RX23E-A Group

Example of IO-Link device (temperature sensor)

Summary

This application note describes a sample program that uses the Renesas Solution Starter Kit for RX23E-A (RSSKRX23E-A) and the IA Sensor Network Connector Board to achieve IO-Link communication with the RX23E-A. For IO-Link communication, the IO-Link stack made by TMG is used.

Target Device

RX23E-A, ZIOL2401 (IO-Link PHY)

When applying this application note to other microcontrollers, please modify it to suit the specifications of the microcontroller and evaluate it thoroughly.

Contents

1. Overview 5

2. Related Documents 6

3. Environment for Operation Confirmation 7

4. Sample application overview 10
4.1 Overview of overall processing flow 11
4.2 Flowchart of App_Temperature function 13
4.3 Flowchart of App_Cmd function 19
4.4 Operating mode and switching state 21
4.4.1 Window mode 22
4.4.2 Two point mode 23
4.5 IO-Link communication specifications 24
4.5.1 Bit rate 24
4.5.2 SIO mode 24
4.5.3 Process data (Input) 24
4.5.4 Parameters 26
4.6 Peripheral Functions and Pins Used 28
4.7 Program Configuration 29
4.7.1 File Configuration 29
4.7.2 Functions 30
4.7.2.1 main.c 30
4.7.2.2 r_temperature_measurementc 31
4.7.2.3 r_led_api.c 32
4.7.2.4 IOLinkMain.c 33

5. Operation check procedure 35
5.1 IP address setting of PC 35
5.2 Launch IO-Link Device Tool V5.1 - PE 36
5.3 Update IO-Link Device Catalog 37
5.4 Update IO-Link Master Catalog 38
5.5 Check for Catalog updates 39
5.6 Setup of IO-Link communication 40
5.7 IOSK RX23E-A board sensor demo for IO-Link device tools 45
5.7.1 Common tab 46
5.7.2 Process Data tab 47
5.7.3 Identification tab 48
5.7.4 Observation tab 49
5.7.5 Parameter tab 50
5.7.6 Scope tab 51
5.7.7 Device Parameter Change (Teach-In / Read) 52
5.7.7.1 Parameter setting in Teach Values (block write mode) 52
5.7.7.2 Parameter setting in Teach Values (direct write mode) 53
5.7.7.3 Parameter settings in the Standard Command 54

5.7.7.4 Parameter reading 55
5.7.8 Restore to factory settings 55
5.7.9 Generic tab 55
5.7.10 IODD tab 55

1. Overview

This application note is based on the application note "RX23E-A group temperature measurement example using thermocouples" (R01AN4747) and explains how to realize IO-Link communication using the IO-Link stack made by TMG.

In this example, the RSSKRX23E-A board and IA Sensor Network Connector Board are used as IO-Link devices, and the IO-Link Master Development Kit (RZ / N1S-IO-LINK-M board) made by Renesas Electronics is used as the IO-Link master. Communication with the IO-Link master uses "IO-Link Device Tool V5.1 – PE" made by TMG. "IO-Link Device Tool V5.1-PE" is application software that runs on a Windows PC.

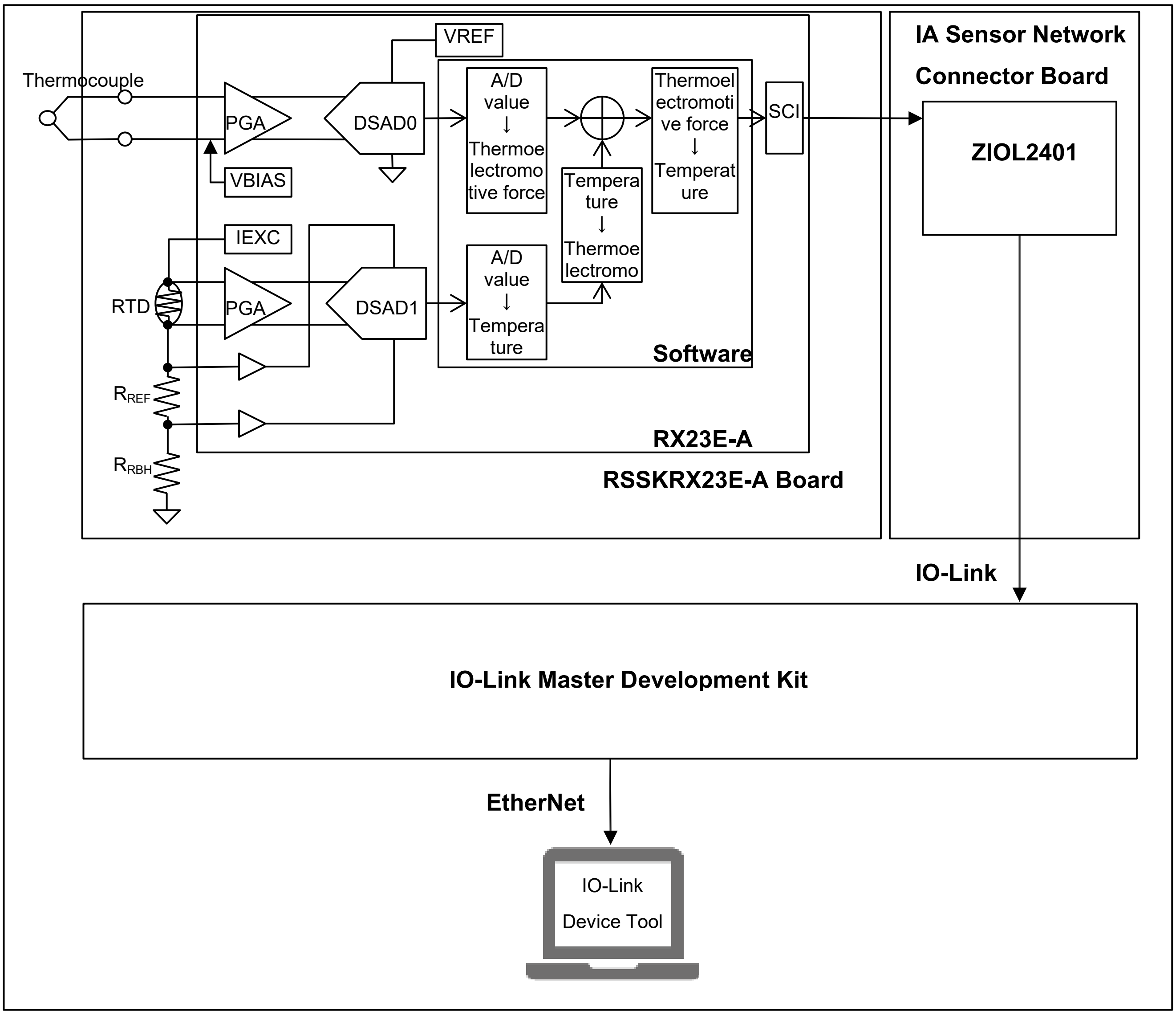


Figure 1-1 System configuration in this application note

2. Related Documents

- RX23E-A Group User's Manual: Hardware R01UH0801
- RSSKRX23E-A User's Manual: Board R20UT4542
- Application Note RX23E-A group temperature measurement example using thermocouples R01AN4747

3. Environment for Operation Confirmation

The environment for operation confirmation is given in Table 3-1. An image of a device with this configuration is shown in Table 3-1. An image of a device with this configuration is shown in Figure 3-1.

Table 3-1 Environment for Operation Confirmation

Item	Description
CPU board	RSSKRX23E-A board (RTK0ESXB10C00001BJ) It connects to the IA Sensor Network Connector Board and operates as an IO-Link device.
MCU	RX23E-A（R5F523E6ADFL） Power voltage (VCC, AVCC0)：5V Operating frequency (ICLK)：32MHz Peripheral operating frequency (PCLKB)：32MHz DSAD operating frequency (f_{DR})：4MHz DSAD modulator clock frequency (f_{MOD})：0.5MHz
RTD (on board)	Vishay PTS060301B100RP100
Thermocouple	Labfacility Ltd XE-3505-001
IDE	Renesas e^2 studio 2020-07 Renesas Smart Configurator 20.7.0.v20200629-0858
Tool Chain	Renesas CC-RX V3.02.00
Library	IO-Link stack made by TMG
Emulator	E1 Emulator
IO-Link communication board	IA Sensor Network Connector Board (RTK0EF0085B00001BJ) It connects to the RSSKRX23E-A board and operates as an IO-Link device.
IO-Link Line Driver	ZIOL2401
IO-Link Master	RZ/N1S-IO-LINK-M board
IO-Link Tool	IO-Link Device Tool V5.1 - PE made by TMG
Host PC for IO-Link tools	Windows10 Professional

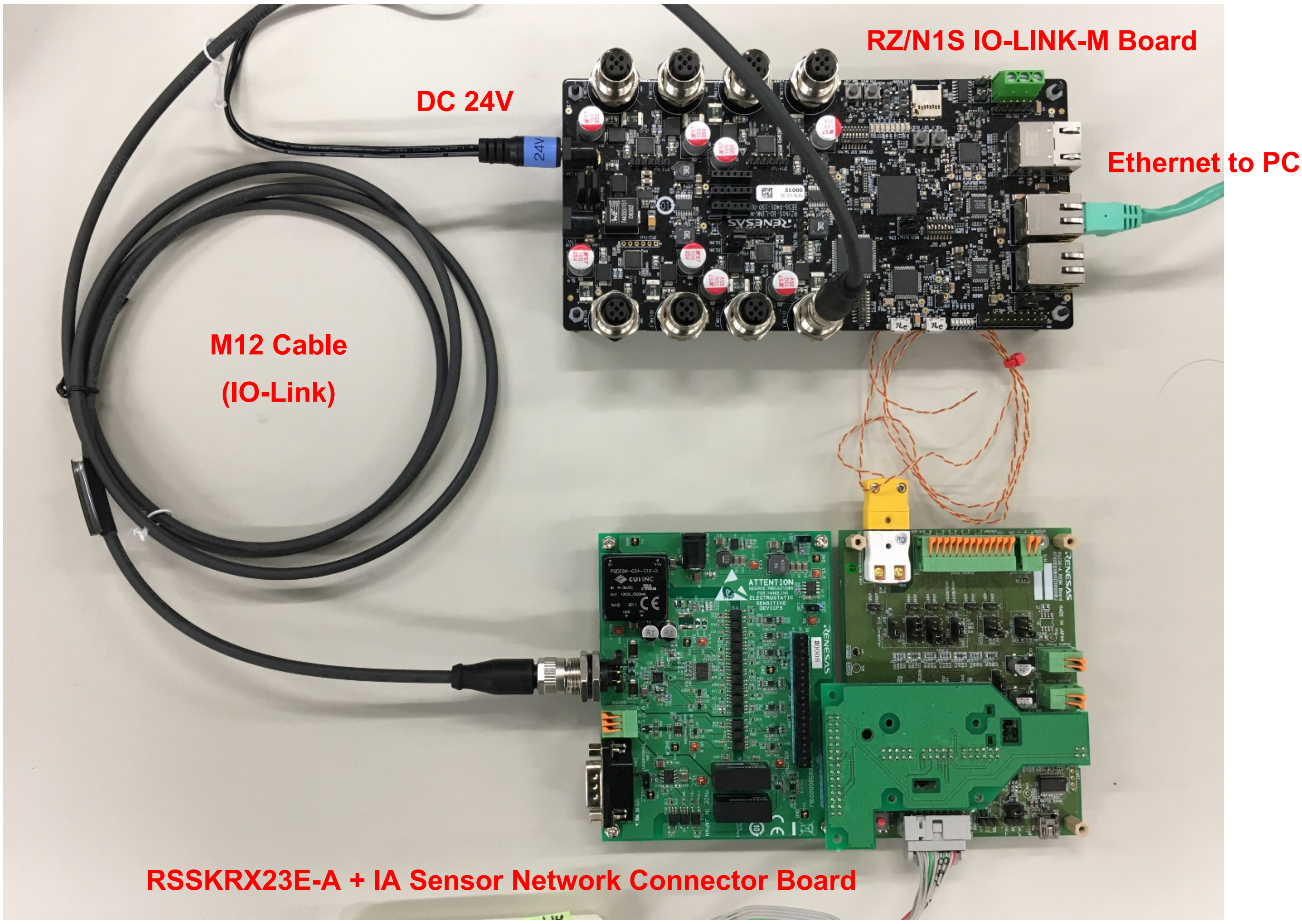


Figure 3-1 Overall system view

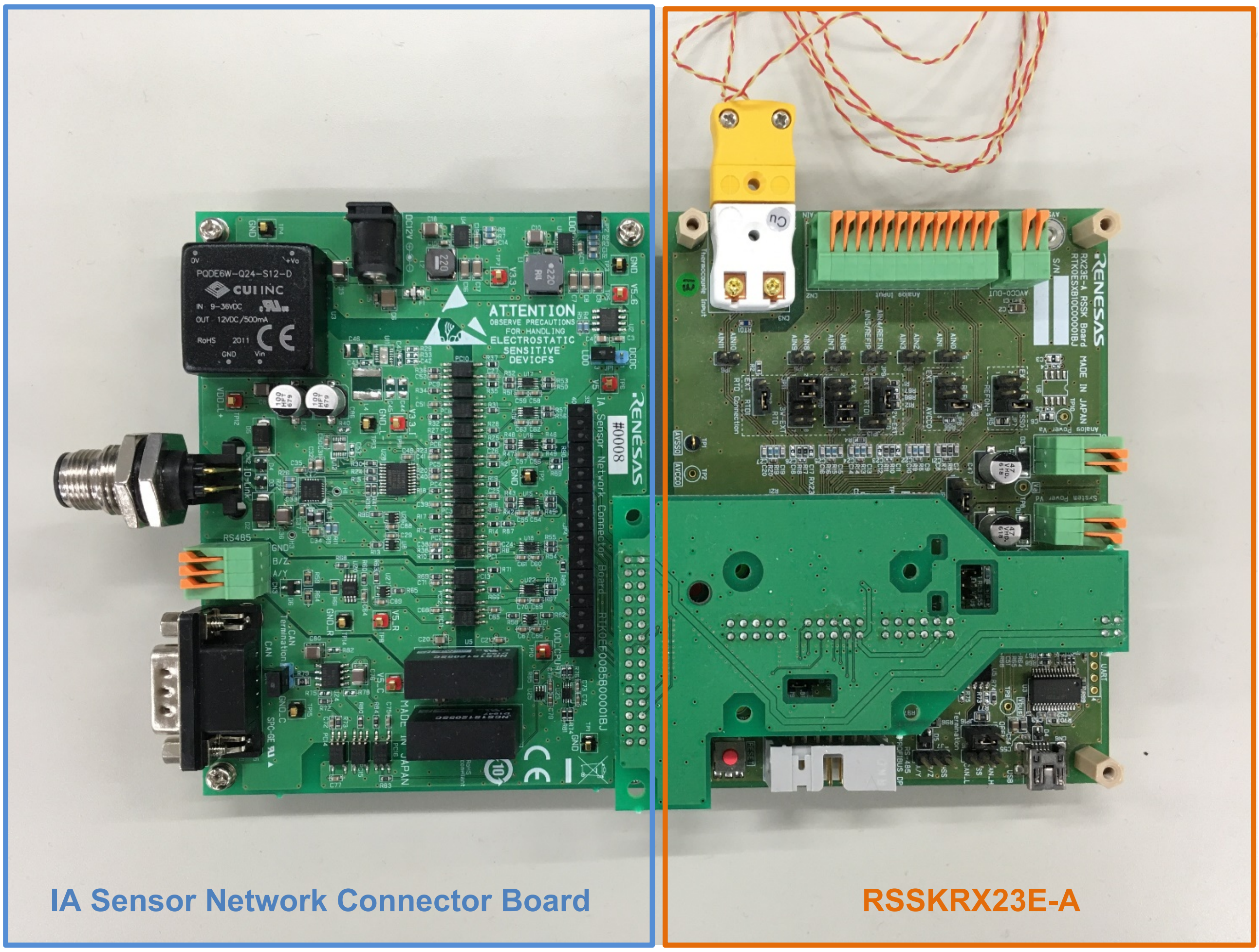


Figure 3-2 RSSKRX23E-A＋IA Sensor Network Connector Board

4. Sample application overview

The IO-Link device used in this sample application is equipped with a smart sensor profile and receives information about switching modes and thresholds through the teaching process. The IO-Link device periodically (once every 100 [ms]) measures the temperature and determines the threshold, and transmits the information over IO-Link. The information to be transmitted (process data) consists of temperature data and switching status (threshold judgment result).

For more information on smart sensor profiles, refer to the documents related to IO-Link smart sensor profiles that can be downloaded from https://io-link.com/en/.

4.1 Overview of overall processing flow

Figure 4-1 shows the process flow of this sample application.

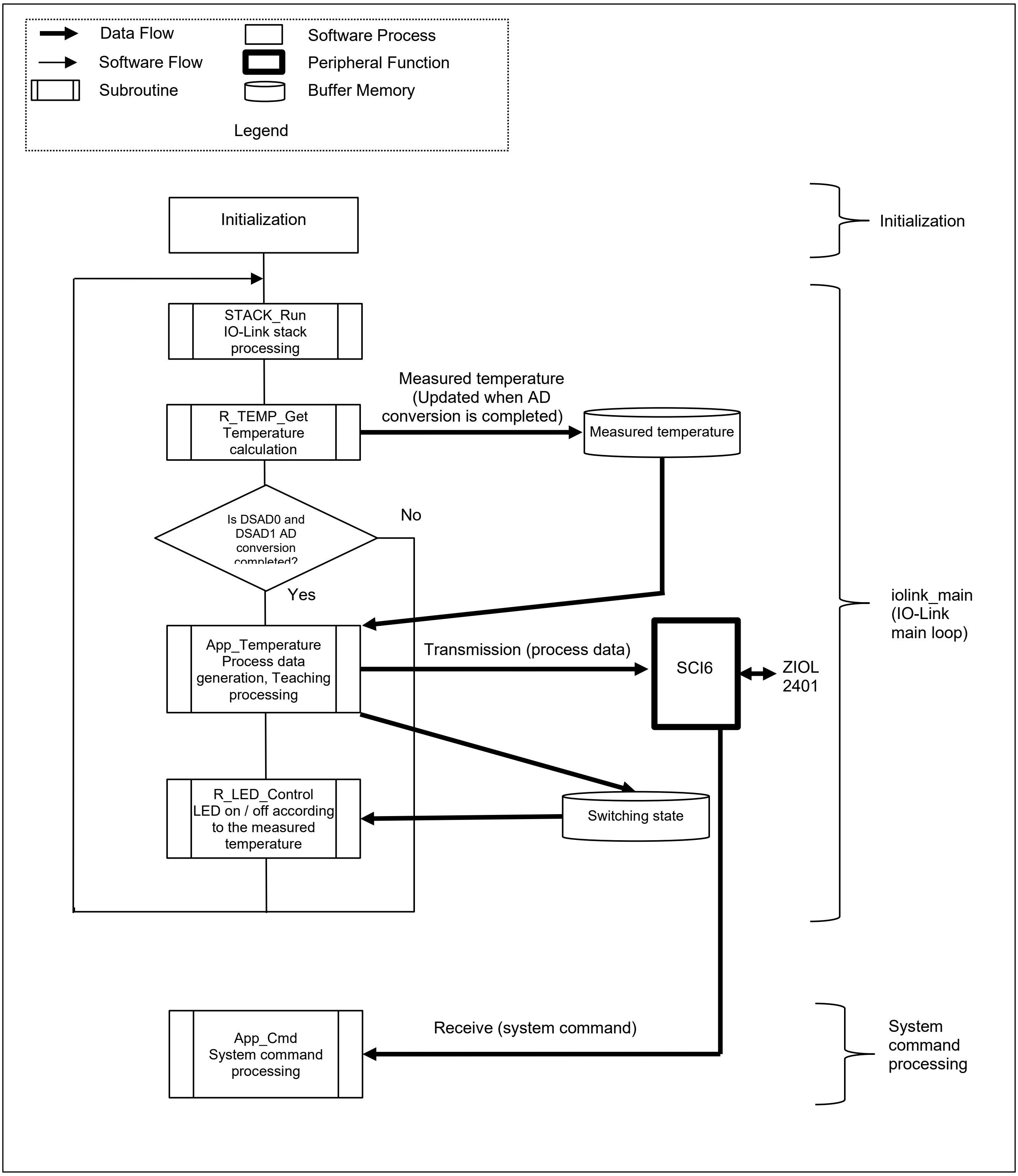


Figure 4-1 the process flow of this sample application

The following provides an overview of each process.

1. Initialization

   Enable synchronous start of DSAD0 and DSAD1.

   Start of the A/D conversion of DSAD0 and DSAD1.

   Initialize the Flash module of FIT.

   Read data from the data flash and store it in a variable.

2. iolink_main

   The main loop of the IO-Link sample application. The following is an overview of each process.

   A) The execution of the Run function of the IO-Link stack (STACK_Run)

      Execute the STACK_Run function of the API provided by the IO-Link stack. This function should be run periodically.

   B) Temperature calculation (R_TEMP_Get)

      When the A/D conversion of DSAD0 and DSAD1 is completed, the temperature is calculated from the A/D value and the processes of C) and D) are performed. If the A/D conversion is not completed, it returns to A).

   C) Process data generation, teaching process (App_Temperature)

      Generate process data from the measured temperature.

      If the Teaching command is being executed, the parameters are checked, and if it is out of the valid range, the Teaching command fails. If it is within the valid range, the Teaching command is successful and the SP1 or SP2 settings are updated according to the command.

      The switching status is judged from the operation mode and the measured temperature. When the Teaching command is being executed (Teach_Result is other than IDLE or SUCCESS), the switching status is not determined and the switching state is regarded as OFF.

      When ConfigLogic (logical setting of switching state) is set to Inverted, the bits of the switching state are inverted.

      Execute AL_SetInputReq of the API provided by the IO-Link stack. As an argument, pass a pointer to the process data created in the previous steps.

      For more details on this process, see Figure 4-2 to Figure 4-5.

   D) LED on / off (R_LED_Control)

      Since the switching state can be obtained as a return value in the process of C), pass this to the argument of R_LED_Control. LED1 turns on when the switching state is ON, and ED1 turns off when the switching state is OFF.

3. System Command Processing (App_Cmd)

   It is executed from the IO-Link stack when a system command is received on the IO-Link. Only accepts when the Teaching command is not running (IDLE state). Only two commands are supported, the SP1 Teaching command (65) and the SP2 Teaching command (66). When these are received, it is assumed that the Teaching command of SP1 or SP2 is being executed.

4.2 Flowchart of App_Temperature function

Figure 4-2 to Figure 4-5 show the flowchart of the App_Temperature function.

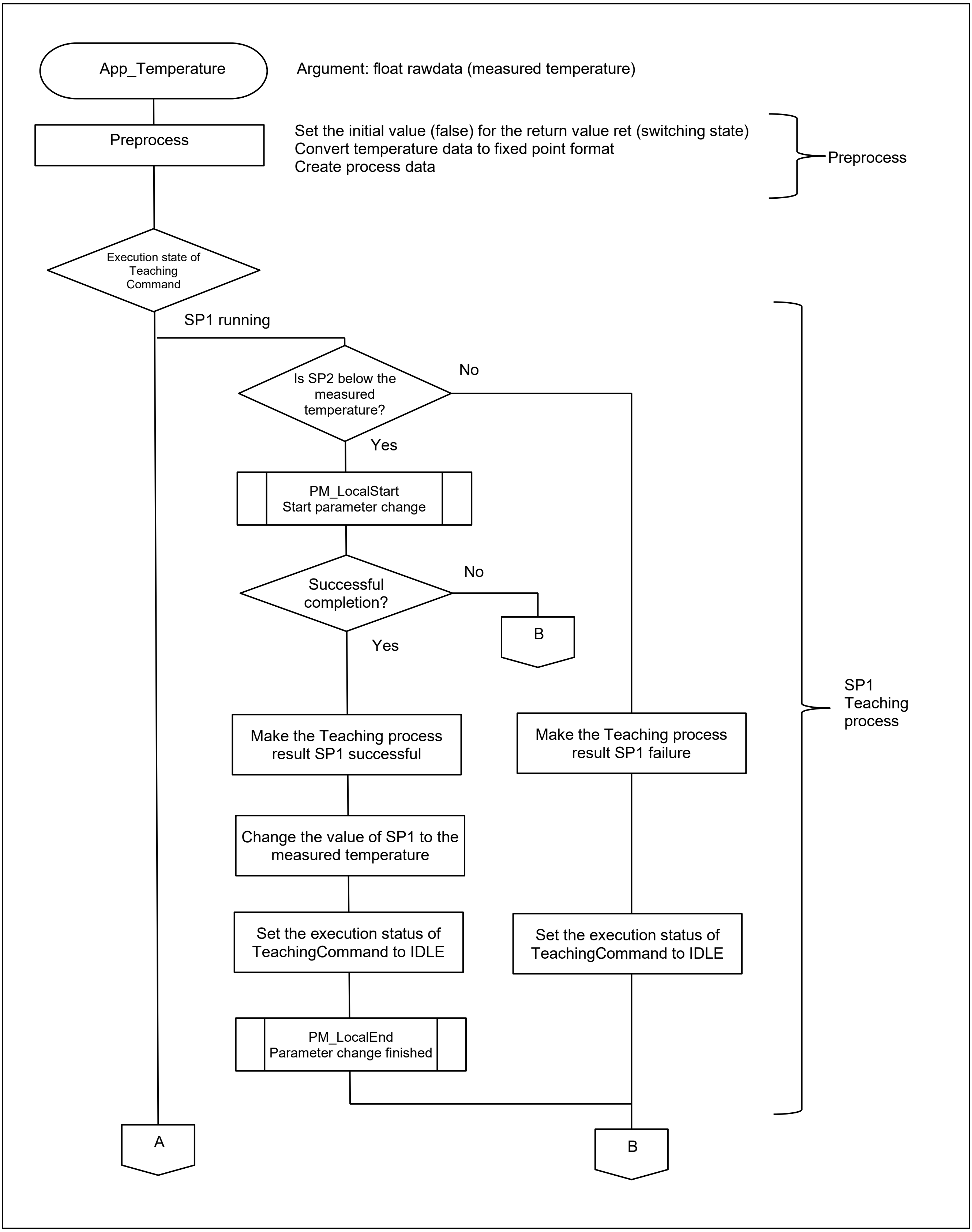


Figure 4-2 App_Temperature Flowchart (1/4)

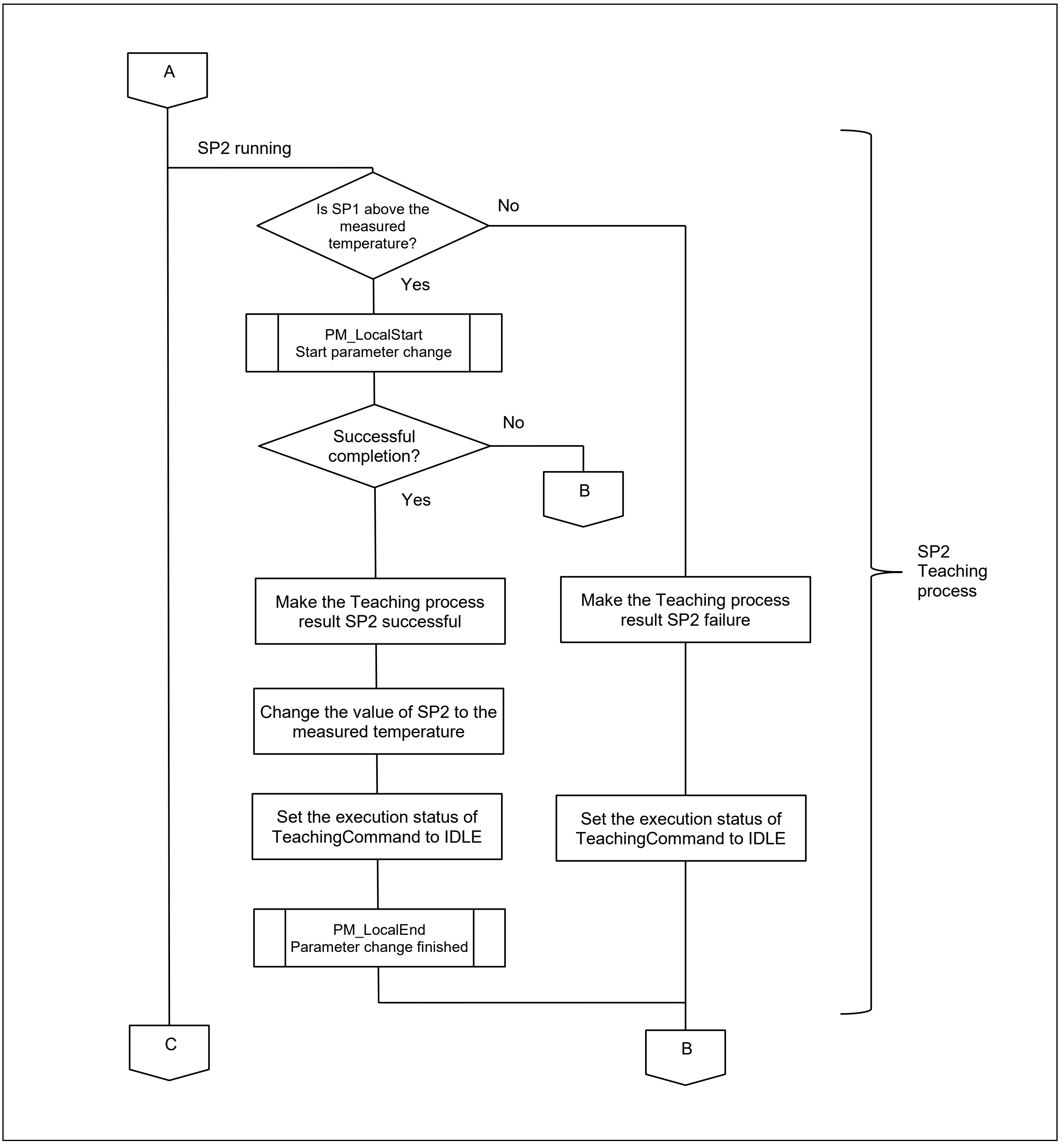


Figure 4-3 App_Temperature Flowchart (2/4)

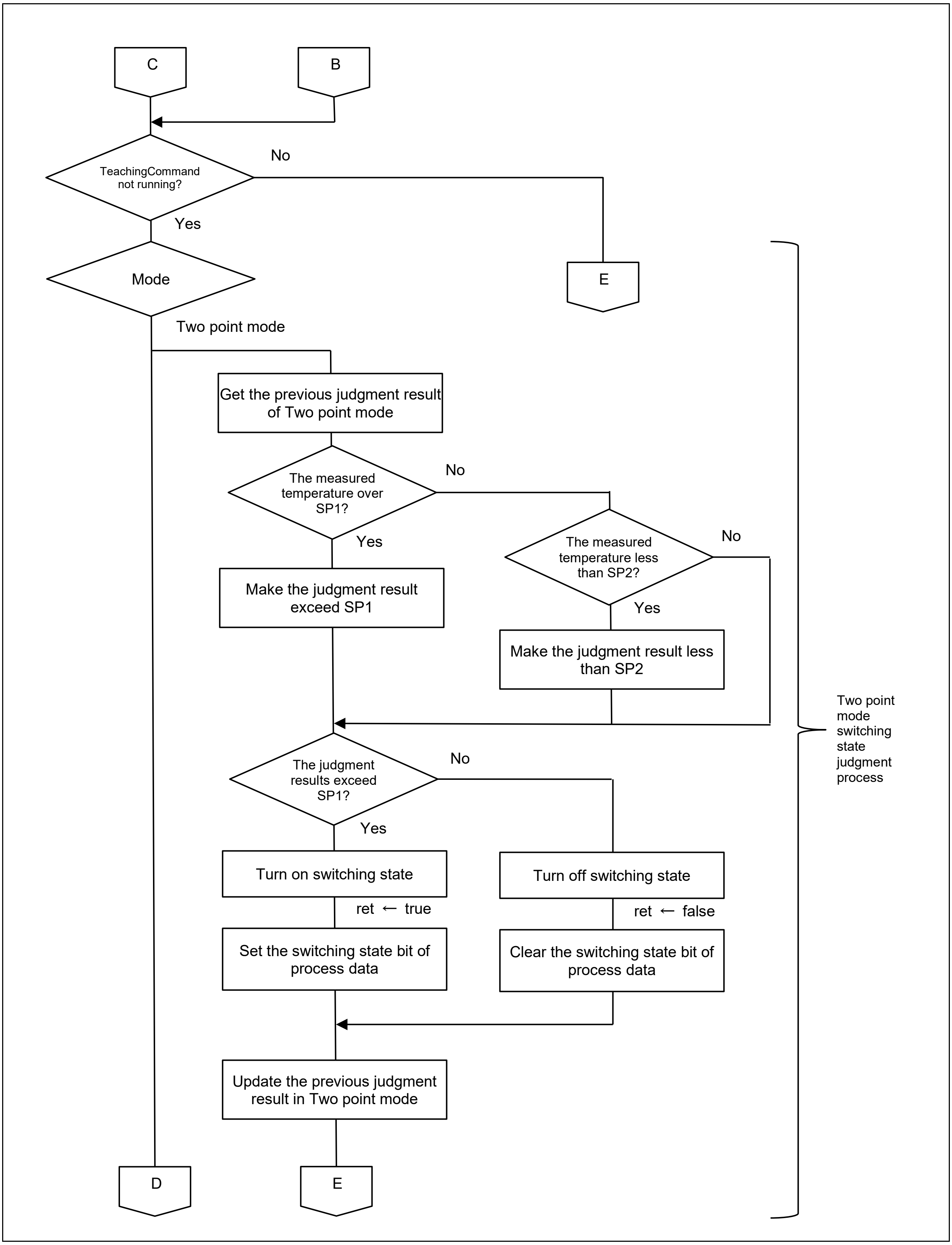


Figure 4-4 App_Temperature Flowchart (3/4)

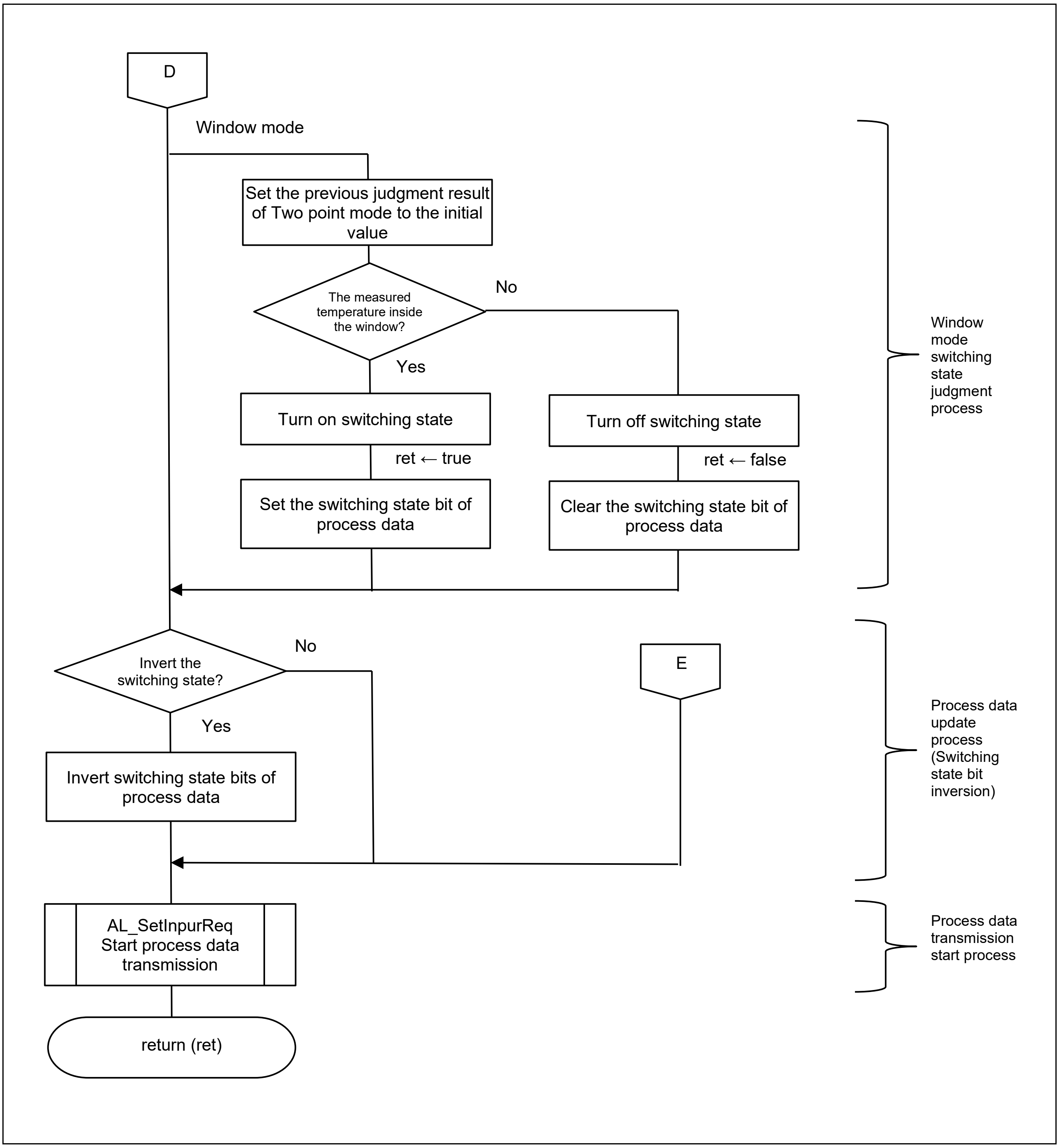


Figure 4-5 App_Temperature Flowchart (4/4)

The outline of each process is described.

1. Preprocess

   Set the initial value (false: OFF) to the return value ret (switching state).

   Multiply the measured temperature (float) passed as an argument by 100, cast to int16_t, and convert to fixed-point format in 0.01 [℃] units.

   Create process data.

   ProcessData.PDIn[16]: Store the upper 8 bits of temperature data.

   ProcessData.PDIn[17]: Store the lower 8 bits of temperature data.

   ProcessData.PDIn[18]: The least significant bit is in the switching state. The initial value is 0.

2. SP1 Teaching Process*Note

   When the execution status of TeachingCommand is "SP1 TeachingCommand", the following processing is performed.

   Since SP1≧SP2 must be satisfied, if the measured temperature <SP2, the result of the Teaching process (ParSetStatic.V_TeachResult) is regarded as SP1 failure.

   Otherwise, the result of the Teaching process is assumed to be SP1 success, and the measured temperature (int16_t) is copied to SP1 (ParSet.V_SetPointValues.SP1) and the TeachingCommand execution state is left idle.

3. SP2 Teaching Process*Note

   When the execution status of TeachingCommand is "SP2 TeachingCommand", the following processing is performed.

   Since SP1≧SP2 must be satisfied, if SP1< the measured temperature, the result of the Teaching process (ParSetStatic.V_TeachResult) is regarded as SP2 failure.

   Otherwise, the result of the Teaching process is assumed to be SP2 success, and the measured temperature (int16_t) is copied to SP2 (ParSet.V_SetPointValues.SP1) and the TeachingCommand execution state is left idle.

4. Two point mode switching state judgment process

   If TeachingCommand is not running (idle or successful) and the operating mode (ParSet.V_SetPointConfig.Mode) is Two point mode, the following processing is performed.

   Acquire the previous judgment result of Two point mode (initial value is less than SP2).

   If SP1 < measured temperature, the judgment result is considered to exceed SP1.

   If SP2 > measured temperature, the judgment result is less than SP2.

   Otherwise, the judgment result will be the previous judgment result.

   If the judgment result exceeds SP1, the switching state is set to ON and ProcessData.PDIn [18] is set to 1.

   If the judgment result is less than SP2, the switching state is set to OFF and ProcessData.PDIn [18] is set to 0.

   Update the previous judgment result in Two point mode to the current judgment result.

5. Window mode switching state judgment process

   If TeachingCommand is not running (idle or successful) and the operating mode (ParSet.V_SetPointConfig.Mode) is Two point mode, the following processing is performed.

   Set the previous judgment result in Two point mode to less than the initial value of SP2.

   When SP1≧measured temperature T [℃]≧SP2, the switching state is set to ON and ProcessData.PDIn [18] is set to 1.

   Otherwise, the switching state is turned off and ProcessData.PDIn [18] is set to 0.

6. Process data update process (switching state bit inversion)

   If the switching state logic setting (ParSet.V_SetPointConfig.Logic) is 1 (Inverted), the following processing is performed.

   Invert the least significant bit of ProcessData.PDIn [18].

7. Process data transmission start process

   Pass the pointer of the process data (ProcessData.PDIn) created so far to the AL_SetInputReq function provided by the IO-Link stack, and start the transmission of process data to the IO-Link master.

*Note:

Before changing ParSet.V_SetPointValues, execute the PM_LocalStart function provided by the IO-Link stack and make sure the return value is True. After the change, you must execute the PM_LocalEnd function. If the return value of the PM_LocalStart function is false, exit the App_Temperature function without changing the execution status of ParSetStatic.V_TeachResult, Parset.V_SetPointValues, and TeachingCommand, and retry the Teaching process in the next post-temperature measurement process.

4.3 Flowchart of App_Cmd function

Figure 4-6 shows the flowchart of the App_Cmd function.

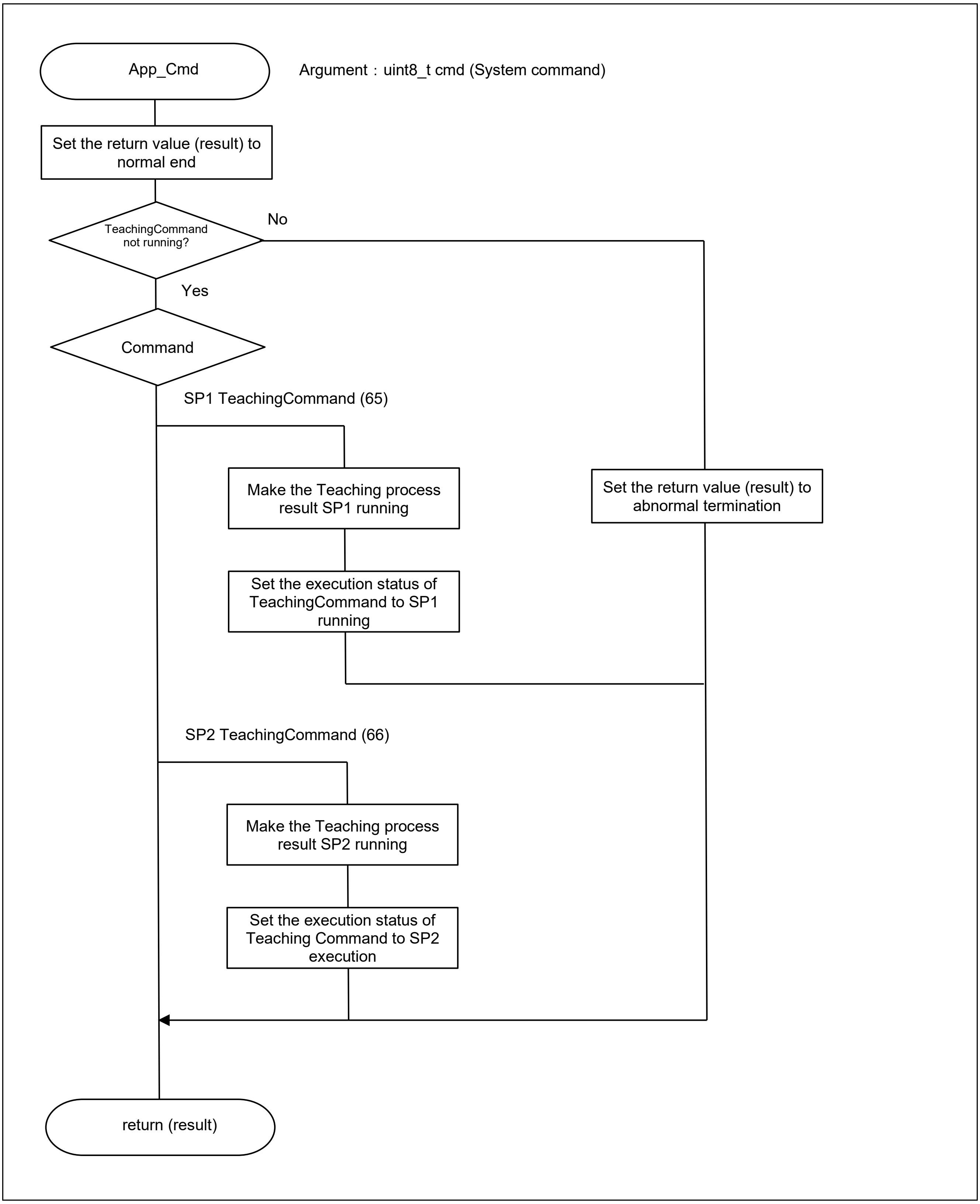


Figure 4-6 App_Cmd Flowchart

The outline of each process is described.

If TeachingCommand is being executed (other than IDLE), no processing is performed and an abnormal termination is returned.

If TeachingCommand is not being executed (IDLE), the following processing is performed.

In the case of TeachingCommand (65) of SP1, the result of Teaching processing is "SP1 is running", and the execution status of TeachingCommand is also "SP2 is running".

In the case of TeachingCommand (66) of SP2, the result of Teaching processing is "SP2 is running", and the execution status of TeachingCommand is also "SP2 is running".

Since the App_Temperature function monitors the execution status of TeachingCommand and executes Teaching processing as needed, the Teaching processing is executed by the App_Temperature function after receiving the system command on IO-Link and executing the App_Cmd function.

4.4 Operating mode and switching state

The operation mode and switching state of this sample application are described. This sample application operates in two modes, Windows mode and Two point mode. The LED is turned on / off according to the switching status so that the switching status can be confirmed by the LED in addition to the process data.

4.4.1 Window mode

This is the operation mode in which the switching state is ON if the temperature [℃] is within the specified range (between SP1 and SP2). The specific judgment is as follows. FIgure 4-7 shows an operation image.

Param1(SP1)≧measured temperature T [℃]≧Param2 (SP2): switching ON

Param1(SP1)<Measured temperature T [℃] or Param2 (SP2)> Measured temperature T [℃]: Switching OFF

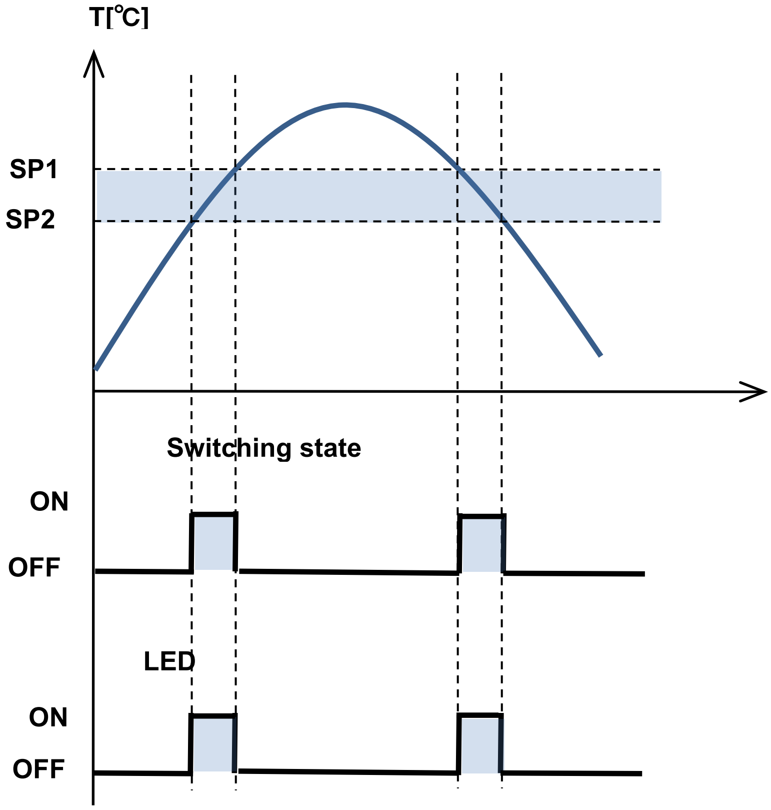


FIgure 4-7 Operation image diagram (Window mode)

4.4.2 Two point mode

This is an operation mode in which the switching state is turned on when the temperature [℃] exceeds the specified value (SP1), and the switching state is turned off when the temperature [℃] falls below the specified value (SP2). The specific judgment is as follows. Figure 4-8 shows an operation image.

Param1 (SP1) <Measured temperature T [℃]: Switching state ON

Param2 (SP2)> Measured temperature T [℃]: Switching state OFF

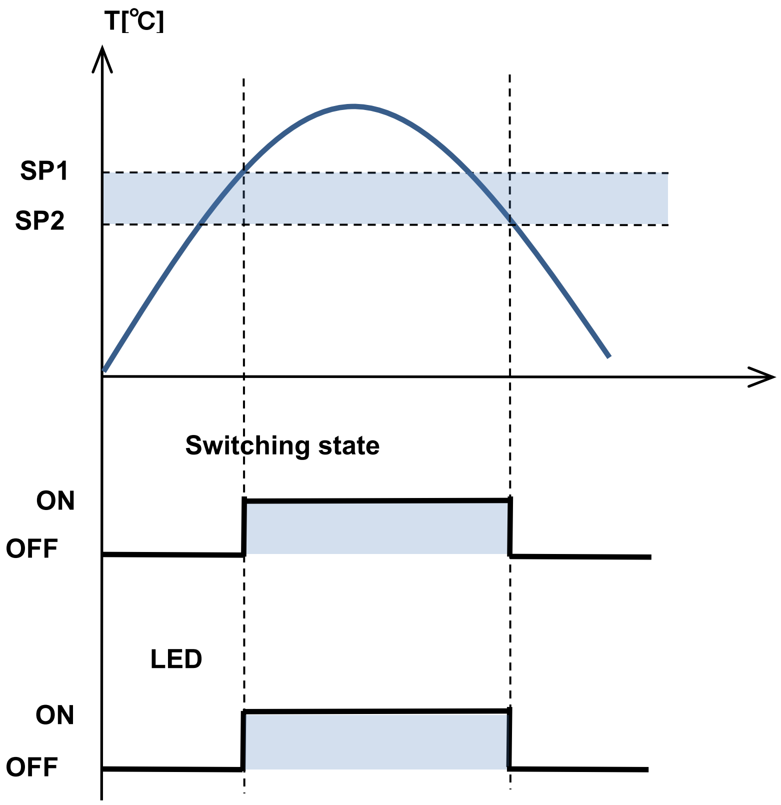


Figure 4-8 Operation image diagram (Two point mode)

4.5 IO-Link communication specifications

Describes the IO-Link communication specifications.

4.5.1 Bit rate

The bit rate is COM2 (38.4 [kbps]).

4.5.2 SIO mode

SIO mode is not supported.

4.5.3 Process data (Input)

The format of the process data (Input: data transmitted from the IO-Link device to the IO-Link master) is shown in Table 4-1.

Table 4-1 Process data (Input) format

Process data (Input) format Data length: 3 [bytes] (24 [bits])				
Subindex	Name	Bit offset	Bit number	Data
1	Switch Point 1	0	1	Switching state
2	Temperature	8	16	temperature (0.01 [℃] unit)

An example of process data (Input) is shown in Table 4-2.

Table 4-2 An example of process data (Input)

PDIn[16]：16 octet								
Subindex	2							
Value	0	0	0	0	1	0	0	1
Example	Upper 8 bits of temperature data (0x09)							
PDIn[17]：17 octet								
Subindex	2							
Value	0	0	1	1	0	1	0	0
Example	Lower 8 bits of temperature data (0x43) Upper + lower = 0x0934 (2356) → 23.56 [℃]							
PDIn[18]：18 octet								
Subindex	1							
Value	0	0	0	0	0	0	0	1
Example	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Switching state

4.5.4 Parameters

The Parameter list that send and receive with master using IO-Link is shown in Table 4-3 and Table 4-4.

Table 4-3 Parameter list (1/2)

Index (Subindex)	Name (Type)	Bit offset (Bit number)	Range (initial value)	R/W	Unit	Overview
Switching Signal Channel1						
60 (1)	Param SP1 (Integer)	16 (16)	-4000 to 15000 (2500) *4	RW	0.01℃ *4	Switching Point1 (SP1) setting*1 SP1≧SP2 must be satisfied.
60 (2)	Param SP2 (Integer)	0 (16)	-4000 to 15000 (2000) *4	RW	0.01℃ *4	Switching Point1 (SP2) setting*1 SP1≧SP2 must be satisfied.
61 (1)	Config Logic (UInteger)	24 (8)	0, 1 (0)	RW	-	Switching state logic setting 0: High active Transmit 1 if ON 1: Low active Transmit 0 if ON
61 (2)	Config Mode (UInteger)	16 (8)	0, 2, 3 (3)	RW	-	Switching state judgment mode setting 0: Deactivated Disabled, switching state is always OFF 2: Window mode*2 3: Two point mode*3

* 1. For SP1 and SP2, refer to 4.4 Operating mode and switching state.

* 2. For Window mode, refer to 4.4.1 Window mode.

* 3. For Two point mode, refer to 4.4.2 Two point mode.

* 4. In Device Tool, set in the range of -40.00 to 150.00. The initial value is displayed as SP1 = 25.00, SP2 = 20.00, refer to 5.7.5 Parameter tab.

Table 4-4 Parameter list (2/2)

Index (Subindex)	Name (Type)	bit offset (Bit number)	Range (initial value)	R/W	Unit	Overview
Teach-In Single Value						
2 (-)	Standard Command (UInteger)	- (-)	65 (-)	WO	-	Set the current measurement temperature to SP1.
2 (-)	Standard Command (UInteger)	- (-)	66 (-)	WO	-	Set the current measurement temperature to SP2.
59 (1)	Teach-In Result：State (UInteger)	0 (4)	0, 1, 2, 3, 4, 5, 7 (0)	RO	-	Result of previous Teach-In command 0: Idle 1: SP1 Success 2: SP2 Success 3: SP12 Success 4: Wait 5: Busy 7: Error
Device Locks						
12 (3)	Device Access Locks Local Parametarization Lock (Boolean)	2 (1)	0, 1 (0)	RW	-	Enable / Disable setting of sensor parameterization function 0: Invalid 1: Valid

4.6 Peripheral Functions and Pins Used

The Peripheral Functions Used in the examples are listed in Table 4-5, and the Pins Used are listed in Table 4-6.

Table 4-5 Peripheral Functions Used

Peripheral function	Use
AFE,DSAD0,DSAD1	Thermocouple, RTD drive, thermocouple A/D conversion (DSAD0), RTD A/D conversion (DSAD1)
IRQ1	ZIOL2401 DC/DC Ready status detection
IRQ4	IO-Link Wake Up detection
SCI6	SPI communication (register setting) and UART communication (IO-Link) with ZIOL2401
P17	ZIOL2401 reset control (reset release by L output)
PB1	ZIOL2401 SPI communication enable / disable control (valid by L output)
PC5	ZIOL2401 UART communication enable / disable control (valid by H output)
PH2	LED1 control

Table 4-6 Pins Used

Pin name	Input/Output	Use
AIN11	Input	Thermocouple + side input terminal
AIN10	Input	Thermocouple - side input terminal
AIN9	Output	RTD excitation current output terminal
AIN7	Input	RTD + side input terminal
AIN6	Input	RTD - side input terminal
AIN5/REF1P	Input	RTD measurement DSAD + side reference voltage
AIN4/REF1N	Input	RTD measurement DSAD - side reference voltage
P17	Output	ZIOL2401 reset control terminal (reset release by L output)
P31/IRQ1	Input	ZIOL2401 DC/DC Ready signal input terminal
PB0/IRQ4	Input	IO-Link WakeUp detection signal input terminal
PB1	Output	ZIOL2401 SPI communication enable / disable control terminal (valid by L output)
PC5/SCK6	Output	ZIOL2401 UART communication enable / disable control terminal (valid by H output) CLK output terminal during SPI communication
PC6/RXD6/SMISO6	Input	UART6 receiving terminal Data receiving terminal during SPI communication
PC7/TXD6/SMOSI6	Output	UART6 transmission terminal Data transmission terminal during SPI communication
PH2	Output	LED1 control

4.7 Program Configuration

4.7.1 File Configuration

The IO-Link stack made by TMG and related files are located in the Library folder. The IOLink stack manuals are located in the Manuals folder inside the Library folder, so refer to them as needed. Table 4-7 shows the file structure in the src2 folder and the StackExtentionsApp folder.

Table 4-7 File Configuration

Folder name, file name	Description
src2	
│ main.c	Main processing
│ r_led_api.c	LED control processing program
│ r_led_api.h	LED control processing API definition
│ r_rtd_api.c	Resistance temperature detector measurement calculation program, temperature vs. resistance value table
│ r_rtd_api.h	Resistance temperature detector measurement calculation API definition
│ r_sensor_common_api.c	Table search, linear interpolation processing program
│ r_sensor_common_api.h	Table search, linear interpolation processing API definition
│ r_temperature_measurement_api.c	Temperature measurement processing program
│ r_temperature_measurement_api.h	Temperature measurement processing API definition
│ r_thermocouple_api.c	Thermocouple measurement calculation program, temperature vs. thermoelectromotive force table
│ r_thermocouple_api.h	Thermocouple measurement calculation API definition
│	
└smc_gen	Smart Configurator generation
├Config_CMT0	
├Config_DMAC0	
├Config_DMAC3	
├Config_DSAD0	
├Config_DSAD1	
├Config_IOL_MTU0	
├Config_PORT	
├Config_SCI1	
├Config_SCI6	
├Config_SCI61	
├general	
├r_bsp	
├r_config	
├r_flash_rx	
└r_pincfg	
IO-Link	
└StackExtentionsApp	
├BSPExtensions.h	BSP extension header
├IOLinkMain.c	IO-Link application program
├IOLinkMain.h	IO-Link application definition
├MemoryManager.h	Memory manager header
├ParameterSet.h	Definition of stack extension
└ProductionSettings.h	Production setting header

4.7.2 Functions

The functions added and changed from the base "Temperature Measurement Example Using a Thermocouple" application note (R01AN4747EJ0110) are described.

4.7.2.1 main.c

[Function name] main

Overview	main function
Header	None
Declaration	void main (void)
Description	Start the operation of DSAD0 and DSAD1 and initialize the IO-Link stack and IO-Link related parameters. After that, the IO-Link application process (iolink_main function) is called periodically.
Argument	None
Return value	None
Remarks	None

4.7.2.2 r_temperature_measurementc

[Function name] R_TEMP_Start

Overview	Start temperature measurement
Header	r_temperature_measurement.h
Declaration	void R_TEMP_Start (void)
Description	Allows DSAD0 and DSAD1 unit-to-unit synchronous start, sets the trigger to a software trigger, and then starts DSAD0 and DSAD1 operation.
Argument	None
Return value	None
Remarks	None

[Function name] R_TEMP_Stop

Overview	Stop temperature measurement
Header	r_temperature_measurement.h
Declaration	void R_TEMP_Stop (void)
Description	Stops the operation of DSAD0 and DSAD1.
Argument	None
Return value	None
Remarks	None

[Function name] R_TEMP_Get

Overview	Get the temperature
Header	r_temperature_measurement.h
Declaration	bool R_TEMP_Get (float *p_temperature)
Description	If the A / D conversion between DSAD0 and DSAD1 is completed (if the temperature measurement is completed), the measured temperature is stored in the address passed as an argument.
Argument	*p_temperature : Temperature [°C] storage area pointer
Return value	Measurement status false: During measurement true: Measurement completed
Remarks	None

4.7.2.3 r_led_api.c

[Function name] R_LED_Control

Overview	Control LED1
Header	r_led_api.h
Declaration	void R_LED_Control (bool state)
Description	The LED turns on / off according to the switching status.
Argument	Switching state false: OFF true: ON
Return value	None
Remarks	None

4.7.2.4 IOLinkMain.c

[Function name] iolink_main

Overview	IO-Link application main processing
Header	IOLinkMain.h
Declaration	uint8_t iolink_main (void)
Description	Executes periodic processing of the IO-Link stack (calls the STACK_Run function) to acquire the temperature. If the temperature measurement is completed, process data will be transmitted via IO-Link.
Argument	None
Return value	IO-Link stack status STACK_STATUS_SIO : IO-Link connection in SIO mode STACK_STATUS_STARTUP : Master detected, device in startup state STACK_STATUS_PREOPERATE : Device is in pre-operation state STACK_STATUS_OPERATE : Device is in operating state STACK_STATUS_DISCONNECTED : Disconnected, device waits for next wakeup in IO-Link mode
Remarks	None

[Function name] App_Cmd

Overview	IO-Link system command processing
Header	IOLinkMain.h
Declaration	uint16_t App_Cmd (uint8_t command)
Description	Performs system command processing.
Argument	System command
Return value	System command processing result RESULT_OK: Successful completion RESULT_ERR_FUNCTION_TEMP_NA: Invalid command
Remarks	None

[Function name] App_Temperature

Overview	IO-Link temperature sensor application processing
Header	None
Declaration	static bool App_Temperature (float rawValue)
Description	Performs the following four processes. 1. Create process data 2. Teaching process (threshold setting process) 3. Threshold judgment processing + process data update 4. Start transmitting process data
Argument	Measured temperature [℃]
Return value	Threshold judgment result false: Out of valid range (OFF) true: Within the effective range (ON)
Remarks	This function is an internal function called from the iolink_main function. The return value is false while the Teaching process is being executed.

5. Operation check procedure

The procedure for checking the operation is described on the assumption that IO-Link Device Tool V5.1 – PE made by TMG is installed on the Windows PC. See Figure 3-1 for the connection of each device.

5.1 IP address setting of PC

The IP address of the RZ / N1S-IOLINK-M board is 192.168.0.225, so set the PC to such as 192.168.0.12.

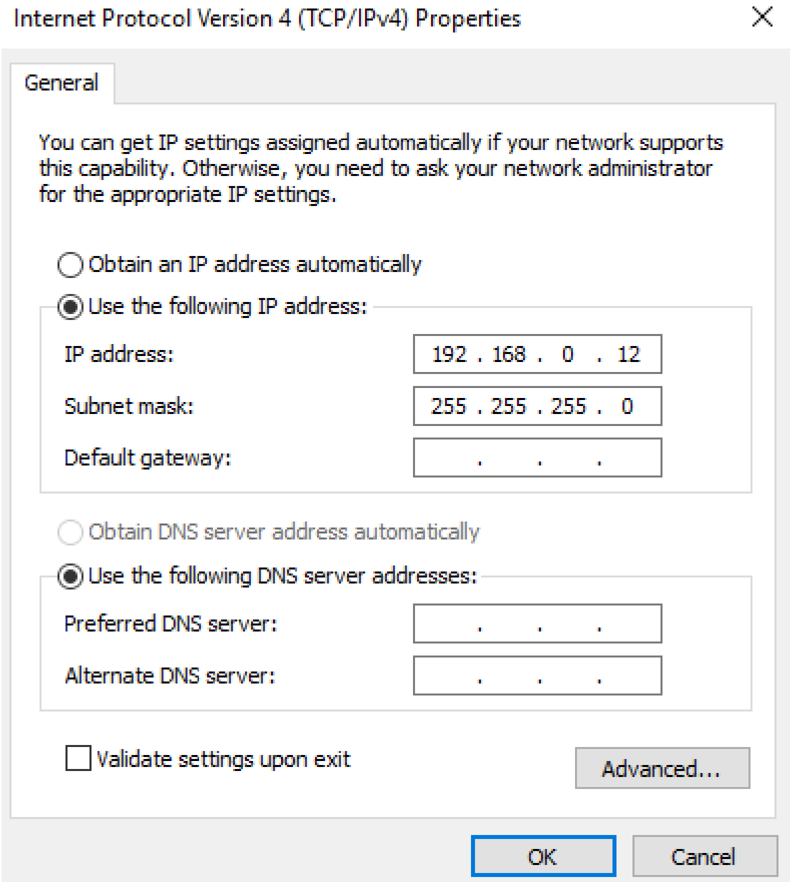


Figure 5-1 IP address setting of PC

5.2 Launch IO-Link Device Tool V5.1 - PE

Topology displays the topology of the IO-Link device from the PC.

Device Catalog shows all installed devices.

Figuire 5-2 IO-Link Device Tool V5.1 – PE

5.3 Update IO-Link Device Catalog

1. Select "Options" from the menu bar, and then select "Import IODD (IO Device Description)".
2. Enter the path or select the folder that contains the IODD file. The IODD file of this sample application is located in the "IODD" folder in the included sample project file. IODD files are auto-detected, as shown in Figure 5-3.

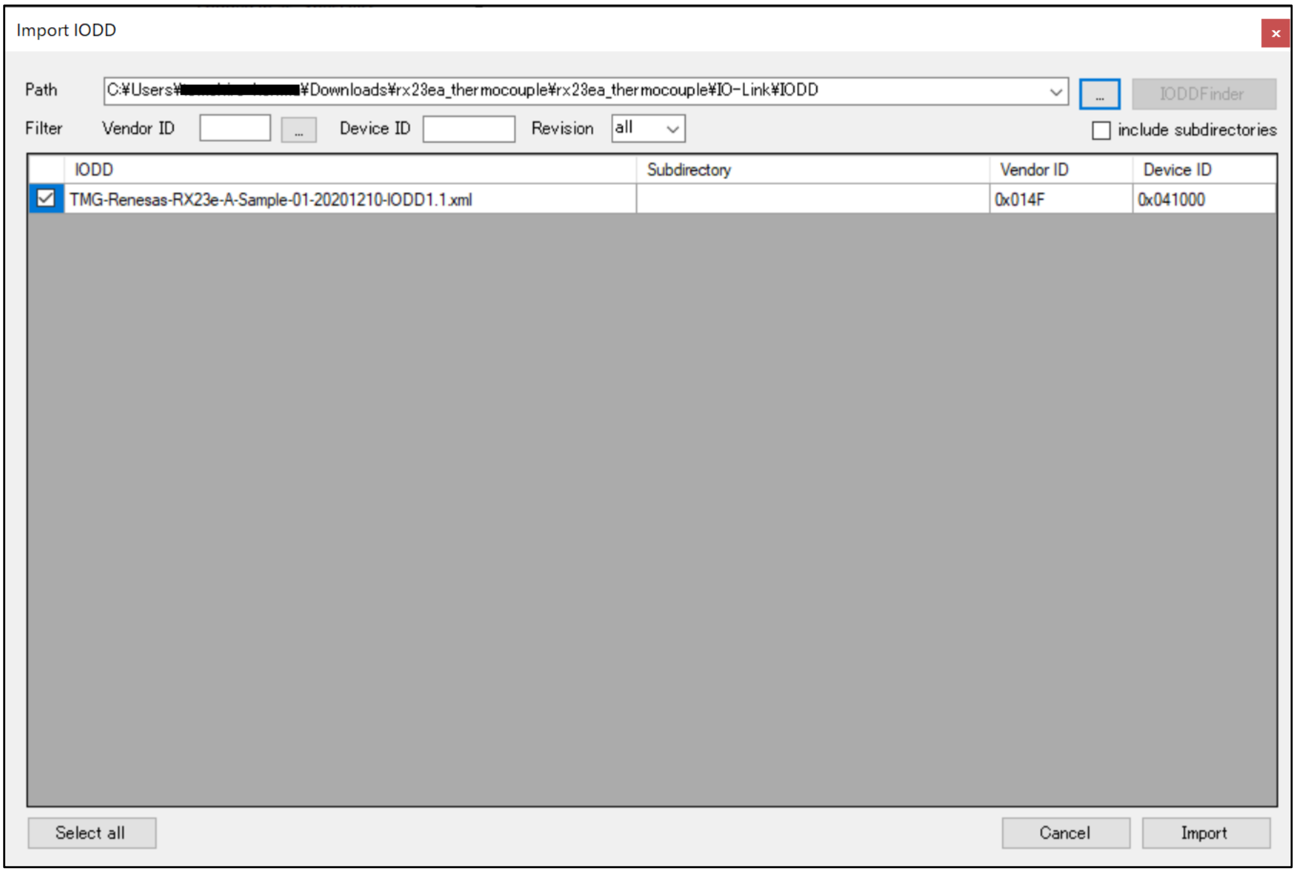


Figure 5-3 IO-Link Device Tool V5.1 – PE (Loading IODD file)

3. Select "Import".

5.4 Update IO-Link Master Catalog

1. Select "Options" on the menu bar and then select "Import IOLM (IO-Link Master Description)".
2. Enter the path or select the folder that contains the IOLM zip file. The IOLM file of this sample application is located in the "IOLM" folder in the included sample project file. The IOLM file is auto-detected, as shown in Figure 5-4.

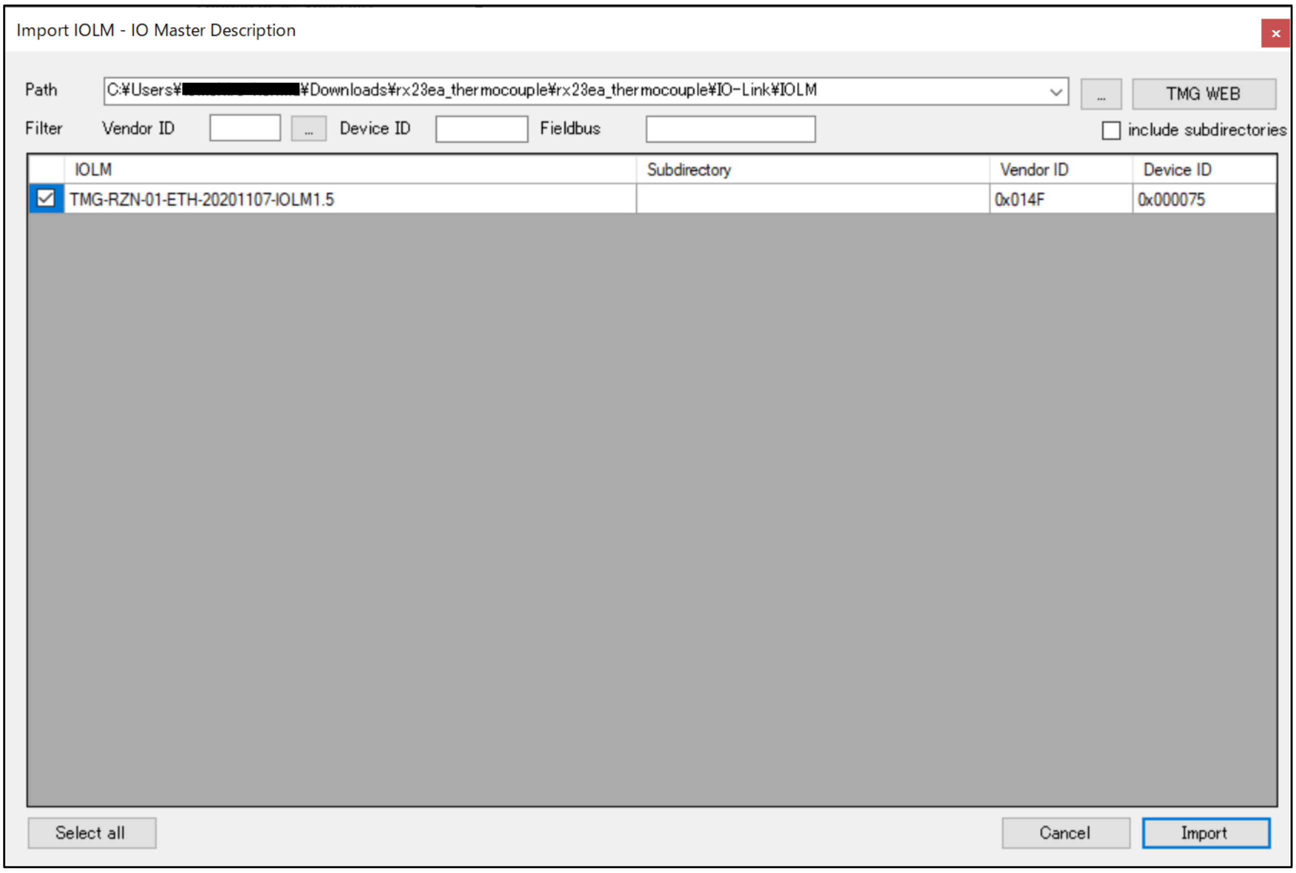


Figure 5-4 IO-Link Device Tool V5.1 – PE (Loading IOLM file)

> Note:
>
> For Path, specify the folder containing the unzipped IOLM zip file. If you specify the unzipped folder, the IOLM file will not be searched.

3. Select "Import".

5.5 Check for Catalog updates

When the Catalog of the update is successful, the TMG TE GmbH vendor and RSSK RX23E-A board will display as the "RX23e-A Starter kit" in the IO-Link Devices section of the Catalog. RZ / N1S IO-Link Master is displayed under the Master as a PROFINET device.

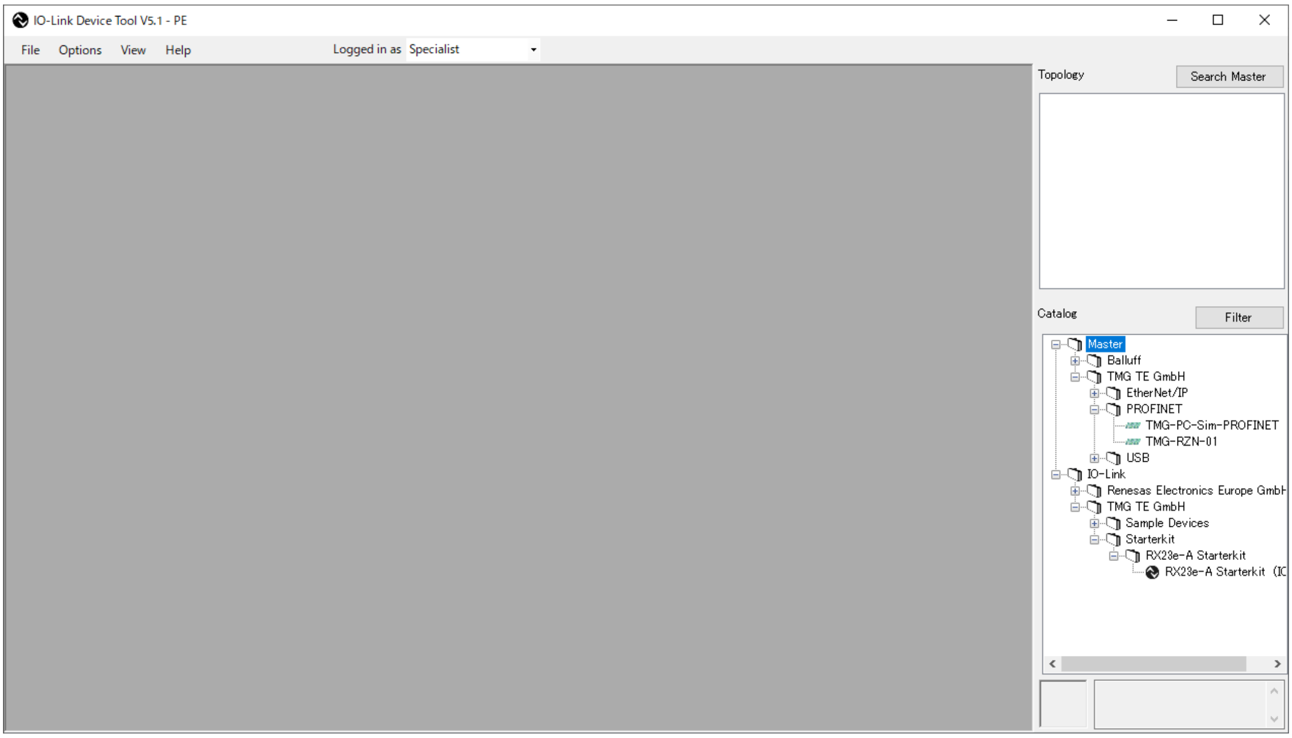


Figure 5-5 IO-Link Device Tool V5.1 – PE (Check for Catalog updates)

5.6 Setup of IO-Link communication

1. Click the "Search Master" button at the top right of the window. As shown in Figure 5-6, RZ / N1SIOlink will be displayed in the "Master Discovery" window.

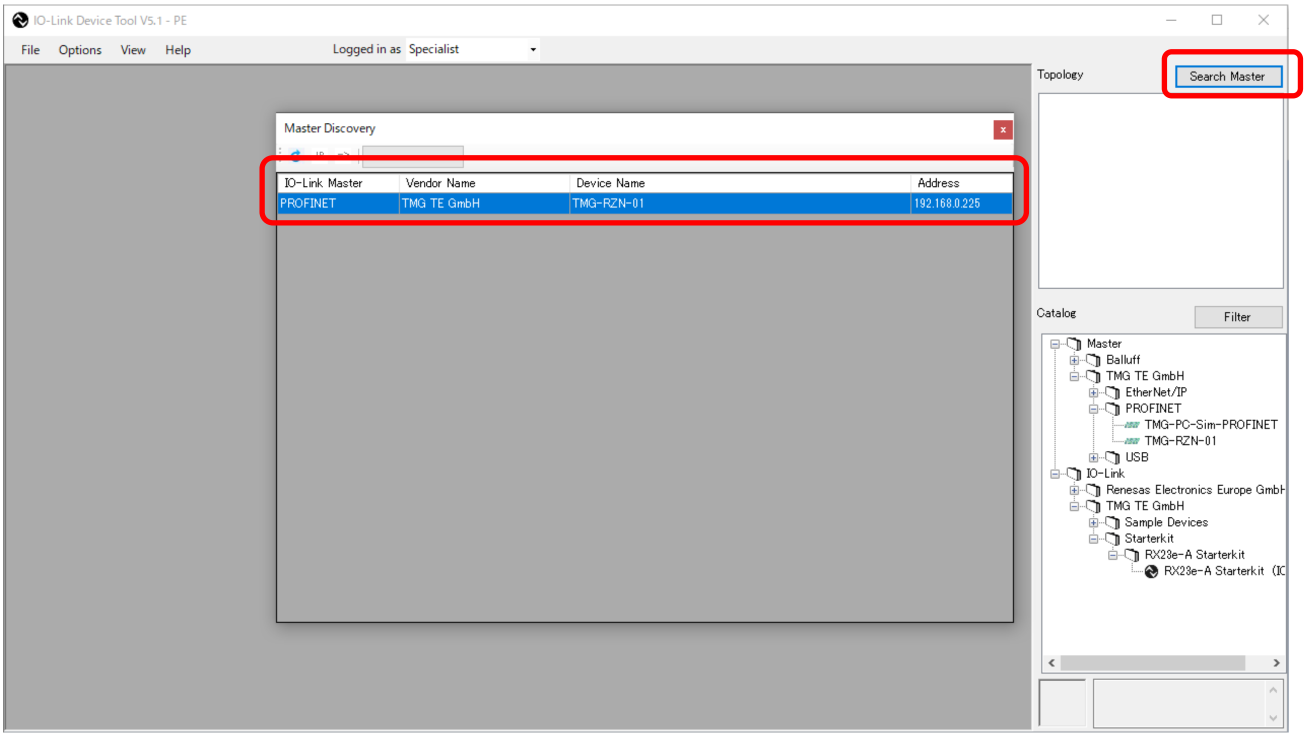


Figure 5-6 IO-Link Device Tool V5.1 – PE (Search for IO-Link master)

2. Double-click the device name displayed in the “Master Discovery” window.

3. Click the "Go Online" button to activate the connection between the master and the device.

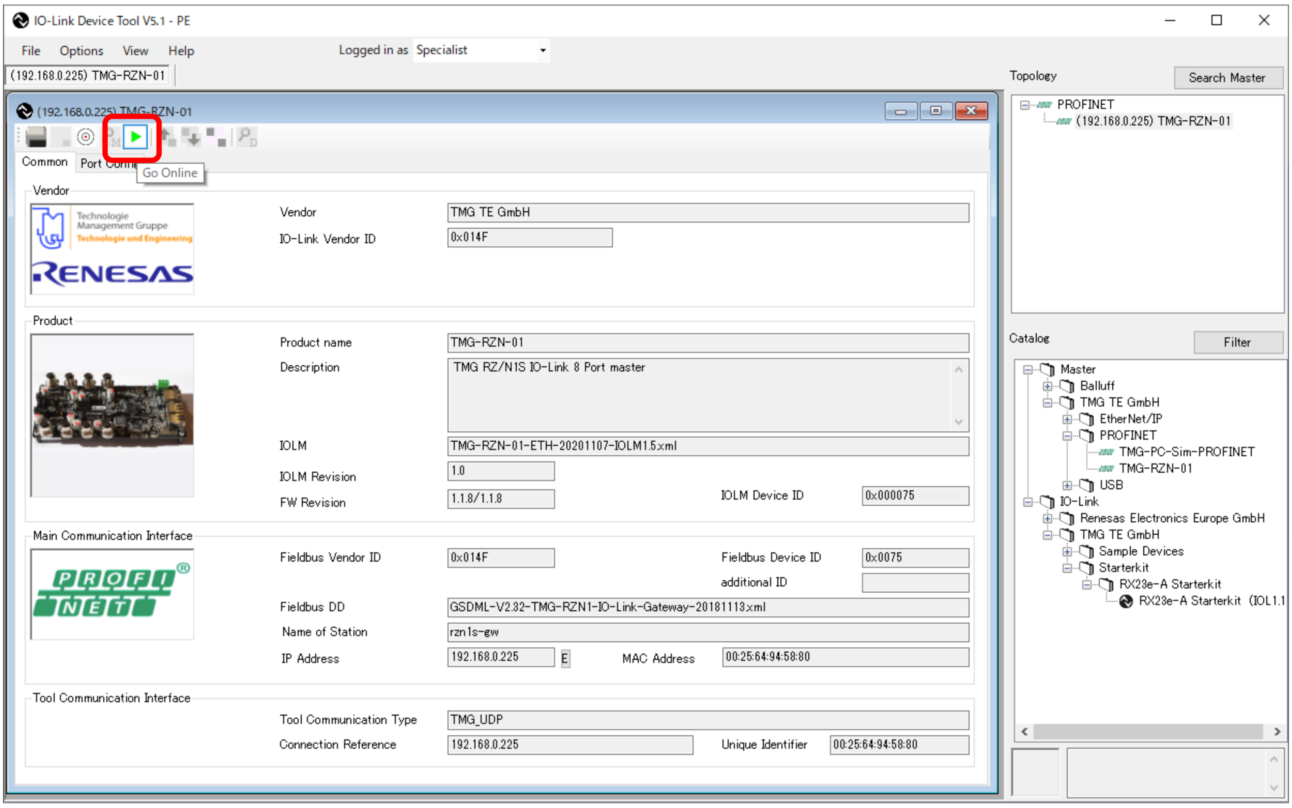


Figure 5-7 IO-Link Device Tool V5.1 – PE (Set to Online state)

When the connection between the master and the device is activated, the "Go Online" button is replaced with a red circle button and the "Check Devices" button is enabled.

4. Click the "Check Devices" button to detect connected devices.

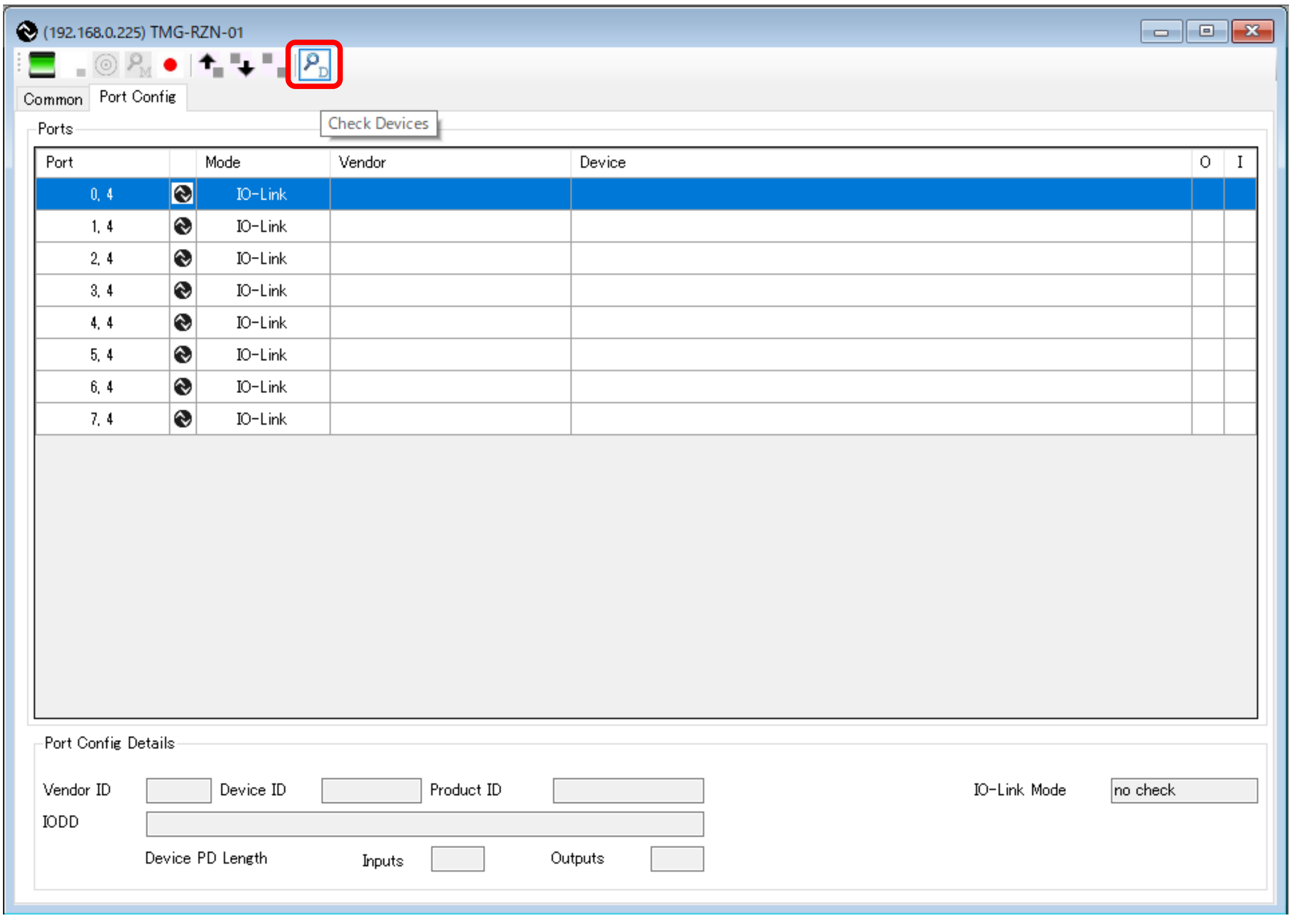


Figure 5-8 IO-Link Device Tool V5.1 – PE (Checking IO-Link device)

"Check Devices" window is displayed, IO-Link RX23e-A Starterkit, which is connected to a port of the Master will be displayed.

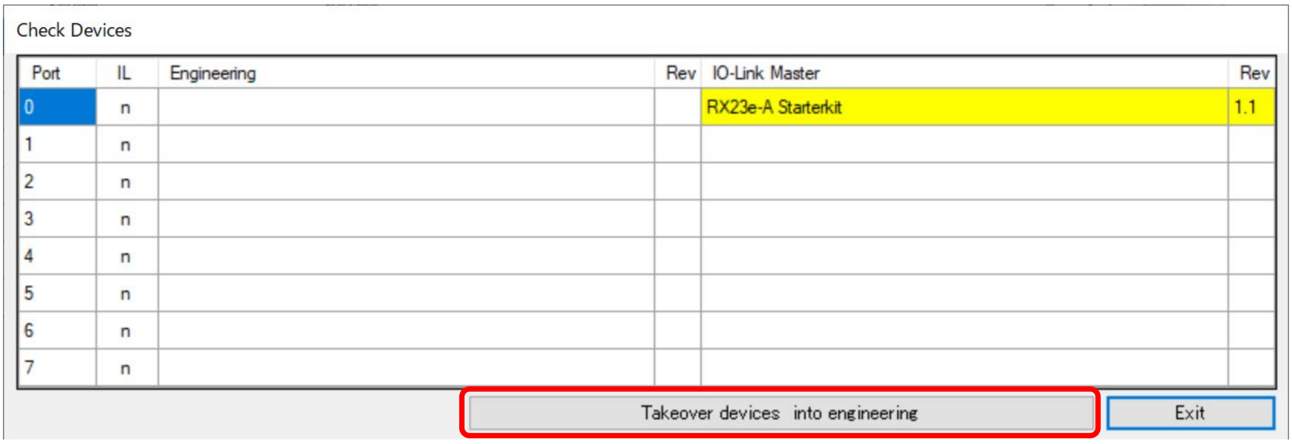


Figure 5-9 IO-Link Device Tool V5.1 – PE (IO-Link device detection)

5. Click the "Takeover devices to engineering" button. When the connection between the master and the device is successful, the display of the devices that are connected to the port of the IO-Link Master is updated.

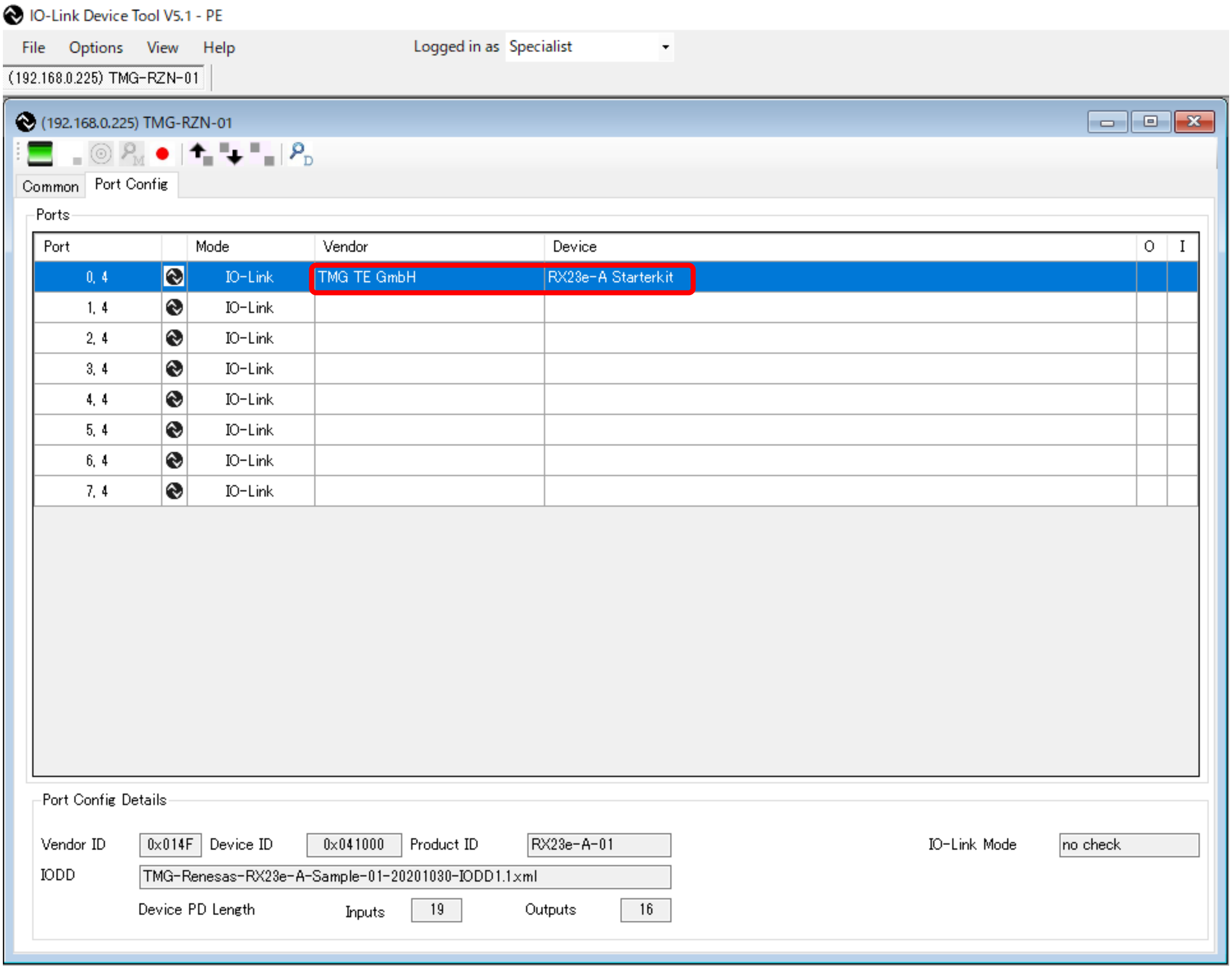


Figure 5-10 IO-Link Device Tool V5.1 – PE (Successful connection between IO-Link master and device)

6. Double-click "RX23e-A Starterkit" to access the sensor description page.

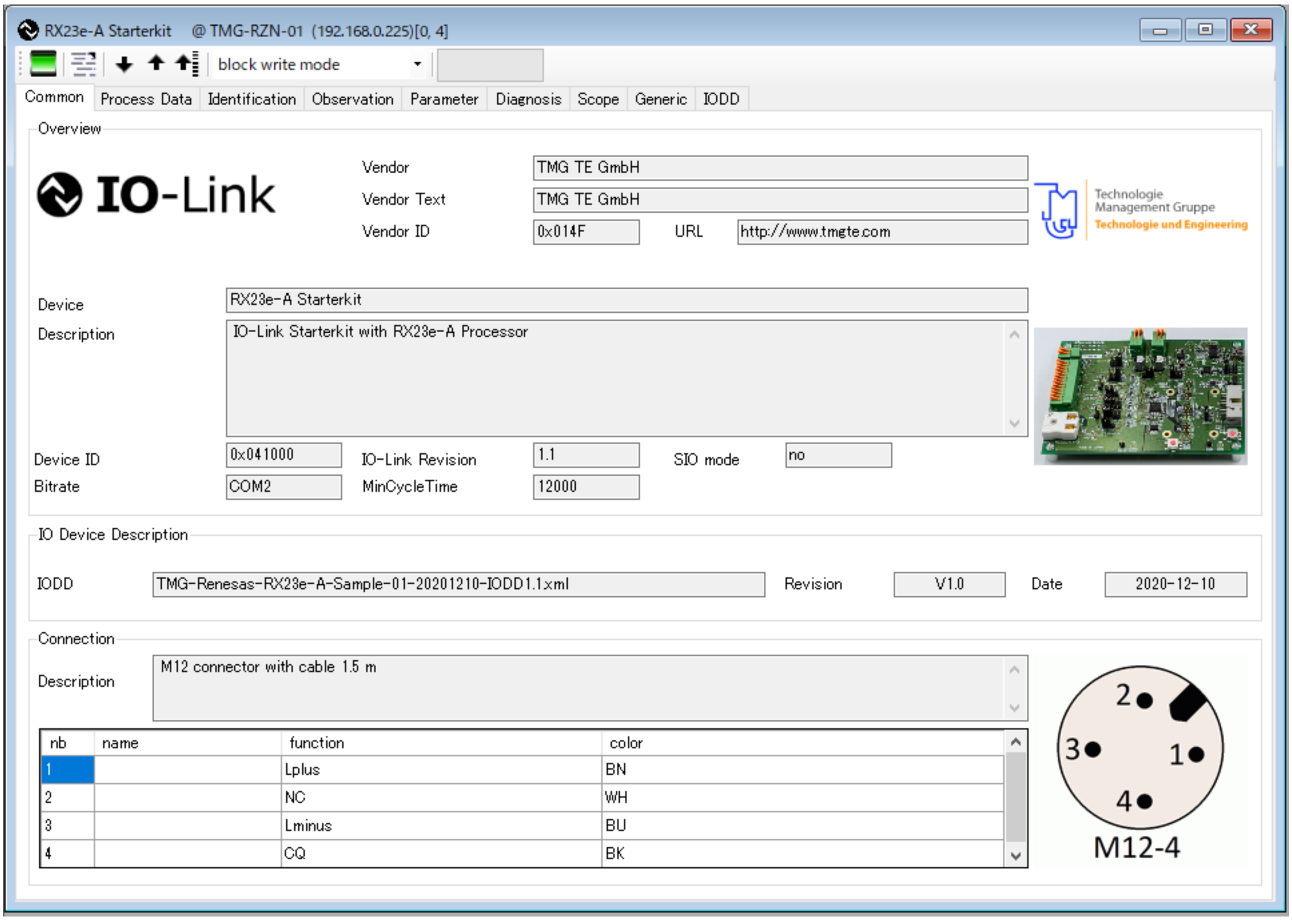
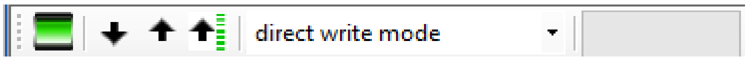


Figure 5-11 IO-Link Device Tool V5.1 – PE (RX23e-A Starterkit Common tab)

5.7 IOSK RX23E-A board sensor demo for IO-Link device tools

Learn how to operate an IO-Link device (temperature sensor) using the RSSKRX23E-A board from the IO-Link Device Tools GUI. The button in Figure 5-12 is located in the upper left corner of the IODD window.

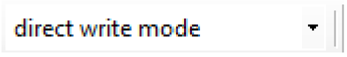


 : System operating status

 : Download variable changes to your device

 : Upload variable information from device

 : Enable dynamic variable updates



: direct write mode -> Automatically download modified variables to device

block write mode -> Manually download the modified variables to device

Figure 5-12 IO-Link Device Tool V5.1 – PE (Device operation toolbar)

5.7.1 Common tab

As shown in Figure 5-11, the Common tab displays the general description information about the device as follows.

Vendor name: TMG TE GmbH

Vendor text: TMG TE GmbH

Vendor ID: 0x014F

URL: www.tmgte.com

Device name: RX23e-A Starterkit

Device Description: IO-Link Starterkit with RX23e-A Processor

Device ID: 0x00041000

IO-Link revision: 1.1

SIO mode support: No

Communication baud rate: COM2

Minimum sensor cycle time: 5000 [us]

IO-Link device photo

Connection description

Pin array of M12 connector

5.7.2 Process Data tab

The Process Data tab displays Process data inputs and Process data outputs.

1. Process data inputs
   - Sensor measure value
     Measured temperature [0.01℃]
   - Switch Point 1
     Switching state (true or false)
2. Process data outputs

   Not supported.

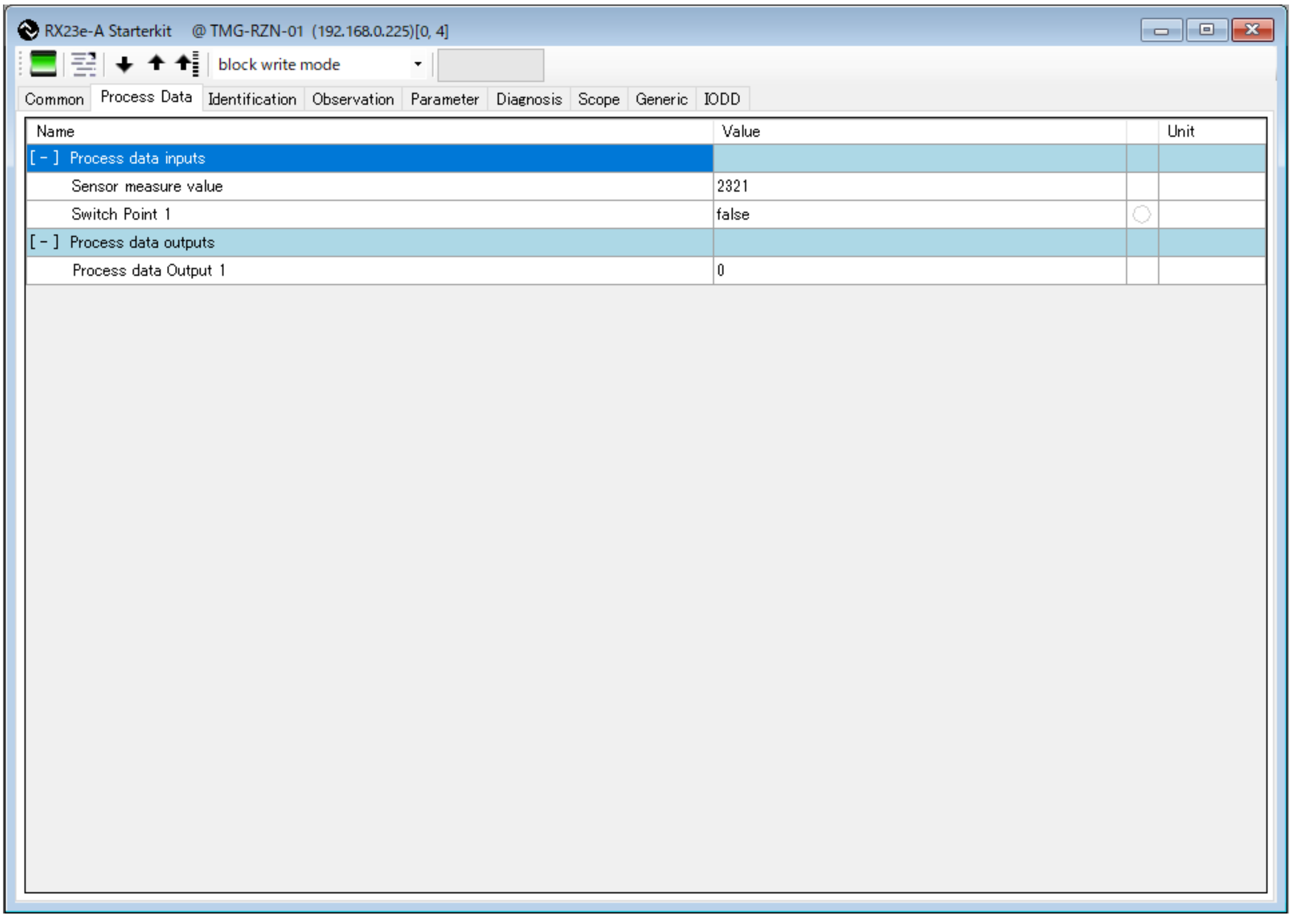


Figure 5-13 IO-Link Device Tool V5.1 – PE (Process Data tab)

5.7.3 Identification tab

The Identification tab, the user can to read the identification information such as the following that are stored on the device.

1. Device information
2. Application-specific information
3. Revision information

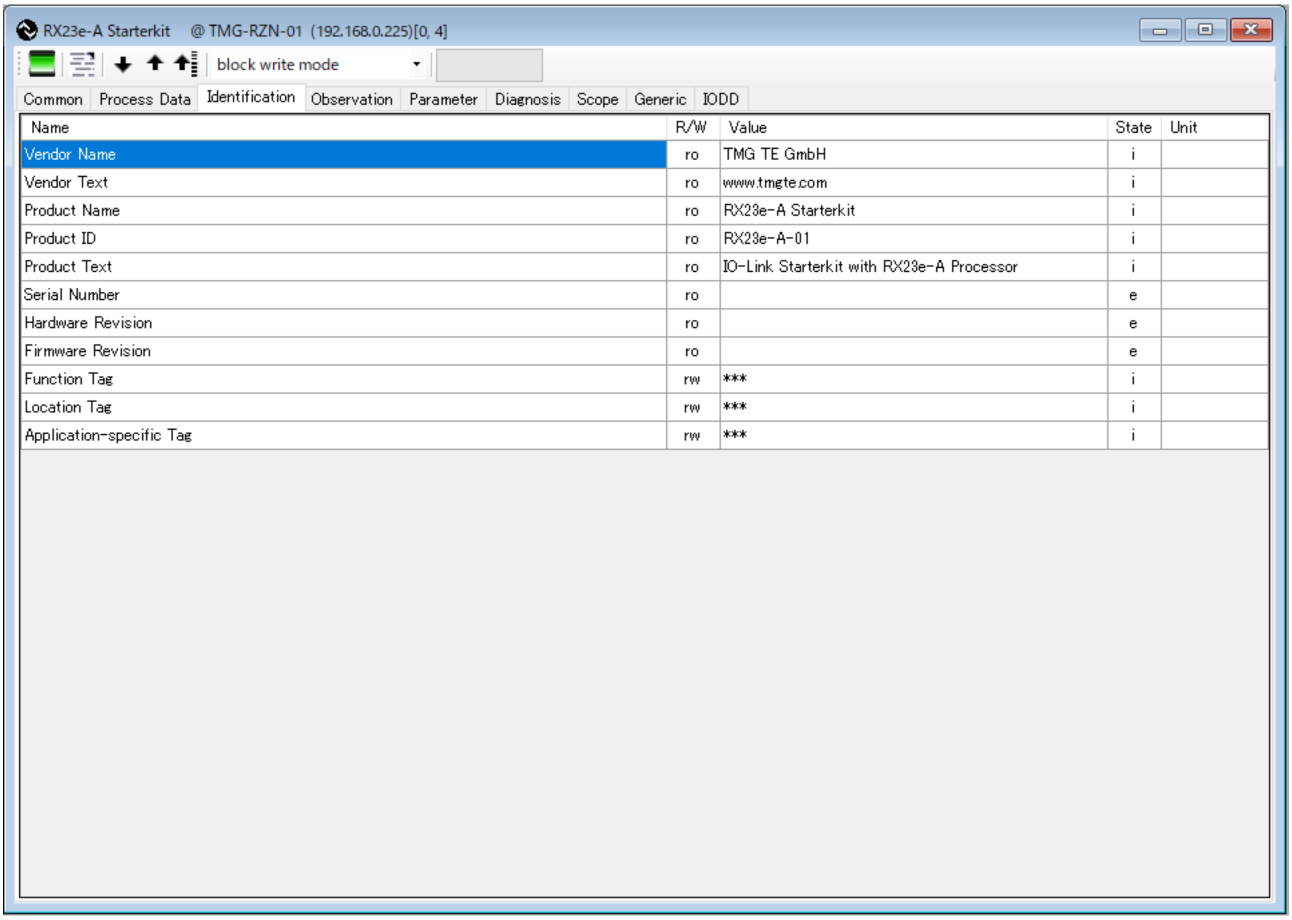


Figure 5-14 IO-Link Device Tool V5.1 – PE (Identification tab)

5.7.4 Observation tab

The Observation tab displays the measurements taken by the device. The user can see the temperature information measured by the device.

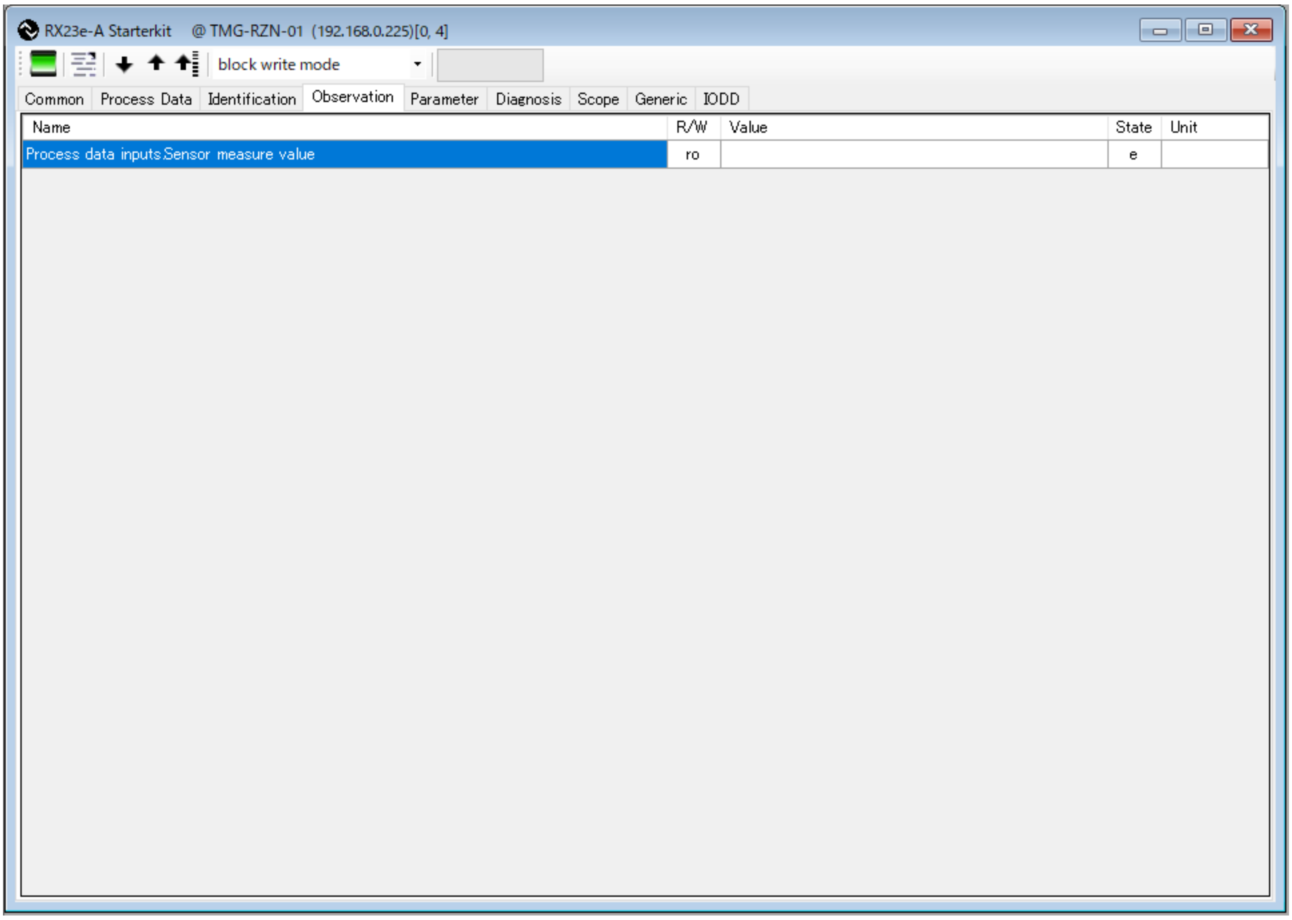


Figure 5-15 IO-Link Device Tool V5.1 – PE (Observation tab)

5.7.5 Parameter tab

The Parameter tab displays the device's parameter settings and allows the user to read the settings from the device. You can also write new settings to the device. See Table 4-3 and Table 4-4 for the meanings and functions of the parameters. For the device parameter setting method, refer to 5.7.7 Device Parameter Change (Teach-In / Read) ".

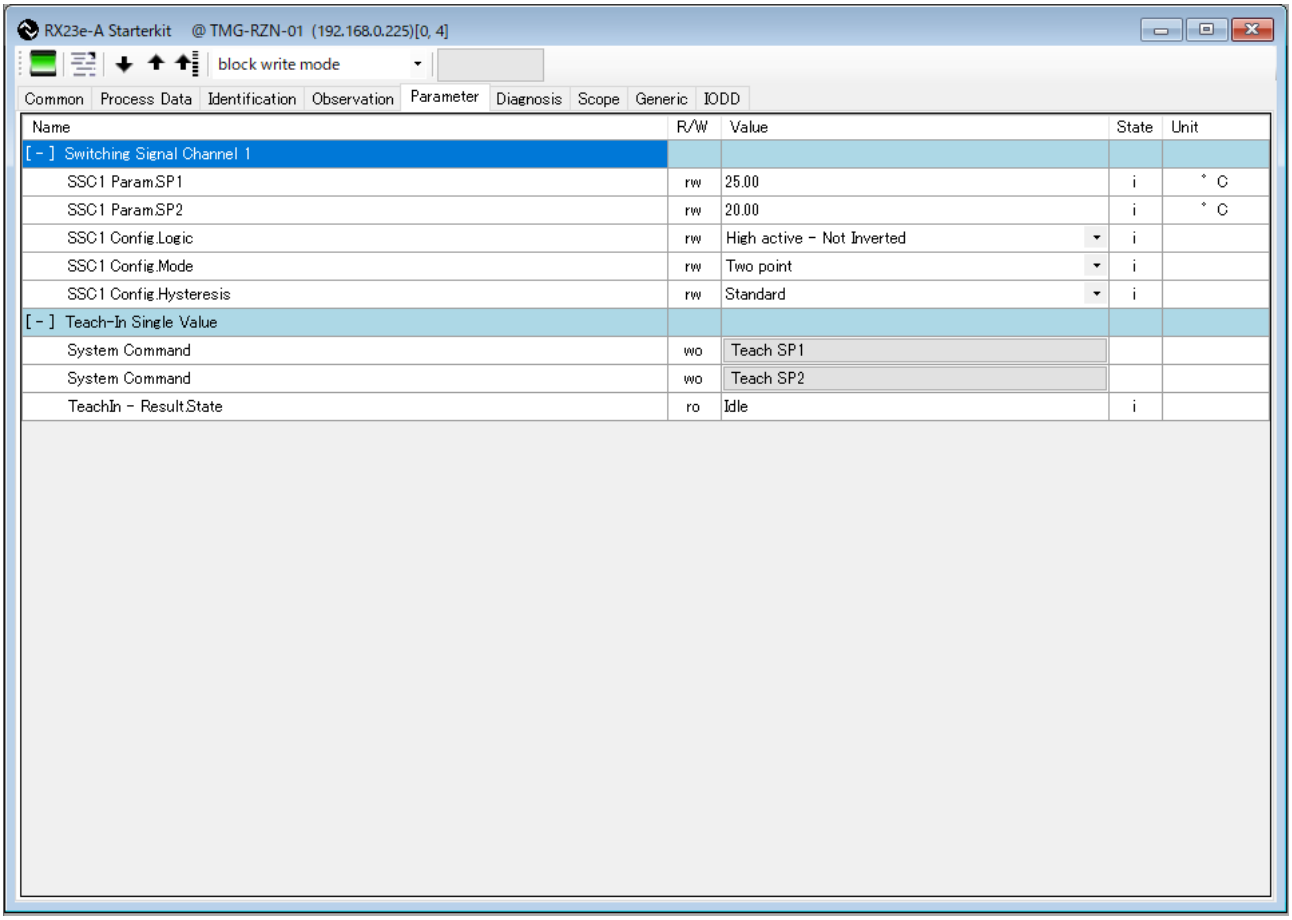


Figure 5-16 IO-Link Device Tool V5.1 – PE (Parameter tab)

5.7.6 Scope tab

The Scope tab allows you to visualize process data.

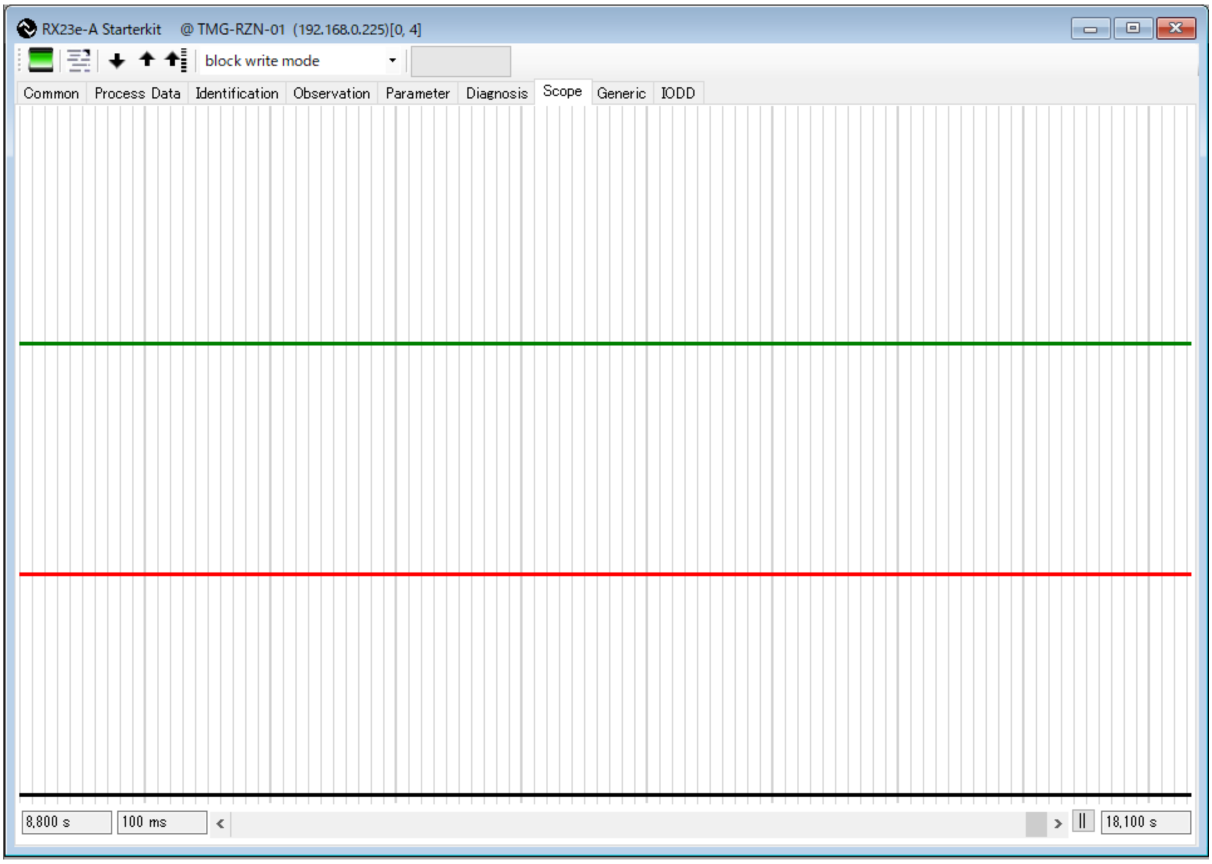


Figure 5-17 IO-Link Device Tool V5.1 – PE (Scope tab)

The user can set the Scope configuration by right-clicking in the Scope area. The following window will appear where you can edit the settings.

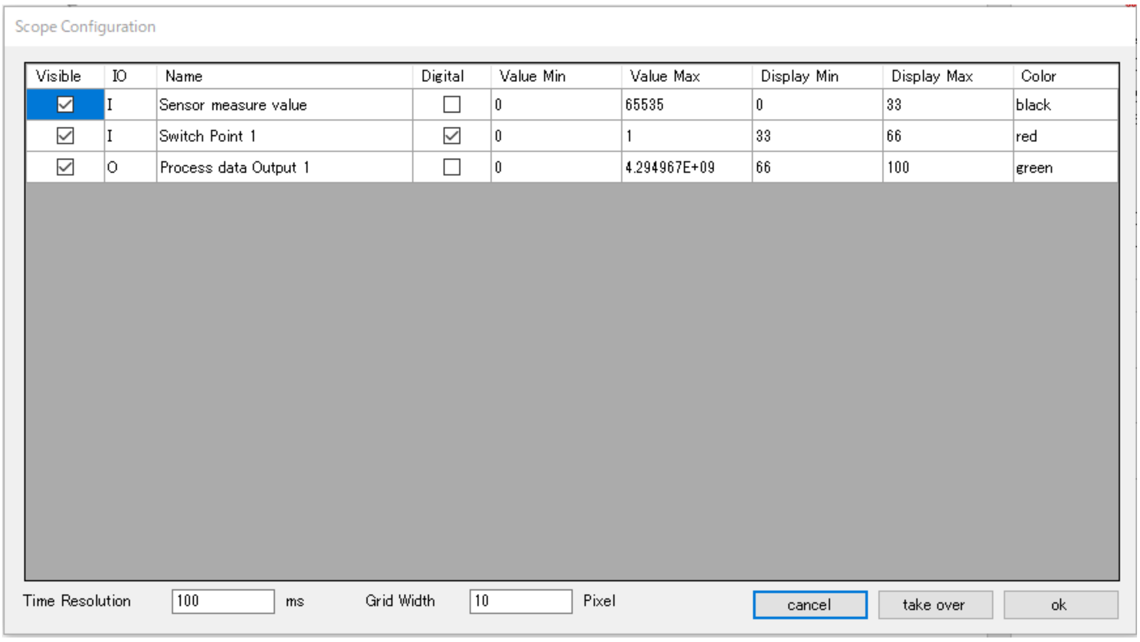


Figure 5-18 IO-Link Device Tool V5.1 – PE (Scope tab, Set Scale/Parameter)

5.7.7 Device Parameter Change (Teach-In / Read)

When the user opens the Parameter tab, the device-specific parameters are set to the default values in the Value column. These values are recorded in the IODD file. See Figure 5-16 for more information.

When setting SetPoint (SP1, SP2, etc.) on a device, make sure that "Device Access Locks" is set to false. If true, parameter setting on the device will fail. The SetPoint parameter can be set using the "Teach Values" parameter or the "Standard Command".

The IO-Link device tool allows the user to select the write mode and set the parameters.

5.7.7.1 Parameter setting in Teach Values (block write mode)

In this mode, you set the parameters on the device by changing the parameters and then clicking the "Write all changed values to device" button.

1. Change the write mode to "block write mode".
2. To set the SetPoint, click the Value field of the parameter “SSC1 Param SP1” or “SSC1 Param SP2”.
3. Enter a number and press “Enter”. The yellow shading of "Status" indicates that the parameter has not yet been set on the device.
4. Every time you change a parameter, you need to click the "Write all changed values to device" button, as shown in Figure 5-19.
5. The green shading of “Status” indicates that the parameter has been set on the device and synchronization has occurred between the master and the device (refer to Figure 5-20).

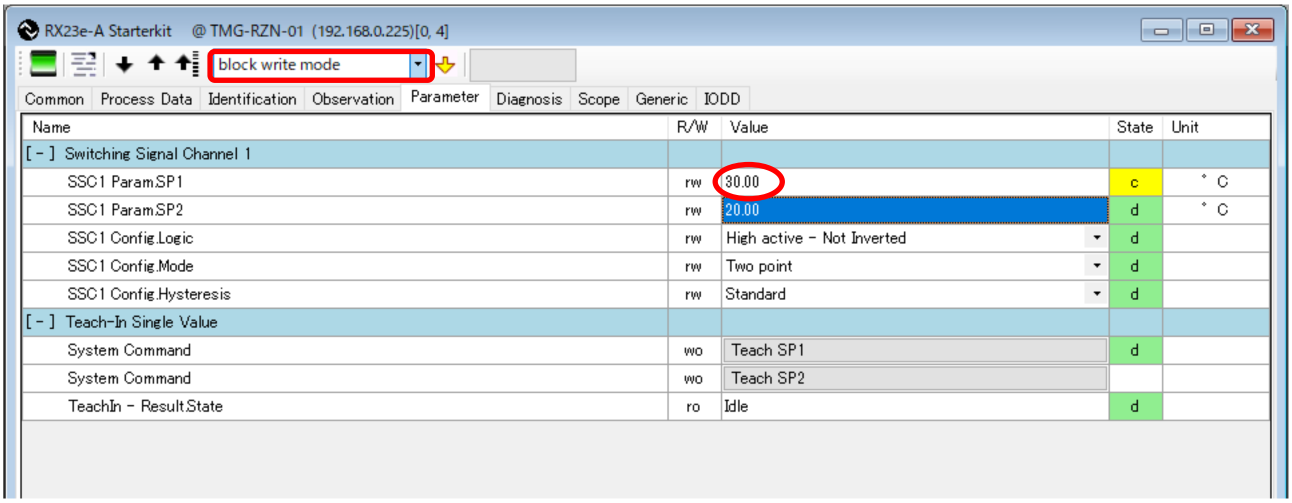


Figure 5-19 IO-Link Device Tool V5.1 – PE (Parameter setting in block write mode)

5.7.7.2 Parameter setting in Teach Values (direct write mode)

In this mode, parameter changes are automatically set on the device.

1. Change the write mode to "direct write mode".
2. To set the SetPoint, click the Value field of the parameter “SSC1 Param SP1” or “SSC1 Param SP2”.
3. Enter a number and press “Enter”.
4. The “Status” turns yellow and then green, indicating that the parameters have been set on the device and synchronization has taken place between the master and the device.

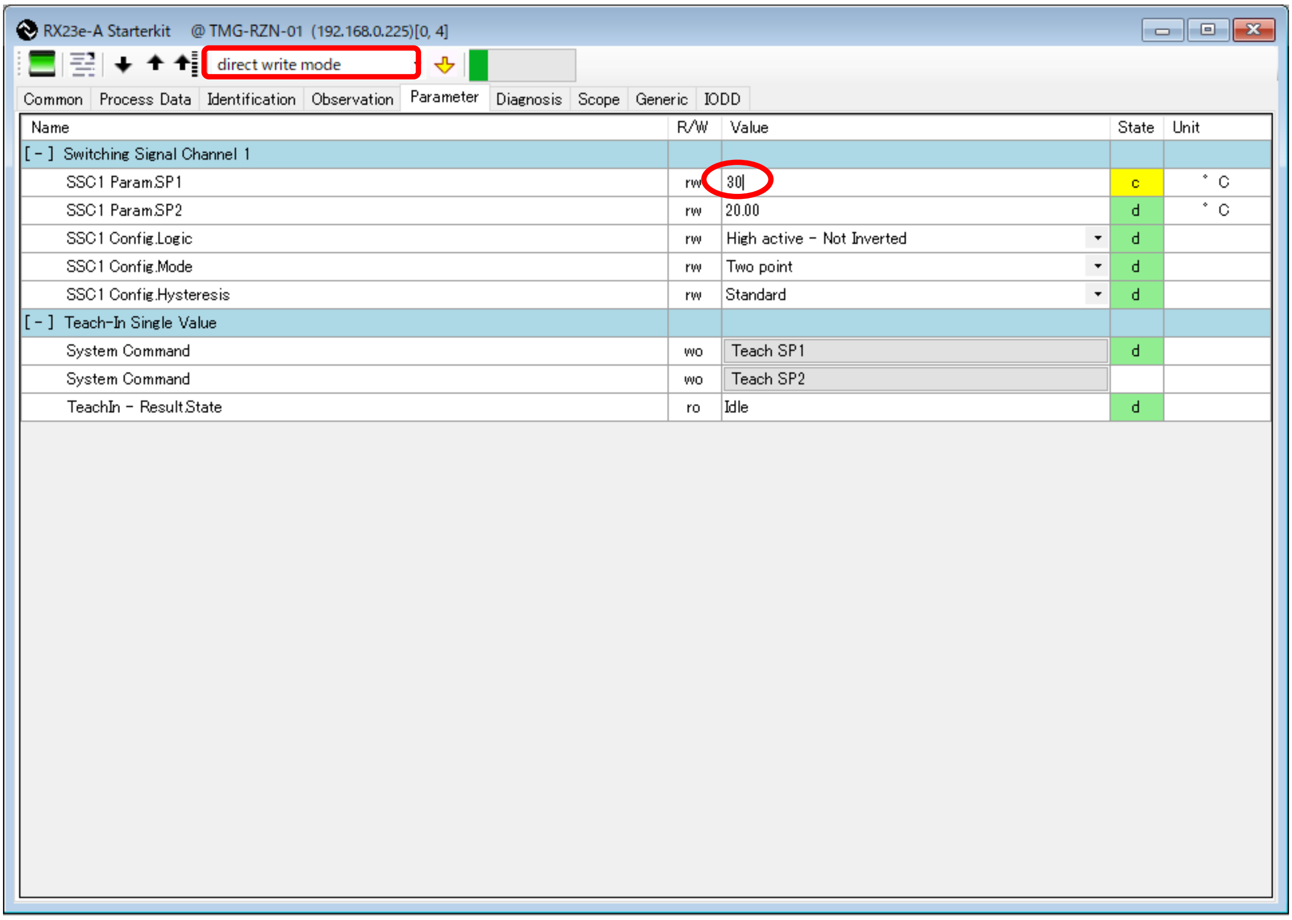


Figure 5-20 IO-Link Device Tool V5.1 – PE (Parameter setting in direct write mode)

5.7.7.3 Parameter settings in the Standard Command

Use the Standard Command to set the SetPoint.

1. Set the temperature of the device to be measured to the temperature you want to set as SetPoint.
2. Click the "Teach SP1" or "Teach SP2" button in the Value field of the Parameter tab.

   The measured temperature is automatically set to "SSC1 Param SP1" or "SSC1 Param SP2", and the Teach-In result is also "SP1 Success" or "SP2 Success".

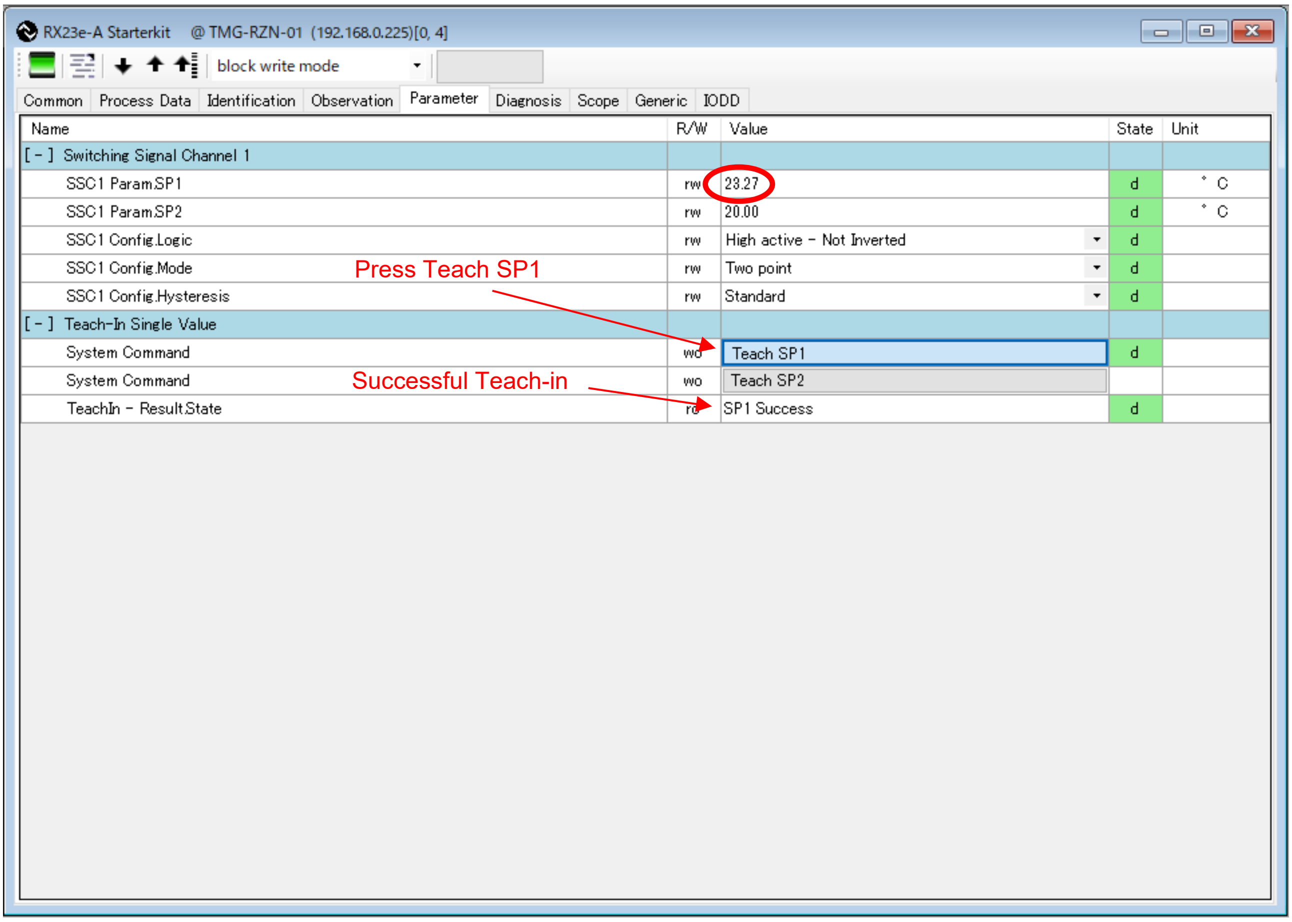


Figure 5-21 IO-Link Device Tool V5.1 – PE (Parameter settings in the Standard Command)

5.7.7.4 Parameter reading

The user can read the current parameters written to the device by clicking the "Upload the variable information from device" button. Please refer to Figure 5-12 for button "Upload the variable information from the device."

5.7.8 Restore to factory settings

Restores the device settings to factory settings.

1. Go to the “Diagnosis” tab.
2. Click the “Restore Factory Setting” button.

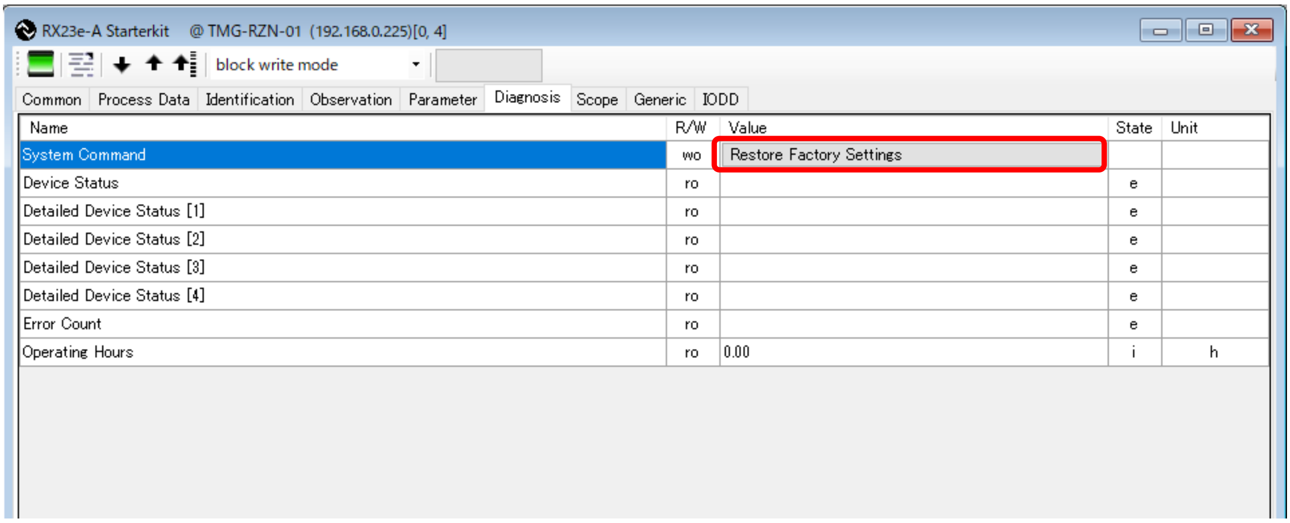


Figure 5-22 IO-Link Device Tool V5.1 – PE (Restore to factory settings)

5.7.9 Generic tab

Use this tab to work with IO-Link devices without IODD files. The data is displayed as RAW data and the address is set via the index and subindex. The message box on the IO-Link port displays the results of manually processed read / write requests.

5.7.10 IODD tab

IODD information is displayed on this tab.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jan. 15, 2021	-	First edition

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.