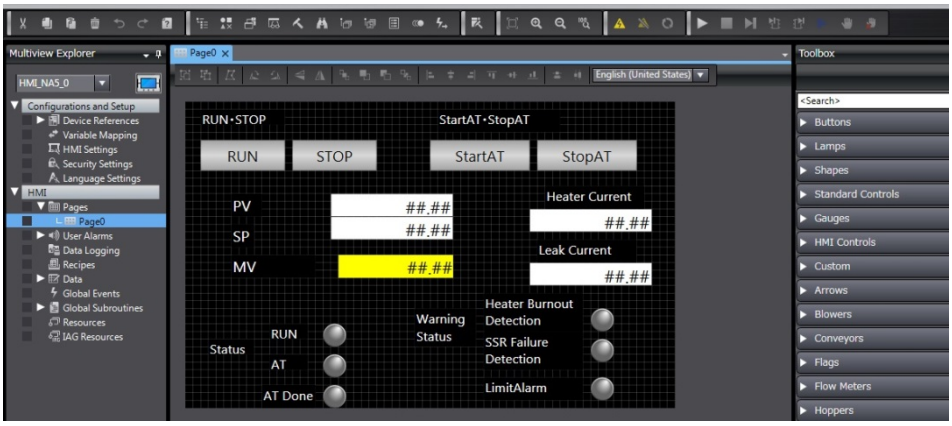
This sample program is meant to operate a temperature control system with minimum requirements. Creating the program for actual equipment requires additional programs: interlock programs related to the EtherCAT communications and the operation of the equipment, and programs related to the input to/output from other devices and the control sequence.

3.7. Creating the Window for the Programmable Terminal

Insert the Programmable Terminal to your project on the Sysmac Studio and create the Programmable Terminal window to be used for operation and display.

1	In the menu bar of the Sysmac Studio, click *Insert*, then point to *HMI* to select *NA5*. 
2	In the **Add Device** Dialog Box, make the settings as shown below. Then, click the **OK** Button. Category: HMI Device: NA5-12W101□ Version: 1.01 
3	Create a mapping from the global variables of the Controller to those of the Programmable Terminal. For details, see the manual of the Programmable Terminal. For the manual number, refer to *Related Manuals*. In the Multiview Explorer, double-click **Variable Mapping** under **Configurations and Setup**. Or, right-click **Variable Mapping** under **Configurations and Setup**, then select *Edit* from the menu. 

4	The **Variable Mapping** window is displayed. 
5	Click *new_Controller_0*, then right-click *User Variables* to select *Create Device Variable* rom the menu. 
6	The global variables of the Controller are mapped to those of the Programmable Terminal. 
7	Create the Programmable Terminal window. Create the window based on the processing that you want to perform from the Programmable Terminal. For details, see the manual of the Programmable Terminal. For the manual number, refer to *Related Manuals*. As an example, this Guide shows the window to manage the following tasks:

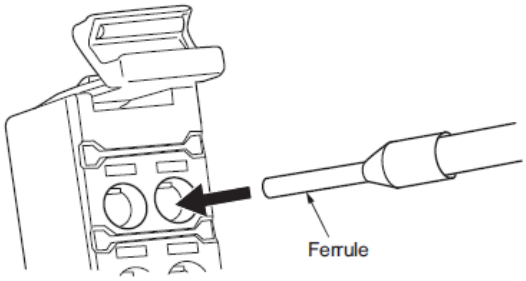
	・Switch: start/ stop of operation ・Switch: execution/stop of autotuning ・Display: operating status ・Display: current value of the control target ・Setting and display: set point of the control target ・Display: manipulated variable ・Display: heater current ・Display: status of heater burnout detection, SSR failure detection and Limit Alarm 

3.8. Wiring the Units

Carry out the wiring of NX Units. Here is an example of wiring to build the system described in this Guide.

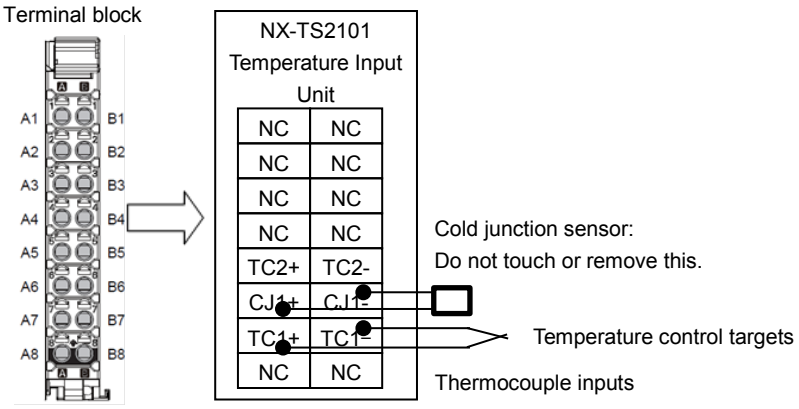
Additional Information

・ The Temperature Input Unit and Heater Burnout Detection Unit use screwless clamping terminal blocks. The use of ferrules makes wiring an easy matter of inserting them. The screwless design greatly reduces wiring work.



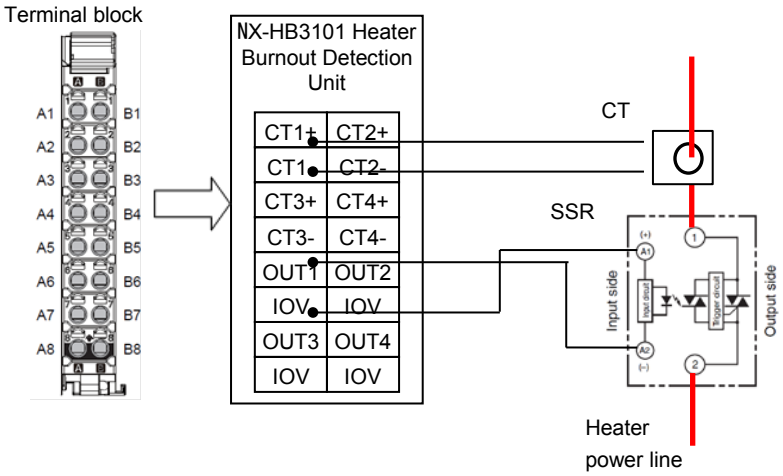
3.8.1. Wiring of the Temperature Input Unit

Carry out the wiring of the Temperature Input Unit as in the figure below. Connect the thermocouple inputs that measure the temperature of the temperature control targets to TC1+ and TC1- of the Temperature Input Unit.



3.8.2. Wiring of the Heater Burnout Detection Unit

Carry out the wiring of the Heater Burnout Detection Unit as in the figure below. Connect the CT that measures the heater current to CT1+ and CT1- of the Heater Burnout Detection Unit. Connect OUT1 and IOV to the input terminals of the SSR that turns ON/OFF a heater serving for heating control.



Additional Information

- The I/O power for control outputs of the Heater Burnout Detection Unit is supplied through the Communications Coupler Unit, or through the NX bus connector from the I/O power supply terminal of an Additional I/O Power Supply Unit. When using control outputs of the Heater Burnout Detection Unit, be sure to supply I/O power to the Communications Coupler Unit or an Additional I/O Power Supply Unit.
- The polarity of the internal I/O common terminal for control outputs from the NX-HB3101 is NPN. Inside the NX-HB3101, the common side (0 VDC) is internally connected to 0 VDC of the I/O power supply through the NX bus.

3.9. Transfer to the CPU Unit and Slave Terminal

Transfer the Unit configuration and settings and the device variables to the CPU Unit; transfer the Unit operation settings to the Temperature Input Unit and Heater Burnout

Detection Unit.

1	From the menu bar of the Sysmac Studio, select *Controller*, then *Online*
2	Select *Controller*, then *Transfer*, then *To Controller*.
3	Clear the selection of the *Do not transfer the following. (All items are not transferred.)* Check Box. Click the **Execute** Button.

3.10. Transfer of the Window Data to the Programmable Terminal

Transfer the window data you created to the actual device, or the Programmable Terminal.

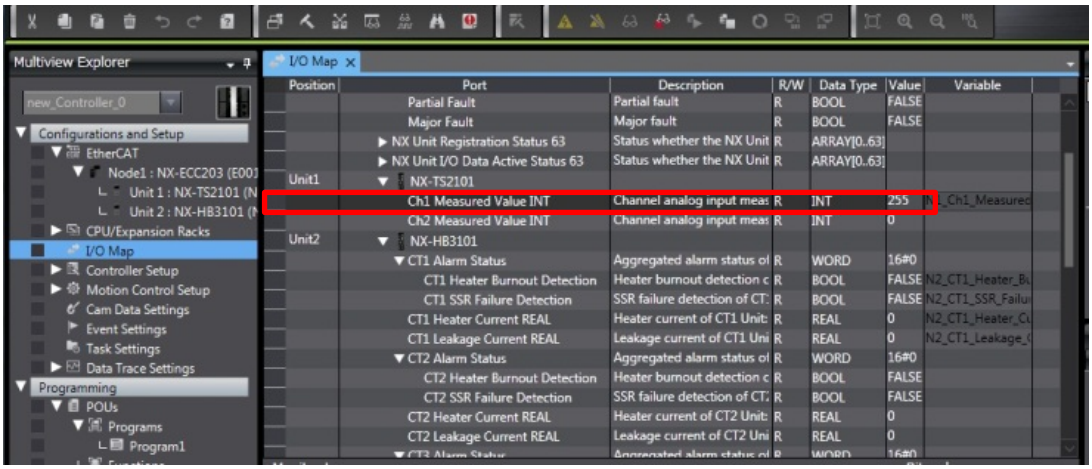
1	From the menu of the Sysmac Studio, select *HMI*, then *Online*.
2	From the menu of the Sysmac Studio, select *HMI*, then *Synchronization*, then *NA Device*.
3	In the synchronization window, click the **To HMI** Button to transfer the settings.

3.11. Test Operation

Perform a test operation to check the operation of the Temperature Input Unit and Heater Burnout Detection Unit, and enable autotuning for temperature control. Before proceeding the following steps, from the menu bar of the Sysmac Studio, select *Controller*, then *Online* to enter an online state, and select *Controller*, then *Monitor* to activate the monitoring.

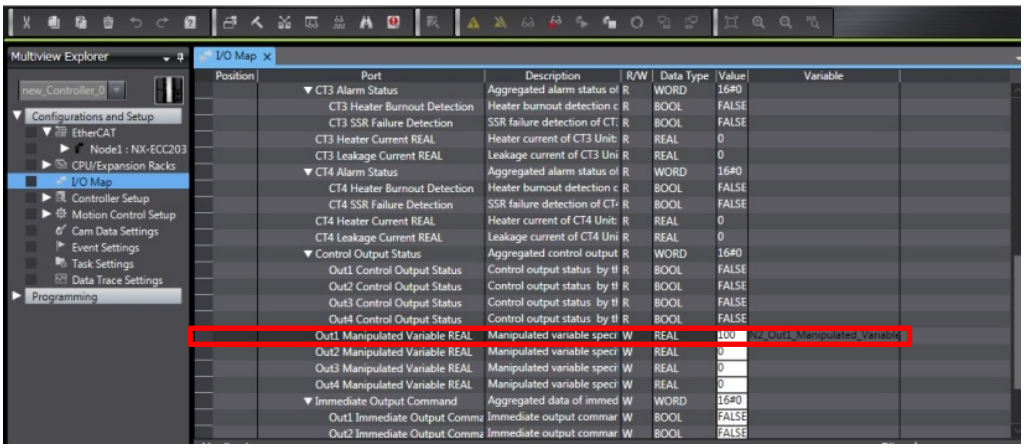
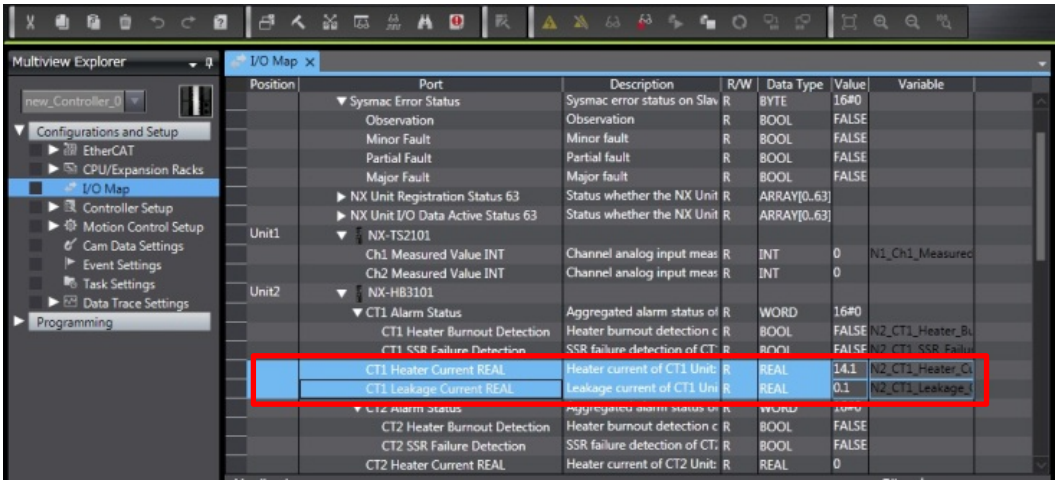
3.11.1. Check on the Temperature Input Unit

The measured temperature of the control target, which is inputted to the Temperature Input Unit, appears on *Ch1 Measured Value INT* of the I/O port. The measurement value can be checked in the I/O Map Tab Page or Watch Tab Page. The example below indicates a measurement value of 25.5°C.



3.11.2. Check on the Heater Burnout Detection Unit

Check the wiring on the SSR and valve connected to the control output. Then, check the CT input and verify the heater current.

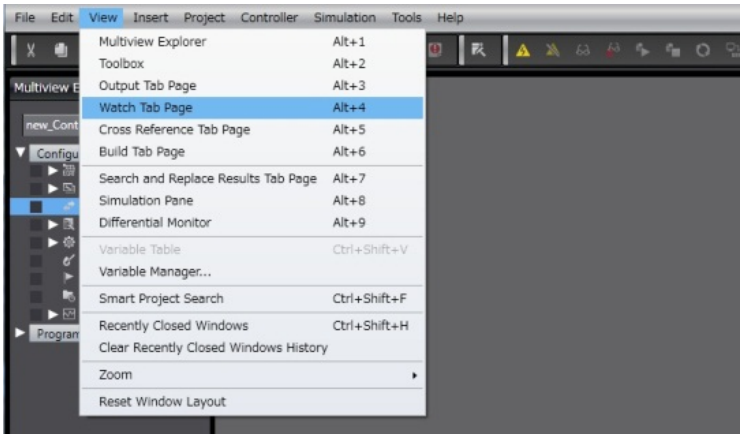
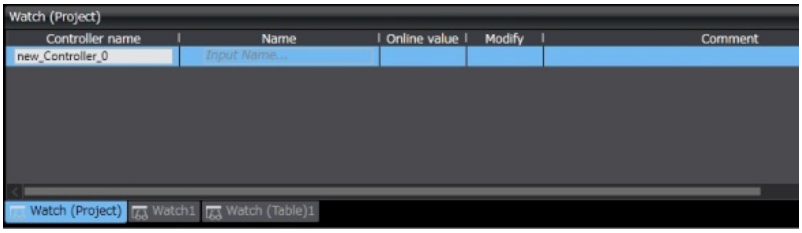
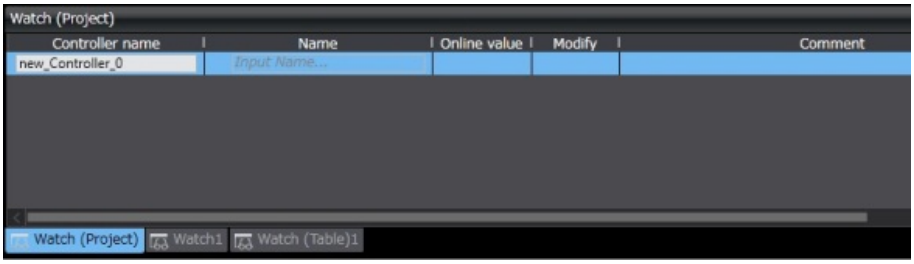
1	Check the wiring on the SSR and valve connected to the control output. From the I/O Map Tab Page or Watch Tab Page, change the values of *Out1 Manipulated Variable REAL* and *Out2 Manipulated Variable REAL*, both of which are the I/O ports for control output manipulated variables of the Heater Burnout Detection Unit, to change the control output forcefully. A value of 0 indicates an always OFF state and a value of 100 indicates an always ON state, so check that the SSR and valve operate accordingly in both states. In the example below, *Out1 Manipulated Variable REAL* is set to 100, so control output 1 is always ON.  For this check, you can also use I/O checking of the EtherCAT Slave Terminal.
2	Check the input to the CT. The current flowing the heater power line is read as Heater Current when the control output is ON and as Leakage Current when the control output is OFF, each being displayed on *CT1_Heater_Current_REAL* and *CT1_Leakage_Current_REAL* of the I/O ports. The read values can be checked in the I/O Map Tab Page or Watch Tab Page. The example below indicates a Heater Current of 14.1 A and a Leakage Current of 0.1A. 
3	If the Heater Current and Leakage Current that are read by the actual device are different from the values used in calculating the alarm currents described in 3.4.1. Alarm Current Calculation, calculate the alarm currents based on the current value measured by the actual device to reset Heater Burnout Detection Current and SSR Failure Detection Current. Follow the description in *3.4. Unit Operation Settings for the Heater Burnout Detection Unit* to reset them.

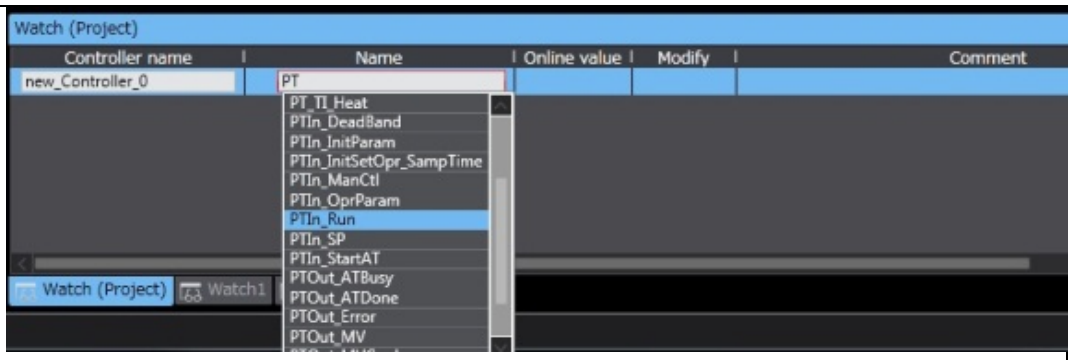
Additional Information

While the Controller is online, if setting values of the setting items other than Heater Burnout Detection Current and SSR Failure Detection Current are changed simultaneously and transferred, the restarting process of the Unit will take place. If this Unit restarting is not desirable, change only the values of Heater Burnout Detection Current and SSR Failure Detection Current before transfer.

3.11.3. Autotuning

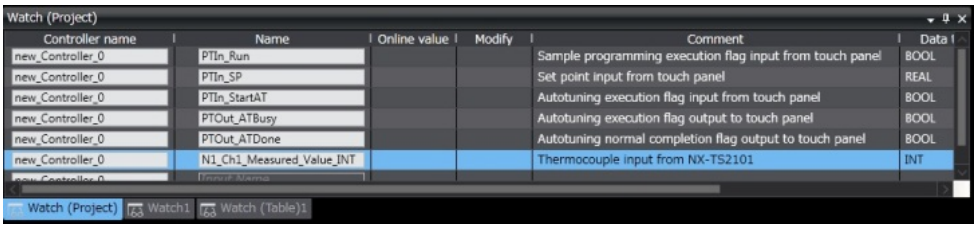
Once you have checked the proper operation of the temperature sensor, CT, SSR, and valve that are connected to the NX Unit, set the set point for the temperature control target, and then perform autotuning to obtain the optimal PID constants. This Guide shows an example using the Sysmac Studio. Before you proceed, make sure that the Controller is online.

1	From the menu bar of the Sysmac Studio, select *View*, then *Watch Tab Page*.  In Watch Tab Page (Project), Watch Tab Page1, and Watch Tab Page (Table)1 that appear, select *Watch Tab Page (Project)*. 
2	Enter *Name* of the variables that need to be manipulated for autotuning. To enter a new name, click *Input Name...* to enter the characters.  As you enter characters, the possible variable names are displayed in the list. Select the desired variable name to register it.



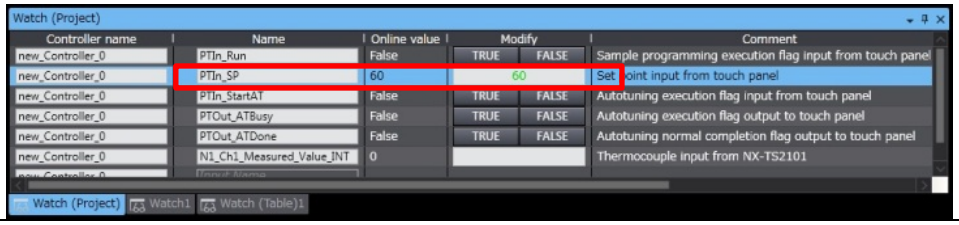
The following variables are entered in the example in this Guide:

- PTIn_Run: Input of sample program execution from the Programmable Terminal
- PTIn_SP: Input of the set point from the Programmable Terminal
- PTIn_StartAT: Input of autotuning execution from the Programmable Terminal
- PTOut_ATBusy: Output flag of autotuning being executed from the Programmable Terminal
- PTOut_ATDone: Output flag of autotuning completed successfully from the Programmable Terminal.
- N1_Ch1_Measured_Value_INT: Measured temperature of the control target from the NX-TS2101



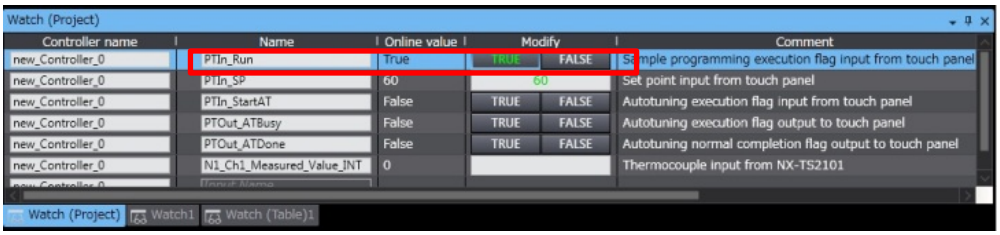
3 Enter the set point for the control target.

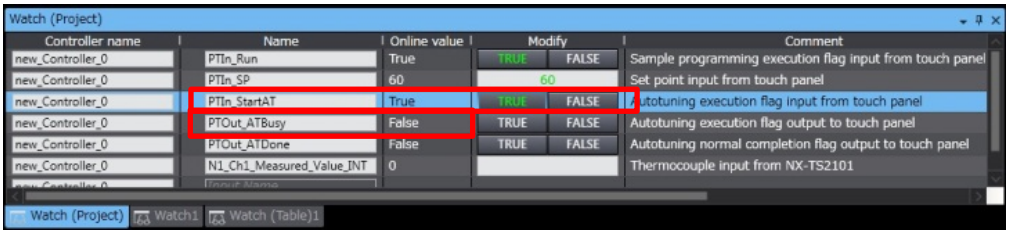
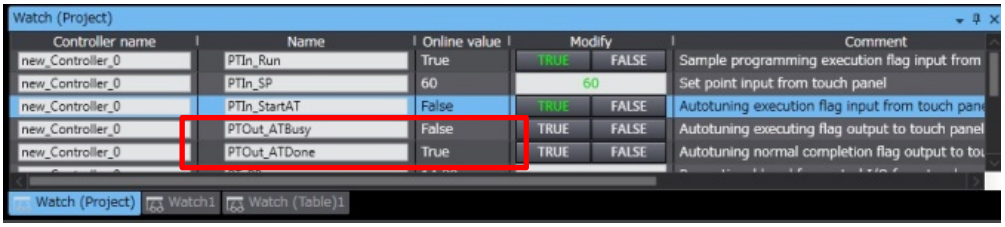
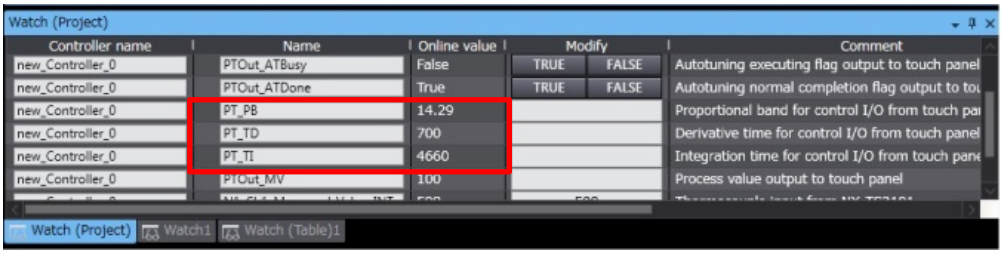
Enter 60 in the change field of *PTIn_SP*, then click. The Online value of *PTIn_SP* will be 60.



4 Start the sample program.

Click *TRUE* in the change field of *PTIn_Run*. The sample program starts to run, and the Online value of *PTIn_Run* turns to *True*.



5	Execute autotuning. Click *TRUE* of the change field of *PTIn_StartAT*. Autotuning starts. The Online value of PTIn_StartAT turns to *True*, and at the same time, the monitor value of *PTOut_ATBusy* turns to *True*.  *PTOut_ATBusy* is *True* while Autotuning is in progress.
6	Upon the completion of the autotuning, the monitor value of *PTOut_ATBusy* turns to *False* and the monitor value of *PTOut_ATDone* turns to *True*, which starts heating/cooling PID operation using the obtained PID constants without being stopped.  To check the obtained PID constants, enter the variable name of the PID constants in the **Name** field, and see the monitor value field. The figure below is an example that shows the PID constants serving for heating control, each indicating the following values: P: Proportional band for heating control 14.29 [%FS] I: Integration time for heating control 4660 [ms] D: Derivative time for heating control 700 [ms] 

You can use the feature of changing current values on the ST Editor to change and monitor the current values of variables and perform autotuning as well. For details, *7-2 Operations Used for Both Online and Offline Debugging* described in the *Sysmac Studio Version 1 Operation Manual* (W504).

If the Programmable Terminal is active and the window is prepared to allow autotuning to be

executed and the set point to be set from the Programmable Terminal, you can perform autotuning from the window of the Programmable Controller. For details, refer to the manual of the Programmable Terminal. For the manual number, see *Related Manuals*.

4. Programming Example

4.1. Specifications

Temperature control is performed by the following specification by a sample program of this Guide.

項目名	仕様
Control method	PID control
Alarm	Upper/Lower Limit Alarm Heater Burnout Alarm SSR Failure Alarm
The behavior which is at the time of warning occurrence.	Upper/Lower Limit Alarm: Output warning and temperature control is continued. Heater Burnout Alarm: Output warning and temperature control is stopped. SSR Failure Alarm: Output warning and temperature control is stopped.
Set point	60℃
Upper limit of temperature	70℃
Lower limit of temperature	0℃
Hysteresis of upper/lower limit alarm	5℃
Control cycle	2 seconds
Heater Burnout Detection Current	7 A
SSR Failure Detection Current	7 A

4.2. Settings for Unit

Refer to the *3.3 Unit Operation Settings for the Temperature Input Unit* for setting of the Temperature Input unit. And refer to the *3.4 Unit Operation Settings for the Heater Burnout Detection Unit* for setting of the Heater Burnout Detection Unit.

4.3. I/O Allocation Settings

To achieve this sample program, The I/O allocation settings for Temperature Input Unit and Heater Burnout Detection Unit are provided below.

For the parameters that are not listed in the table below, leave them at the default settings.

● I/O Allocation Settings for Temperature Input Unit

The I/O allocation settings for the Temperature Input Unit are given in the following table. These are the default allocation settings.

I/O	I/O entry mapping name	I/O entry to allocate	
		I/O entry name	Description
Input	Input Data Set 1	Ch1 Measured Value INT	Channel measured value （INT）

● I/O Allocation Settings for Heater Burnout Detection Unit

The I/O allocation settings for the Heater Burnout Detection Unit are given in the following table. These are the default allocation settings.

I/O	I/O entry mapping name	I/O entry to allocate	
		I/O entry name	Description
Outputs	Output Data Set 1	Out1 Manipulated Variable REAL	Manipulated variable specified for Out1 Unit: %
Inputs	Input Data Set 1	CT1 Alarm Status	This word contains all of the alarm status for CT1.
		CT1 Heater Current REAL	CT1 heater current Unit: Amperes
		CT1 Leakage Current REAL	CT1 leakage current Unit: Amperes

4.4. I/O Map

The settings of variables for the Temperature Input Unit and Heater Burnout Detection Unit to allocate to the I/O map are provided below.

Unit	I/O port name	Description	Variable	Variable comment	Variable type
NX-TS2101	Ch1 Measured Value INT	Channel measured value（INT）	N1_Ch1_Measured_Value_INT	Thermocouple input from NX-TS2101	Global variable
NX-HB3101	CT1 Heater Burnout Detection	CT1 Heater Burnout Detection	N2_CT1_Heater_Burnout_Detection	Heater burnout detection flag	Global variable
	CT1 SSR Failure Detection	CT1 SSR Failure Detection	N2_CT1_SSR_Failure_Detection	SSR failure Detection flag	Global variable
	Out1 Manipulated Variable REAL	Manipulate-d variable specified for Out1 Unit: %	N2_Out1_Manipulated_Variable_REAL	Manipulated variable	Global variable
	N2_CT1_Heater_Current_REAL	The REAL heater current for CT1. Unit: Amperes	N2_CT1_Heater_Current_REAL	CT1 Heater Current	Global variable
	N2_CT1_Leakage_Current_REAL	The REAL leakage current for CT1 Unit: Amperes	N2_CT1_Leakage_Current_REAL	CT1 Leakage Current	Global variable

4.5. Sample programs

4.5.1. Variables Used in Programming

- Global variable

Name	Data type	Default	AT	Retained	Network Publish	Comment
N2_CT1_Heater_Burnout_Detection	BOOL	-	ECAT://node#[1,2]/CT1 Alarm Status/CT1 Heater Burnout Detection	FALSE	Do not publish	Heater burnout detection flag
N2_CT1_SSR_Failure_Detection	BOOL	-	ECAT://node#[1,2]/CT1 Alarm Status/CT1 SSR Failure Detection	FALSE	Do not publish	SSR failure detection flag
N2_Out1_Manipulated_Variable_REAL	REAL	-	ECAT://node#[1,2]/Out1 Manipulated Variable REAL	FALSE	Do not publish	Manipulated variable
N1_Ch1_Measured_Value_INT	INT	-	ECAT://node#[1,1]/Ch1 Measured Value INT	FALSE	Do not publish	Thermocouple input from NX-TS2101
N2_CT1_Heater_Current_REAL	REAL	-	ECAT://node#[1,2]/CT1 Heater Current REAL	FALSE	Do not publish	CT1 heater current
N2_CT1_Leakage_Current_REAL	REAL	-	ECAT://node#[1,2]/CT1 Leakage Current REAL	FALSE	Do not publish	CT1 leakage current
PTIn_Run	BOOL	FALSE	-	TRUE	Input	Sample programming execution flag input from Programmable Terminal
PTIn_ManCtl	BOOL	FALSE	-	TRUE	Input	Manual/auto control flag input from Programmable Terminal
PTIn_SP	REAL	60	-	TRUE	Input	Set point input from Programmable Terminal
PTIn_StartAT	BOOL	FALSE	-	TRUE	Input	Autotuning execution flag input from Programmable Terminal

Name	Data type	Default	AT	Retained	Network Publish	Comment
PTIn_InitParam	_sINIT_SET_PARAMS	(SampTime:=T#100ms,RngLowLmt := 0.0,RngUpLmt :=100.0,DirOpr := FALSE)	-	TRUE	Input	Initial setting parameter input from Programmable Terminal
PTIn_InitSetOpr_SampTime	LINT	100	-	TRUE	Input	PID sampling period input from Programmable Terminal (unit: ms)
PTIn_OprSetParam	_sOPR_SET_PARAMS	(MVLowLmt :=0.0, MVUpLmt := 100, ManResetVal := 0.0, MVTrackSw := False, MVTrackVal := 0.0, StopMV := 0.0, ErrorMV := 0.0, Alpha := 0.65, ATCalcGain := 1.0, ATHystrs := 0.2)	-	TRUE	Input	Operation setting parameter input from Programmable Terminal
PTOut_PV	REAL	-	-	FALSE	Output	Process value output to Programmable Terminal
PT_PB	REAL	1	-	TRUE	Input	Proportional band for control I/O from Programmable Terminal
PT_TI	LINT	1000	-	TRUE	Input	Integration time for control I/O from Programmable Terminal (unit: ms)
PT_TD	LINT	1000	-	TRUE	Input	Derivative time for control I/O from Programmable Terminal (unit: ms)
PT_ManMV	REAL	0	-	TRUE	Input	Manual manipulated variable I/O from Programmable Terminal

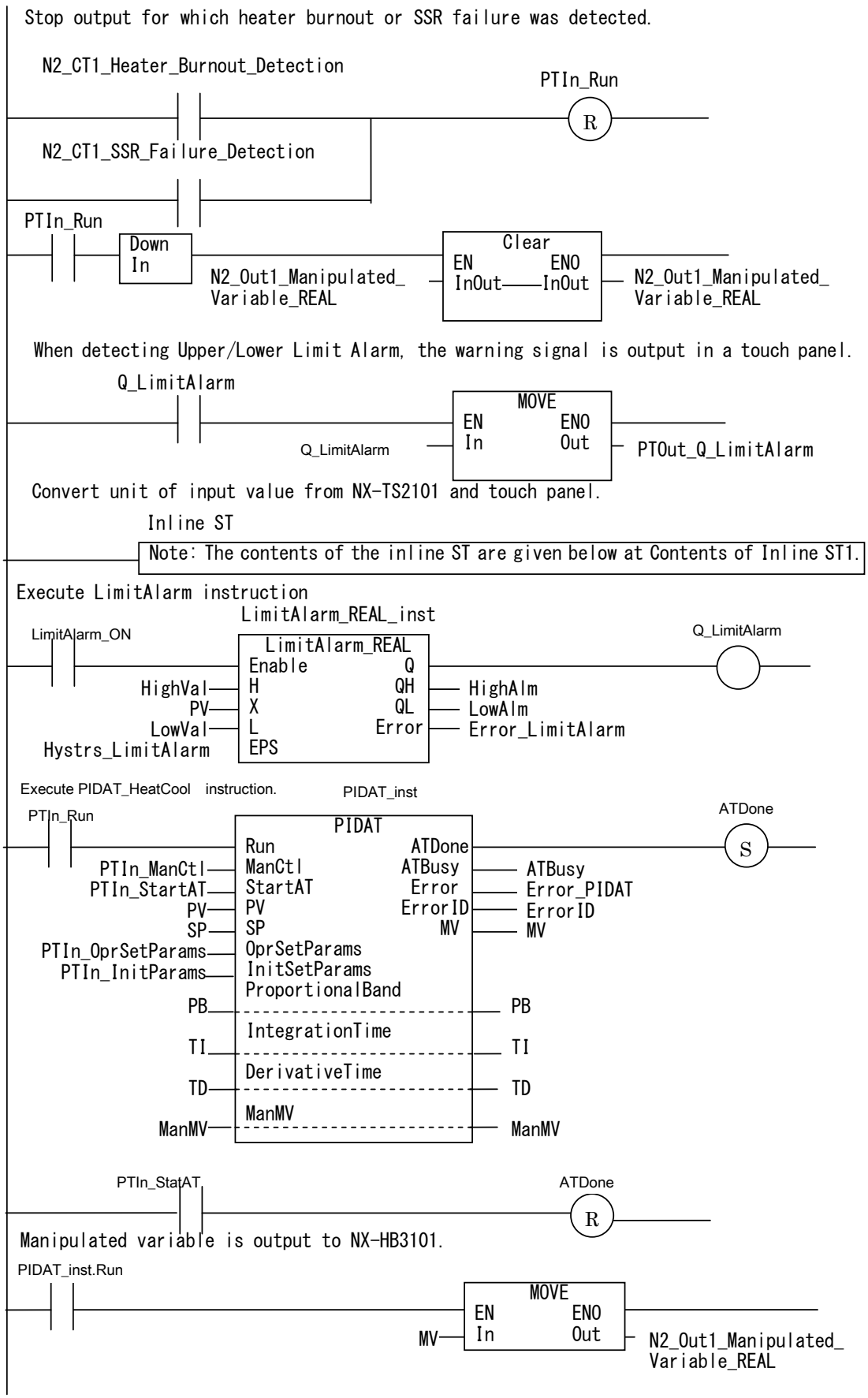
Name	Data type	Default	AT	Retained	Network Publish	Comment
PTOut_ATDone	BOOL	FALSE	-	FALSE	Output	Autotuning normal completion flag output to Programmable Terminal
PTOut_ATBusy	BOOL	FALSE	-	FALSE	Output	Autotuning executing flag output to Programmable Terminal
PTOut_Error	BOOL	FALSE	-	FALSE	Output	Error flag output to Programmable Terminal
PTOut_MV	REAL	0	-	FALSE	Output	Manipulated variable output to Programmable Terminal
PTOut_Q_LimitAlarm	BOOL	FALSE	-	FALSE	Output	Upper/Lower Limit Alarm output to Programmable Terminal

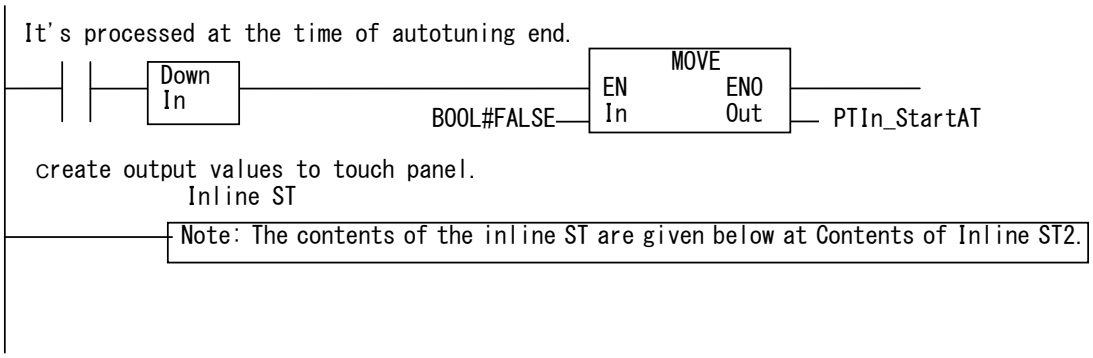
- Internal Variables

Name	Data type	Default	Comment
PB	REAL	0	Proportional band
MV	REAL	0	Manipulated variable
PIDAT_inst	PIDAT		Instance of PIDAT instruction
ATDone	BOOL	FALSE	Autotuning normal completion flag
TI	TIME	T#0s	Integration time
TD	TIME	T#0s	Derivative time
ManMV	REAL	0	Manual manipulated variable
PV	REAL	0	Process value
LimitAlarm_REAL_inst	LimitAlarm_REAL		Instance of PIDAT LimitAlarm instruction
HighAlm	BOOL	FALSE	Upper Limit Alarm
LowAlm	BOOL	FALSE	Lower Limit Alarm
Q_LimitAlarm	BOOL	FALSE	Upper/Lower Limit Alarm flag
HighVal	REAL	70	Upper Limit Value
Hystrs_LimitAlarm	REAL	5	Hysteresis of Upper/Lower Limit Alarm
LowVal	REAL	0	Lower Limit Value
LimitAlarm_ON	BOOL	TRUE	Upper/Lower LimitAlarm execution flag
Error_LimitAlarm	BOOL	FALSE	Error flag of LimitAlarm instruction
F_TRIG_inst1	F_TRIG		Instance1 of F_TRIG instruction * It's used at the time of ST Programming.
F_TRIG_inst2	F_TRIG		Instance2 of F_TRIG instruction * It's used at the time of ST.

Name	Data type	Default	Comment
ATfinsh	BOOL	FALSE	The flag to return ATDone flag to FALSE at the time of autotuning end. * It's used at the time of ST.
MVinital	BOOL	FALSE	The flag to set a zero as regulated condition of the Heater Burnout Detection unit at the time of program initiation * It's used at the time of ST.

4.5.2. Ladder Programming





●Contents of Inline ST 1

```
// Convert unit of input value from NX-TS2101 and Programmable Terminal.
PV:=INT_TO_REAL(N1_Ch1_Measured_Value_INT)/REAL#10.0;
PTIn_InitParam.SampTime:=NanoSecToTime(PTIn_InitSetOpr_SampTime*1000000);
PB:=PT_PB;
TI:=NanoSecToTime(PT_TI*1000000);
TD:=NanoSecToTime(PT_TD*1000000);
ManMV:=PT_ManMV
```

●Contents of Inline ST 2

```
// Create output values to Programmable Terminal.
PTOut_PV:=PV;
PTOut_ATDone:=ATDone;
PTOut_ATBusy:=PIDAT_inst.ATBusy;
PTOut_Error:=PIDAT_inst.Error;
PTOut_MV:=PIDAT_inst.MV;
PT_PB:= PB;
PT_TI:=TimeToNanoSec(TI)/1000000;
PT_TD:=TimeToNanoSec(TD)/1000000;
PT_ManMV :=ManMV;
```

4.5.3. ST

```
// Stop output for which heater burnout or SSR failure was detected.
IF N2_CT1_Heater_Burnout_Detection = TRUE OR N2_CT1_SSR_Failure_Detection = TRUE THEN;
    PTIn_Run   := FALSE;
    PTIn_StartAT   := FALSE;
END_IF;

//The value of the manipulated variable output to the Heater Burnout Detection unit //is initialized at the time of a program start.
F_TRIG_inst1(PTIn_Run, MVinital);
IF MVinital=TRUE THEN;
    Clear(N2_Out1_Manipulated_Variable_REAL);
END_IF;

//When detecting Upper/Lower Limit Alarm, the warning signal is output.
IF Q_LimitAlarm = TRUE THEN;
    PTOut_Q_LimitAlarm:=    LimitAlarm_REAL_inst.Q;
END_IF;

// Convert unit of input value from NX-TS2101 and Programmable Terminal.
PV :=   INT_TO_REAL(N1_Ch1_Measured_Value_INT)/REAL#10.0;
PTIn_InitParam.SampTime := NanoSecToTime(PTIn_InitSetOpr_SampTime*1000000);
PB := PT_PB;
TI := NanoSecToTime(PT_TI*1000000);
TD:= NanoSecToTime(PT_TD*1000000);
ManMV    := PT_ManMV;

//Execute LimitAlarm instruction.
LimitAlarm_REAL_inst(
Enable :=LimitAlarm_ON,
H :=HighVal,
X :=PV,
L :=LowVal,
EPS :=Hystrs_LimitAlarm,
Q =>Q_LimitAlarm,
QH =>HighAlm,
QL =>LowAlm,
Error =>Error_LimitAlarm);

//Execute PIDAT_HeatCool   instruction.
PIDAT_inst(Run:=PTIn_Run,
        ManCtl:=PTIn_ManCtl,
        StartAT:=PTIn_StartAT,
```

```
        PV:=PV,
        SP:=PTIn_SP,
        OprSetParams:=PTIn_OprParams,
        InitSetParams:=PTIn_InitParam,
        ProportionalBand:=PB,
        IntegrationTime:=TI,
        DerivativeTime:=TD,
        ManMV:=ManMV,
        MV=>MV);

//The value of ATDone output to a Programmable Terminal is kept in FALSE at the time of autotuning
//execution.
IF PTIn_StartAT = TRUE THEN;
    ATDone:=  FALSE;
END_IF;

// Manipulated variable is output to NX-HB3101.
N2_Out1_Manipulated_Variable_REAL := MV;

// It's processed at the time of autotuning end.
F_TRIG_inst2(PIDAT_inst.ATBusy, ATfinsh);
IF ATfinsh=TRUE THEN;
    PTIn_StartAT:=  FALSE;
END_IF;

//Create output values to Programmable Terminal.
PTOut_PV :=         PV;
PTOut_ATDone :=     PIDAT_inst.ATDone;
PTOut_ATBusy :=     PIDAT_inst.ATBusy;
PTOut_Error   :=    PIDAT_inst.Error;
PTOut_MV  :=   PIDAT_inst.MV;
PT_PB :=  PB;
PT_TI :=   TimeToNanoSec(TI)/1000000;
PT_TD :=  TimeToNanoSec(TD)/1000000;
PT_ManMV    :=      ManMV;
```

Note: Do not use this document to operate the Unit.

OMRON Corporation **Industrial Automation Company**

Tokyo, JAPAN

Contact: www.ia.omron.com

Regional Headquarters

OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388

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

OMRON ASIA PACIFIC PTE. LTD.
No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967
Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:

Cat. No. P105-E1-01 0916 (0916)