RA2E1 Group

Examples of IO-Link Solutions

Introduction

This application note describes a sample application to realize IO-Link communication with RA2E1 using the Evaluation Kit for RA2E1 Microcontroller Group (EK-RA2E1), IA Sensor Network Connector Board, ZSSC3240 Evaluation Board, and EK-RA2E1 Change Board. IO-Link is a communication technology for sensors and actuators that complies with IEC61131-9. For IO-Link communication, the IO-Link stack manufactured by TMG is used.

Target Device

- RA2E1, ZIOL2401 (IO-Link PHY), ZSSC3240

When applying this application note to other microcontrollers, modify any applicable configurations and settings according to the specifications of the microcontroller and evaluate any differences thoroughly.

Contents

1. Overview ... 3

2. Confirmed operation environment ... 4

3. Hardware Configuration ... 5
3.1 IO-Link Master ... 6
3.2 IA Sensor Network Connector Board (IO-Link Communication Board) ... 6
3.3 EK-RA2E1 Board ... 7
3.4 ZSSC3240 Evaluation Board ... 8
3.5 EK-RA2E1 Change Board ... 8
3.6 Power supply ... 9
3.6.1 Connect the 5V Service Power Supply ... 10

4. Sample Application Overview ... 11
4.1 Overview of the Overall Processing Flow ... 12
4.2 Flowchart of App_SensorValue Function ... 14
4.3 Operating Mode and Switching State ... 19
4.4 IO-Link communication ... 20
4.4.1 Bit Rate ... 20
4.4.2 SIO Mode ... 20
4.4.3 Process Data (PDIn) ... 20
4.4.4 Parameters ... 21
4.5 Peripheral features and Terminals to be Used ... 22
4.6 Program Structure ... 24
4.6.1 File Structure ... 24
4.6.2 Function List ... 25

5. Sample Project Execution Method 28
5.1 Importing Sample Projects 28
5.2 Build the Project 30
5.3 Setting up a Debug Connection between the EK-RA2E1 Board and the Host PC 32
5.4 Writing Sample Projects to the MCU 32
5.5 Start the Program 34

6. IO-Link Device Tool Usage and Functions 35
6.1 IO-Link Device Tool V5.1 - Start PE 35
6.2 Update IO-Link Device Catalog 36
6.3 Update IO-Link Master Catalog 36
6.4 Check for Catalog Updates 39
6.5 Setup for IO-Link communication 39
6.6 EK-RA2E1 Board Sensor Demo of IO-Link Device Tool 41
6.6.1 Common Tab 42
6.6.2 Process Data tab 44
6.6.3 Identification Tab 45
6.6.4 Observation Tab 46
6.6.5 Parameter Tab 47
6.6.6 Scope tab 48
6.6.7 Change Device Parameters (Teach-In/Read) 49
6.6.8 IODD Tab 54

7. Examples of Use 55
7.1 Reset IO-Link parameters to default values 55
7.2 Operating Methods 55

8. How to Change ZSSC Settings 59
8.1 USB-SERIAL Conversion Cable Connection 60
8.2 Check the Virtual COM Port Number 61
8.3 Start the ZSSC GUI 62
8.4 Establish a Communication Path 63
8.4.1 Open Port 63
8.4.2 Interface 64
8.4.3 Scan I2C-Bus 65
8.5 Measurement (Corr. Measurement) 66
8.6 NVM Operation 71
8.6.1 Read NVM 71
8.6.2 Write NVM 73
8.6.3 Write CRC 74
8.6.4 Return ADC resolution set value 75

9. References 75

Revision History 76

1. Overview

This application note describes how to realize IO-Link communication using IO-Link stack manufactured by TMG.

In this example, EK-RA2E1 board and IA Sensor Network Connector Board are used as IO-Link devices, and IO-Link-Master02-USB manufactured by Pepperl + Fuchs is used as IO-Link master. For communication with the IO-Link master, "IO-Link Device Tool V5.1 - PE" provided by TMG is used. "IO-Link Device Tool V5.1 - PE" is an application software that runs on a Windows PC.

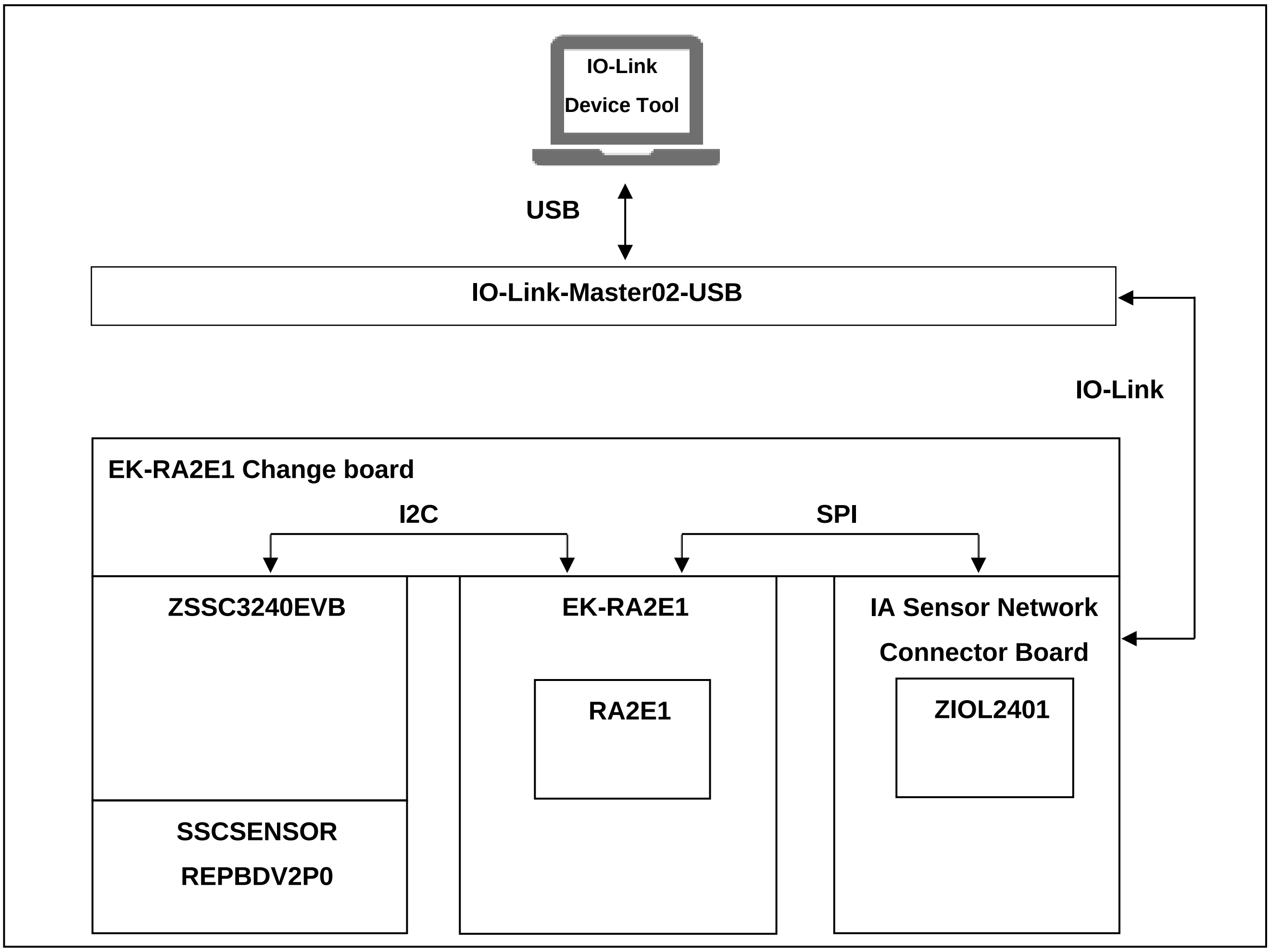


Figure 1. System Configuration in this Application Note

2. Confirmed operation environment

The confirmed operation environment is shown in Table 1. The configuration of the device is described in section 3 Hardware .

Table 1. Development Environment

Item	Description
CPU board	EK-RA2E1 board
MCU	R7FA2E1A92DFM
IDE	Renesas e^2 studio 2022-04 (22.4.0)
Tool Chain	GCC ARM Embedded 10.3.1.20210824
FSP	3.7.0
Library	IO-Link stack manufactured by TMG
Emulator	SEGGER J-Link OB
IO-Link communication board	IA Sensor Network Connector Board (RTK0EF0085B00001BJ) It is connected to the EK-RA2E1 board and operates as an IO-Link device.
IO-Link Line Driver	ZIOL2401
IO-Link Master	Pepperl+Fuchs IO-Link-Master02-USB Device Type Manager[1]
IO-Link Tool	IO-Link Device Tool V5.1 – PE
Host PC for IO-Link Tool	Windows10 Professional
ZSSC3240EVK	ZSSC3240 Evaluation Kit
ZSSC3240EVB	SSC evaluation board with ZSSC3240
SSCCOMMBOARDV4P1C	CB (SSC communication board) CB H/W Version (SSC CB V4.1) CB F/W Version (V4.19 © ZMD AG 2020 – CB || SVN: 11903) * This board is included in the ZSSC3240EVK, but is not used in this application. The EK-RA2E1 board is the pipe for communication between the PC and the ZSSC3240EVB.
SSCSENSORREPBDV2P0	SRB (Sensor Replacement Board)
ZSSC GUI	ZSSC3240 Evaluation Software (manual COMport selection, recommended for older PC) v3.02
EK Change Board EX-EK-RA2E1	It connects EK-RA2E1, ZSSC 3240EVB, and IA Sensor Network Connector Board.
USB-SERIAL conversion cable	FT232RL built-in USB-SERIAL conversion cable

Note[1]: If IO-Link-Master02-USB is not detected in the IO-Link Device Tool master search (section 6.5), reinstall the software “Device Type Manager”.

3. Hardware Configuration

This application uses the IO-Link master and IO-Link device. IO-Link master is composed of Pepperl+Fuchs IO-Link-Master02-USB, and IO-Link device is composed of ZSSC3240 Evaluation Board, EK-RA2E1 Board, IA Sensor Network Connector Board and EK-RA2E1 Change Board to connect them. The connection of each board is shown below.

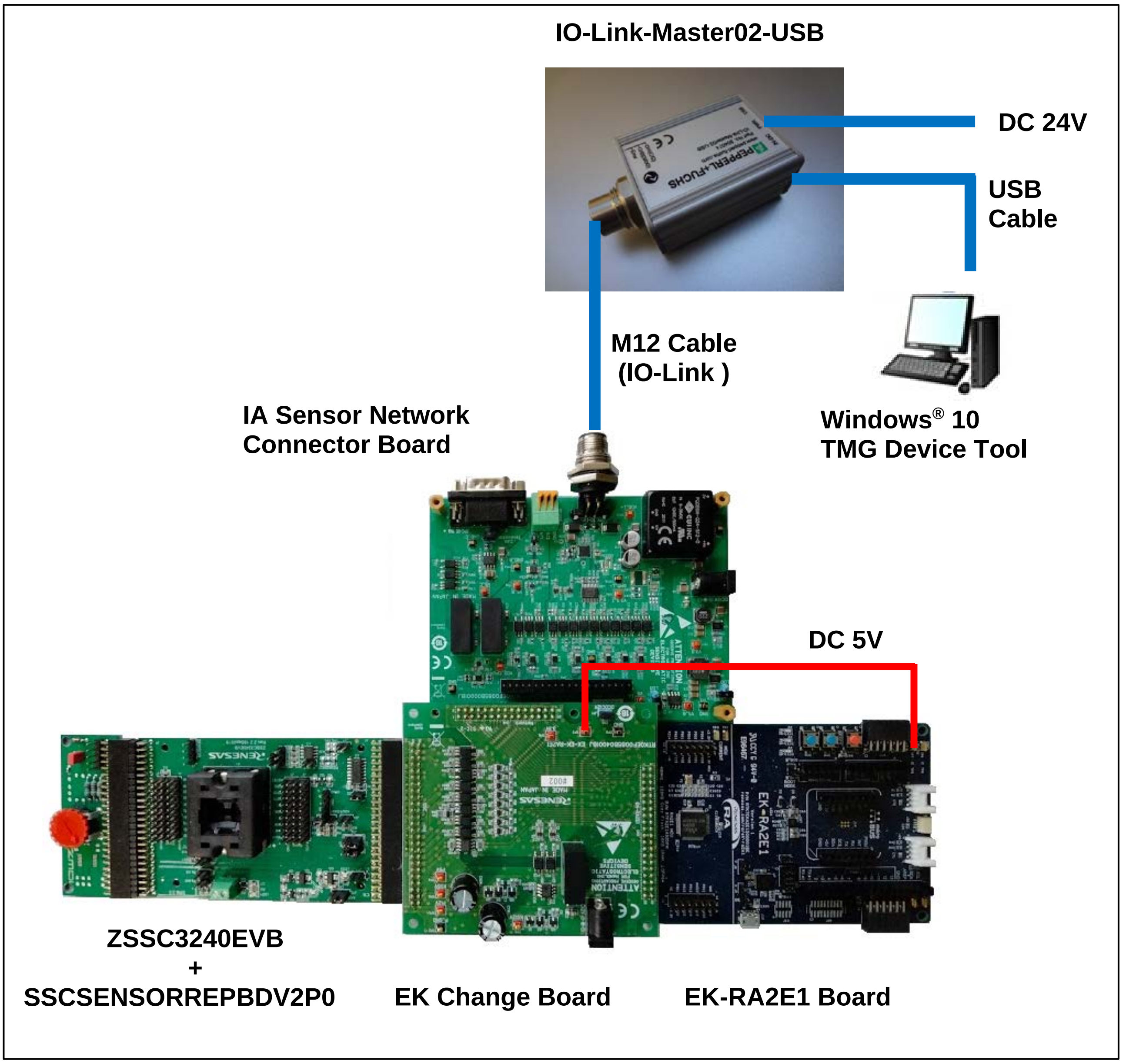


Figure 2. Hardware Configuration

3.1 IO-Link Master

In this application note, Pepperl+Fuchs IO-Link-Master02-USB is used.

Figure 3. I-O Link Master Connector Description

Connect DC 24V to DC jack for power supply, and the USB cable to the USB terminal.

Connect IO-Link cable M12 to the connector.

3.2 IA Sensor Network Connector Board (IO-Link Communication Board)

This board supports "IO-Link", "RS485" and "CAN", but only "IO-Link" will be used in this application note.

Power is supplied from the IO-Link master board through the IO-Link cable.

Jumper is set to JP1: 1-2 and JP2: Open.

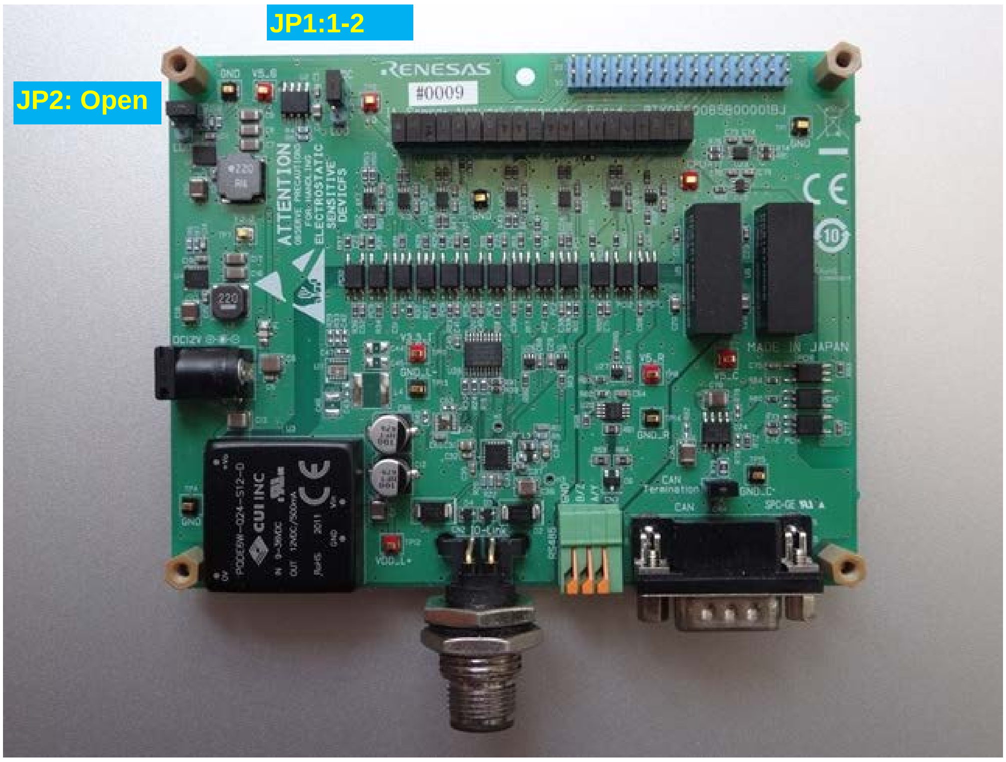


Figure 4. IO Link Communication Board

Note: Connecting the M12 cable to the IO-Link communication board while power is supplied to the IO-Link master may damage the IO-Link communication board. M12 cable connection to the IO-Link communication board should be made with no power supplied to the IO-Link master.

3.3 EK-RA2E1 Board

Write the sample project firmware to this board to control the measurement with ZSSC and IO-Link communication with the IO-Link master.

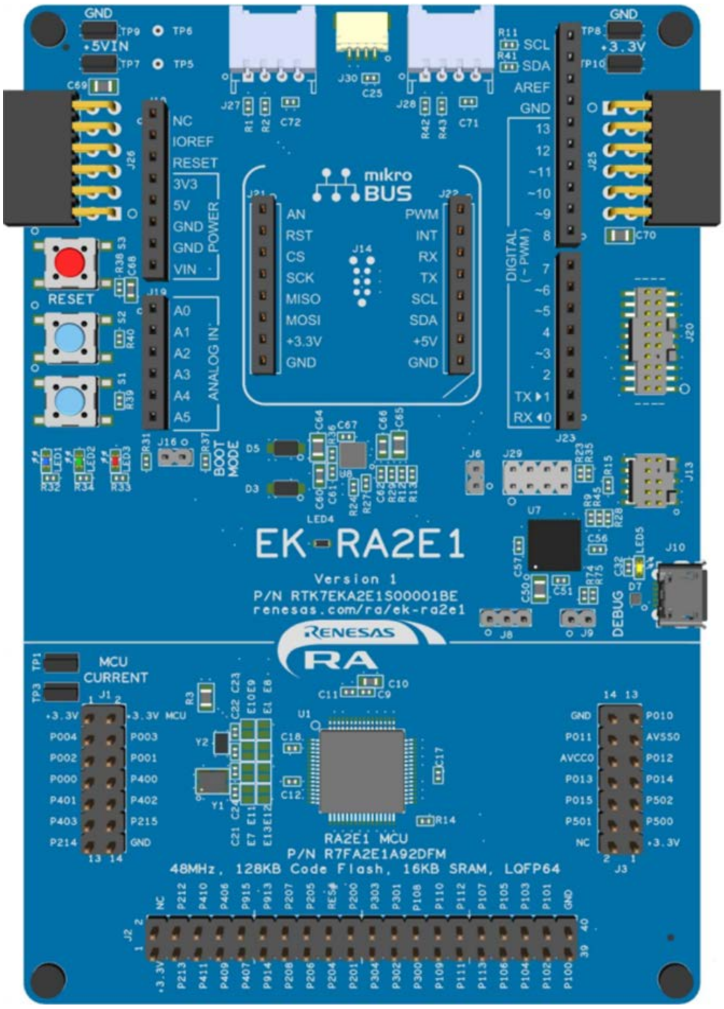


Figure 5. EK-RA2E1 Board

3.4 ZSSC3240 Evaluation Board

The ZSSC3240 Evaluation Board (hereafter referred to as “ZSSC EVB”) and the Sensor Replacement board (hereafter referred to as SRB) are used. There are three types of communication methods, I2C, SPI, and OWI, but in this application note, only I2C is covered.

To use the ZSSC EVB with I2C, connect 1-2 of J21/J22 to the jumper on the board.

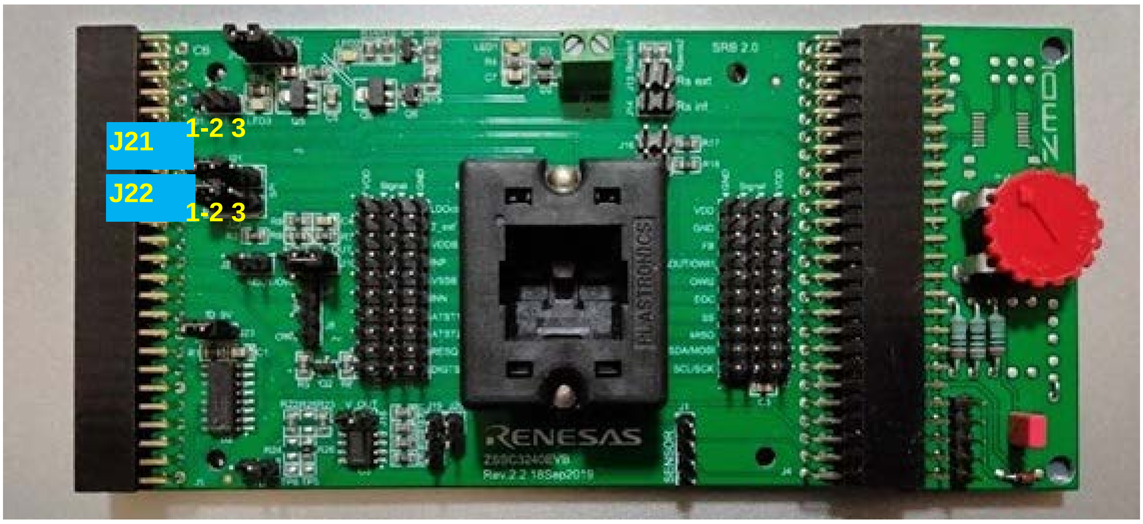


Figure 6. ZSSC3240 Evaluation Board (I2C Setting)

3.5 EK-RA2E1 Change Board

This conversion board is used to connect [ZSSC3240 Evaluation Board] [EK-RA2E1 Board] [IA Sensor Network Connector Board].

The power is supplied from the IO-Link communication board through the I/F connector, and the ZSSC EVB is also powered from this board. The EK-RA2E1 board can also be powered by connecting the 5V service output of this board to the + 5VIN terminal of the EK-RA2E1 board by cable.

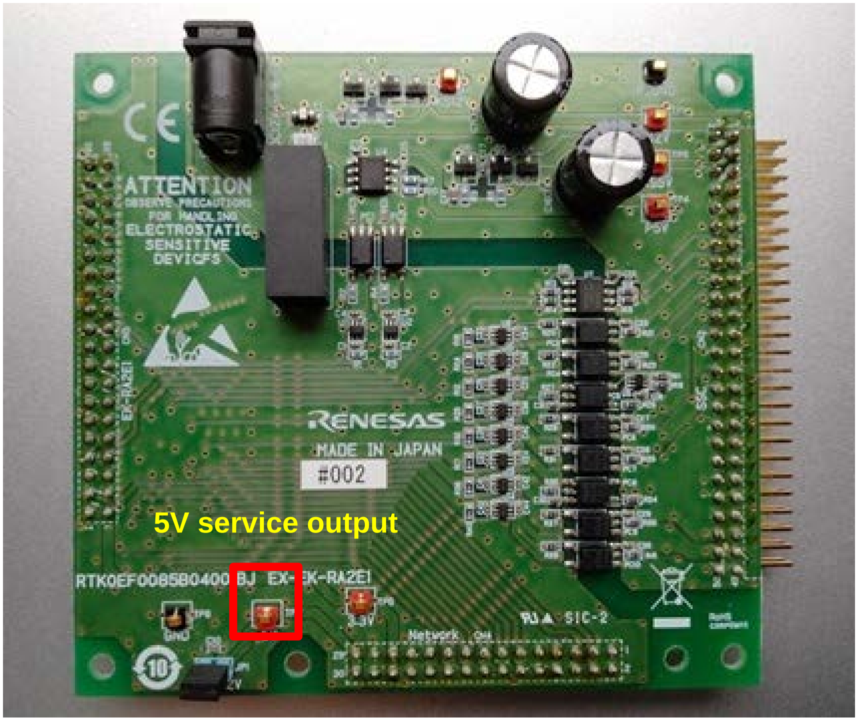


Figure 7. EK-RA2E1 Change Board

3.6 Power supply

The power supply to each board is supplied from the IO-Link-Master02-USB through the IO-Link M12 Cable.

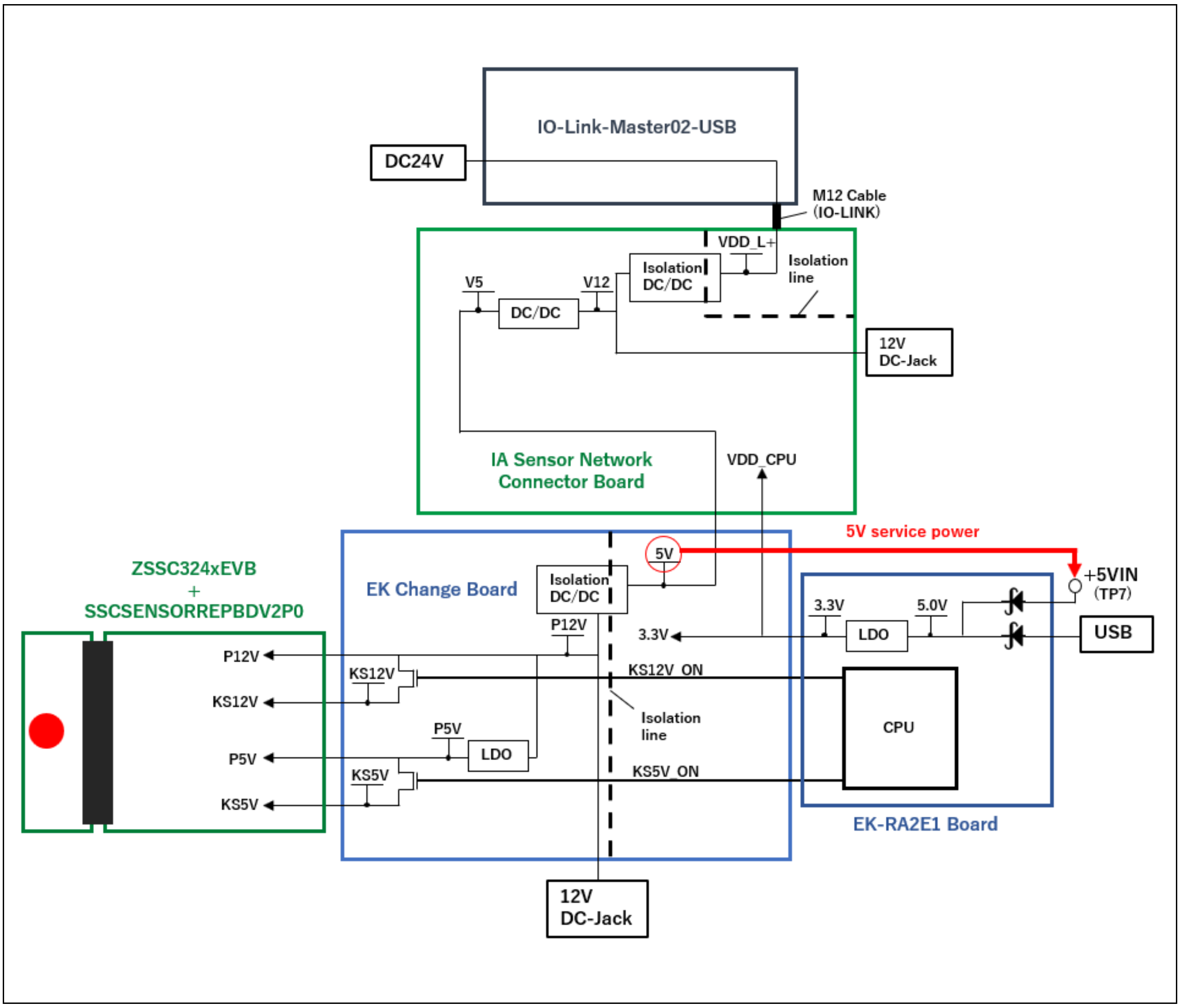


Figure 8. Power Supply

3.6.1 Connect the 5V Service Power Supply

Connect 5V(TP7) on the conversion board to +5VIN(TP7) on the EK-RA2E1 Board to supply power to the EK-RA2E1 Board.

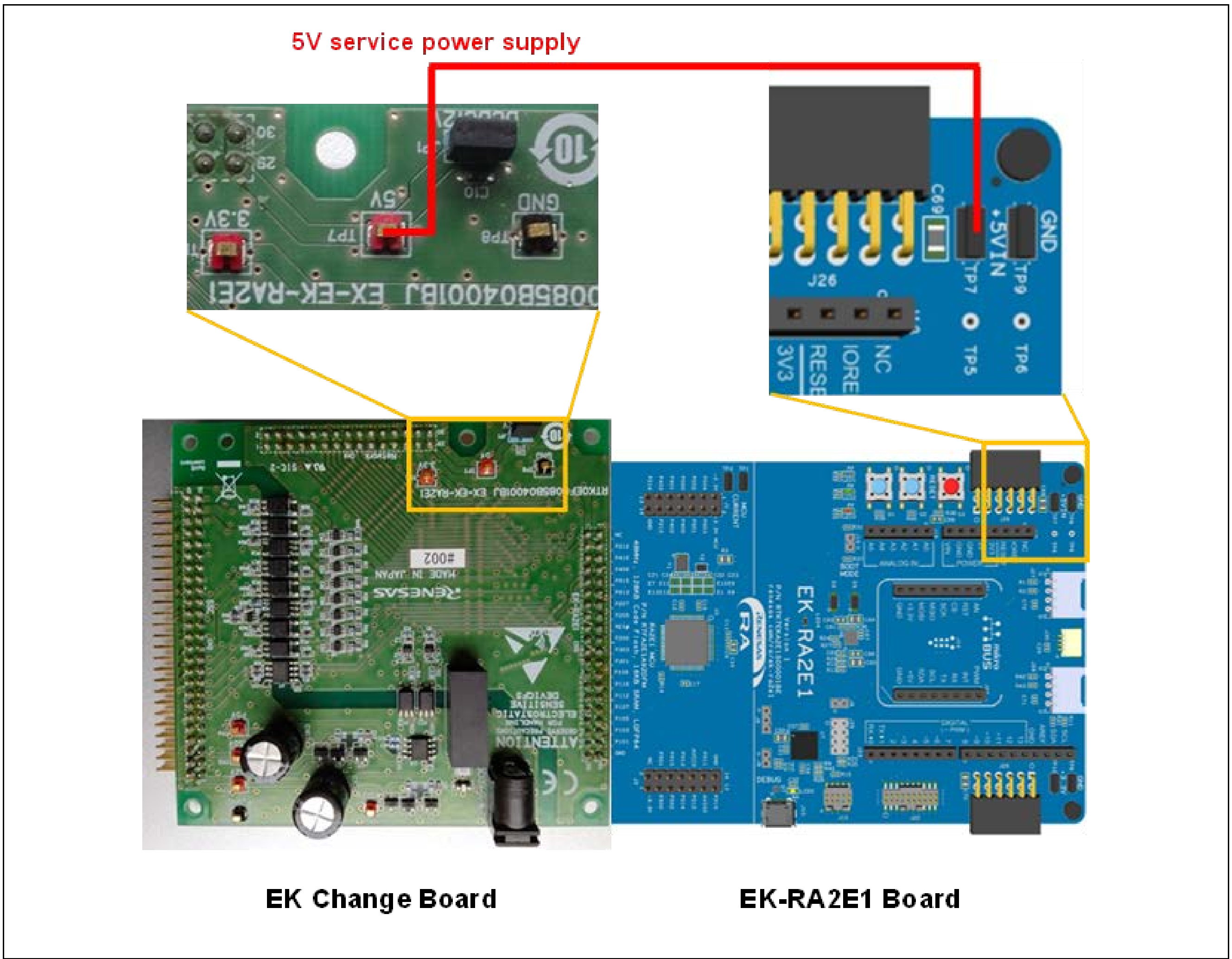


Figure 9. Connect the 5V Service Power Supply

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 a teaching process. The IO-Link device executes the measurement and threshold judgment periodically (once every 100 ms) and sends the information to the master via IO-Link communication. The information to be sent (process data) consists of measurement values and switching states (threshold judgment results).

For more information on the smart sensor profile, please refer to the documentation related to the IO-Link smart sensor profile, which can be downloaded from https://io-link.com/en/.

This is an overview of the software components that make up the sample application.

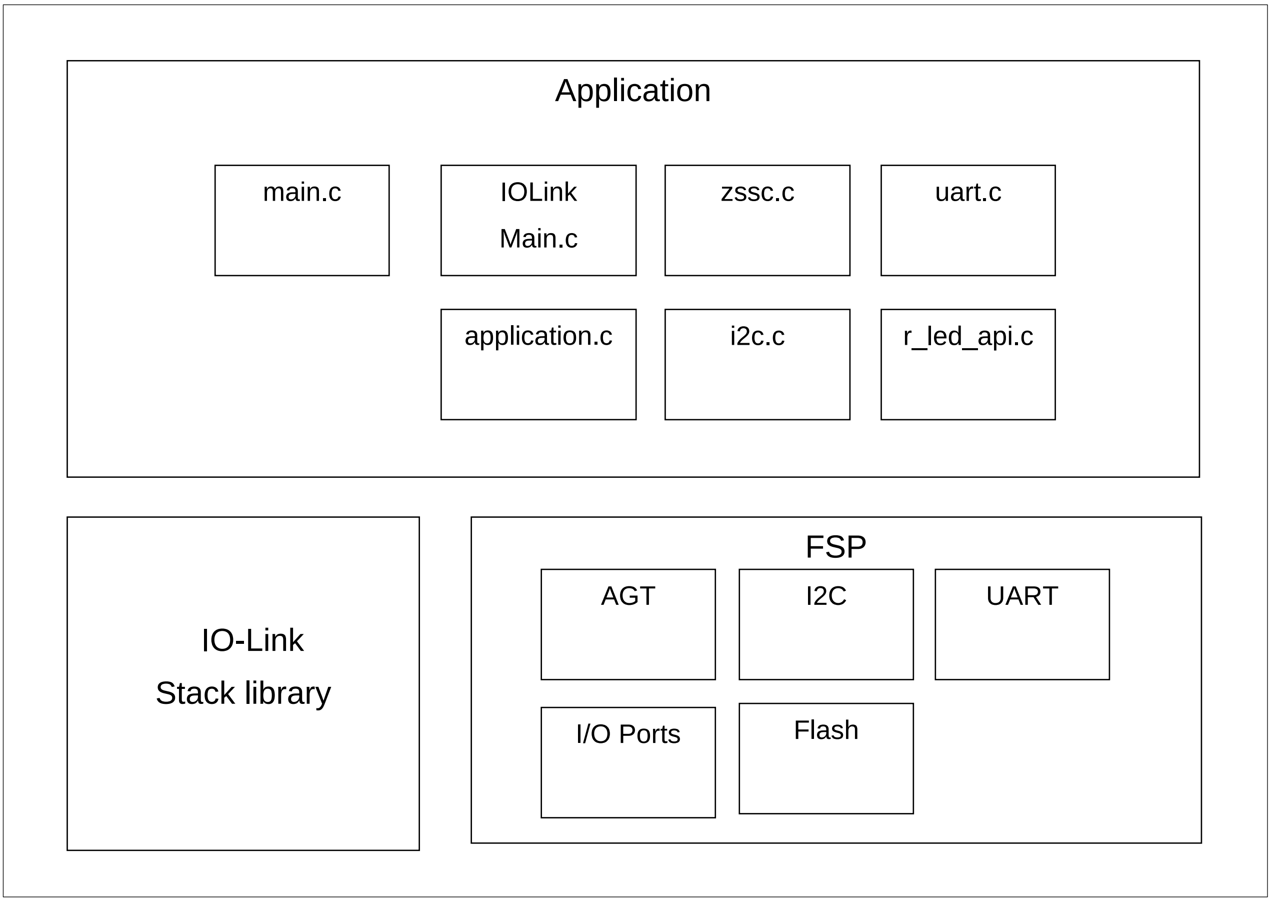


Figure 10. Software Components

4.1 Overview of the Overall Processing Flow

The processing flow of the sample application is described.

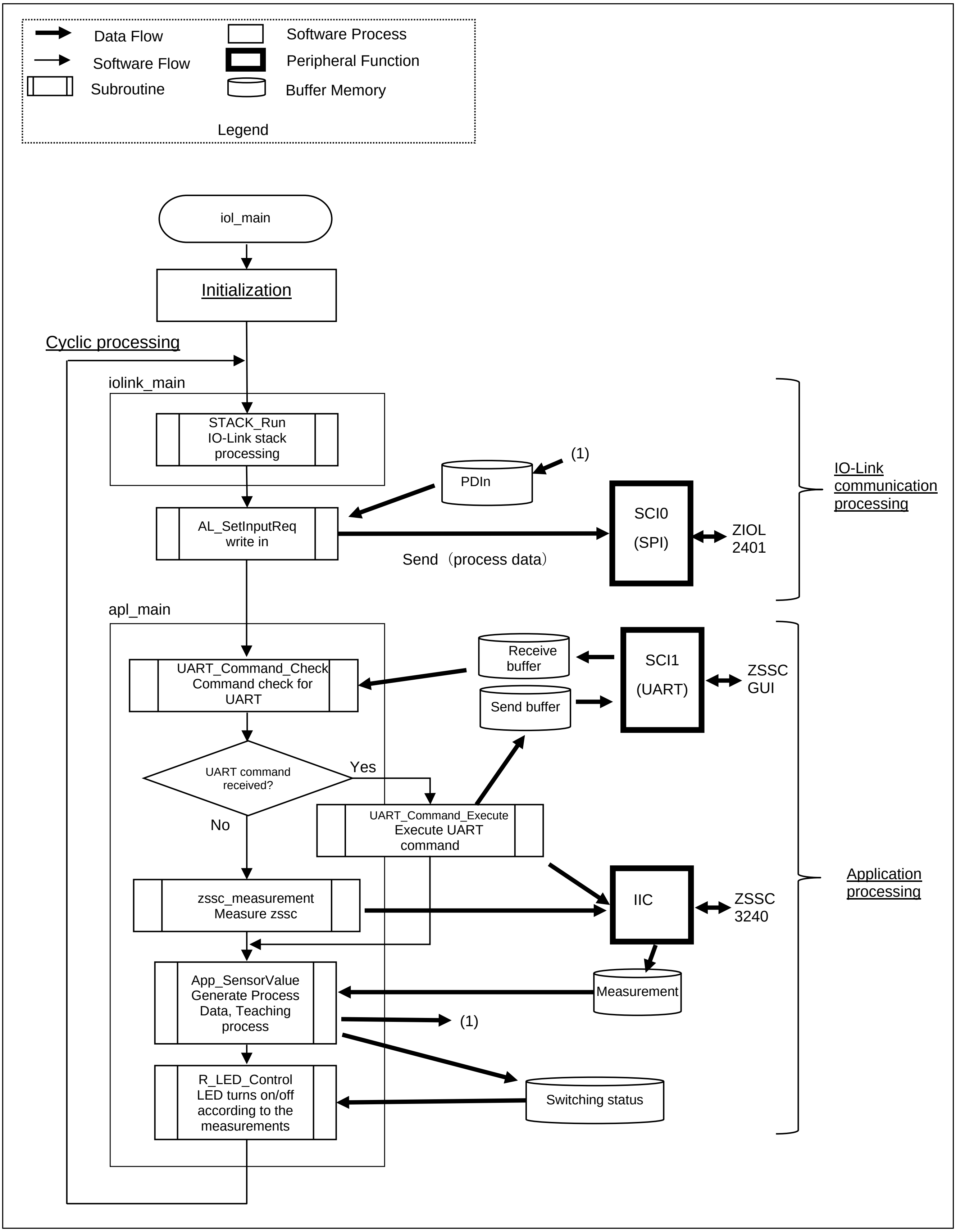


Figure 11. Processing Flow of the Sample Application

The outline of each process is described as follows:

1. Initialization
   Initializes IO-Link, UART, I2C, and resets ZSSC. It also reads ZSSC NVM data and stores it in a variable.

2. IO-Link communication processing
   A. Run STACK_Run (iolink_main)
      Execute STACK_Run, which is the API provided by the IO-Link stack.

   B. Process data writing process
      Execute AL_SetInputReq, which is the API provided by the IO-Link stack. Pass a pointer to the process data as an argument.

3. Application processing
   A. Execution of UART command reception check function (UART_Command_Check)
      Check UART command reception.

   B. State transition depending on whether UART commands are received
      If a command is received, the UART_Command_Execute function is executed to process the UART command.
      If the UART command is an NVM setting command, the NVM data is read and stored in a variable.

   C. Measurement processing
      If UART command is not received, zssc_measurement is called and ZSSC measurement is executed.
      If a measurement command has been received from ZSSC GUI, it issues a measurement command in UART_Commnad_Execute and executes the measurement.
      In either case, the measurement result is stored in a global variable (measurement).

   D. Generate Process Data, Teaching process (App_SensorValue)
      Generates process data from the measurements.
      If the Teaching command is being executed, a parameter check is performed, and if it is outside the valid range, the Teaching command fails. If it is within the valid range, the Teaching command succeeds and the SP1 or SP2 setting value is updated according to the command.

      The switching state is judged based on the operation mode and the measurements. when the Teaching command is being executed (Teach_Result is other than IDLE or SUCCESS), the switching state is not judged and the switching state is OFF.

      If ConfigLogic (logical setting of the switching state) is set to Inverted, the switching state bit is inverted.

      For details of this process, refer to section, 4.2 Flowchart of App_SensorValue Function.

   E. LED on/off (R_LED_Control)
      Pass the switching status to the argument of R_LED_Control. If the switching status is ON, the LED is turned on; if OFF, the LED is turned off.

4. Cyclic processing
   Return to step 2 and repeat the cyclic processing.

4.2 Flowchart of App_SensorValue Function

The flowchart of the App_SensorValue function is shown in Figure 12.

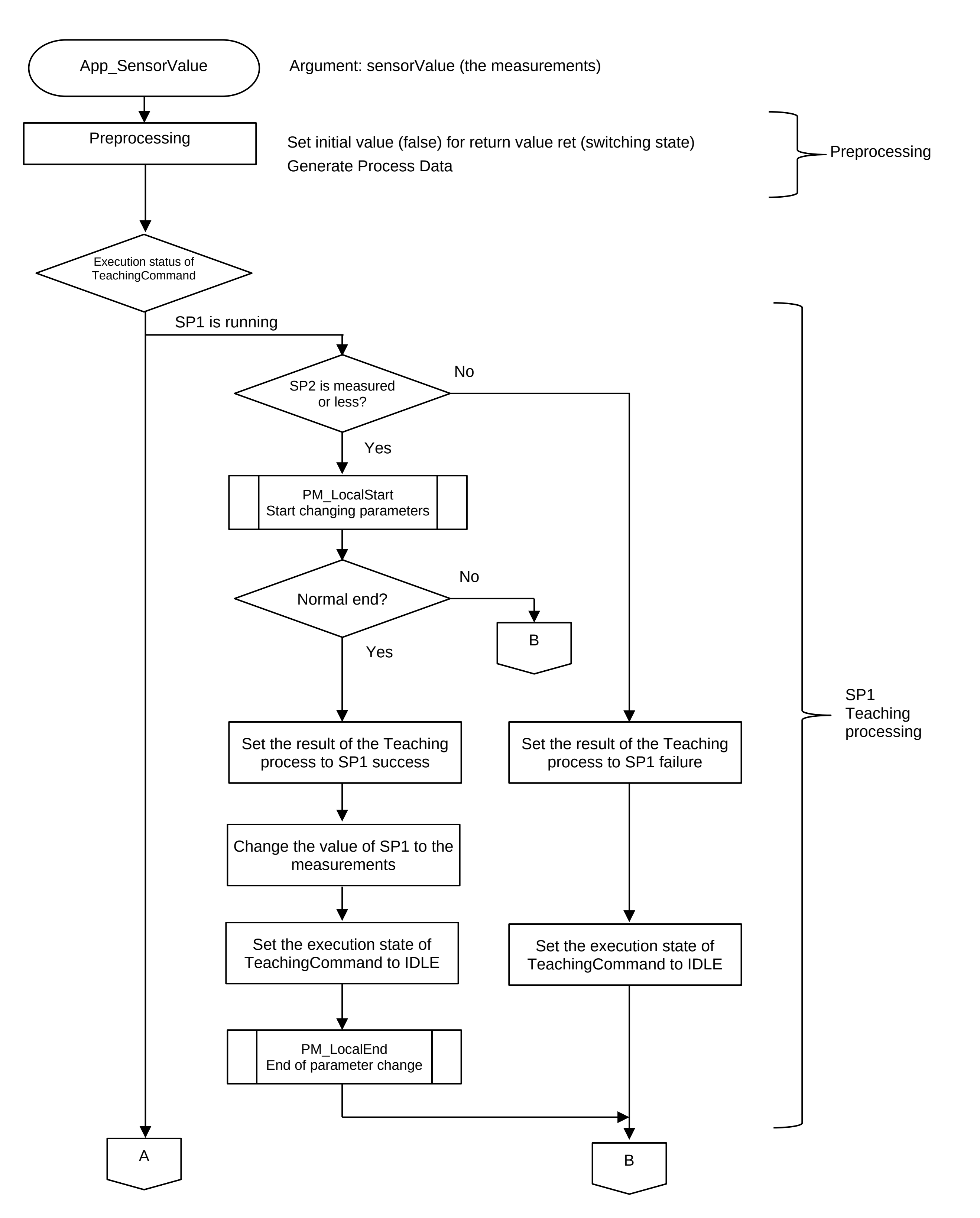


Figure 12. App_SensorValue Flowchart (1/4)

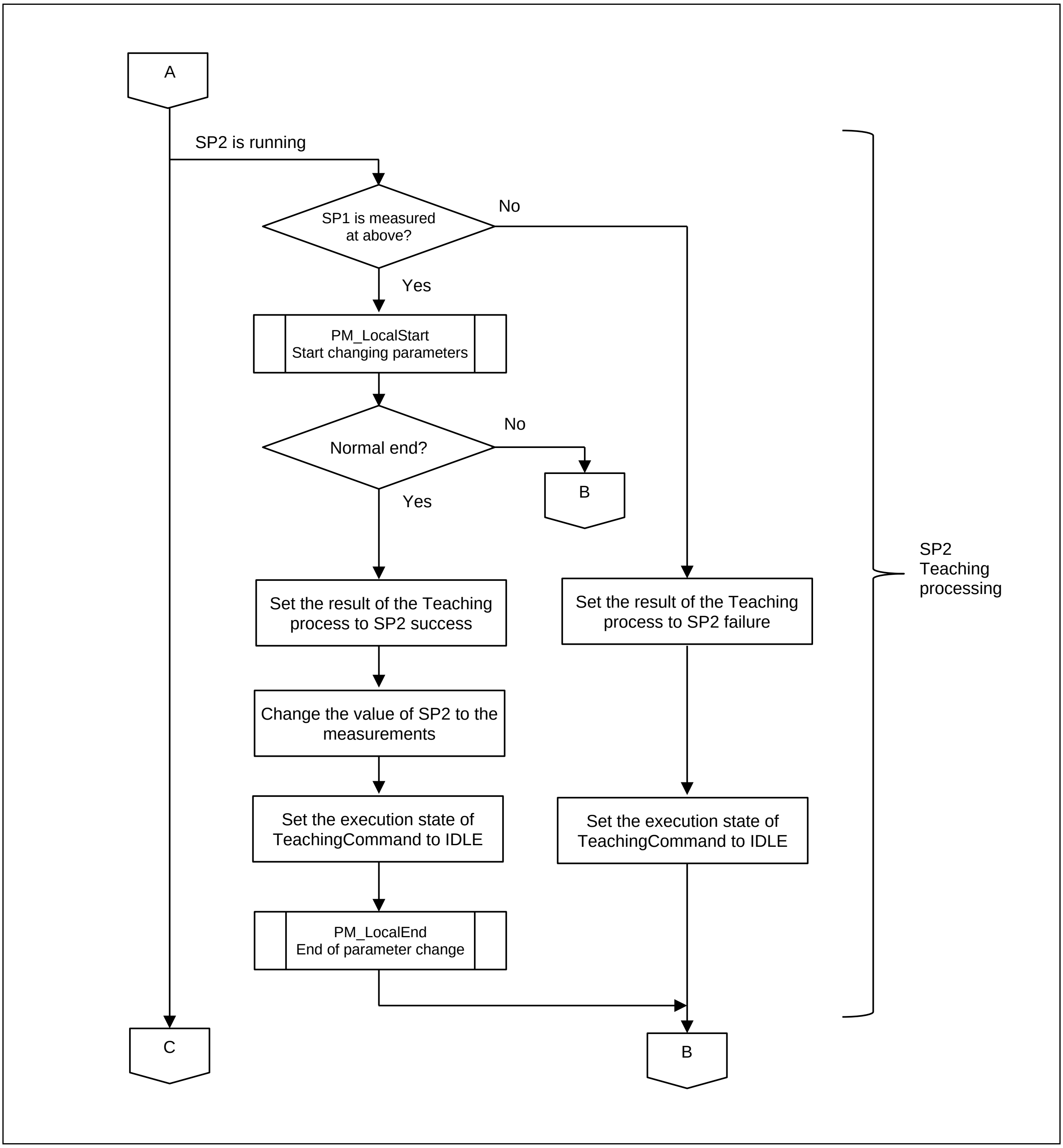


Figure 13. App_SensorValue Flowchart (2/4)

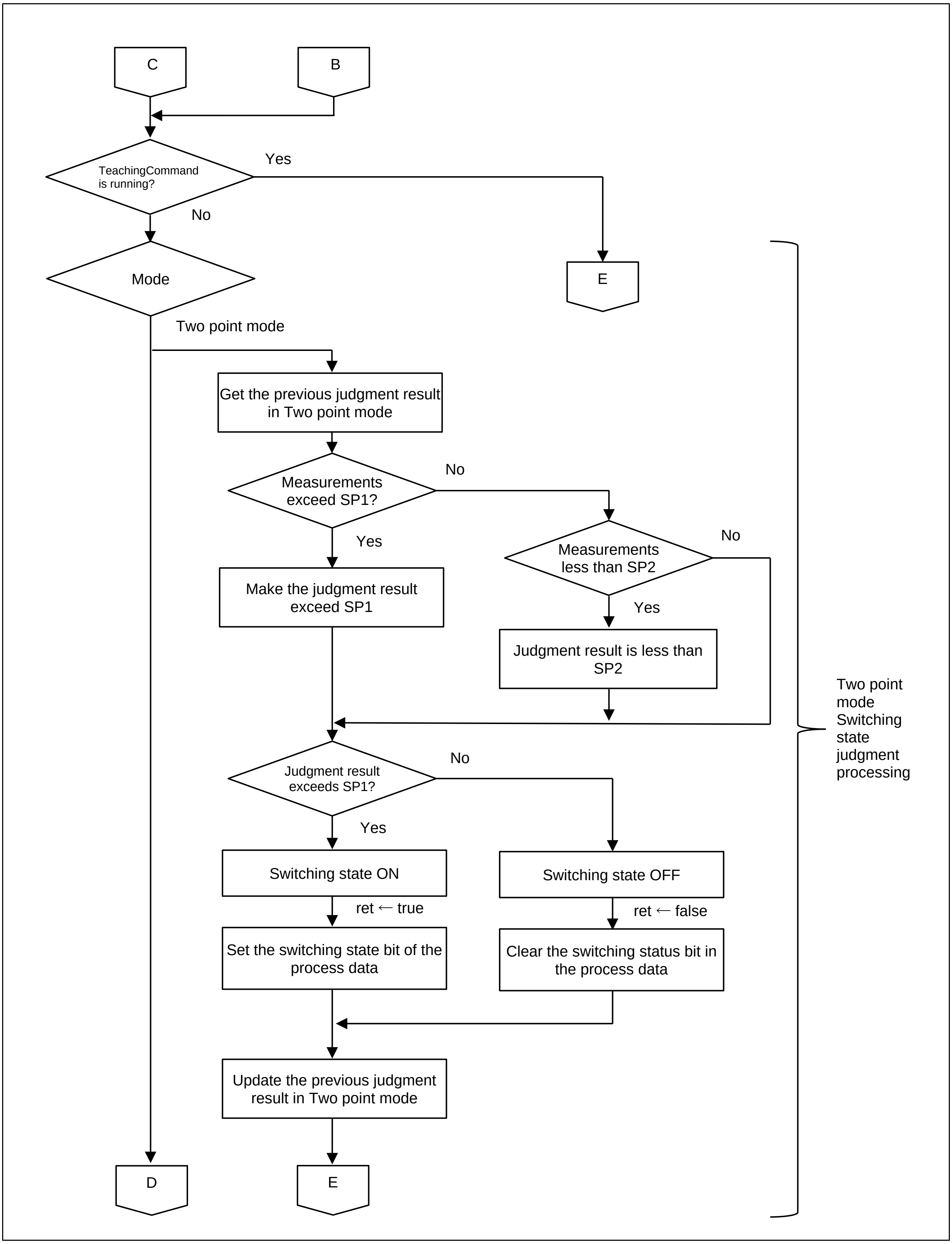


Figure 14. App_SensorValue Flowchart (3/4)

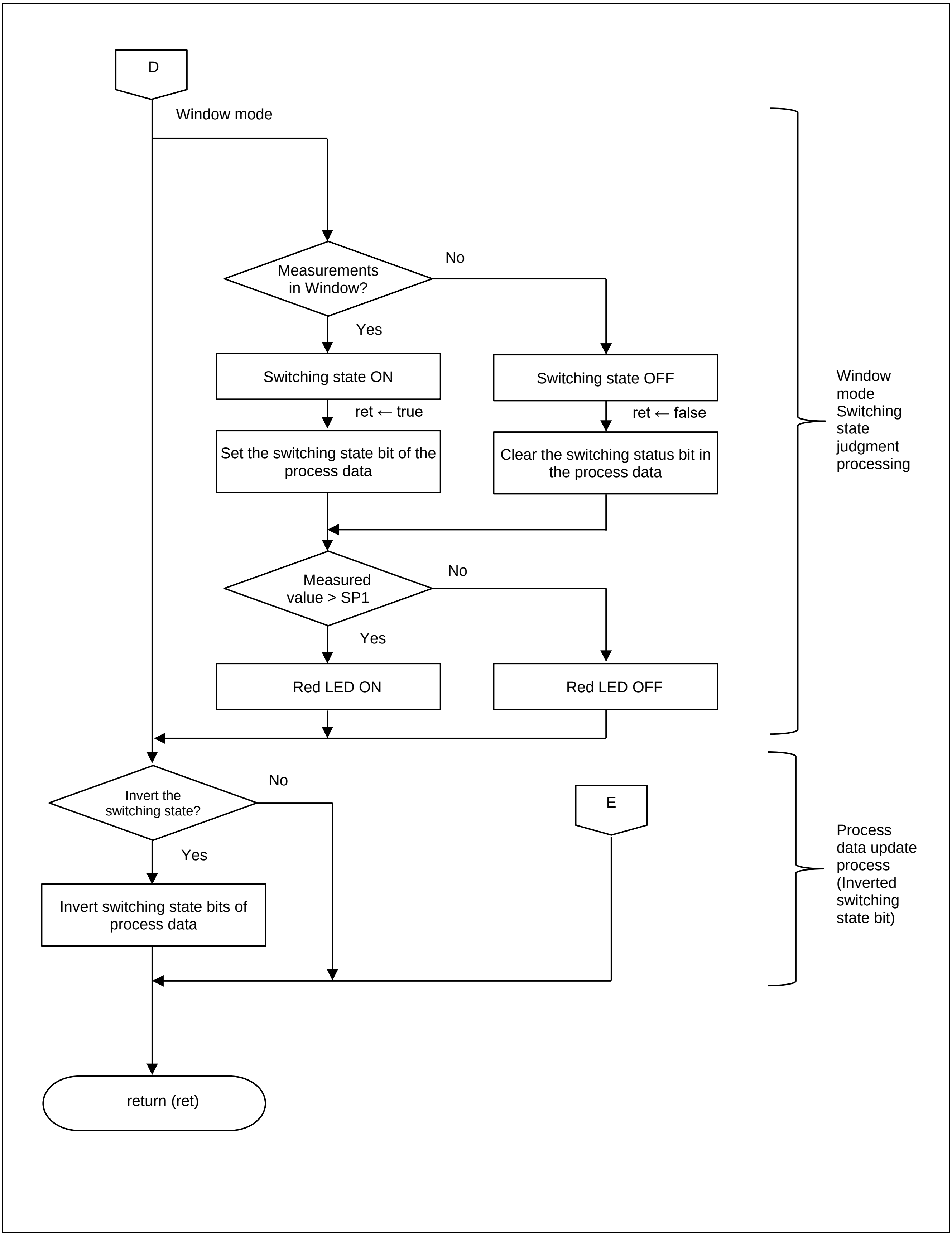


Figure 15. App_SensorValue Flowchart (4/4)

The outline of each process is described as follows:

1. Preprocessing
   - Set the initial value (false: OFF) to the return value ret (switching state).
   - Generate Process Data.
   - Stores the lower 24 bits of the sensor value (32 bits) passed in the argument in ProcessData.
   - Stores the sensor value [16:23] bit in ProcessData.PDIn[1].
   - Stores the sensor value [8:15] bit in ProcessData.PDIn[2].
   - Stores the sensor value [0:7] bit in ProcessData.PDIn[3].
   - Stores the Switching Point1 detection status in the least significant bit of ProcessData.PDIn[5].

2. SP1 Teaching processing*note

   When the execution status of TeachingCommand is "TeachingCommand of SP1", the following process is performed.
   - Since SP1 ≥ SP2 must be satisfied, the result of the Teaching process (ParSetStatic.V_TeachResult) is set as SP1 failure when the measurements < SP2.
   - In other case, the result of Teaching process is set as SP1 success, the measurements are copied to SP1 (ParSet.V_SetPointValues.SP1), and the execution status of TeachingCommand is set to idle.

3. SP2 Teaching processing *note

   When the execution status of TeachingCommand is "TeachingCommand of SP2", the following process is performed.
   - Since SP1≥SP2 must be satisfied, the result of the Teaching process (ParSetStatic.V_TeachResult) is set as SP2 failure when SP1 < the measurements.
   - In other case, the result of Teaching process is set as SP2 success, the measurements are copied to SP2 (ParSet.V_SetPointValues.SP2), and the execution status of TeachingCommand is set to idle.

4. Two point mode Switching state judgment processing

   When TeachingCommand is not running (idle or success) and the operation mode (ParSet.V_SetPointConfig.Mode) is Two point mode, the following process is performed.
   - Get the previous judgment result of Two point mode (initial value is less than SP2).
   - If SP1< measurements, the judgment result is SP1 exceeded.
   - If SP2> measurements, the judgment result is less than SP2.
   - In other cases, the judgment result is the previous judgment result.
   - If the judgment result exceeds SP1, the switching state is set to ON and ProcessData.PDIn[5] is set to 1.
   - If the judgment result is less than SP2, the switching state is set to OFF and ProcessData.PDIn[5] is set to 0.
   - Update the previous judgment result in the Two point mode to the current judgment result.

5. Window mode Switching state judgment processing

   When TeachingCommand is not running (idle state or success) and the operation mode (ParSet.V_SetPointConfig.Mode) is Window mode, the following process is performed.
   - If SP1≥measurements ≥SP2, the switching state is set to ON and ProcessData.PDIn[5] is set to 1.
   - In other cases, the switching state is set to OFF and ProcessData.PDIn[5] is set to 0.

   Red LED processing:
   - If SP1 < current sensor value, the red LED turns on.
   - If SP1 ≥ current sensor value, the red LED turns off.

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

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

   Invert the least significant bit of ProcessData.PDIn[5].

Note: When changing ParSet.V_SetPointValues, execute the PM_LocalStart function provided by the IO-Link stack, make sure that the return value is True, and then make the change. After the change, it is necessary to execute the PM_LocalEnd function. If the return value of the PM_LocalStart function is false, exit the App_SensorValue function without changing the execution status of ParSetStatic.V_TeachResult, Parset.V_SetPointValues, and TeachingCommand, and retry the Teaching process in the next post-measurement process.

4.3 Operating Mode and Switching State

The operation mode and switching state of this sample application are described. In order to check the switching status by LED as well as process data, the LED is turned on and off according to the switching status.

- Window mode
  When Switching Point1(SP1) ≥ measurements ≥ Switching Point2 (SP2) is met, the LED(green) turns ON.
  When measurements > Switching Point1(SP1) is met, LED (red) turns on when the measured value is satisfied.

- Two point mode
  In this mode, the switching state is determined with hysteresis.
  — If the measured value is rising, the threshold is at Switching Point 1 (SP1). Once the measured value > SP1 is met, the LED (green) changes to ON. After that, the state is maintained until the measured value falls below SP2.
  — If the measured value is falling, the threshold is at Switching Point 2 (SP2). Once SP2 > measured value is met, the LED (green) changes to OFF. After that, the state is maintained until the measured value exceeds SP1.

The operation image is described.

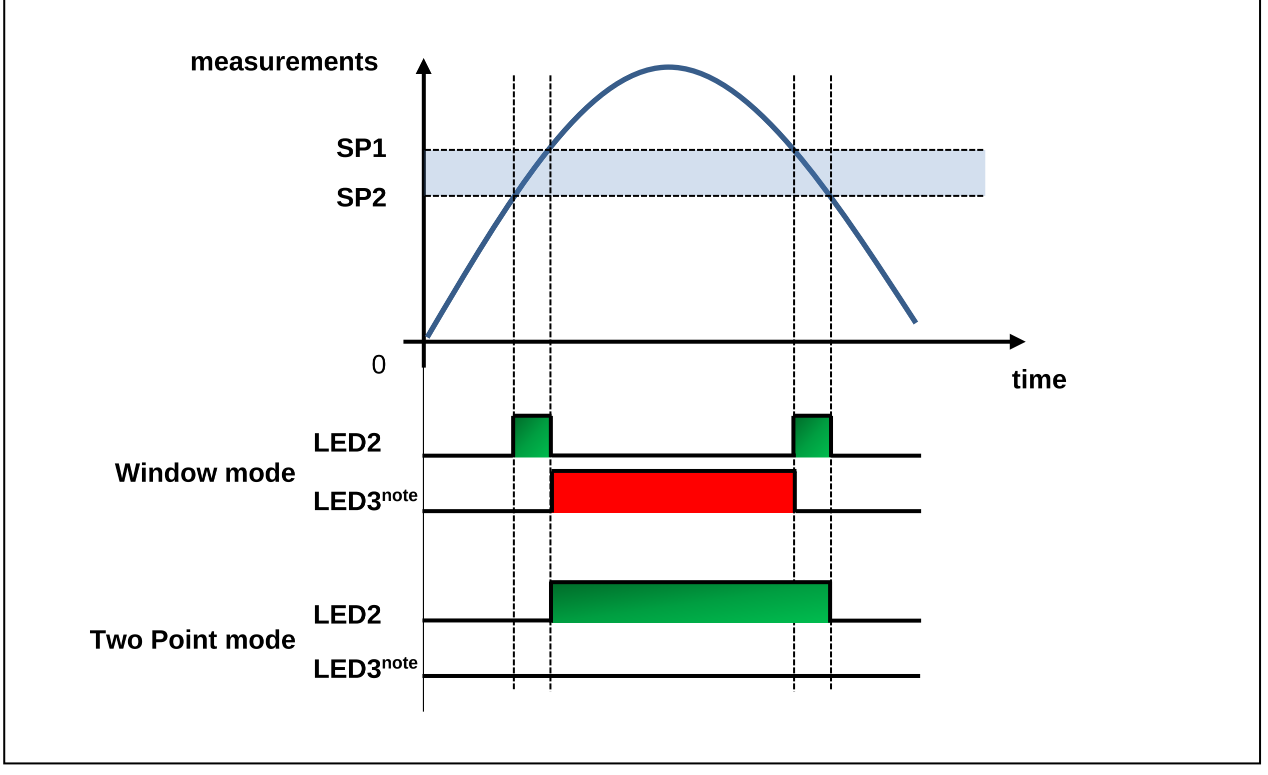


Figure 16. LED Operation Image

Note: The red LED works only in Window mode. Also, the status of the red LED is not reflected in Process Data, so you cannot check the status with the TMG Device Tool.

4.4 IO-Link communication

IO-Link communication specifications are described.

4.4.1 Bit Rate

The bit rate is COM2 (38.4 kbps).

4.4.2 SIO Mode

SIO mode is not supported in this sample application.

4.4.3 Process Data (PDIn)

The measurement information is sent to the IO-Link master as process data. The contents of the process data are described in Table 2.

Table 2. Process Data (PDIn)

Process Data (PDIn)		**Data length : 6 bytes**	
PDIn[n]	**Bit**	**Stored data**	**Details**
PDIn[0]	0-7	Sensor Value(bit24-32)	Stores Bit [24-32] of the sensor value
PDIn[1]	0-7	Sensor Value(bit16-23)	Stores Bit [16-23] of the sensor value
PDIn[2]	0-7	Sensor Value(bit8-15)	Stores Bit [8-15] of the sensor value
PDIn[3]	0-7	Sensor Value(bit0-7)	Stores Bit [0-7] of the sensor value
PDIn[4]	0-7	-	(Not used)
PDIn[5]	1-6 0	- Switching Signal	(Not used) Stores Switching Signal status

Table 3 shows an example of process data when the sensor value is 0x00010B4F and the Switching Signal state is 1.

Table 3. Example of Process Data

-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PDIn[0]								
Value	0	0	0	0	0	0	0	0
PDIn[1]								
Value	0	0	0	0	0	0	0	1
PDIn[2]								
Value	0	0	0	0	1	0	1	1
PDIn[3]								
Value	0	1	0	0	1	1	1	1
PDIn[4]								
Value	0	0	0	0	0	0	0	0
PDIn[5]								
Value	0	0	0	0	0	0	0	1

4.4.4 Parameters

The list of parameters to be sent and received with the master by IO-Link is described in Table 4.

Table 4. List of Setting Parameters

Name (Type)	Number of Bits	Value range (initial value)	R/W	Unit	Overview
		Switching Signal Channel1			
Param SP1 (unsigned int)	32	0x00000000 to 0x00FFFFFF (55000) note2	RW	-	Switching Point1 (SP1) setting note1 SP1≥SP2 must be met.
Param SP2 (unsigned int)	32	0x00000000 to 0x00FFFFFF (45000) note2	RW	-	Switching Point2 (SP2) setting note1 SP1≥SP2 must be met.
Config Logic (unsigned char)	8	0, 1 (0)	RW	-	Switching state logical setting 0: High active Send 1 if ON. 1: Low active Send 0 if ON.
Config Mode note3 (unsigned char)	8	0, 1, 2, 3 (0)	RW	-	Switching state judgment mode setting 0：Deactivated Disabled, switching state is always OFF 1：Single point 2：Window 3：Two point
Hysteresis note4 (unsigned sort)	16	0x0000 to 0xFFFF (0)	R/W	-	Hysteresis
Teach-In Result：state (unsigned char)	8	0, 1, 2, 3, 4, 5, 7 (0)	RO	-	Results of the previous Teach-In command 0: Idle, 1: SP1 Success, 2: SP2 Success 3: SP12 Success, 4: Wait 5: Busy 7: Error

Note 1. Refer to section, 4.3 Operating Mode and Switching State.

2. The Device Tool sets the value in the range of 0 to 16777215 (0xFFFFFF). For the display image, refer to section, 6.6.5 Parameter Tab.
3. The default setting is Deactivated. If using switching judgment, change to Windows or Two point. Do not select Single point as it is not supported by this sample application.
4. Not supported in this sample application. Do not use.

4.5 Peripheral features and Terminals to be Used

The list of peripheral functions used in this sample application is described in Table 5, and the list of pins used is described in
Table 6. List of Terminals Used .

Table 5. List of Peripheral Features to be Used

Peripheral Features	Usage
SCI0 (Simplified SPI)	SPI Master for ZIOL2401 communication(used in IO-Link stack)
SCI1 (UART)	UART communication with PC (ZSSC GUI)
IIC	IIC Master for ZSSC communication
AGT0	Cycle timer for IO-Link communication (used in IO-Link stack)

Table 6. List of Terminals Used

Terminal Name	I/O	Usage
P411/IRQ4	Input	IO_INT(Interrupt occurs when LOW)
P410	Input	WakeUp detection signal input pin of IO-Link
P409/IRQ6	Input	DC/DC Ready signal input pin of ZIOL2401
P408/SCL	Output	IIC clock pin for ZSSC communication
P407/SDA	I/O	IIC data pin for ZSSC communication
P913	Output	Control of EK board LED3
P914	Output	Control of EK board LED2
P401/TXD	Output	UART TXD for PC (ZSSC GUI) communication
P402/RXD	Input	UART RXD for PC (ZSSC GUI) communication
P304	Output	ZSSC KS5V ON/OFF (ON at L output)
P303	Output	ZSSC KS12V ON/OFF (ON at L output)
P302	Output	IO-Link IO_AUX_TX
P301	Output	IO-Link IO_AUX_EN
P106	Output	Reset control terminal of ZIOL2401 (Reset is released by L output)
P105	Output	Reset control terminal of ZSSC (Reset is released by L output)
P104/IRQ1	Input	EOC (End-Of-Conversion signaling) of ZSSC
P103/SSL0	Output	SPI communication enable/disable control pin of ZIOL2401 (enabled by L output)
P102/SCK0	Output	SPI communication SCK0 of ZIOL2401
P101/MOSI0	Output	SPI communication for ZIOL2401 MOSI0
P100/MISO0	Input	SPI communication for ZIOL2401 MISO0

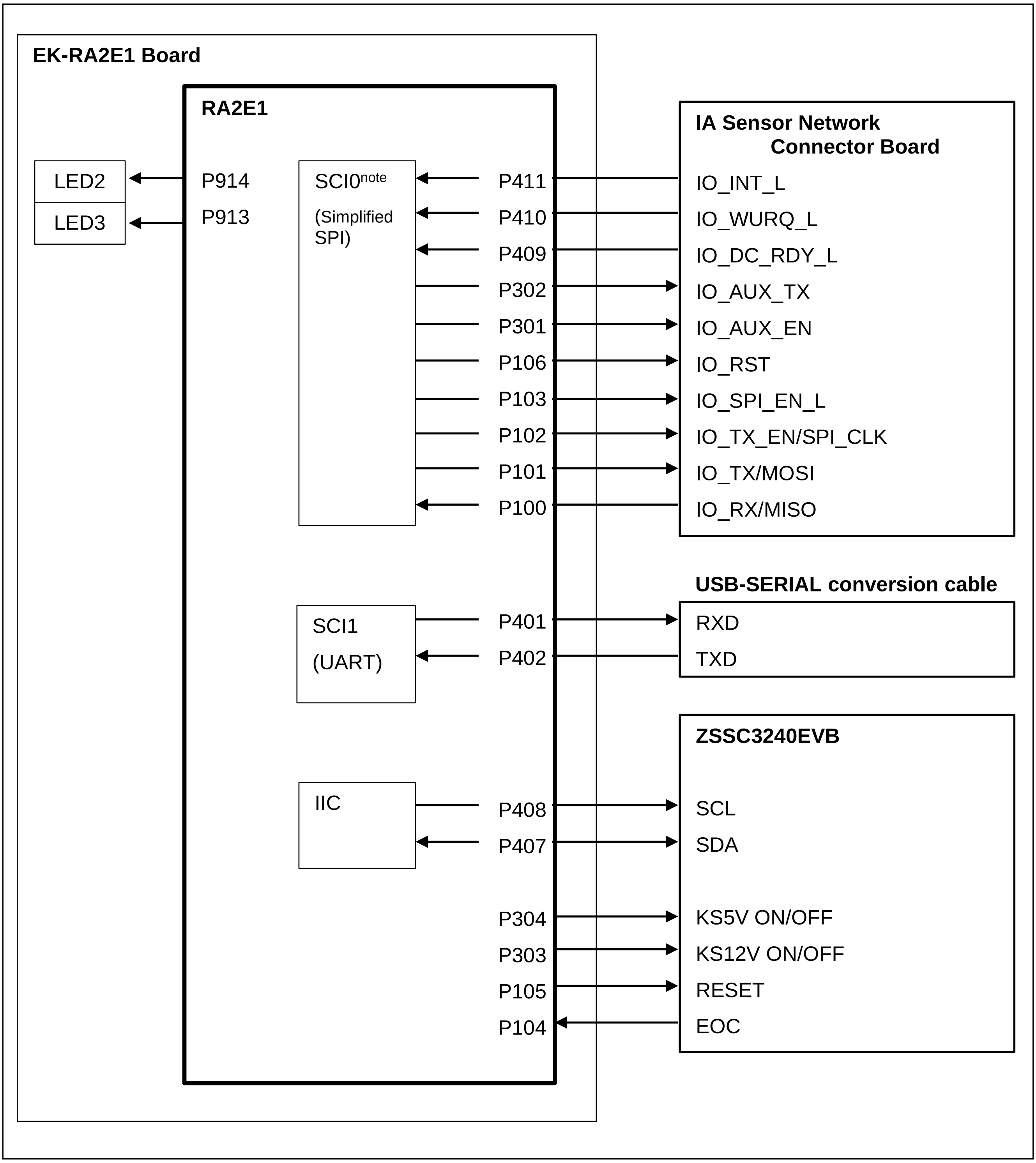


Figure 17. Peripheral Features and Terminals to be Used

Note: The SCI0 simple SPI is controlled by a proprietary driver provided by the IO-Link stack library. It does not use the SCI.SPI driver provided by FSP.

4.6 Program Structure

4.6.1 File Structure

The TMG IO-Link stack and related files are located in the Library folder. The IO-Link stack manuals are located in the Manuals folder. The file structure in the IO-Link folder is described in Table 7 .

Table 7. File Structure

Folder name, File name	Description
IO-Link	
├─Application	Application folder
│ application.c	Application program
│ application.h	Application header
│ define.h	Define header
│ global.h	Global header
│ i2c.c	I2C program
│ i2c.h	I2C header
│ main.c	Application main
│ r_led_api.c	LED control program
│ r_led_api.h	LED control header
│ spi.c	SPI processing program
│ spi.h	SPI header
│ Starterkit_Config.h	Starterkit_Config.h
│ tool.c	tool function
│ tool.h	tool function header
│ uart.c	UART command processing program
│ uart.h	UART command processing header
│ utils.h	Utilities header
│ zssc.c	ZSSC processing program, ZSSC measurement processing
│ zssc.h	ZSSC header
│	
├─BSP	BSP folder for application parts not related to IO-Link
│ BSPStack.h	User-implemented Hardware Abstraction definitions
│ IOLD_Config.h	system specific Defines
│ SystemInit.h	Hardware settings definitions
│ ZIOL2401.h	Functions for ZMDI L2401
│	
├─IODD	IODD file storage folder for IO-Link devices
│ TMG-logo.png	TMG Logo Image File
│ TMG-RA2E1-Starterkit-20210715-IODD1.0.1.xml	IODD1.0.1 File
│ TMG-RA2E1-Starterkit-20210715-IODD1.1.xml	IODD1.1 File
│ TMG-RA2E1-Starterkit-con-pic.png	M12 connector 4Pin image file
│ TMG-RA2E1-Starterkit-icon.png	Board icon image file
│ TMG-RA2E1-Starterkit-pic.png	Board Image File
│	
├─Library	IO-Link stack library, parameter set storage folder
│ BSPInterface.h	Defines for the BSP
│ DeviceAccess.h	Device Access protection
│ DeviceStack.h	IO-Link Device Stack
│ DStorage.h	Data Storage for the DeviceStack
│ DTypes.h	Datatype Definitions

Folder name, File name	Description
│ EventDispatcher.h	functions for EventDispatching
│ libIO_Link_Starterkit_RA2E1_lib.a	IO-Link Library for the starter kit solution
│ ParameterManager.h	Interface to the application framework
│ Profile_common.h	common Declarations for Profiles
│	
├─Manuals	Manual folder
│ TMG IO-Link Device Integration Library Manual.pdf	IO-Link Device Evaluation Library
└─StackExtensionsApp	IO-Link related part storage folder of application
BSPExtensions.h	definitions for Stack Extensions Board Support Package
IOLinkMain.c	IO-Link Application
IOLinkMain.h	definitions for main program
MemoryManager.h	definitions for Stack Extensions Memory Manager
ParameterSet.h	Definitions for Stack Extensions
ProductionSettings.h	definitions for Stack Extensions Production Settings

4.6.2 Function List

In the following section, only the functions introduced in section, 4.1 Overview of the Overall Processing Flow are described in detail. For other functions, please refer to the source code of the included sample application project.

4.6.2.1 main.c

[Function name] iol_main

Overview	iol_main function
Header	None
Declaration	int iol_main(void)
Description	It is composed by adding application-specific processing to iol_main() provided by TMG. After initializing the IO-Link stack library and peripheral functions used in the application, it shifts to periodic processing.
Argument	None
Return value	None
Remarks	None

4.6.2.2 IOLinkMain.c

[Function name] iolink_main

Overview	IO-Link main processing
Header	IOLinkMain.h
Declaration	TUnsigned8 iolink_main(void)
Description	Call the STACK_Run function. Periodically calls the IO-Link Stack to get the status.
Argument	None
Return value	State of the IO-Link stack
	STACK_STATUS_SIO :
	IO-Link connection is SIO mode
	STACK_STATUS_STARTUP :

	Master is detected and device is in startup state
	STACK_STATUS_PREOPERATE：
	Device is in pre-operating state
	STACK_STATUS_OPERATE：
	Device is in working state
	STACK_STATUS_DISCONNECTED：
	Disconnected state, device waits for next wake-up in IO-Link mode
Remarks	None

4.6.2.3 application.c

[Function name] apl_main

Overview	Application main
Header	application.h
Declaration	void apl_main(void)
Description	Call UART_Command_Check() to check whether UART command is received. When UART command is received, call UART_Command_Execute() to process the UART command. If there is no UART command reception for 1 second, call zssc_measurement_reset() and then shift to ZSSC measurement processing. zssc_measurement() is called periodically to measure ZSSC as long as there is no UART command reception. If a UART command is received, the ZSSC measurement process is terminated immediately.
Argument	None
Return value	None
Remarks	None

[Function name] App_SensorValue

Overview	IO-Link application processing using sensor values
Header	None
Declaration	static bool App_SensorValue (TUnsigned32 sensorValue)
Description	Execute the following three processes.
	1. Creating process data
	2. Teaching process (threshold setting process)
	3. Switch point processing
Argument	Measurement
Return value	Switching judgment result
	false：
	Out of valid range (OFF)
	true：
	Within valid range (ON)

Remarks	This function is an internal function called by the apl_main function. While the Teaching process is in progress, the return value will be false.

4.6.2.4 uart.c

[Function name] UART_Command_Check

Overview	UART command reception check
Header	uart.h
Declaration	uint8_t UART_Command_Check(uint8_t *buffer)
Description	Check if the command is received from the ZSSC GUI.
Argument	buffer Buffer top pointer
Return value	Reception status 0: Command not received 1: Command received
Remarks	None

[Function name] UART_Command_Execute

Overview	UART command execution
Header	uart.h
Declaration	void UART_Command_Execute(uint8_t *buffer)
Description	Execute the command received from the ZSSC GUI. When a measurement command is executed, the measurement result is stored in the global variable sensor_value.
Argument	buffer Buffer top pointer
Return value	None
Remarks	None

4.6.2.5 zssc.c

[Function name] zssc_measurement

Overview	ZSSC measurement
Header	zssc.h
Declaration	bool zssc_measurement(void)
Description	The AA_{HEX} command performs ZSSC measurement and stores the measurement result in the global variable sensor_value.
Argument	None
Return value	true
Remarks	For the AA_{HEX} command, refer to the following materials. ・ZSSC3240 Evaluation Kit User Manual ・ZSSC3240 Data Sheet

4.6.2.6 r_led_api.c

[Function name] R_LED_Control

Overview	Control of LED
Header	r_led_api.h
Declaration	void R_LED_Control (uint16_t led_pin, bool state)
Description	Turns on / off the LED.
Argument	led_pin LED Pin number
Argument	state LED status false: OFF true: ON
Return value	None
Remarks	None

5. Sample Project Execution Method

This chapter describes the steps for importing a sample project and executing the program.

5.1 Importing Sample Projects

1. Start e^2 studio.
2. Enter a new workspace name in the Workspace dialog box. Then click **Launch**.

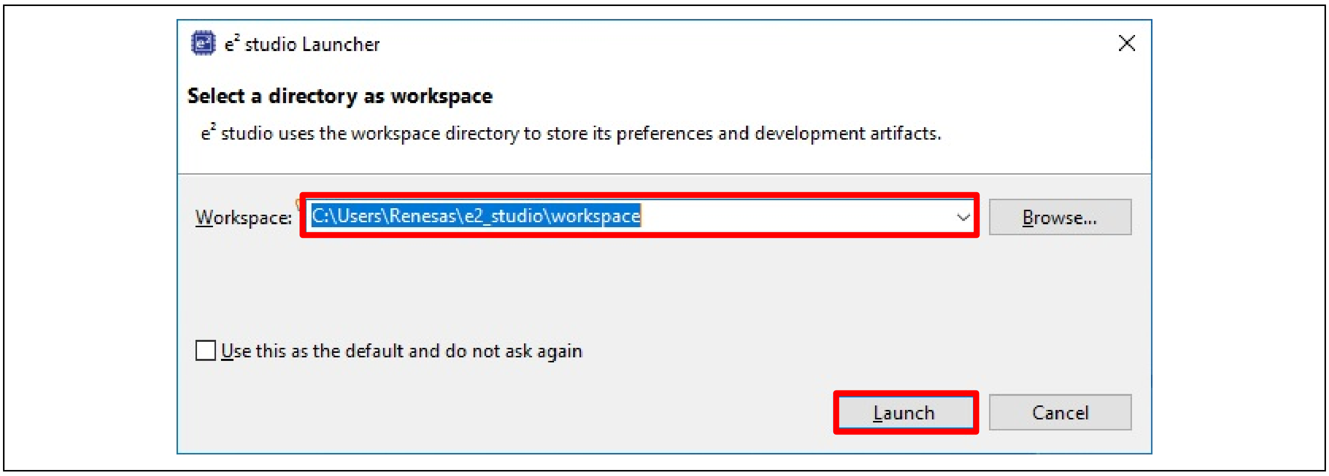


Figure 18. Start Using a New Workspace

3. Click on **Import existing projects**.

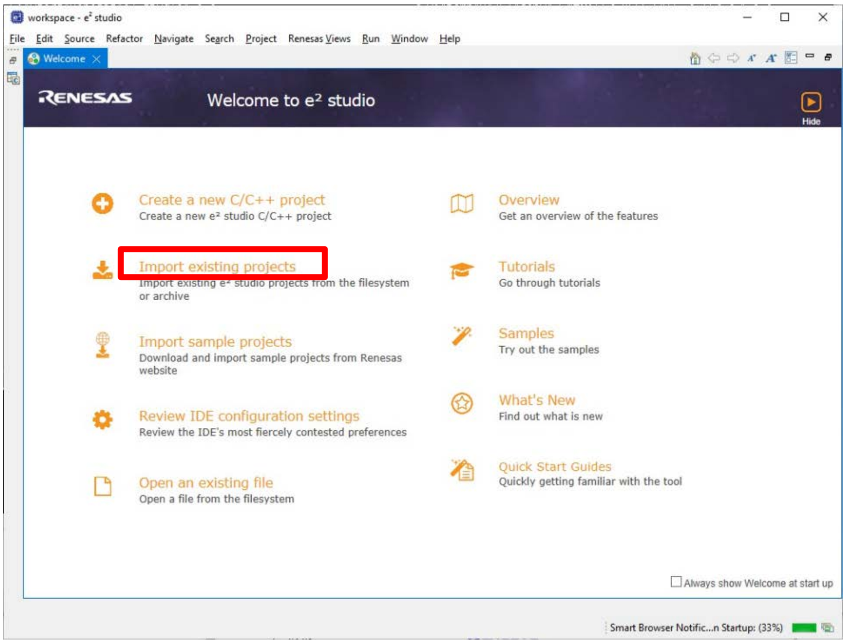


Figure 19. Welcome to e²studio

4. Click Select archive file, then click **Browse** to open the location of the sample project zip file.
5. Select the zip file of the sample project and click **Finish**.

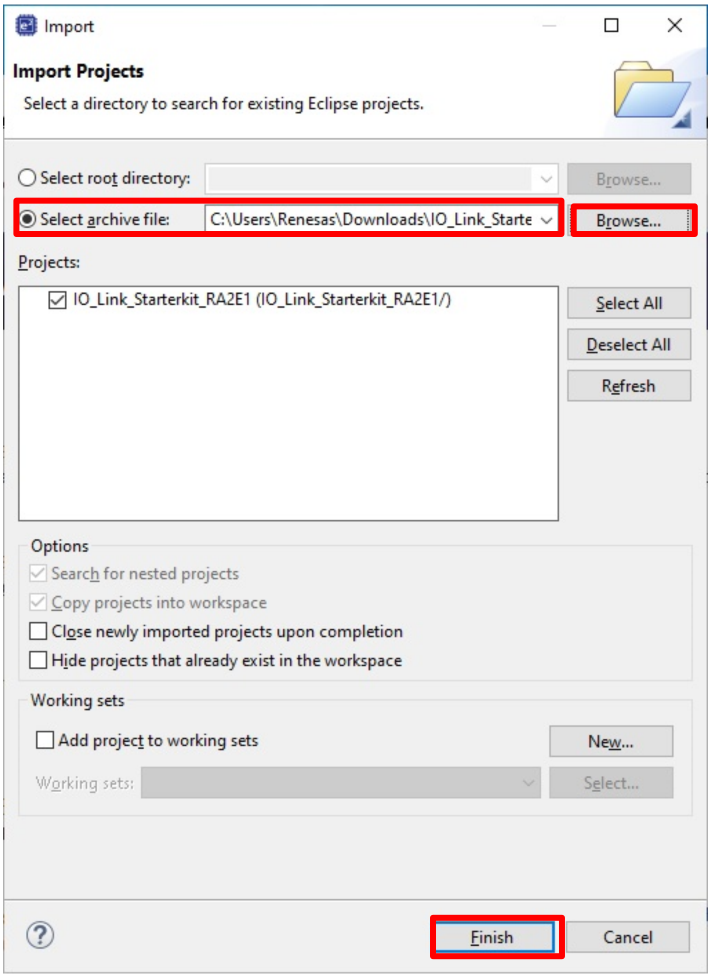


Figure 20. Importing Sample Projects

5.2 Build the Project

1. After the sample project is imported, double-click `configuration.xml` to open the configurator.

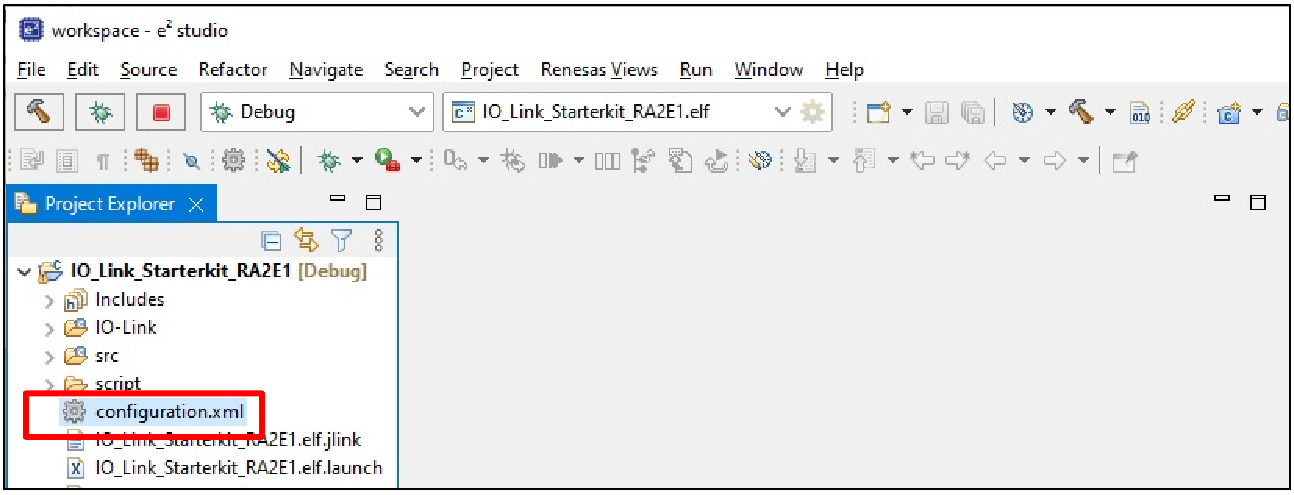


Figure 21. Start the Configurator

2. Click **Generate Project Content** on the **Stacks** tab. The configurator generates the necessary files and adds them to the project.

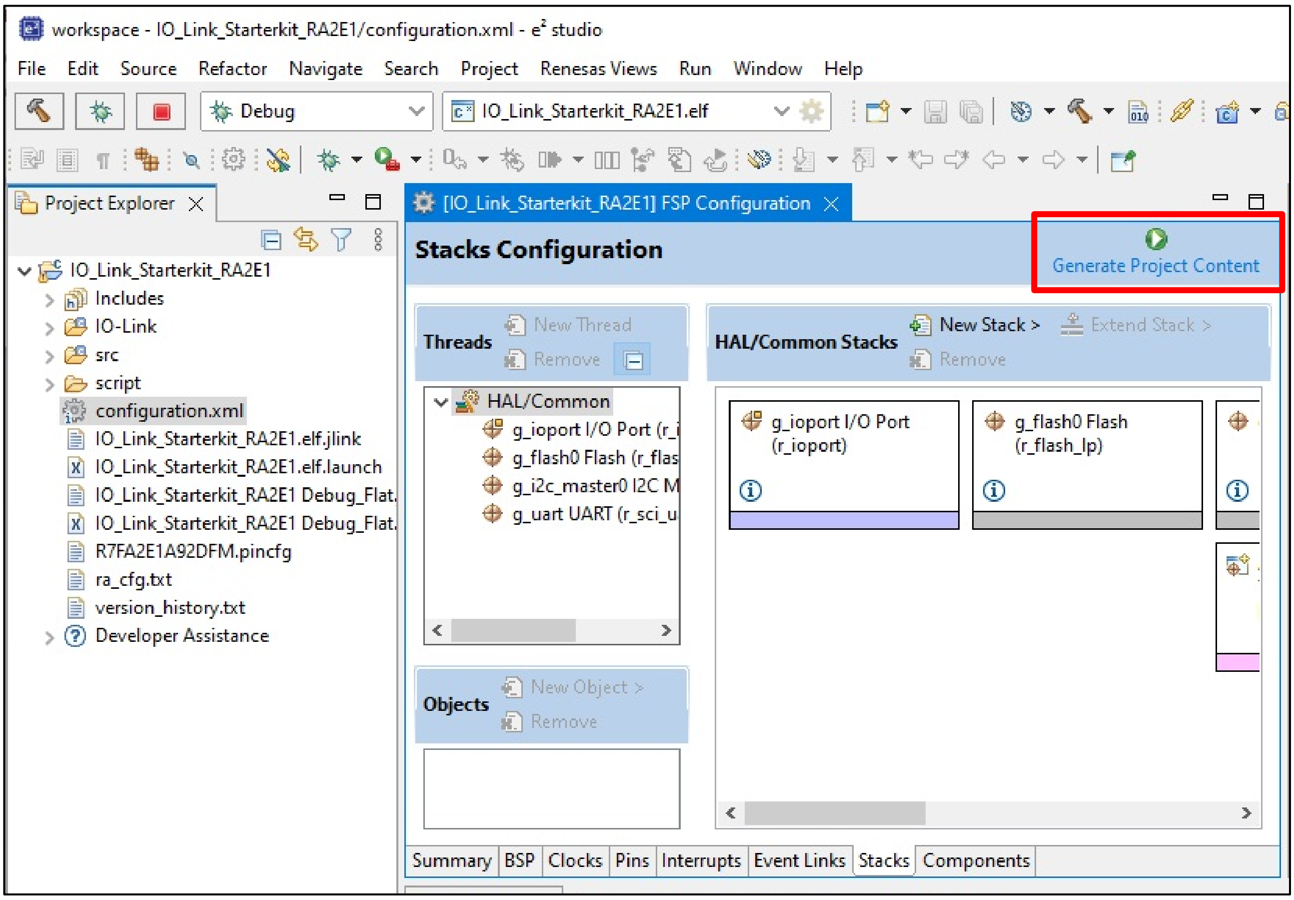


Figure 22. Generate Project Content

3. Click the Build icon to build the project.

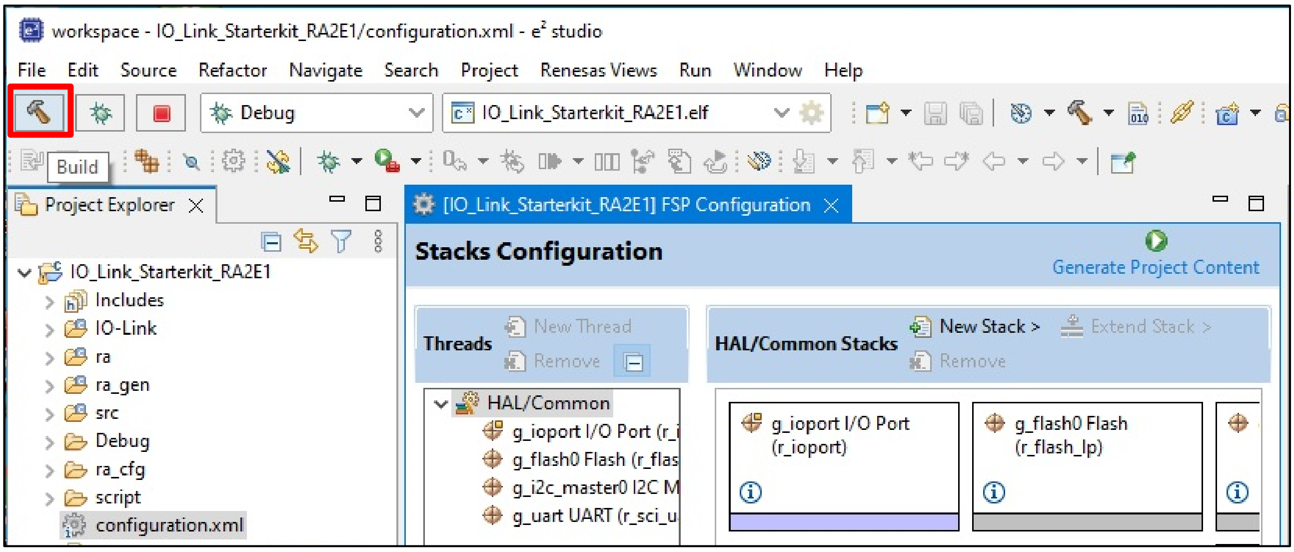


Figure 23. Build the Project

4. When the build finishes successfully, the following output is generated.

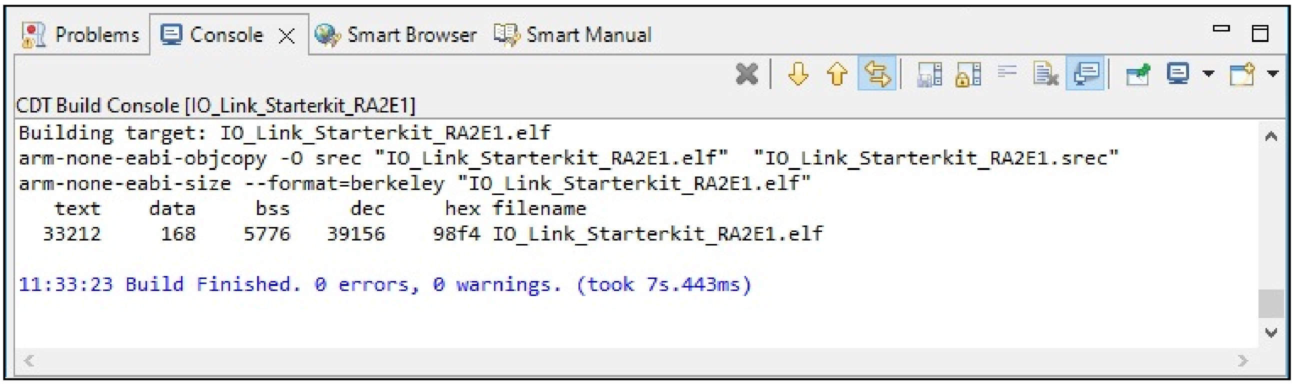


Figure 24. Output on a Successful Build

5.3 Setting up a Debug Connection between the EK-RA2E1 Board and the Host PC

The following graphic shows how to connect the EK-RA2E1 board to a host PC with a USB cable and download the program via the debug interface.

1. Connect the USB cable from the PC to the USB debug port (J10) on the EK-RA2E1 board.
2. Confirm that the debug LED (LED5) is lit orange.

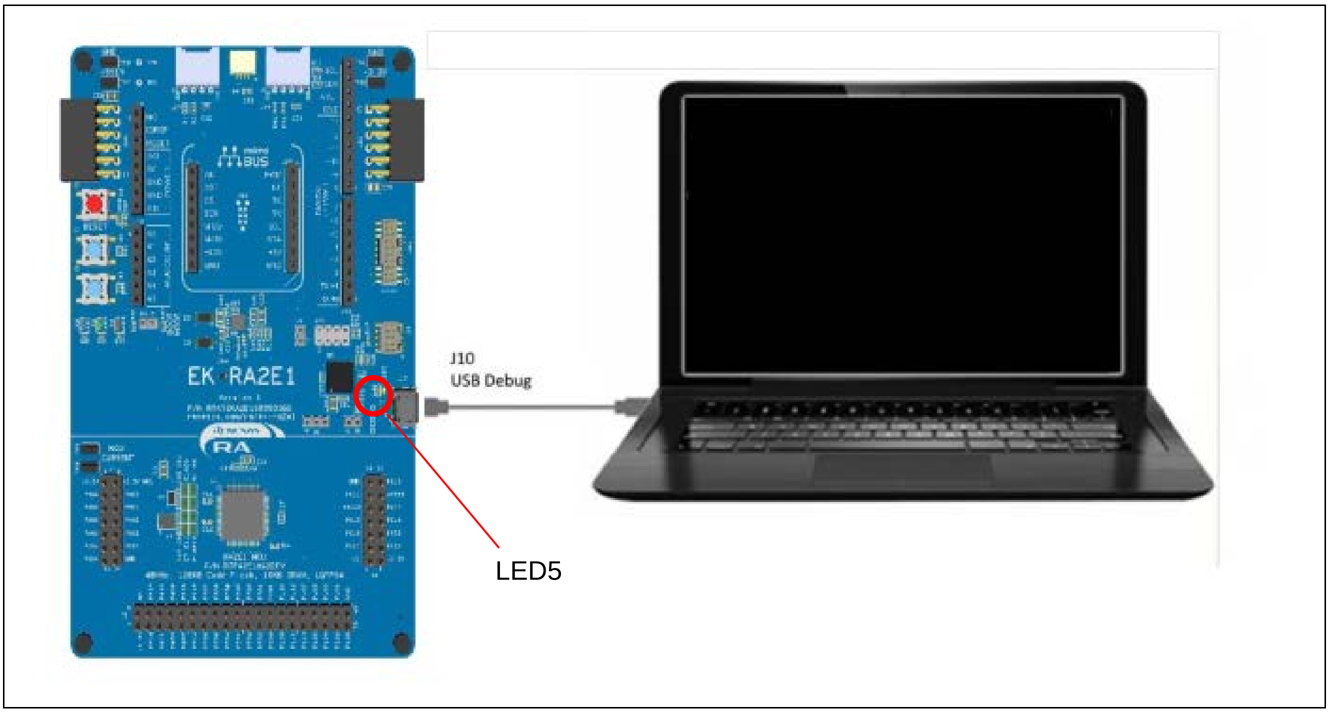


Figure 25. Connect the Board's USB Debug Port (J10) to the Host PC

5.4 Writing Sample Projects to the MCU

1. Click the Debug icon.

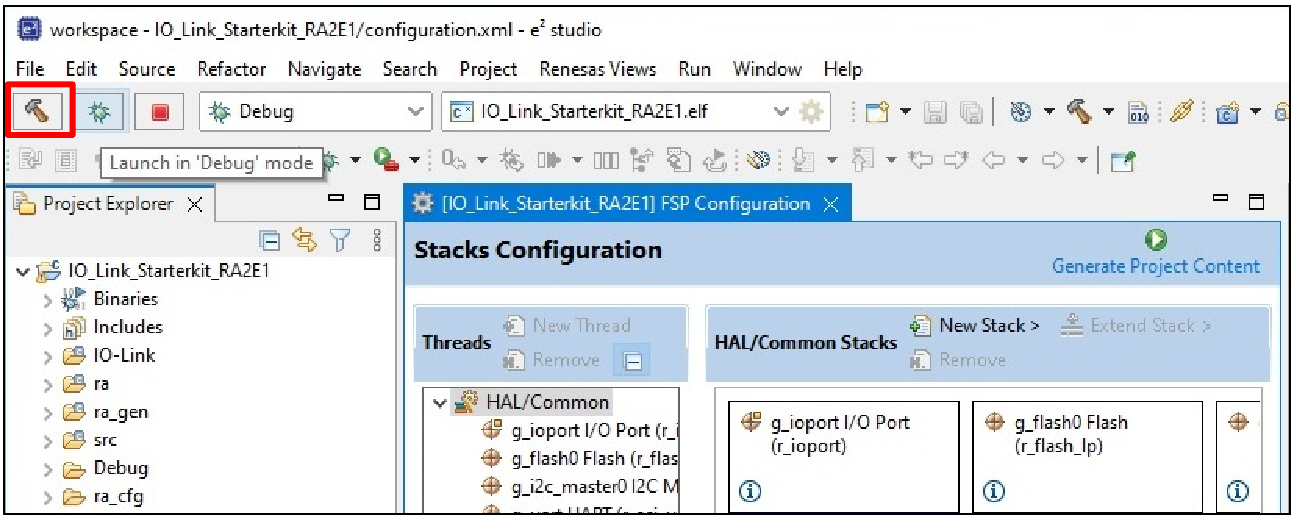


Figure 26. Launch in Debug Mode

2. You may receive a firewall warning for 'e2- server-gdb.exe'. Check the **Private networks, such as my home or work network** checkbox and click **Allow access**.

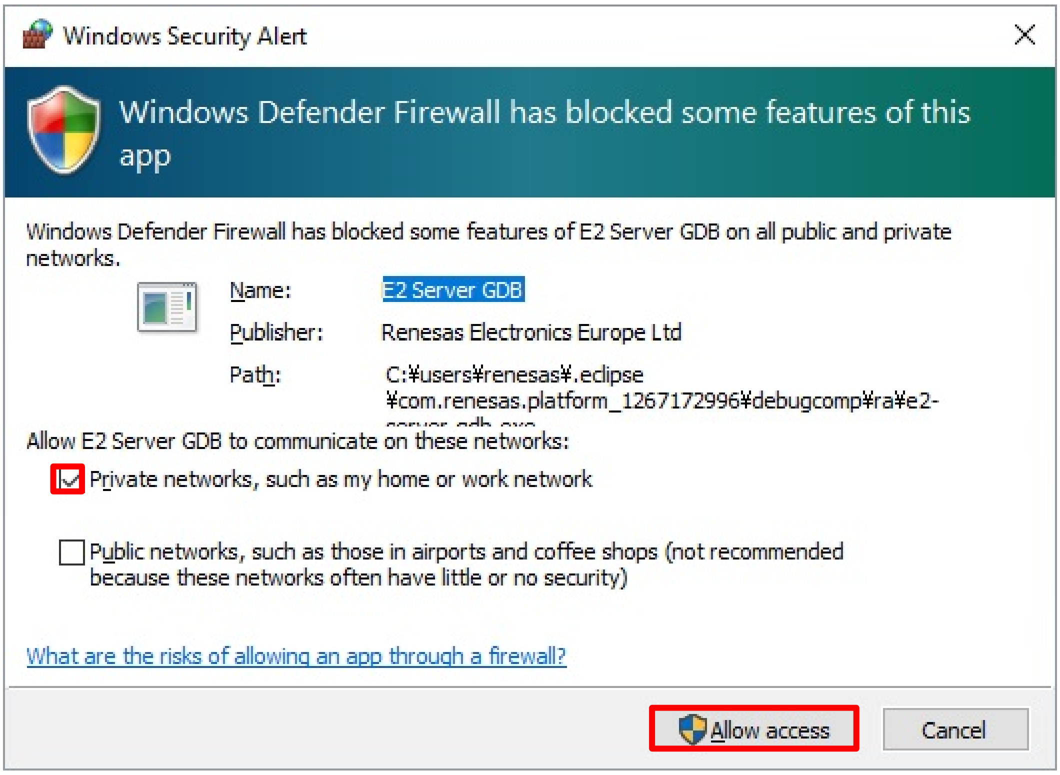


Figure 27. Firewall Warning

3. A dialog may appear prompting you to switch to the debug perspective. Click **Switch** to switch views.

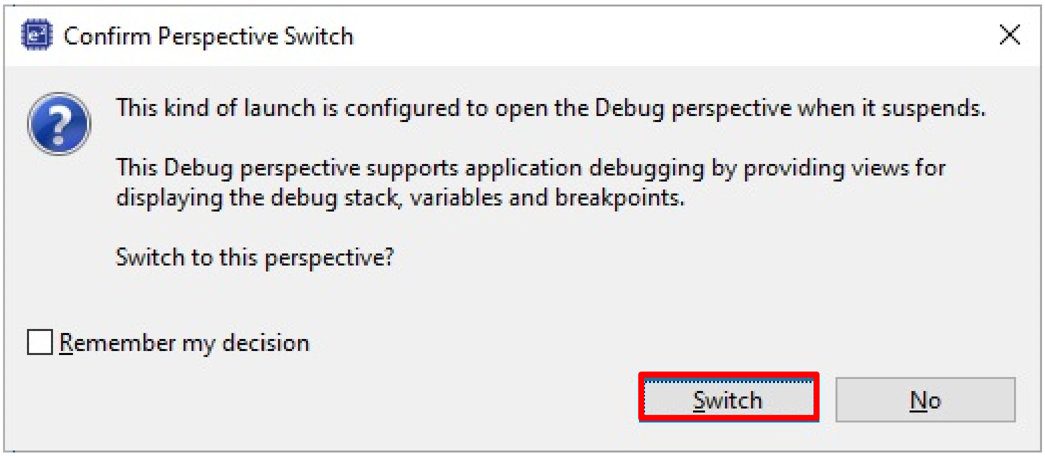


Figure 28. Confirm Perspective Switch

4. The sample project is written to the MCU and the screen changes as shown in Figure 29.

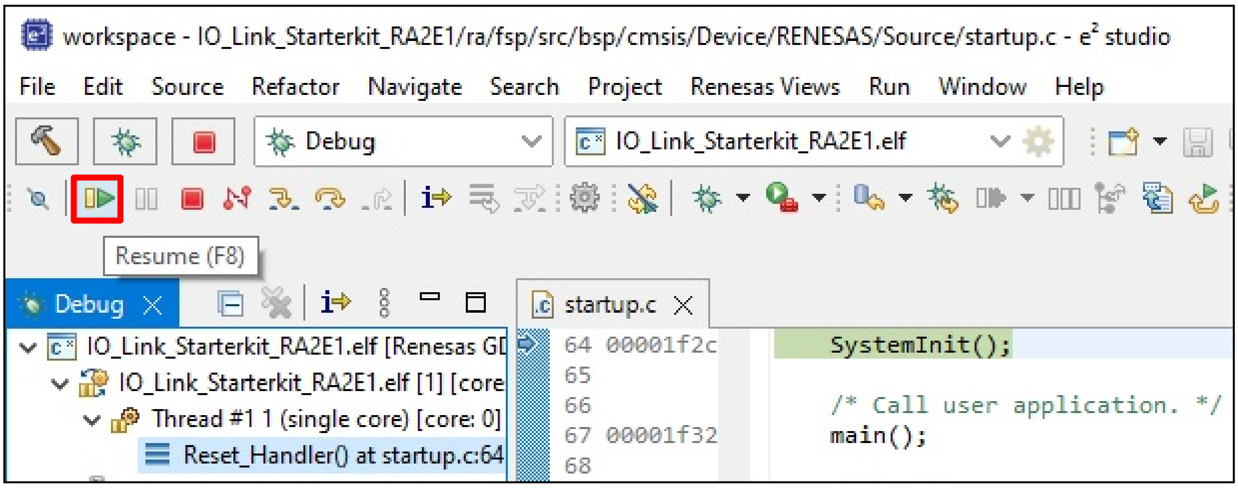


Figure 29. Writing to MCU Completed

5.5 Start the Program

1. In the state shown in Figure 29, click the F8 or Resume icon to start running the program.
2. Stop once at the beginning of the `main()` function. Click F8 or Resume icon again.
3. The program is now running and 'Running' is displayed in e^{2}studio status bar.

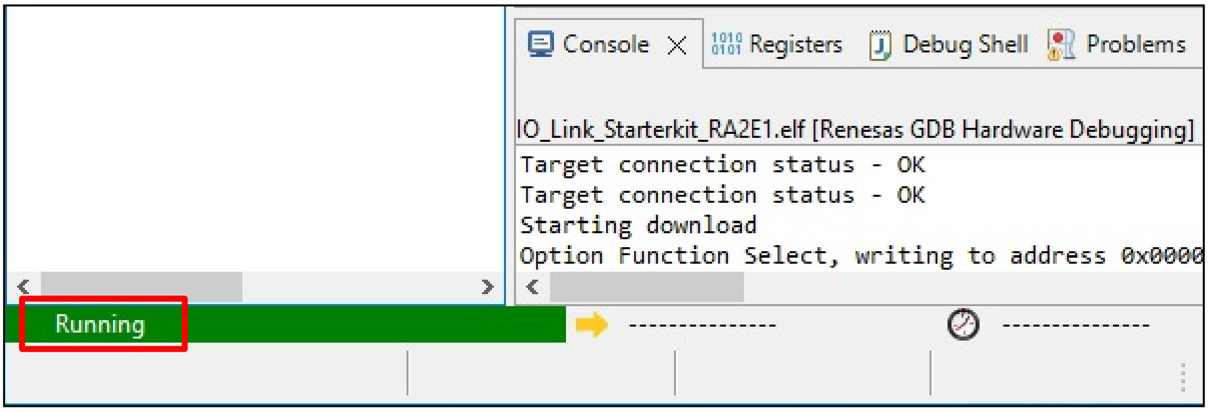


Figure 30. Start the Program

6. IO-Link Device Tool Usage and Functions

In this section, it is assumed that the TMG IO-Link Device Tool V5.1 – PE is installed on the Windows® PC, and the preparation for use and the explanation of the functions are described. Refer to section, 3 Hardware for details.

6.1 IO-Link Device Tool V5.1 - Start PE

The **Topology** pane shown in Figure 31 displays the topology from the PC to the IO-Link device.

The **Device Catalog** shows all devices that have been imported and are available for use.

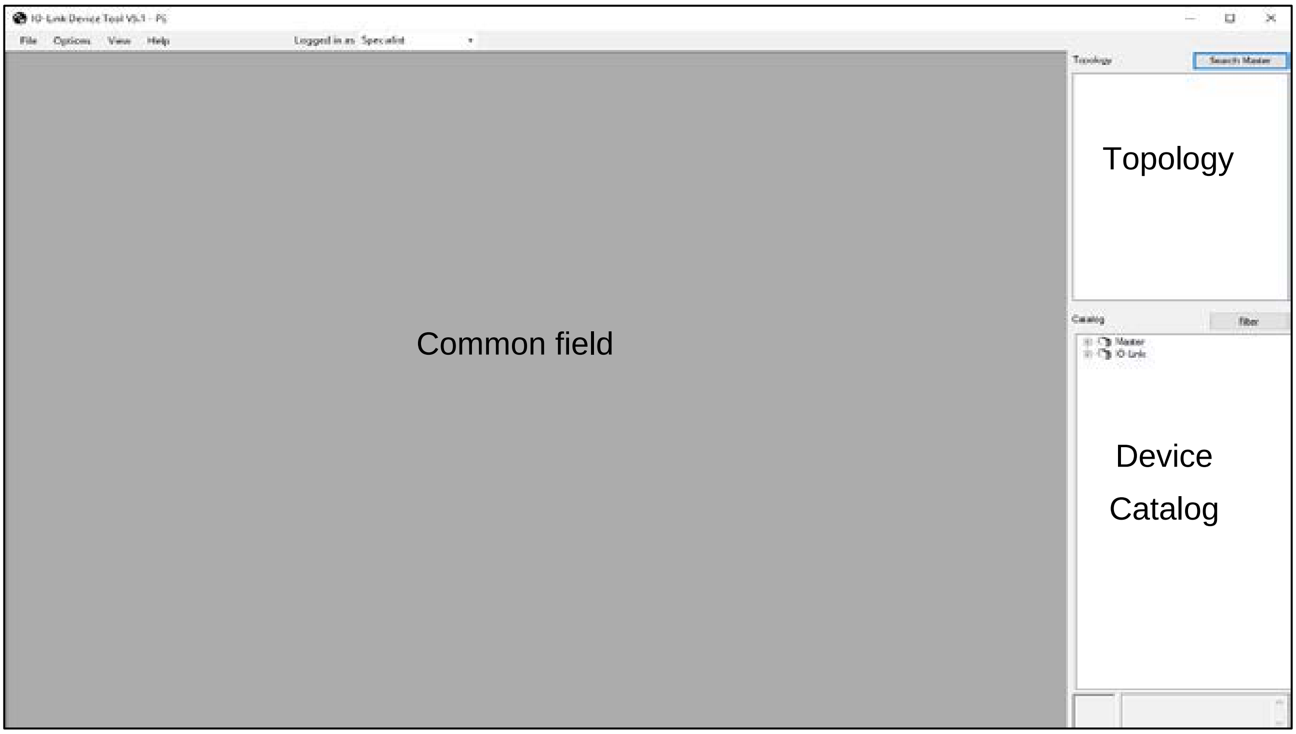


Figure 31. IO-Link Device Tool V5.1 – PE

6.2 Update IO-Link Device Catalog

1. Select **Options** from the menu bar, and then select "Import IODD (IO Device Description)".
2. Enter directly or browse to select the path of the folder where the IODD file is located. The IODD files used in this sample project are stored in the IODD folder of the sample project. By setting the path to this folder as shown in Figure 32, available IODD files can be automatically detected.
3. Check **TMG-RA2E1-Starterkit-20210715-IODD1.1.xml**.

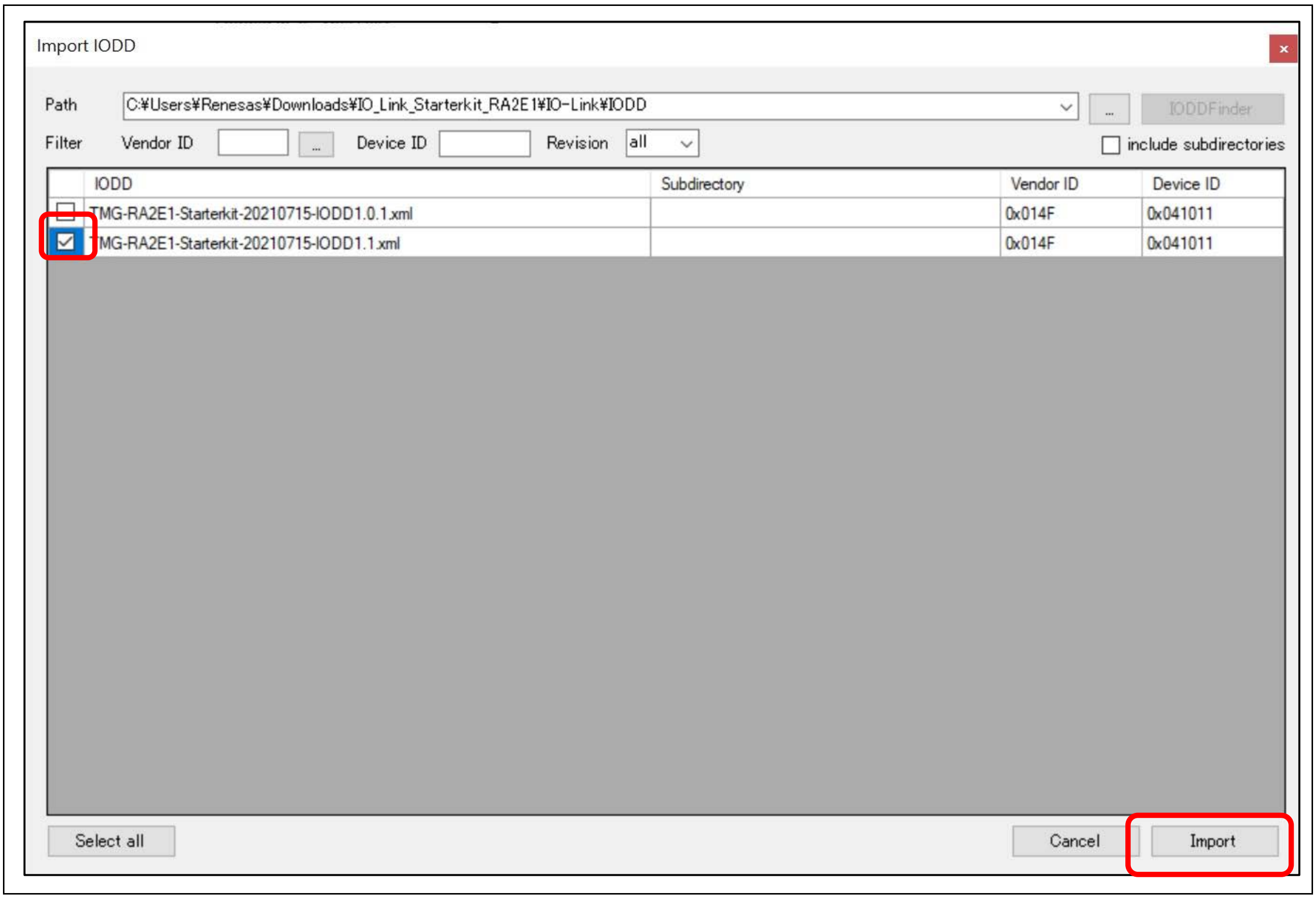


Figure 32. IO-Link Device Tool V5.1 – PE (Load the IODD file)

4. Select **Import**.

6.3 Update IO-Link Master Catalog

1. Select **Options** in the menu bar, and then select **Import IOLM(IO-Link Master Description)**.
2. When Figure 33 is displayed, select **TMG WEB**.

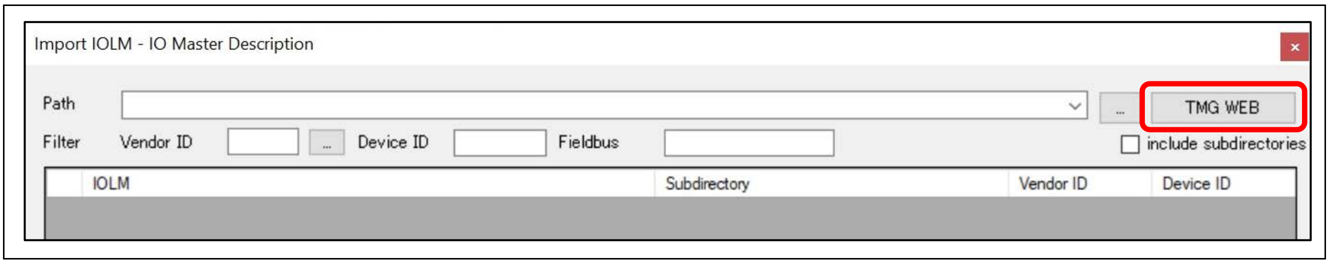


Figure 33. IO-Link Device Tool V5.1 – PE (Import IOLM)

3. When the screen changes and the list of IOLM is displayed, select "**...**" button of Vendor ID.

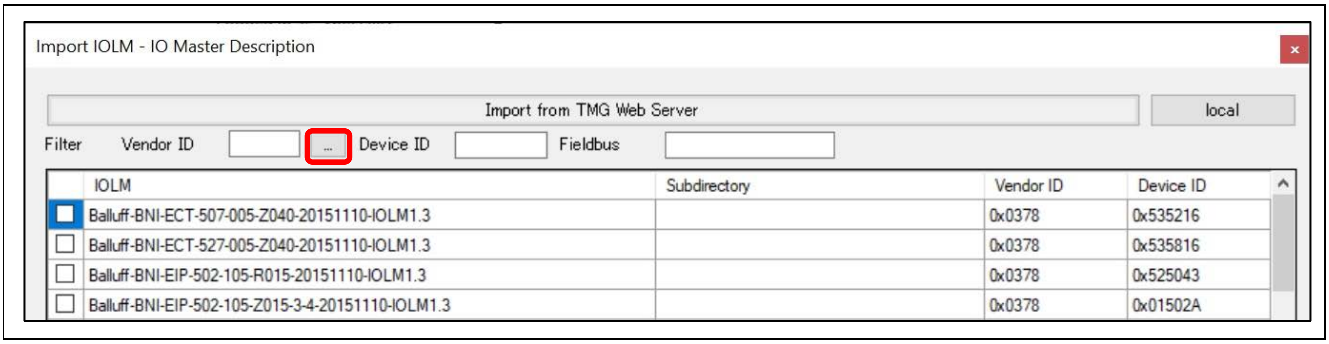


Figure 34. IO-Link Device Tool V5.1 – PE (Vendor ID)

4. When the Vendor ID Table appears, select “**Pepperl + Fuchs GmbH (0x0001)**".

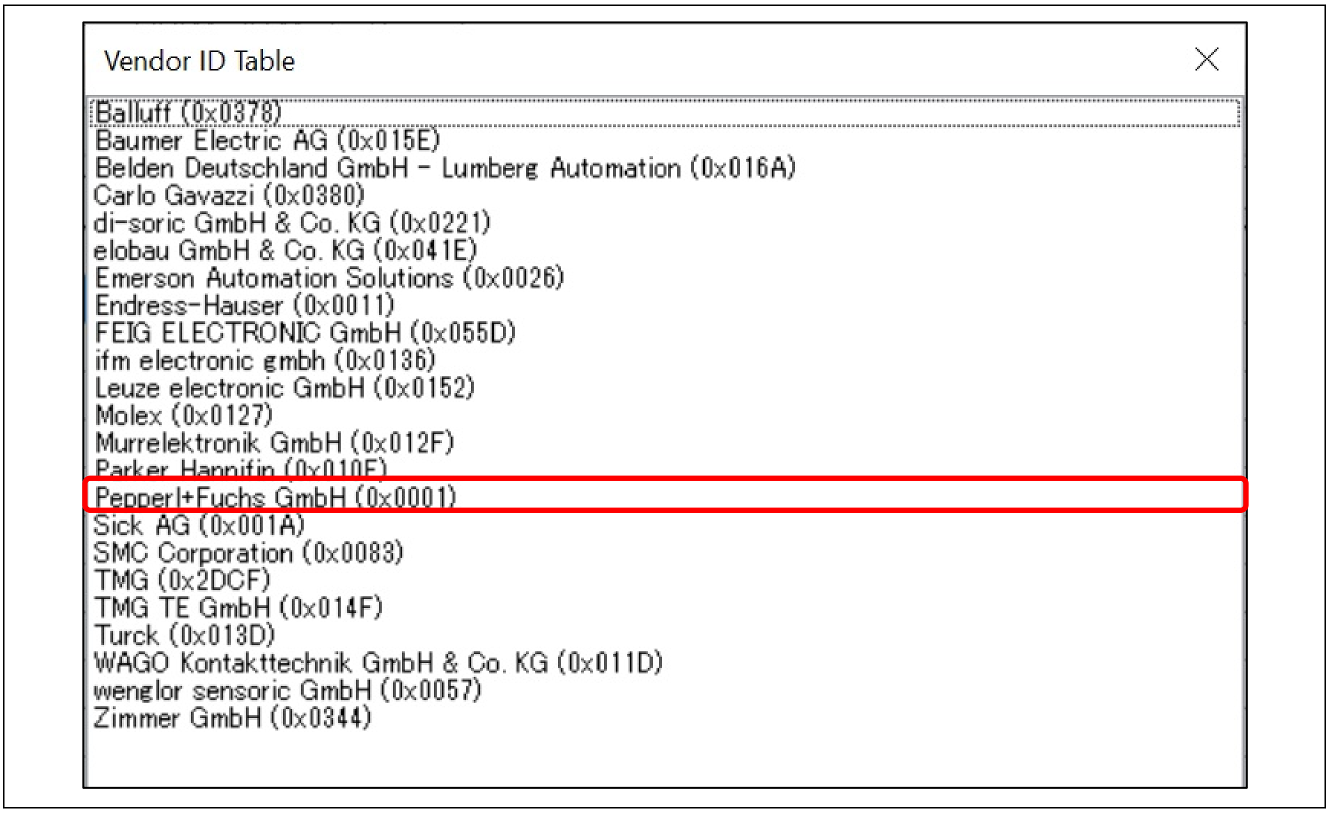


Figure 35. IO-Link Device Tool V5.1 – PE (Vendor ID Table)

4. The available IOLM files are detected.

5. Check "**Pepperl + Fuchs-IO-Link-Master02-USB-20200517-IOLM 1.4**".

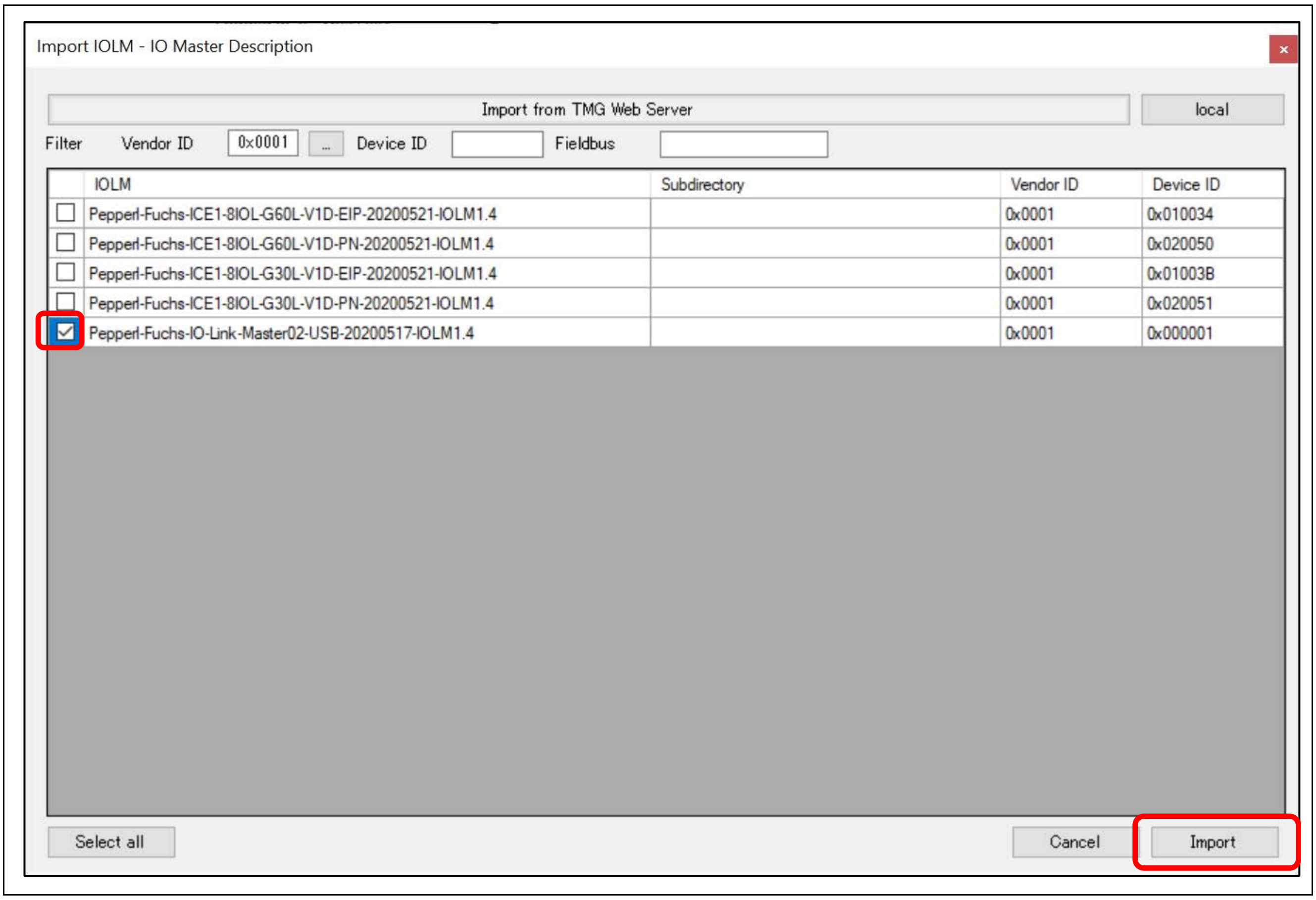


Figure 36. IO-Link Device Tool V5.1 – PE (Load the IOLM file)

6. Select **Import**.

6.4 Check for Catalog Updates

After successfully updating the Catalog in sections 6.2 and 6.3, the TMG TE GmbH vendor and the IO-Link device board with EK-RA2E1 will appear in the IO-Link Devices section of the Catalog as **RA2E1 Starterkit**. The IO-Link-Master02-USB appears under Master.

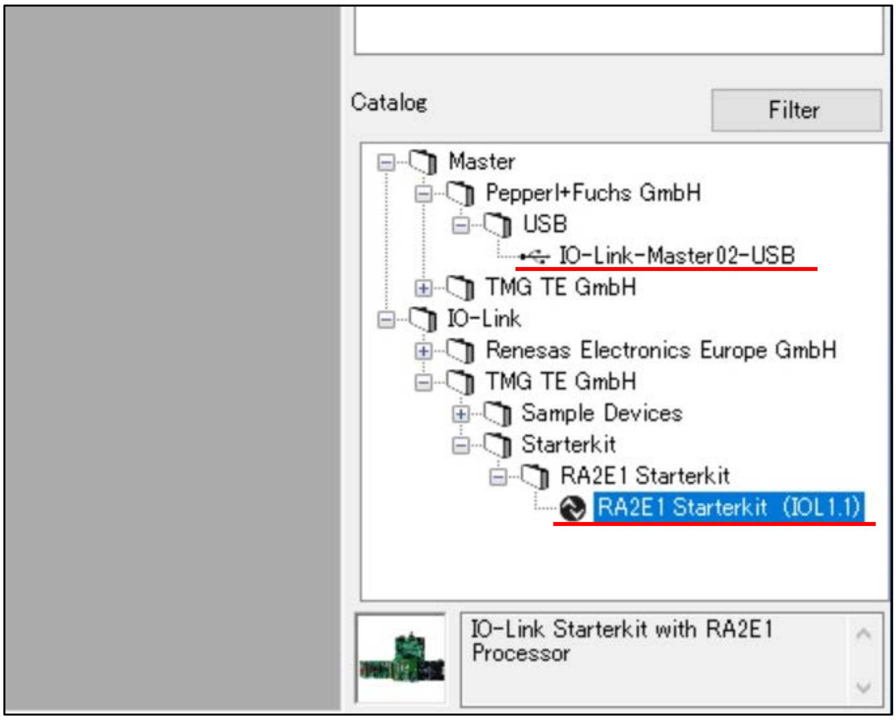


Figure 37. IO-Link Device Tool V5.1 – PE (Check for Catalog Updates)

6.5 Setup for IO-Link communication

1. Click the **Search Master** button in the upper right corner of the window. "**IO-Link-Master02-USB**" will appear in the **Master Discovery** window as shown in Figure 38.

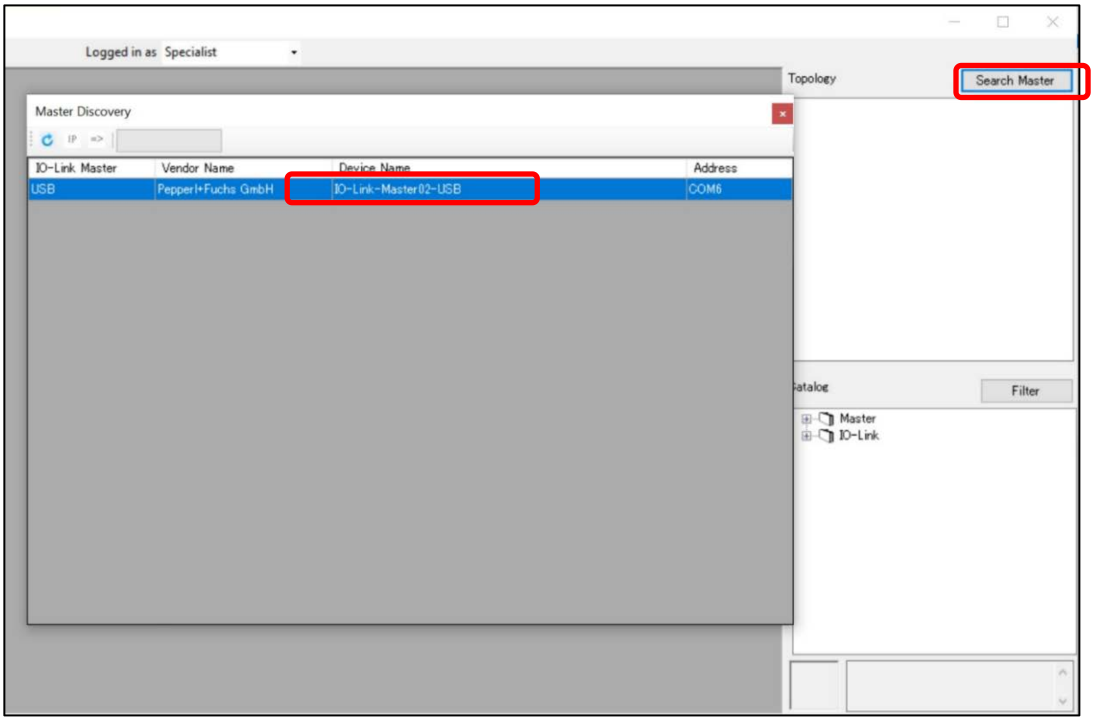


Figure 38. IO-Link Device Tool V5.1 – PE (Search for IO-Link Master)

2. Double-click on the device name displayed in the **Master Discovery** window.
3. Click on the **Go Online** button to activate the connection between the Master and the device.

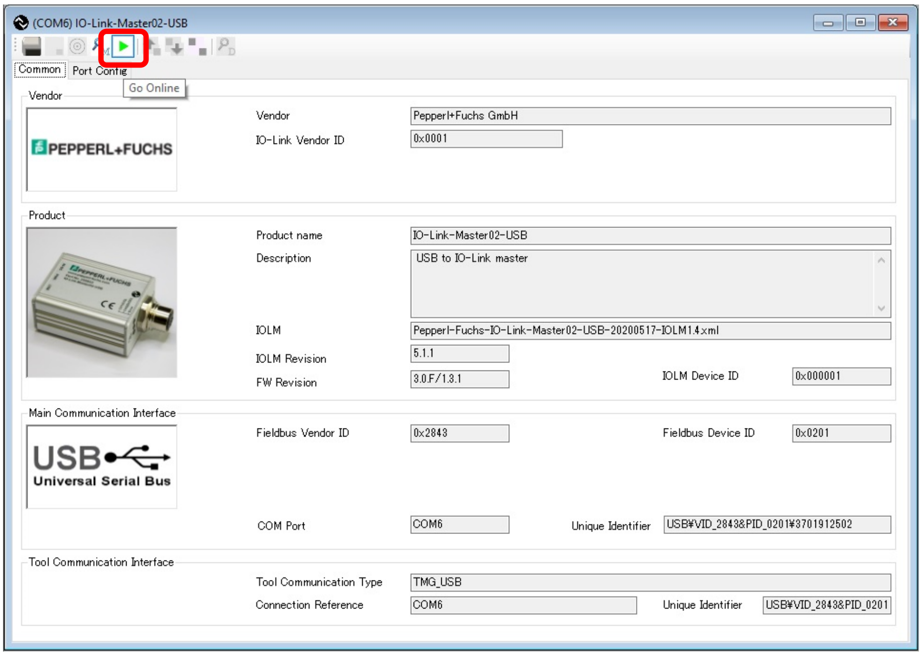


Figure 39. IO-Link Device Tool V5.1 – PE (Set to Online State)

Once the connection between the Master and the device is active, the **Go Online** button will be replaced by a button with a red circle and the **Check Devices** button will be enabled.

4. Click on the **Check Devices** button to detect the connected devices.

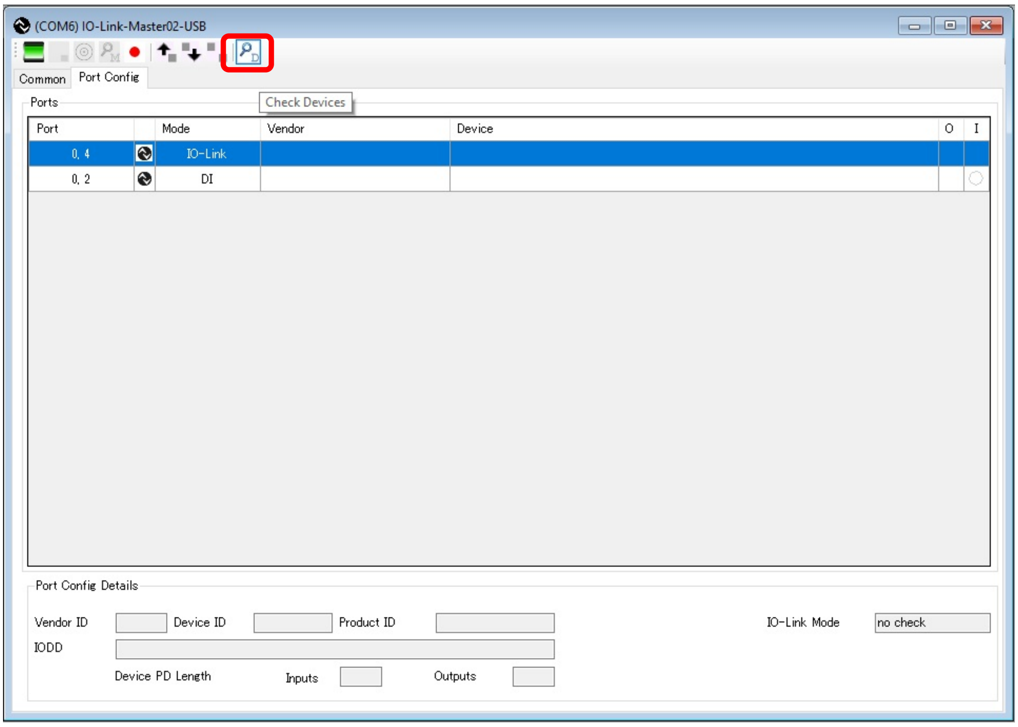


Figure 40. IO-Link Device Tool V5.1 – PE (Check the IO-Link Device)

When the detection is complete, the **Check Devices** window will appear and show the RA2E1 Starterkit connected to the IO-Link Master port.

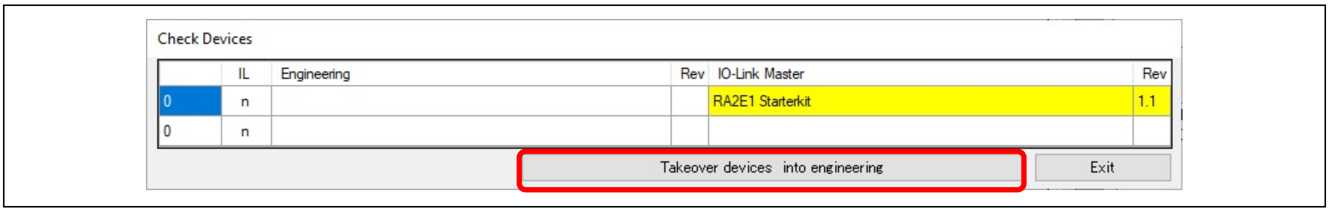


Figure 41. IO-Link Device Tool V5.1 – PE (Detect IO-Link Device)

5. Click on the **Takeover devices to engineering** button.
   When the connection between the master and the device is successful, **RA2E1 Starterkit** is added to the list of devices connected to the IO-Link Master as shown in Figure 42.

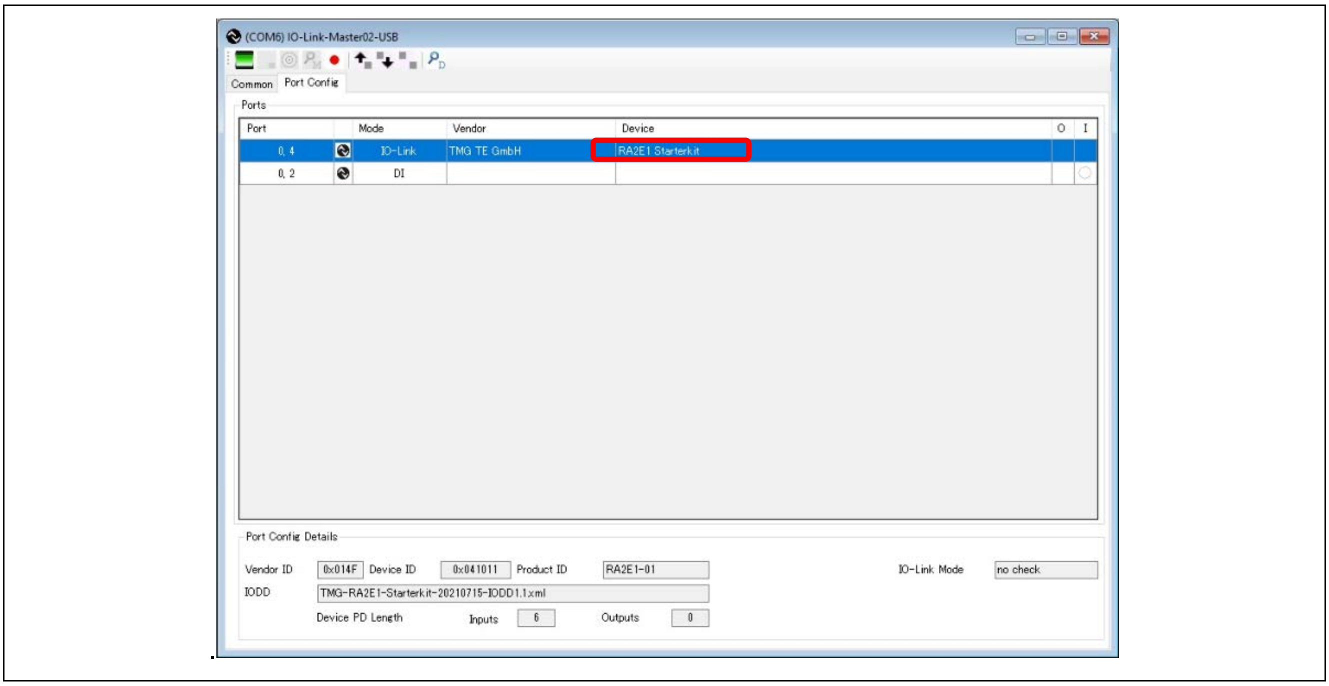


Figure 42. IO-Link Device Tool V5.1 – PE (Connect IO-Link Master and Device Successfully)

6. Double click on **RA2E1 Starterkit** and **Common** will appear.

6.6 EK-RA2E1 Board Sensor Demo of IO-Link Device Tool

This section explains how to operate IO-Link devices using the EK-RA2E1 board from the IO-Link Device Tool. The device operation toolbar is located in the upper left corner of Figure 43.

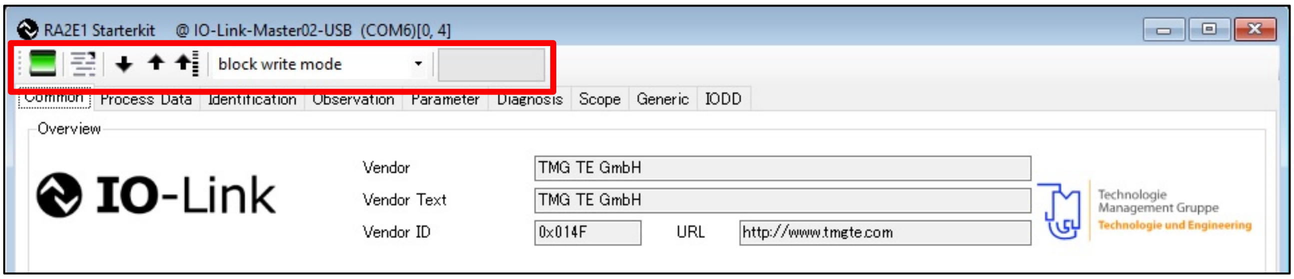
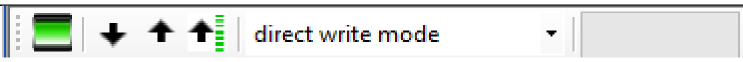


Figure 43. Sensor's Description Page

Figure 44 shows the buttons on the device operation toolbar and their functions.

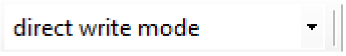



: System Operation Status


: Download the variable changes to the device (Download to device)


: Upload variable information from device (Upload from device)


: Enable dynamic updating of variables



: direct write mode -> Download changed variables to the device automatically

block write mode -> Download changed variables to the device manually

Figure 44. IO-Link Device Tool V5.1 – PE (Device Operation Toolbar)

Note: If the communication status of the master is not possible, the **System Operation Status** icon may be displayed in gray. In this case, reconnecting power to the master board and restart the device tool to improve the situation.

6.6.1 Common Tab

The Common tab displays general descriptive information about the device, as follows.

Vendor: TMG TE GmbH

Vendor Text: TMG TE GmbH

Vendor ID: 0x014F

URL: www.tmgte.com

Device: RA2E1 Starterkit

Description: IO-Link Starterkit with RA2E1 Processor

Device ID: 0x041011

IO-Link Revision: 1.1

SIO mode: No

Communication baud rate: COM2

Minimum cycle time of sensor: 6400[us]

Photo of IO-Link device

Connection description

Pin layout of M12 connector

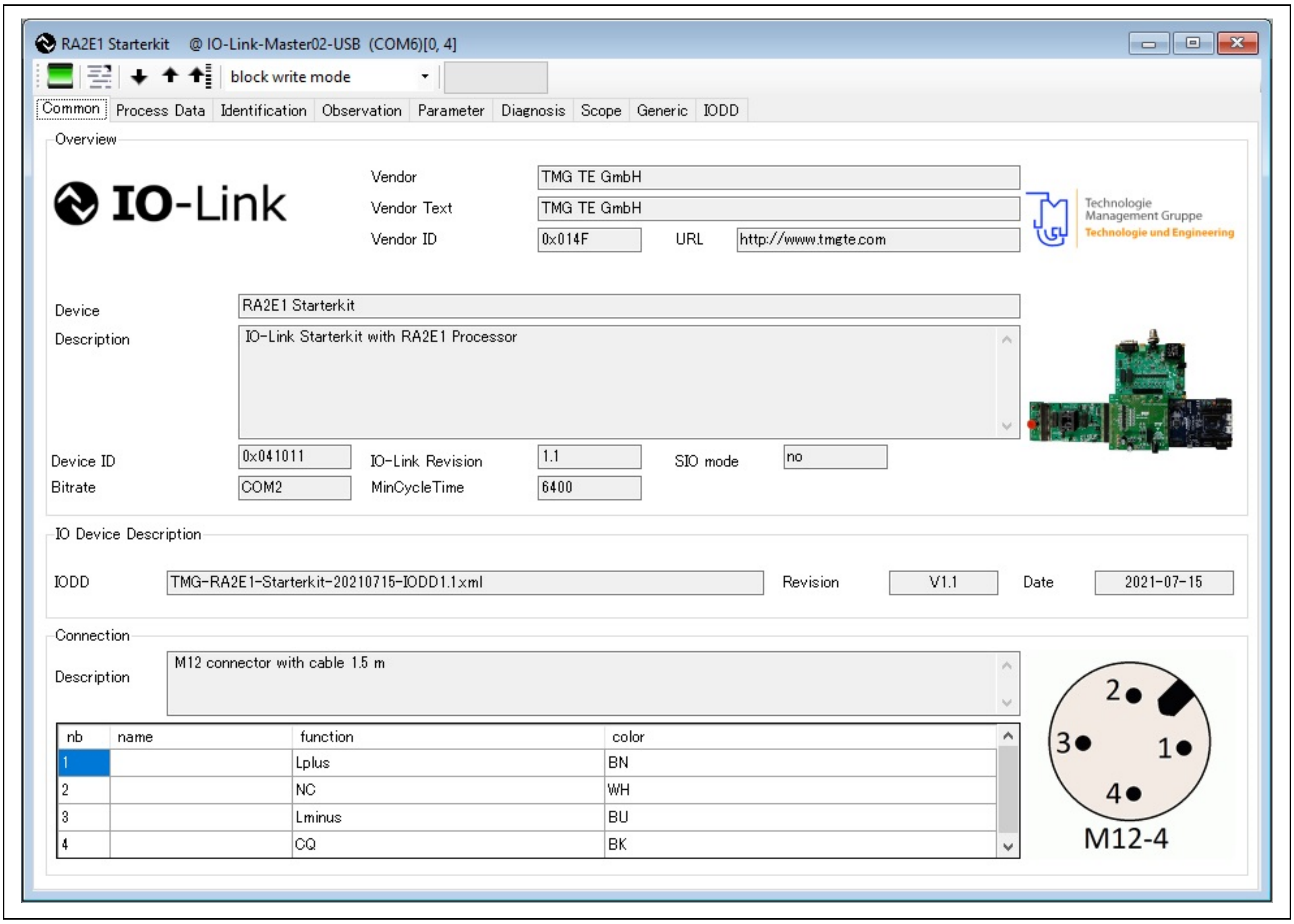


Figure 45. IO-Link Device Tool V5.1 – PE (Common tab)

6.6.2 Process Data tab

The **Process Data** tab displays information on PD Input.

1. PD Input
   - MDC – Measurement Value
     Sensor measurement value
   - MDC – Scale
     Unsupported
   - SSC.1 Switching Signal*
     Switching state (Low or High)
   - SSC.2 Switching Signal
     Unsupported

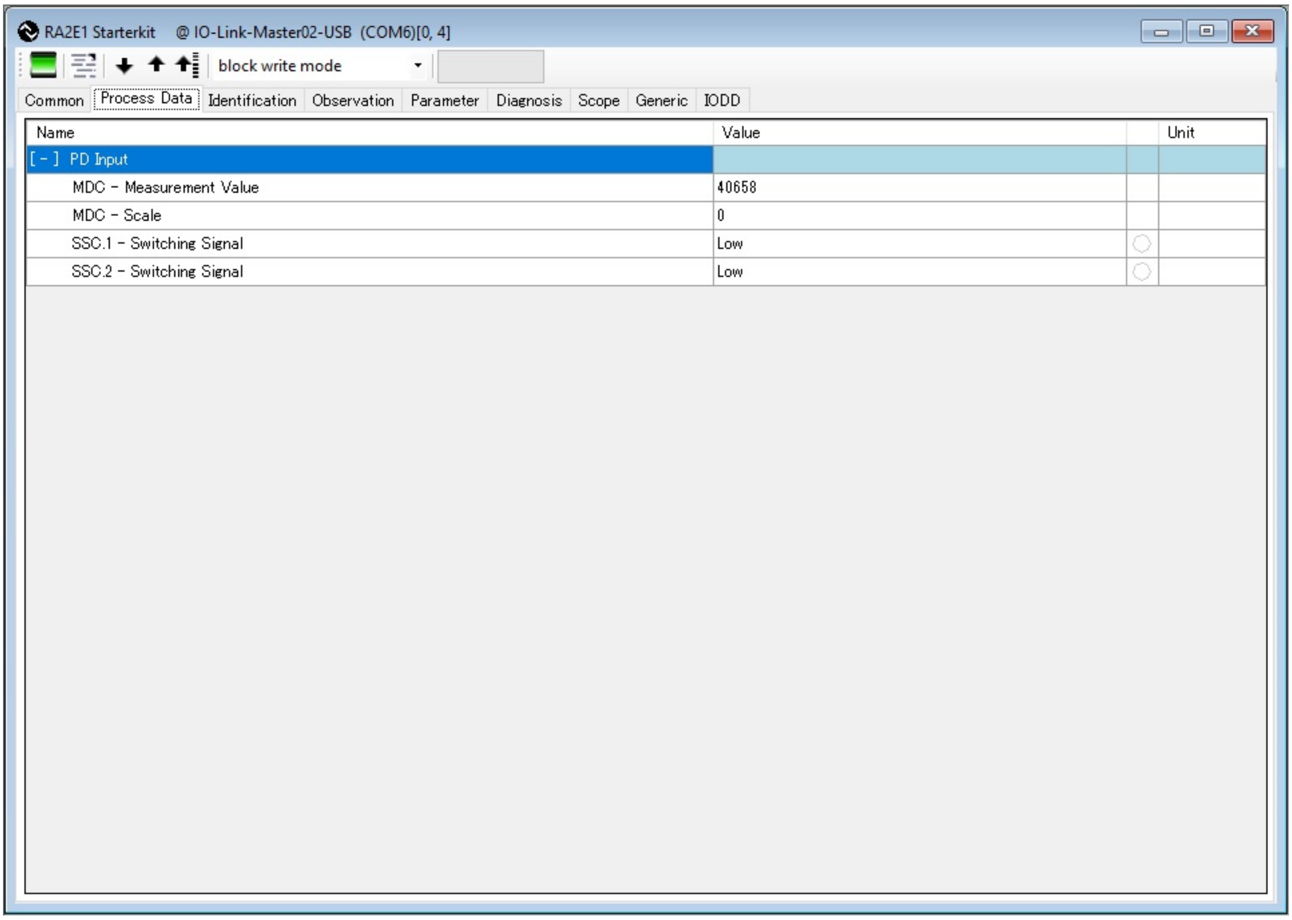


Figure 46. IO-Link Device Tool V5.1 – PE (Process Data Tab)

Note*: SSC.1 - Switching Signal works only with the green LED. Red LED is not supported.

6.6.3 Identification Tab

The Identification tab allows the user to read and review the identification information stored on the device, such as.

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

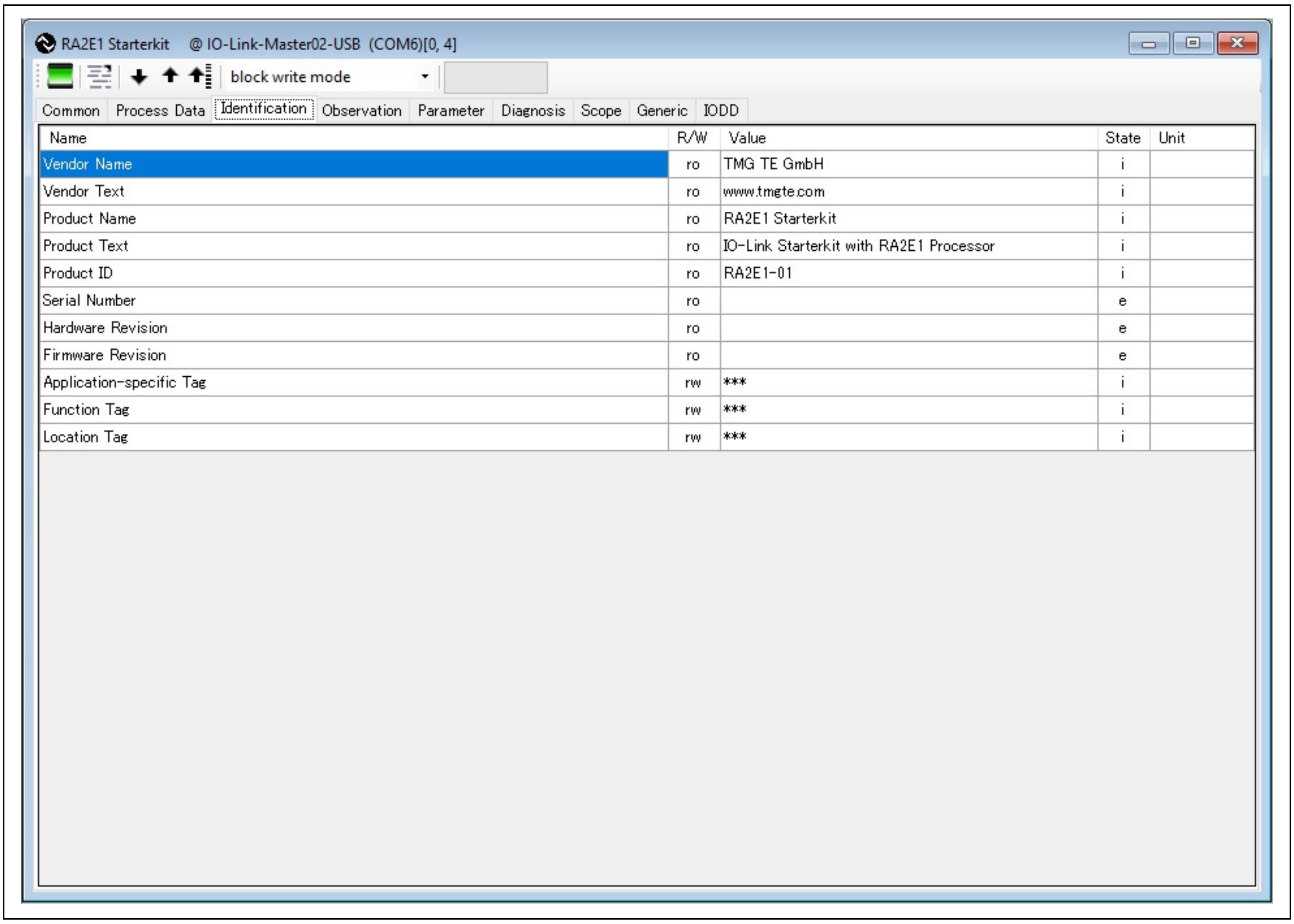


Figure 47. IO-Link Device Tool V5.1 – PE (Identification tab)

6.6.4 Observation Tab

The **Observation** tab displays the measurements taken by the device. The user can review the information measured by the device.

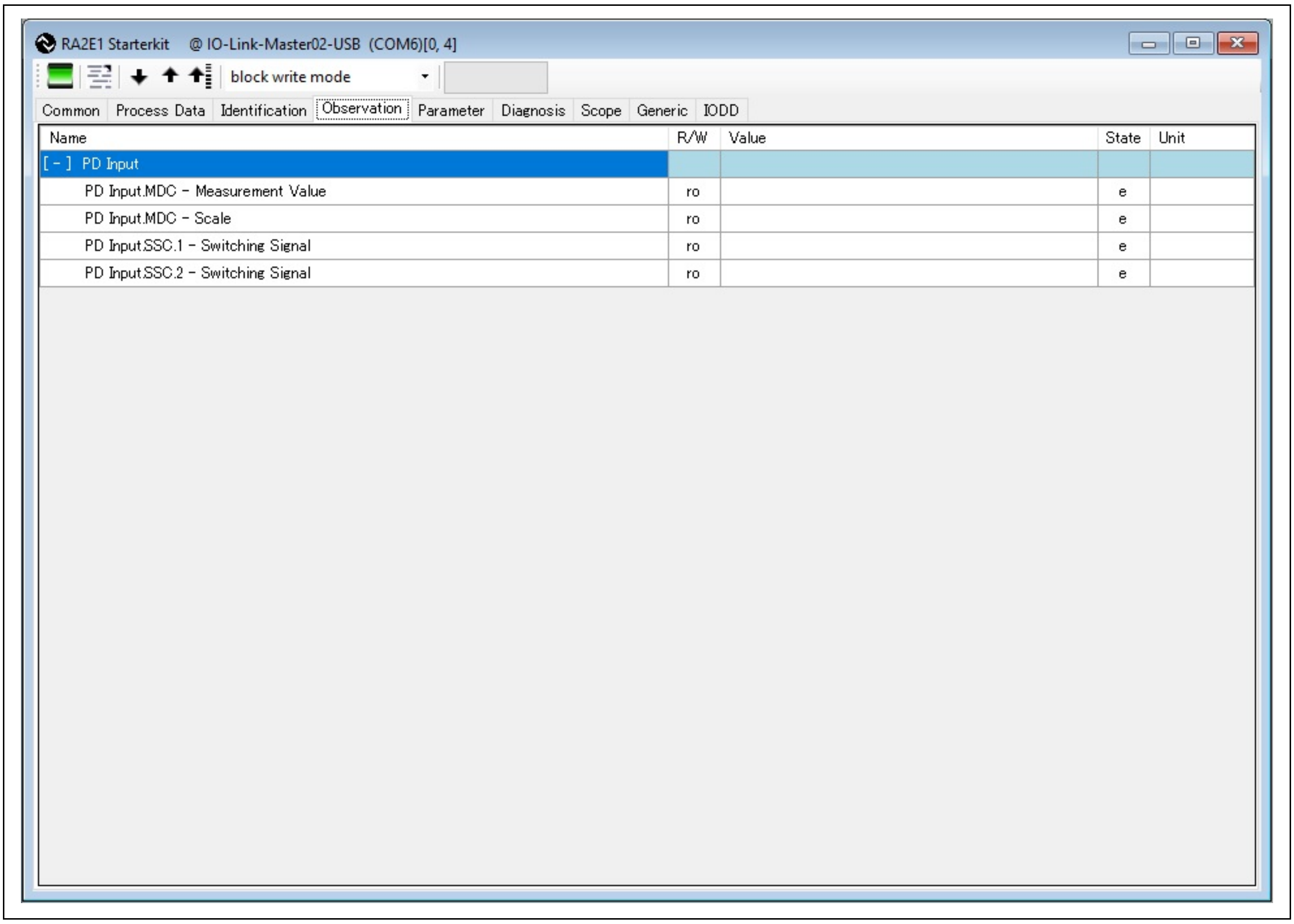


Figure 48. IO-Link Device Tool V5.1 – PE (Observation tab)

6.6.5 Parameter Tab

The **Parameter** tab displays the parameter setting status of the device and allows the user to check the status of the device from here. It also allows the user to write new settings to the device. Please refer to Table 4 for the meaning and features of the parameters. For details on how to set parameters of the device, please refer to section, 6.6.7 Change Device Parameters (Teach-In/Read).

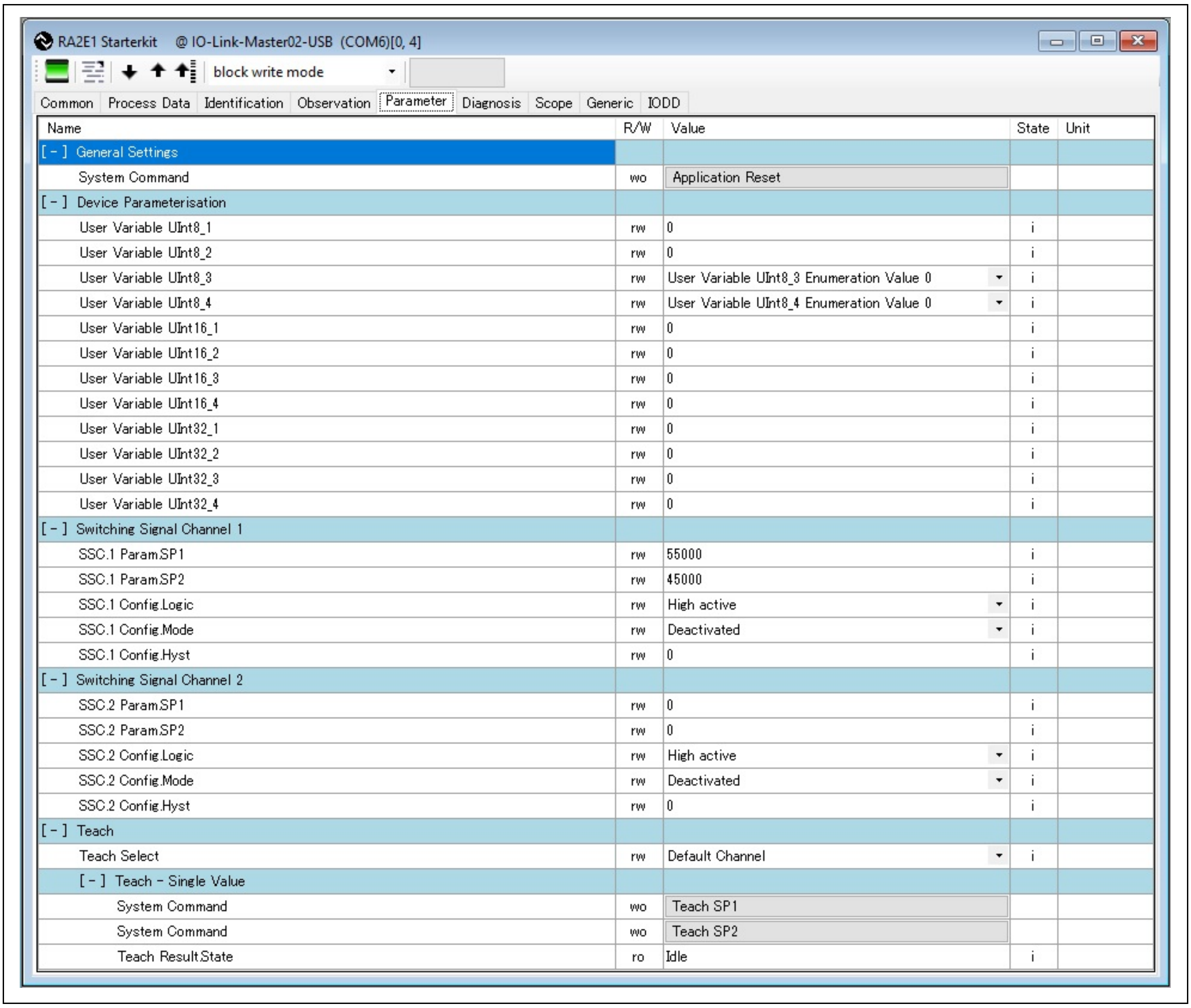


Figure 49. IO-Link Device Tool V5.1 – PE (Parameter tab)

6.6.6 Scope tab

The **Scope** tab allows the user to visualize the processed data.

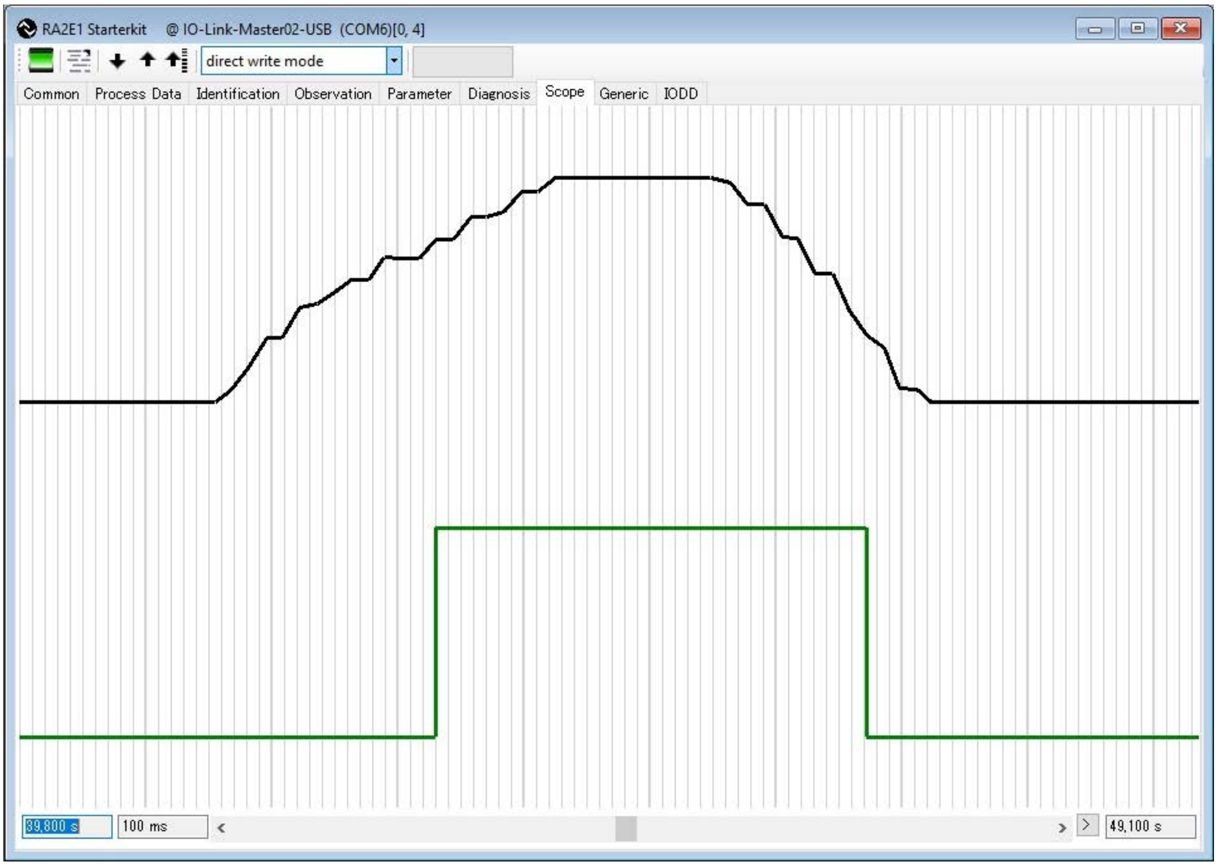
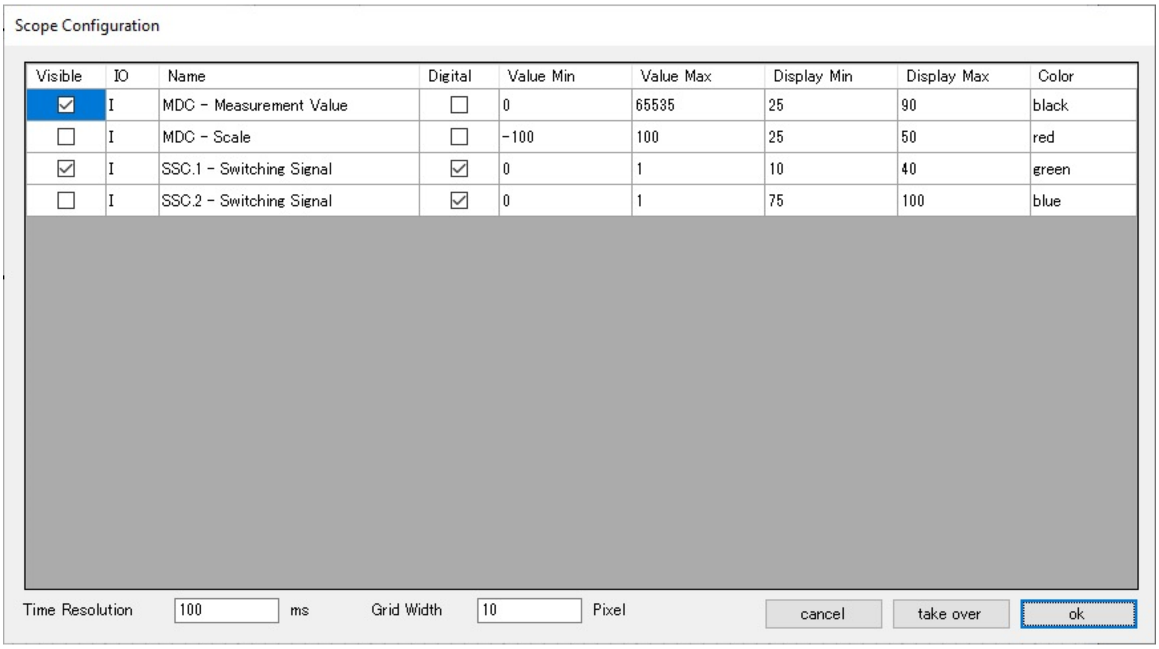


Figure 50. IO-Link Device Tool V5.1 – PE (Scope tab)

The user can open the Scope configuration settings by right-clicking in the Scope area. The settings can also be edited.



Scope Configuration

Visible	IO	Name	Digital	Value Min	Value Max	Display Min	Display Max	Color
☑	I	MDC - Measurement Value	☐	0	65535	25	90	black
☐	I	MDC - Scale	☐	-100	100	25	50	red
☑	I	SSC.1 - Switching Signal	☑	0	1	10	40	green
☐	I	SSC.2 - Switching Signal	☑	0	1	75	100	blue

Time Resolution 100 ms Grid Width 10 Pixel cancel take over ok

Table 8. IO-Link Device Tool V5.1 – PE (Scope tab, Scale/Parameter Settings)

6.6.7 Change Device Parameters (Teach-In/Read)

When the user opens the **Parameter** tab, the device-specific parameters are displayed in the Value column. These values are recorded in the IODD file. For details on the **Parameter** tab, refer to section, 6.6.5 Parameter Tab.

When setting SetPoint (SP1, SP2, so forth) to a device, make sure "Device Access Locks" is set to false; if it is true, setting the parameter to the device will fail. The SetPoint parameter can be set using the "Teach Values" parameter or by using the "Standard Command".

Users can change parameter settings with the IO-Link Device Tool. The settings depend on the write mode. The method is shown below.

6.6.7.1 Set parameters in Teach Values (block write mode)

In this block write mode, the user can change the parameters and then click the **Download to device** button to set the parameters to the device.

1. Change the write mode to **block write mode**.
2. To set the SetPoint, click on the **Value** field of the parameter **SSC.1 Param.SP1** or **SSC.1 Param.SP2**.
3. Enter the values and press **Enter** on the keyboard. The yellow background color of **Statue** indicates that the parameter has not yet been set in the device.
4. To write the parameters, click the Download to device button.
5. The green background color of **State** indicates that the synchronization between the master and the device was successfully performed.

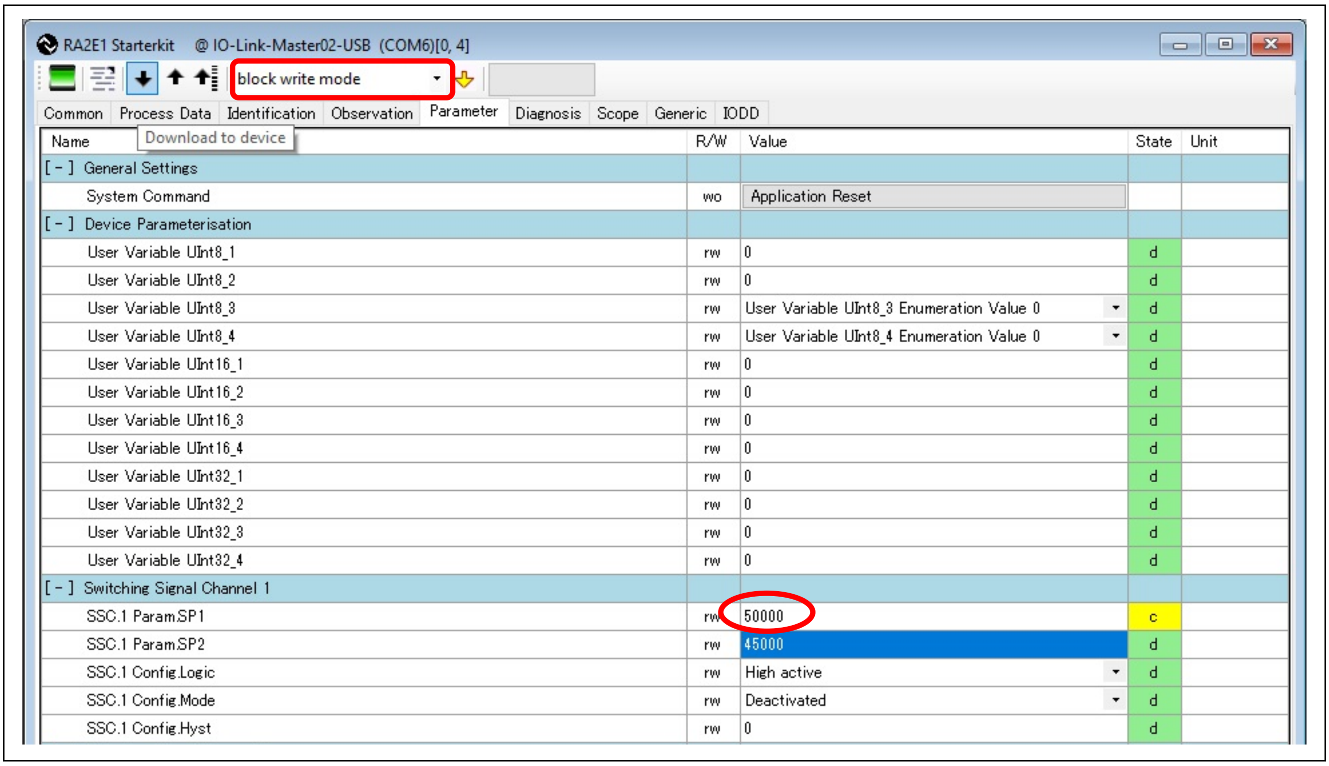


Figure 51. IO-Link Device Tool V5.1 – PE (Set Parameters in Block Write Mode)

6.6.7.2 Parameter Setting Using Direct Write Mode

In this direct write mode, parameter changes are automatically set to the device.

1. Change the write mode to **direct write mode**.
2. To set the SetPoint, click on the **Value** field of the parameter **SSC.1 Param.SP1** or **SSC.1 Param.SP2**.
3. Enter the values and press **Enter** on the keyboard.
4. The **State** will turn yellow and then green to indicate that the parameters have been set in the device and the master and device are synchronized.

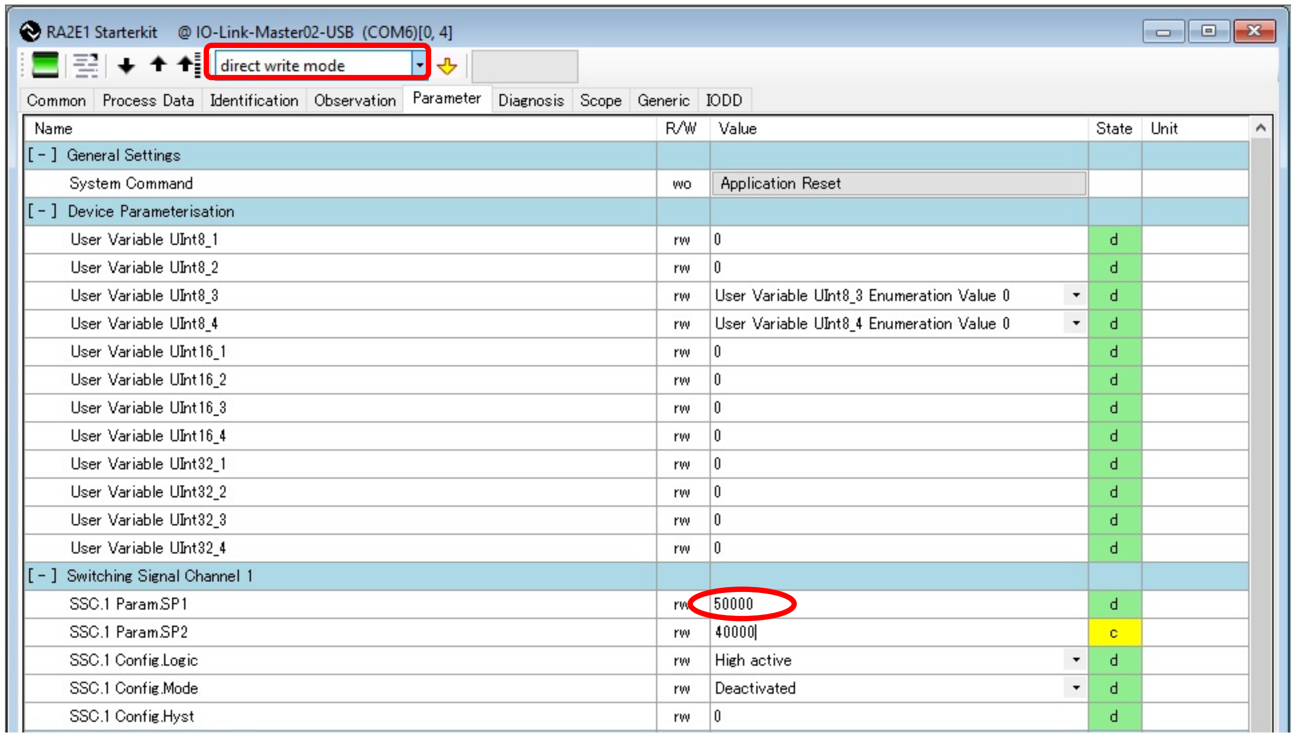


Figure 52. IO-Link Device Tool V5.1 – PE (Set Parameters in Direct Write Mode)

6.6.7.3 Change Operation mode

The operating mode is set to **Deactivated** as the initial value. To enable the switching state function using SetPoint, change the operation mode to **Window** or **Two point**.

For details on the operating modes, refer to section, 4.3 Operating Mode and Switching State.

1. Change the write mode to **block write mode**.
2. Select **Window** or **Two point** in the list of SSC.1 Config.Mode.
3. Click the Download to device button to set the device.

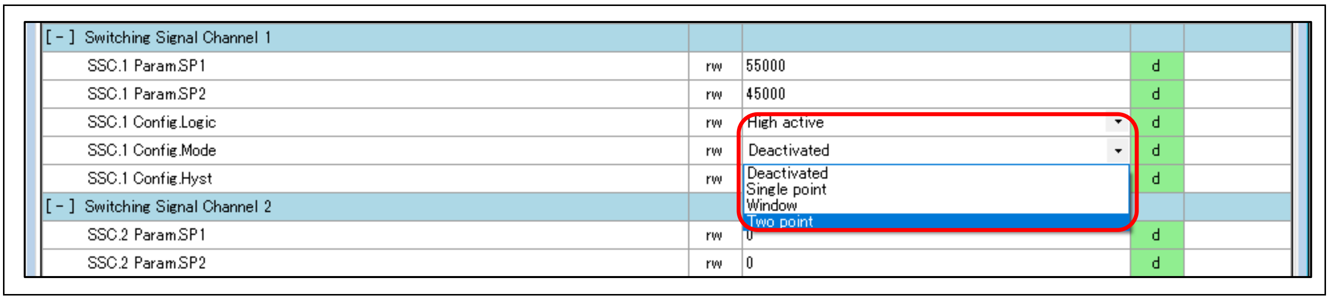


Figure 53. IO-Link Device Tool V5.1 – PE (Change Operation Mode)

Note: Single point is not supported in this sample application. Do not select it.

6.6.7.4 SetPoint settings by Device Measurement Value

Set the value measured by the device to SP1 and SP2 as SetPoint.

1. Click the **Teach SP1** button or the **Teach SP2** button in the Value field of the **Parameter** tab. The measurement value at that time will be automatically set to **SSC.1 Param.SP1** or **SSC.1 Param.SP2**, and the result of Teach Result State will be **SP1 Success** or **SP2 Success**.

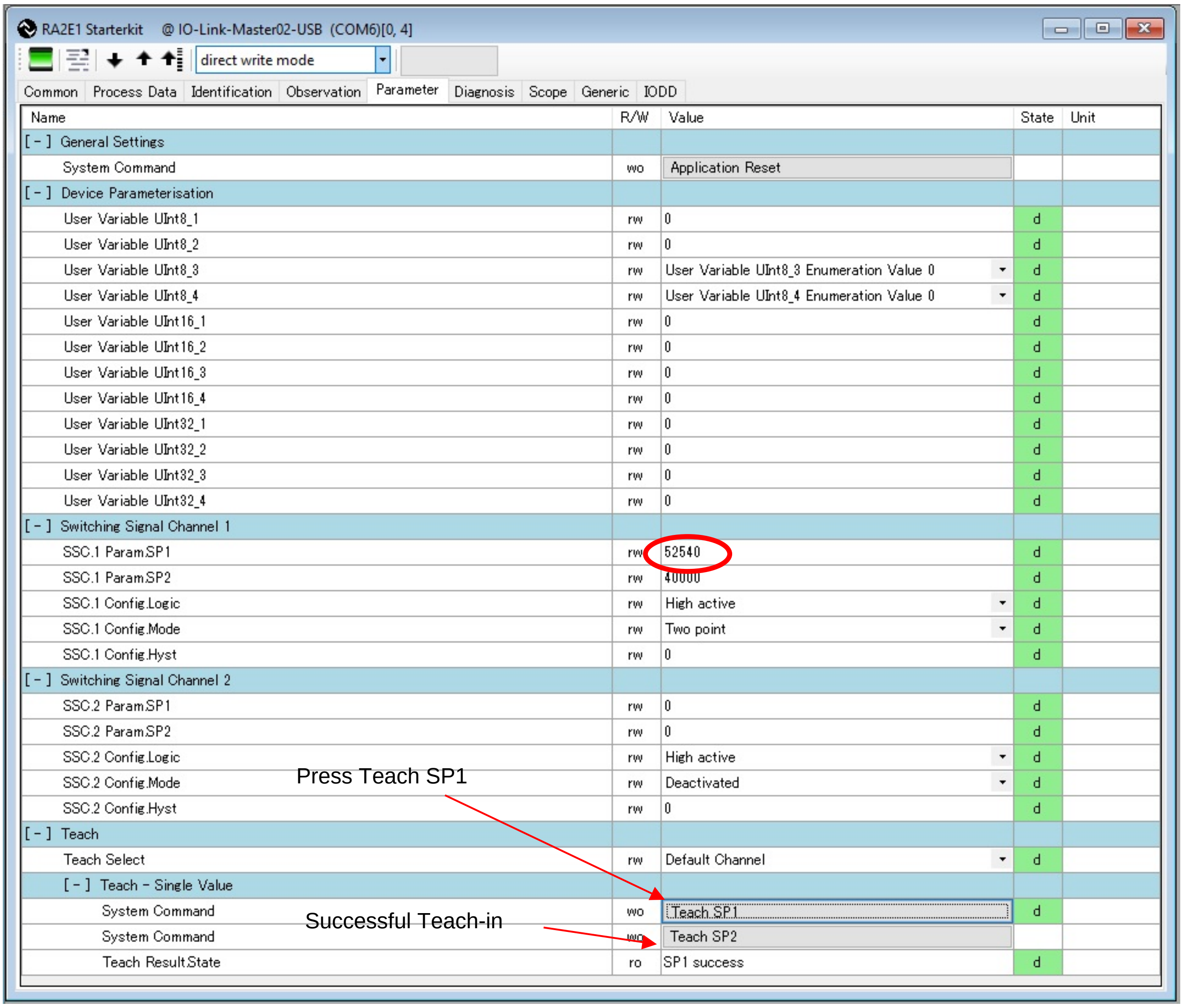


Figure 54. IO-Link Device Tool V5.1 – PE (SetPoint Settings by Device Measurement Value)

Note: Teach Select is not supported in this sample application. Do not change it.

6.6.7.5 Read Parameters

The user can click the Upload from device button to read the current parameters written to the device. For details about the Upload from device button, refer to section, 6.6 EK-RA2E1 Board Sensor Demo of IO-Link Device Tool.

6.6.7.6 Reset Parameters to Default Values

Reset the device settings to default values.

1. Click the **Application Reset** button.

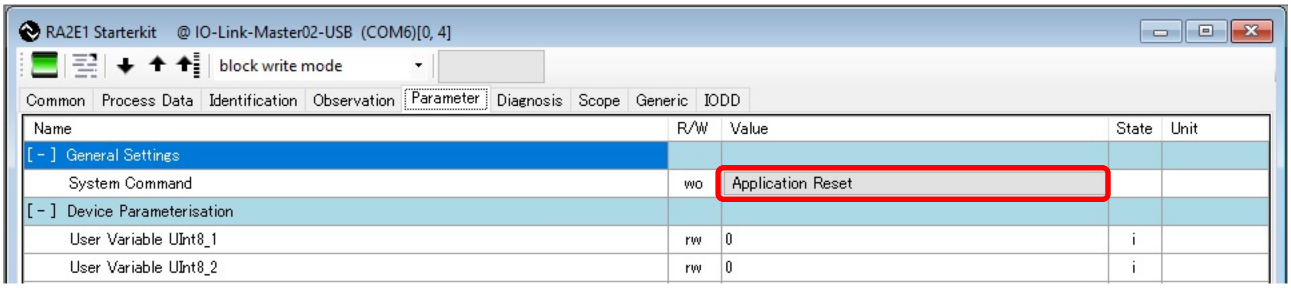


Figure 55. IO-Link Device Tool V5.1 – PE (Application Reset)

Note: When Application Reset is performed, the operation mode is set to the default, Deactivated, and LED2/LED3 are always off. To change the operation mode, refer to section, 6.6.7.3 Change Operation mode for operation.

6.6.8 IODD Tab

This tab displays the IODD information.

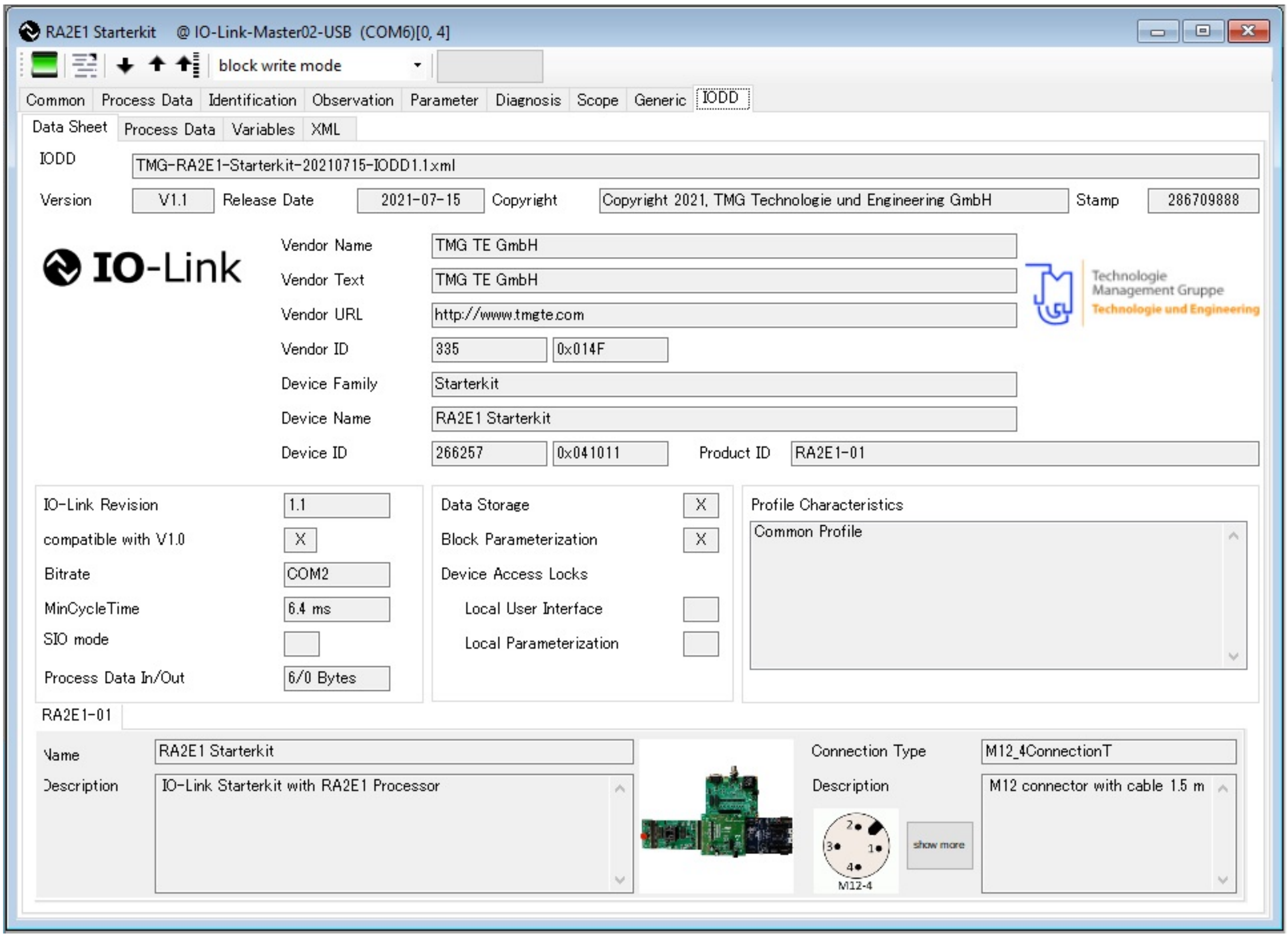


Figure 56. IO-Link Device Tool V5.1 – PE（IODD tab）

7. Examples of Use

This chapter describes an example of a sample application.

The measurement data (Measurement Value) and the switching state during operation are transferred from the IO-Link device to the IO-Link master and PC via IO-Link communication. Open the Process Data tab and Scope tab of the IO-Link Device Tool, and check the changes in the values in subsequent operations.

7.1 Reset IO-Link parameters to default values

Reset the IO-Link parameters of EK-RA2E1 before starting operation, according to section, 6.6.7.6 Reset Parameters to Default Values.

7.2 Operating Methods

This chapter shows an example of operation with the setting values shown below.

Action Mode	Window
SP1	55,000
SP2	45,000
ZSSC resolution	16 bit

The operation steps are as follows:

1. Set the operating mode to Window according to section, 6.6.7.3 Change Operation mode.
2. Turn the SRB dial counterclockwise to set the lowest limit.

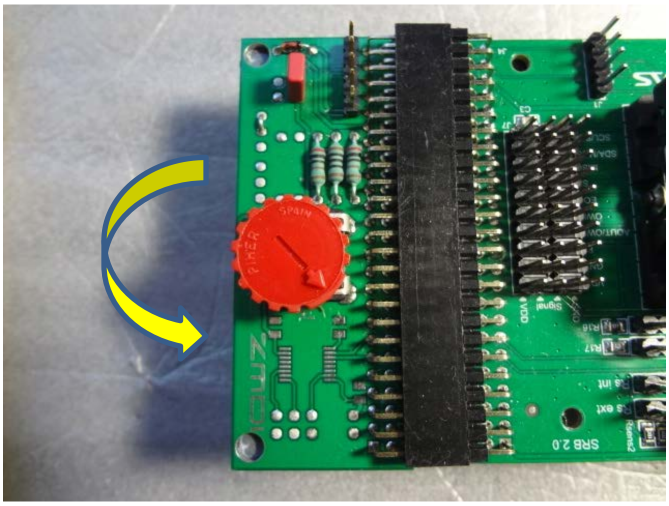


Figure 57. SRB Lower Limit Setting

LED2 is off because the measurements are less than SP2.

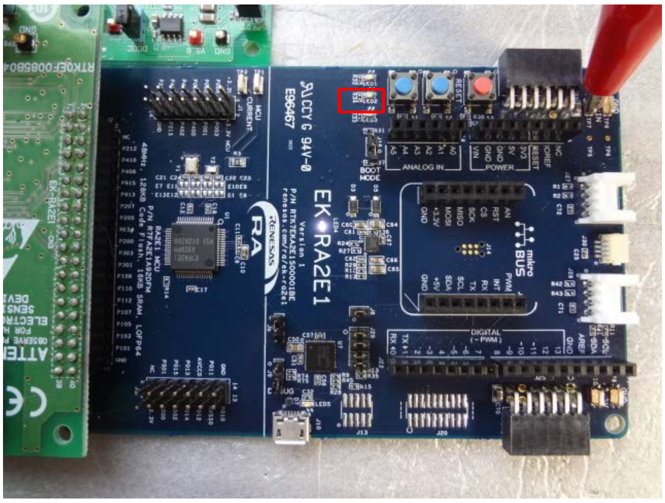


Figure 58. LED Status - Less than SP2

3. Turn the SRB dial clockwise.

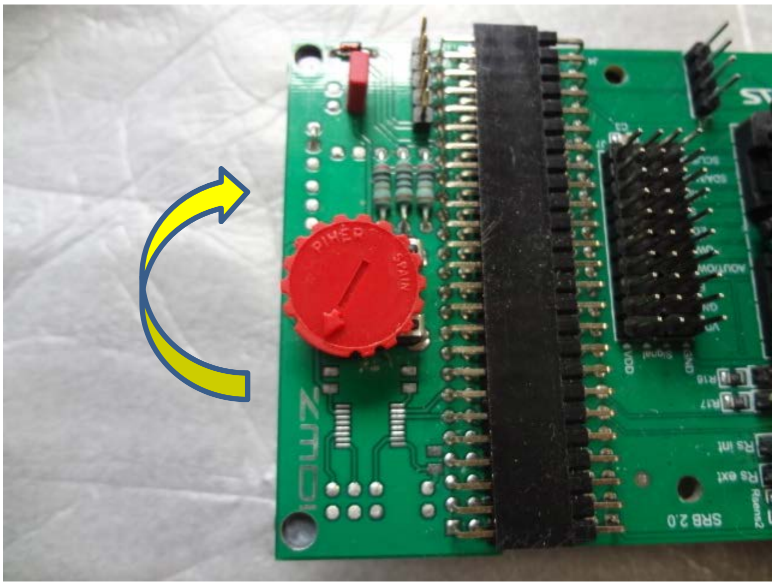


Figure 59. SRB - SP2 Location

LED2 lights up when the measured value is SP2 or higher. The position of the dial at this time is shown in Figure 60.

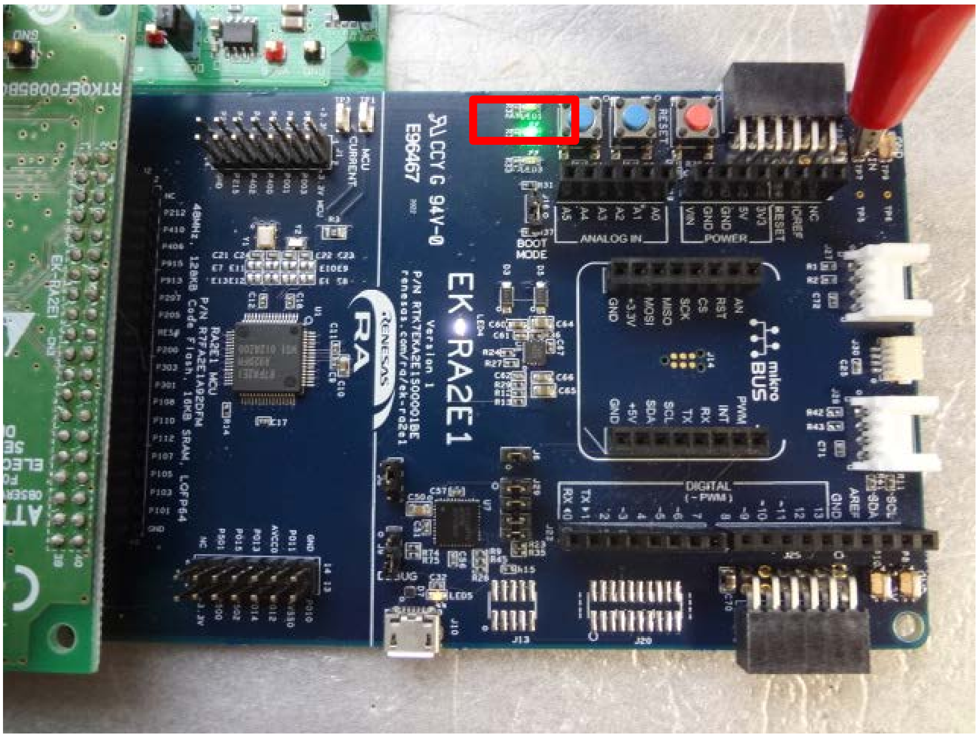


Figure 60. LED Status - SP2 or Higher

4. Turn the SRB dial further clockwise.

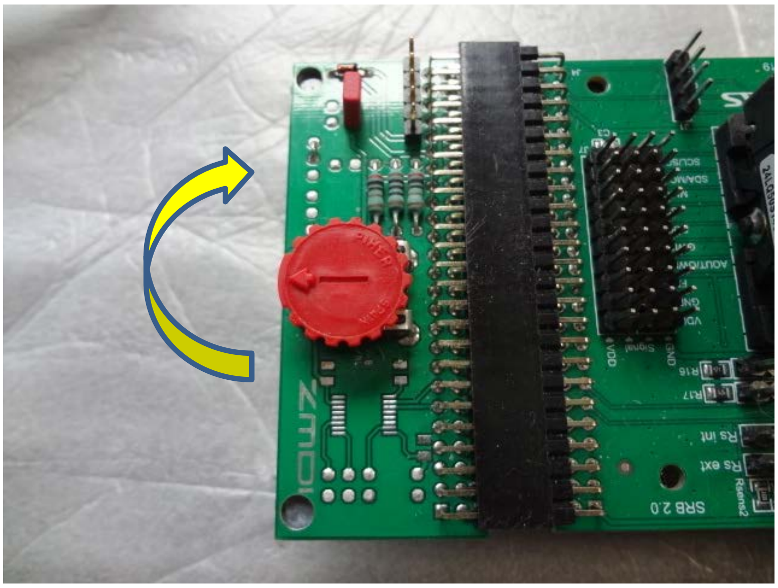


Figure 61. SRB - SP1 Reached

When the measured value exceeds SP1, the LED switches to LED3 (red). The position of the dial at this time is shown in Figure 62.

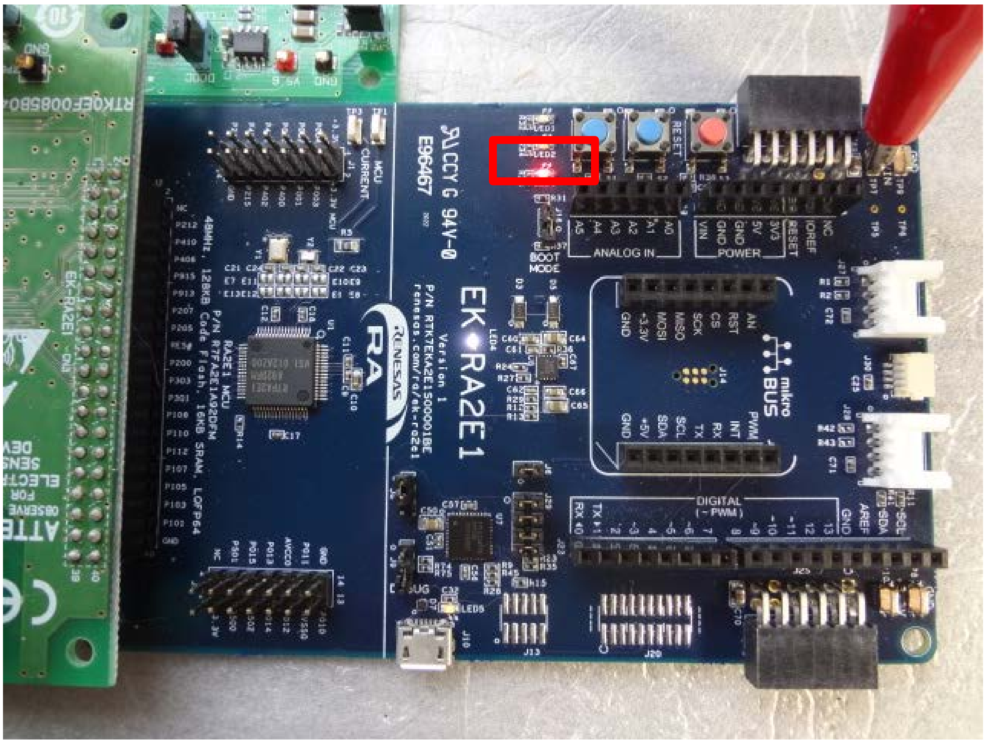


Figure 62. LED Status - SP1 Excess

Note: Measurement with ZSSC is performed with the AA_{HEX} command, which is one of the Measurement Commands of ZSSC. For the AA_{HEX} command, refer to the following materials.
- ZSSC3240 Evaluation Kit User Manual (Table 9. Measurement Commands)
- ZSSC3240 DataSheet (Table 33. Command List)

8. How to Change ZSSC Settings

ZSSC settings are made from the ZSSC GUI on the PC while the Windows PC is connected to the EK-RA2E1 board with a USB-SERIAL conversion cable.

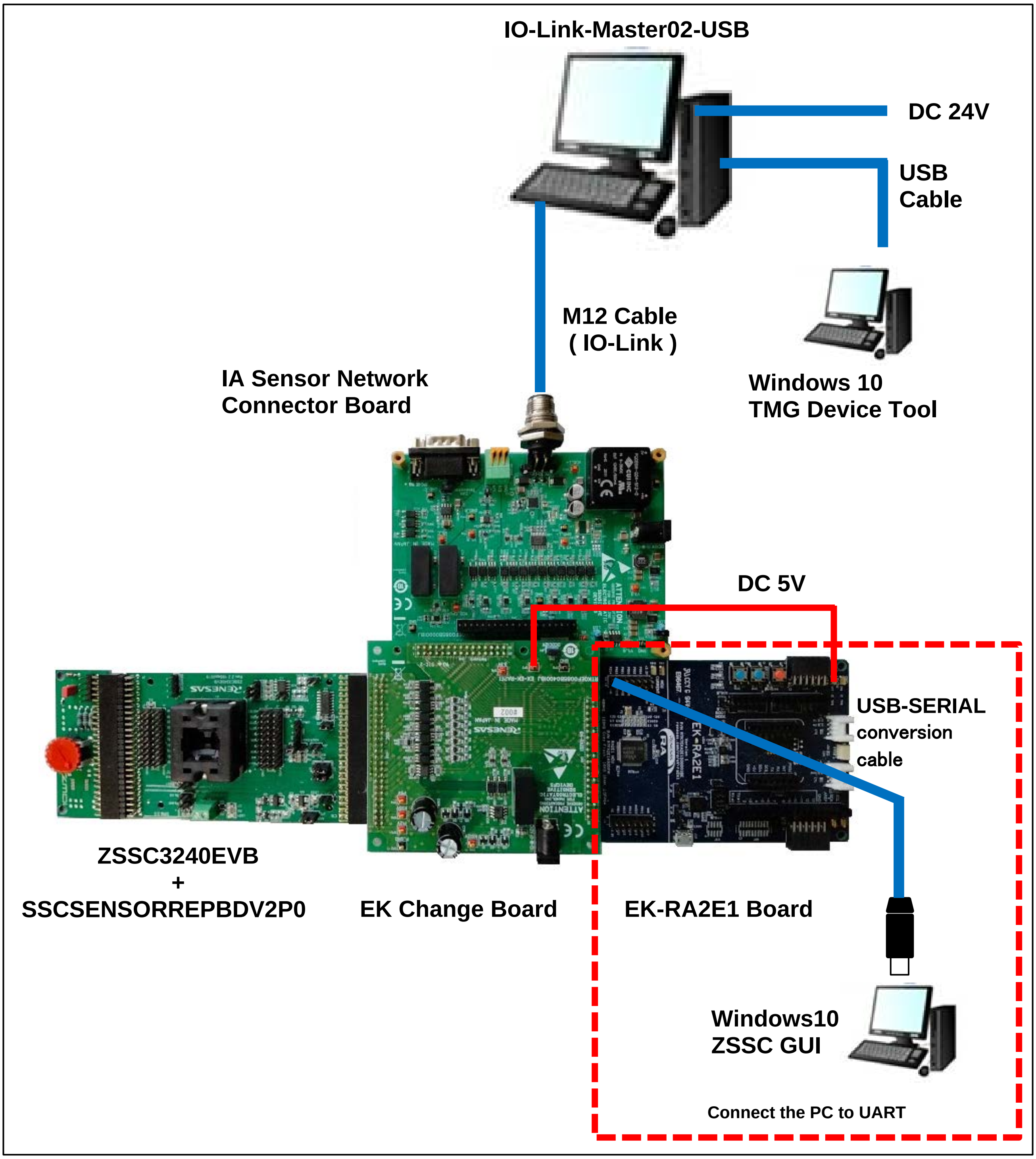


Figure 63. USB-SERIAL Conversion Cable Connection Status

8.1 USB-SERIAL Conversion Cable Connection

Since EK board does not have USB interface for communication with PC, it connects USB-SERIAL conversion cable to J1-9 (P401), J1-10 (P402) and J1-14 (GND) of the J1 connector to realize communication with PC by virtual COM.

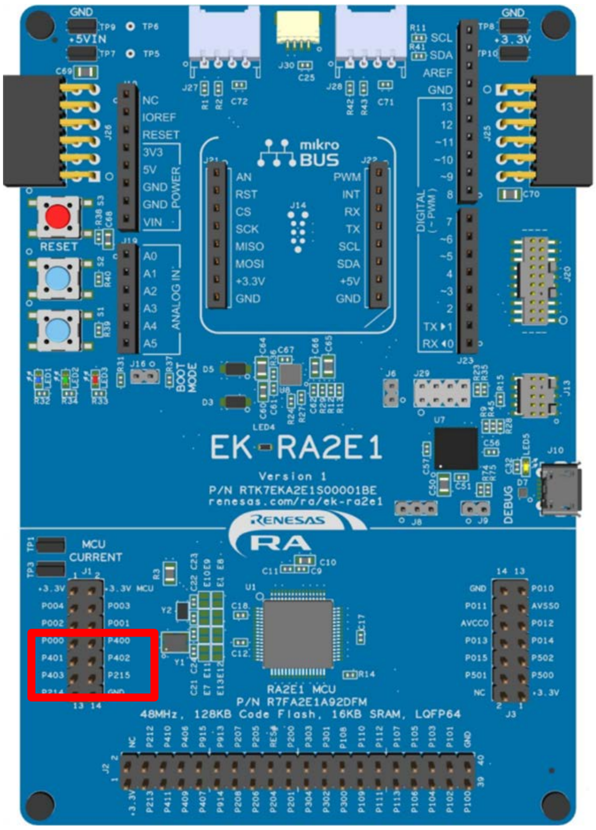


Figure 64. EK-RA2E1 Board J1 Connector

Connect the RXD/TXD/GND lines of the USB-SERIAL conversion cable with built-in FT232RL as follows.

Any cable with built-in features equivalent to FT232RL can be used.

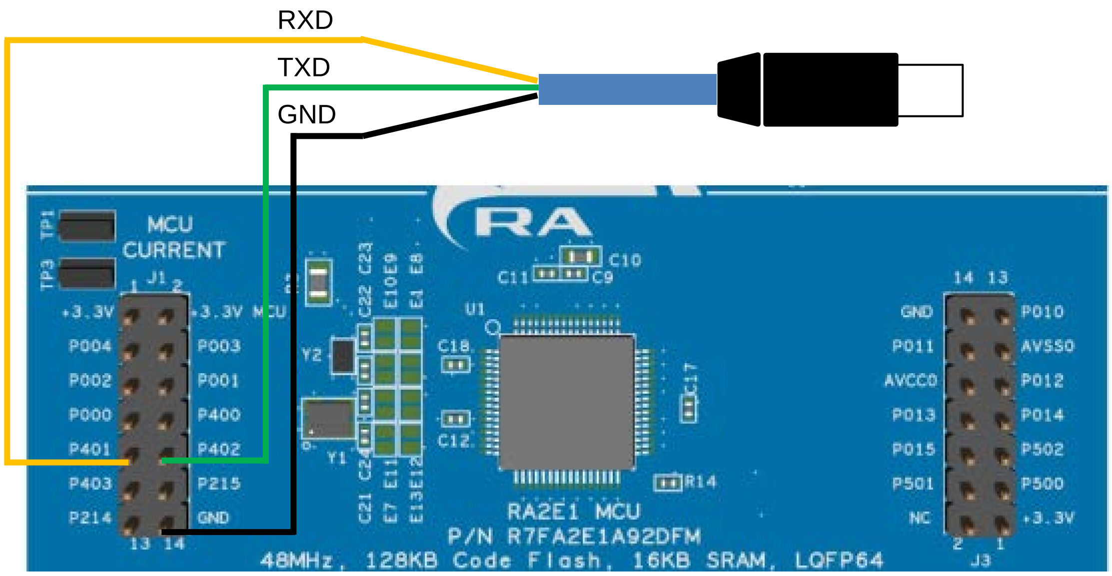


Figure 65. Connecting the USB-SERIAL Conversion Cable

8.2 Check the Virtual COM Port Number

When connecting USB-SERIAL converter cable to PC, the driver is automatically selected and recognized as a COM port.[note1]

Check the virtual COM port number of the USB-SERIAL converter cable in Device Manager of Windows.[note2]

The following is an example when recognized as COM7.

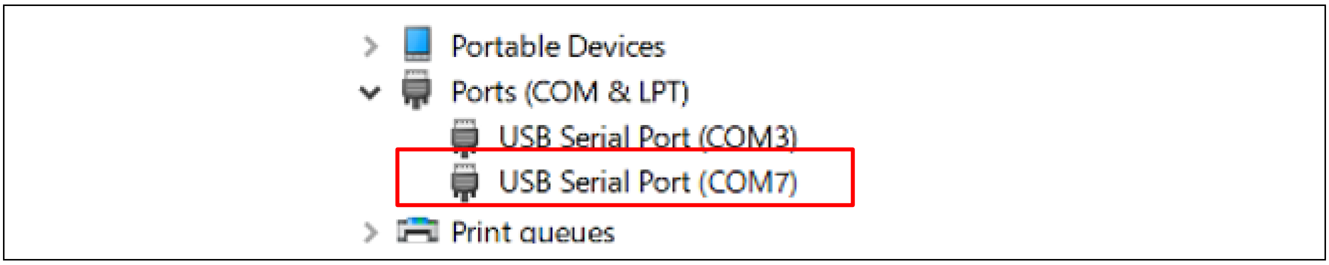


Figure 66. Device Manager

Note 1. When using FT232RL compatible cables, some drivers may not be set automatically. In that case, get the driver and set the driver manually.

2. The name displayed may be different depending on the USB-SERIAL conversion cable used.

8.3 Start the ZSSC GUI

Start the ZSSC GUI. The following graphic shows the ZSSC GUI screen after startup.

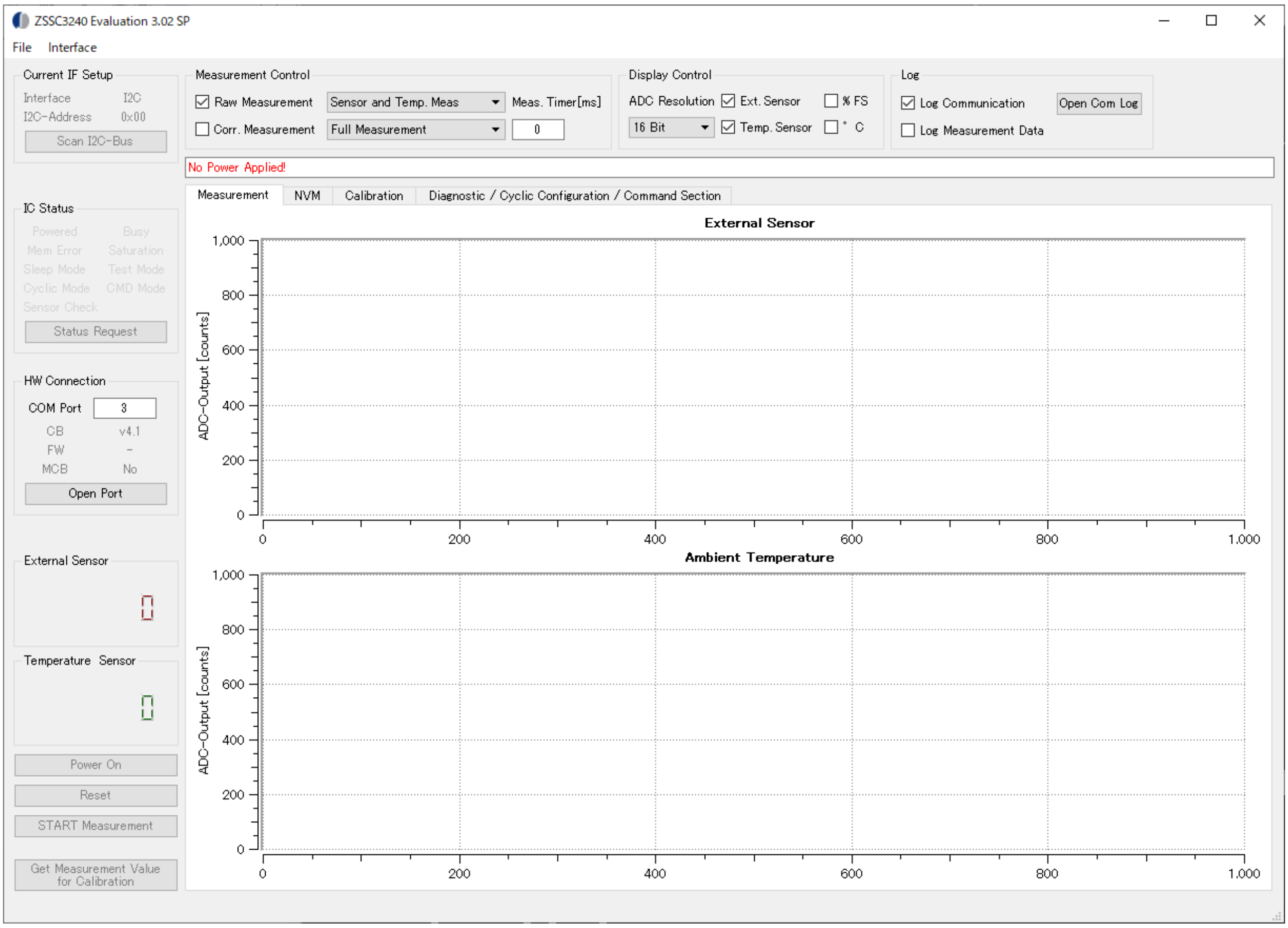


Figure 67. Screen after Starting ZSSC GUI

Note: The GUI is displayed after about 10 seconds of communication negotiation.

Note: Communication negotiation is performed by COM Port default COM3 in the GUI, and an error message is displayed if communication cannot be confirmed.

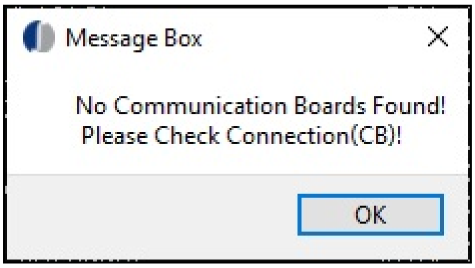


Figure 68. Communication Error Display at GUI Startup

8.4 Establish a Communication Path

8.4.1 Open Port

For **COM Port**, set the COM port number checked in Figure 66, and click the **Open Port** button.

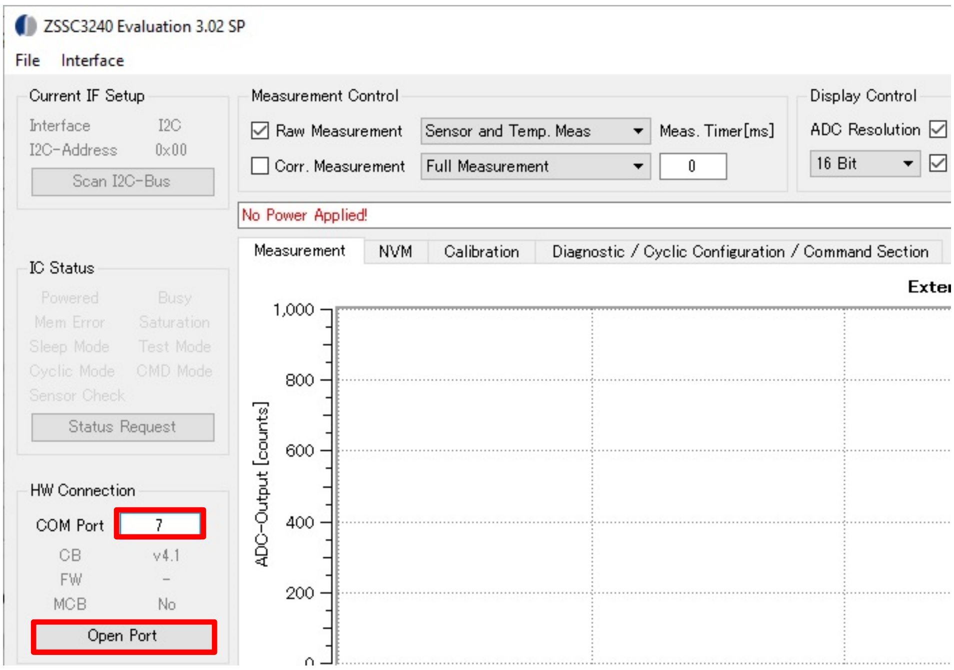


Figure 69. Open Port

Note: If the COM port used is COM3, **Open Port** is automatically selected.

If the Open Port is successful, the GUI screen changes as shown in Figure 70.

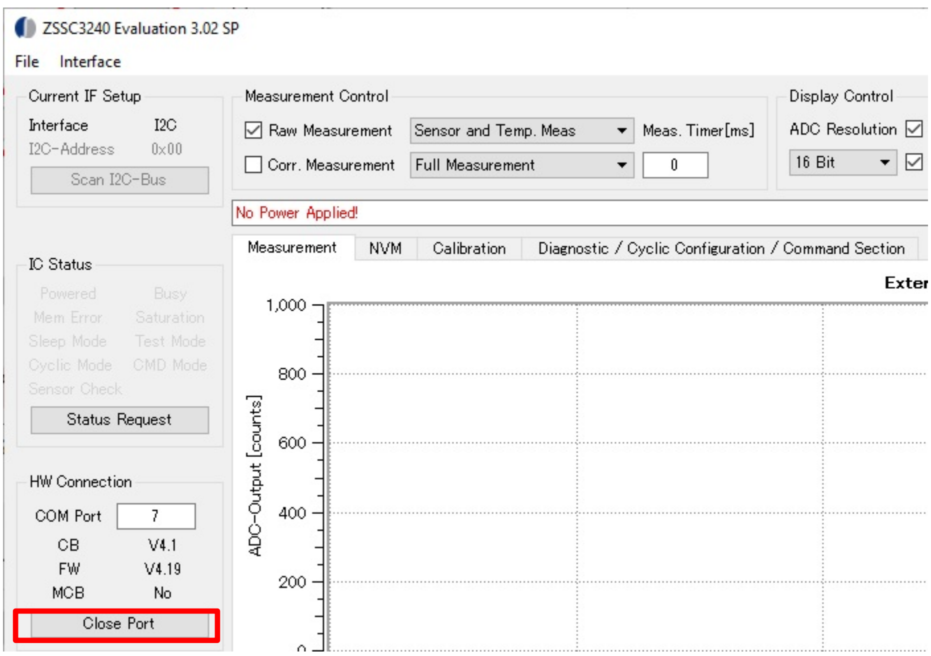


Figure 70. Open Port Success

Note: If communication with the default COM3 is successful and Open Port is automatically selected, the IC Status and other displays may differ from the example shown in Figure 70.

8.4.2 Interface

Select **Interface** > **I2C**. note1

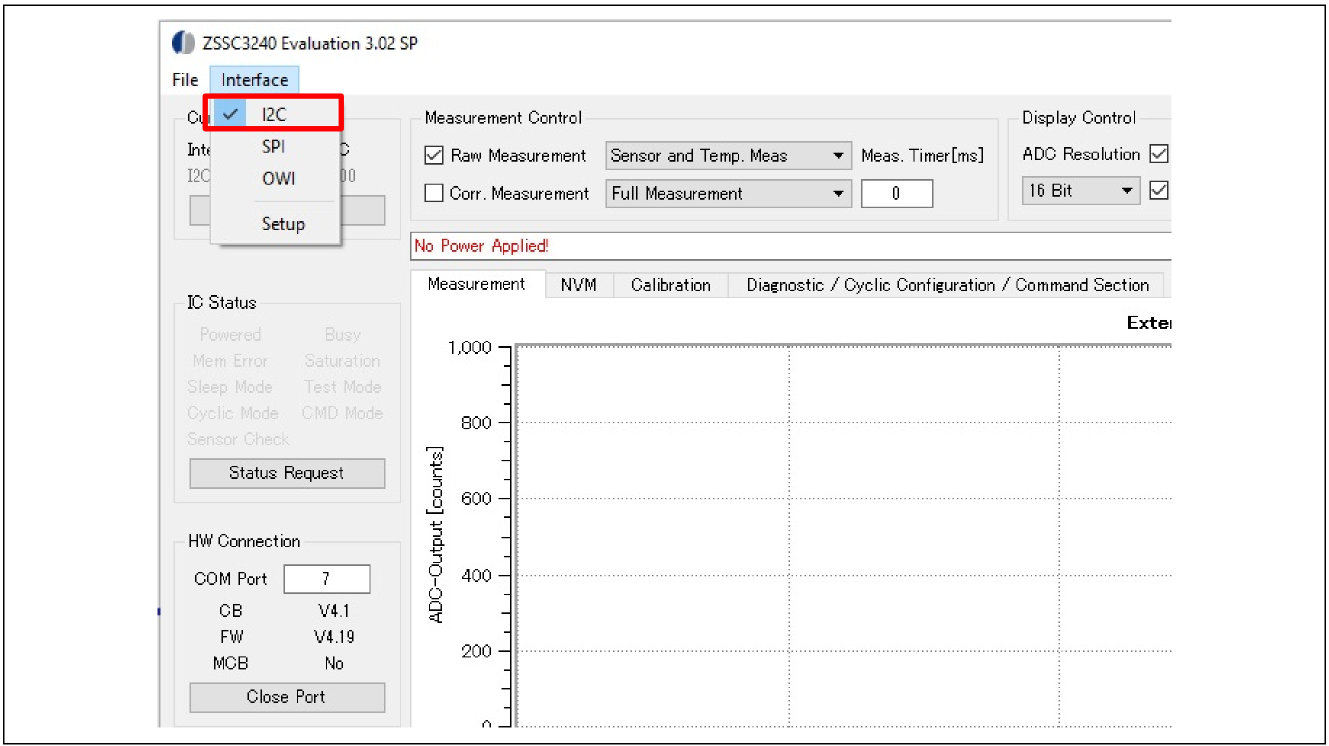


Figure 71. Interface Selection

Confirm that **I2C [400kHz]** is displayed in the Interface of Current IF Setup. note2

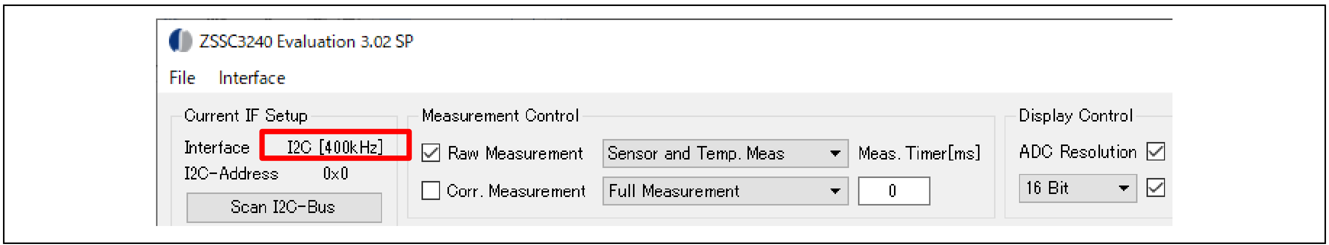


Figure 72. Current IF Setup Interface

Note 1. This sample application operates on I2C only; SPI and OWI cannot be used.

2. If the Interface display does not change to **I2C [400 kHz]** as shown in Figure 72, select **I2C** from the **Interface** menu again.

8.4.3 Scan I2C-Bus

Click **Scan I2C-Bus** button to detect ZSSC I2C devices.

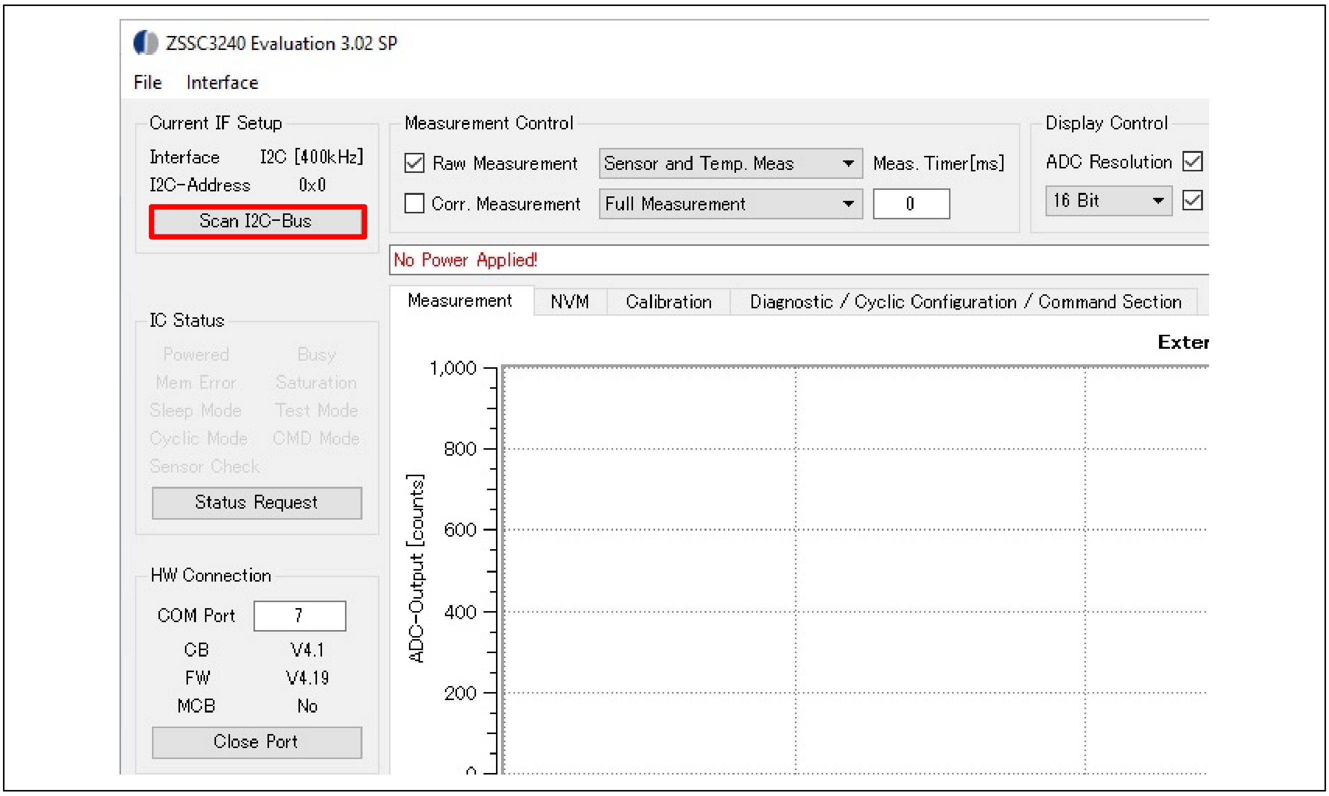


Figure 73. Scan I2C-Bus

When I2C device is detected, the message shown in Figure 74 appears. Communication with the ZSSC3240 is now possible.

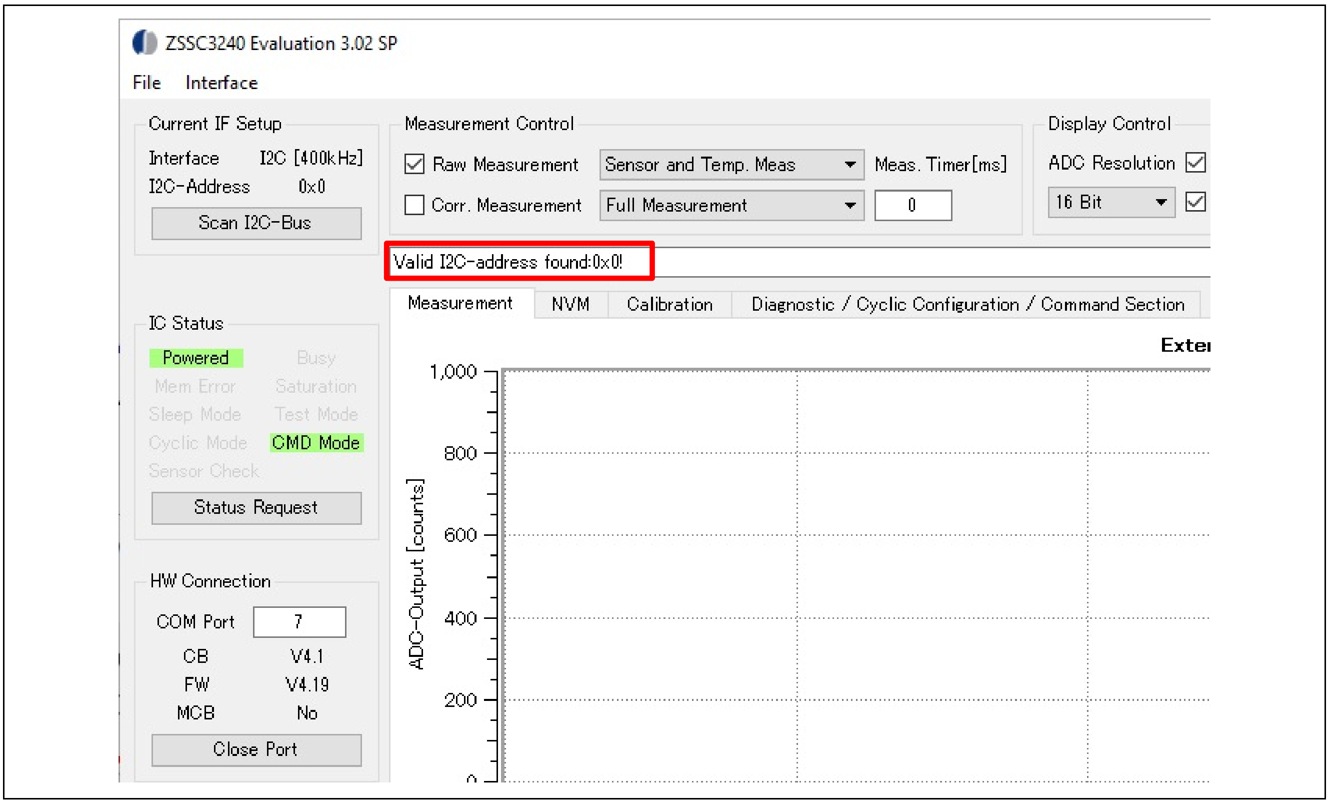


Figure 74. I2C Device Detection

8.5 Measurement (Corr. Measurement)

This chapter provides an example of a measurement using the ZSSC GUI tool based on the setting values shown in the table below.

Action Mode	Two Point
SP1	55,000
SP2	45,000
ZSSC resolution	16bit

Note: Each graph image was captured separately for illustration purposes after [Start Measurement]. They are not based on continuous measurements.

The steps to make a measurement are as follows:

1. Set the operating mode to Two point according to section, 6.6.7.3 Change Operation mode.
2. Check **Corr. Measurement** and select **Full Measurement**. If **Temp. Sensor** is checked, uncheck it.

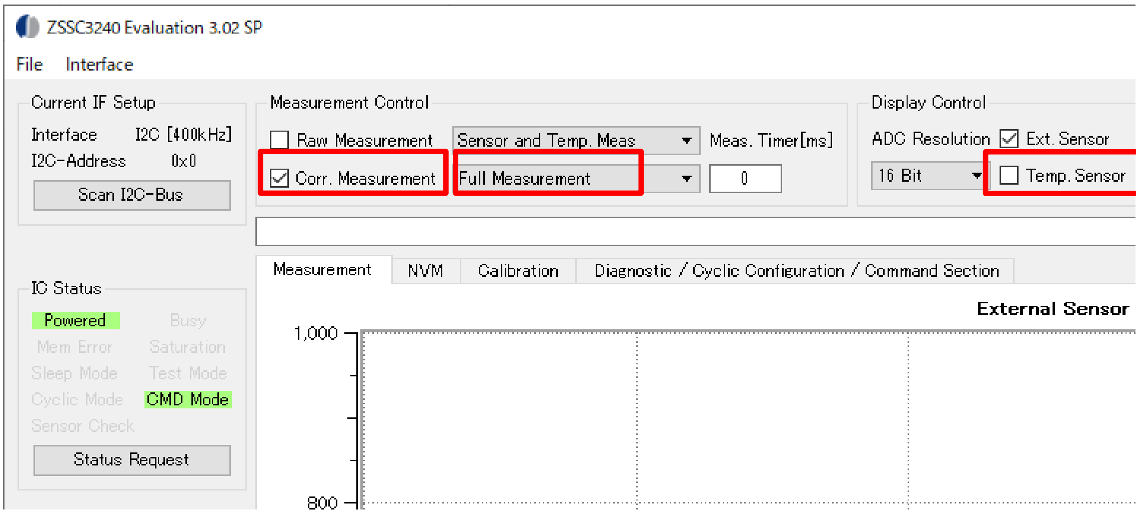


Figure 75. Corr. Measurement

3. Turn the SRB dial counterclockwise to set it to the lowest limit.

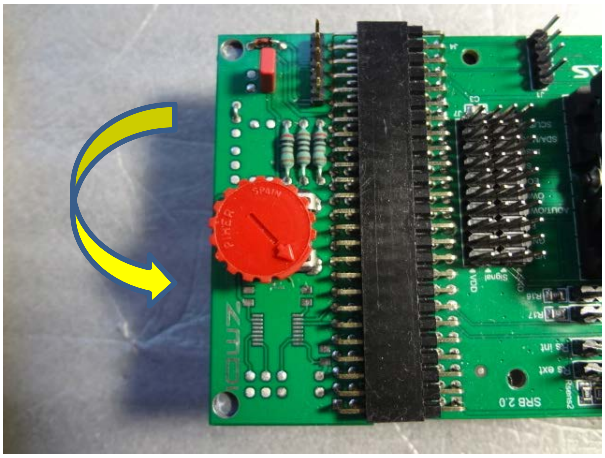


Figure 76. SRB Lower Limit Setting

4. Start measurement
   Click **START Measurement** to start measurement with the sensor connected to the ZSSC3240EVB and begin graph display. After the click operation, the button changes to the **STOP Measurement** display.

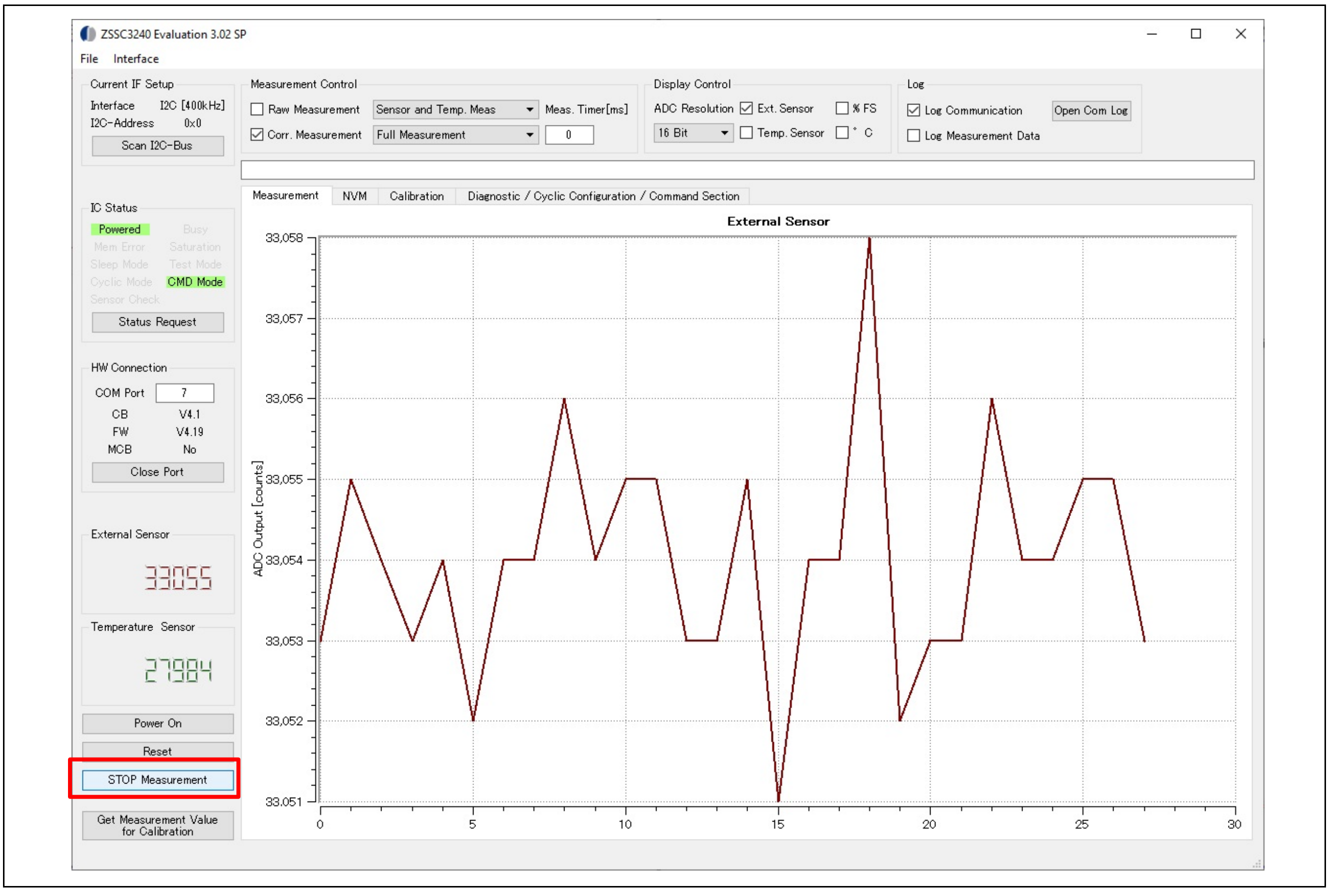


Figure 77. Start Measurement

Note: The graph scale is automatically adjusted according to the measured values. When the dial is not operated immediately after the start of measurement, there is little change in the measurements, so the scale is displayed with a grid unit of 1.

5. Turn the SRB dial clockwise until the External Sensor value is just before SP1 (55000).

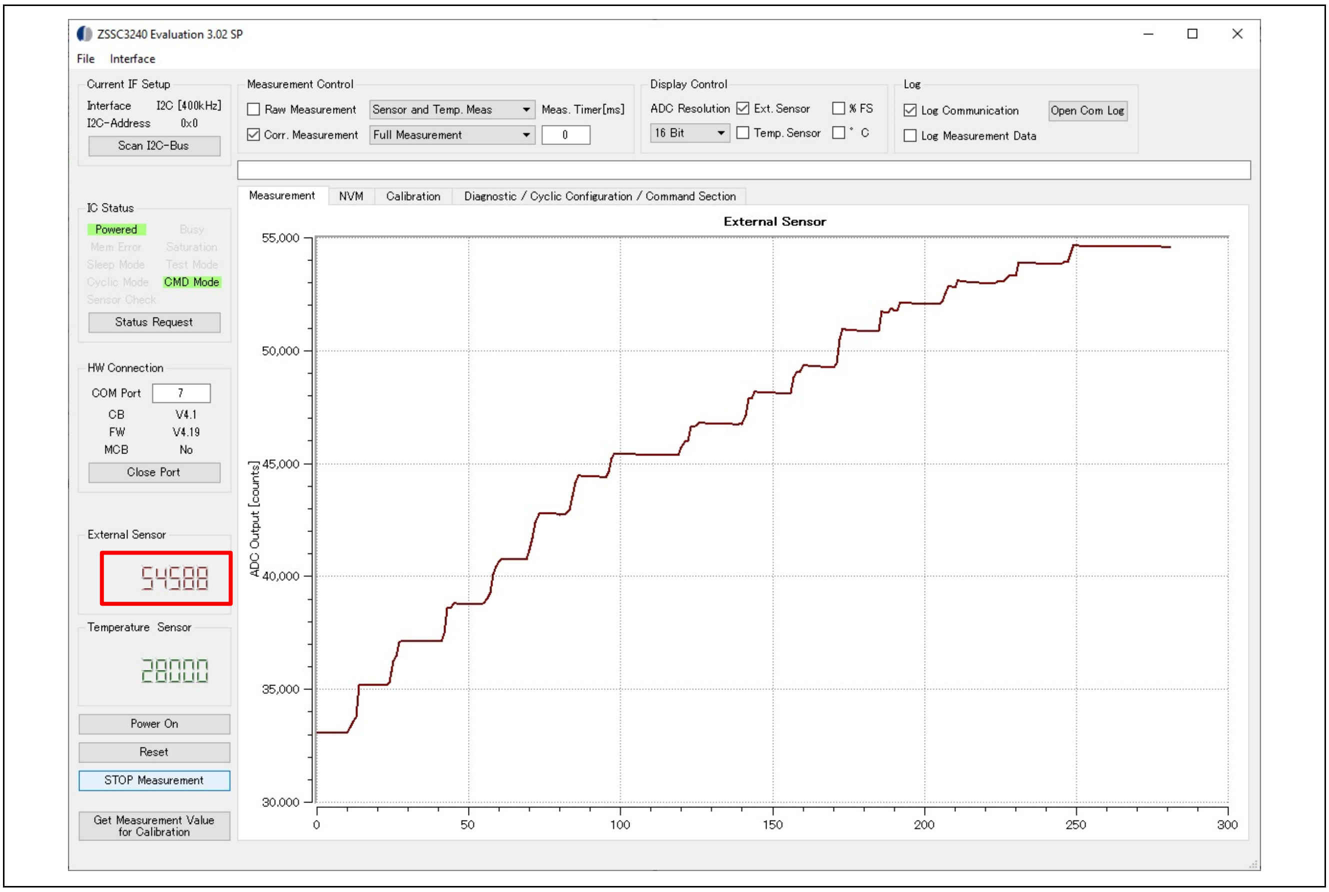


Figure 78. Measurement - SP1 Front

LED2 is off because the measured value is SP1 or less.

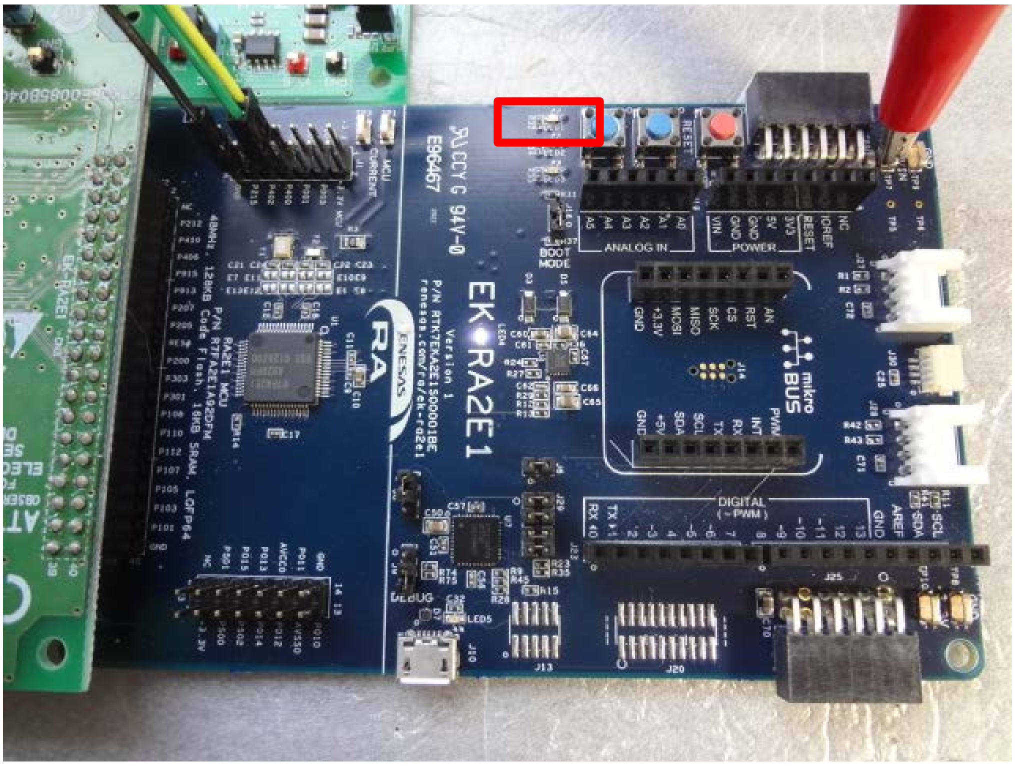


Figure 79. LED Status - SP1 or Less

6. Turn the SRB dial clockwise until the External Sensor value exceeds SP1 (55000).

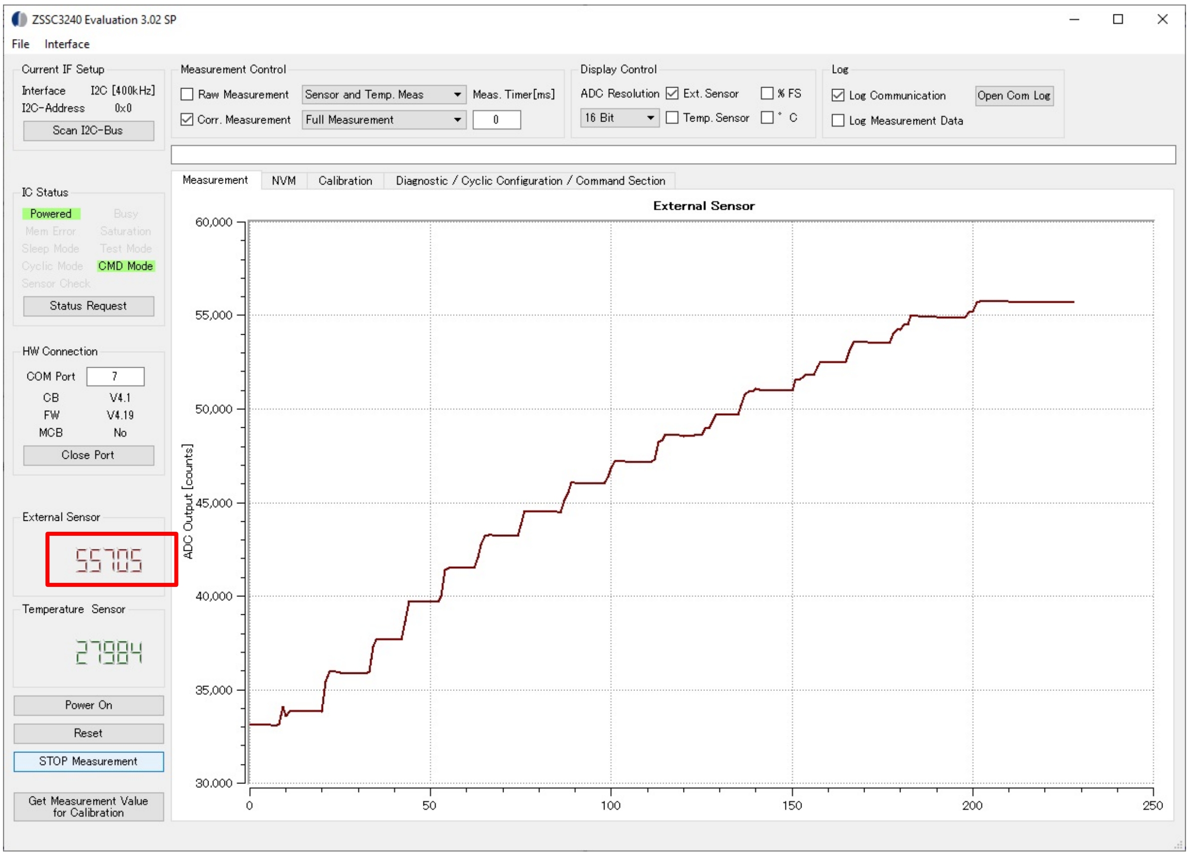


Figure 80. Measurement - SP1 Excess

LED2 lights up when the measured value exceeds SP1.

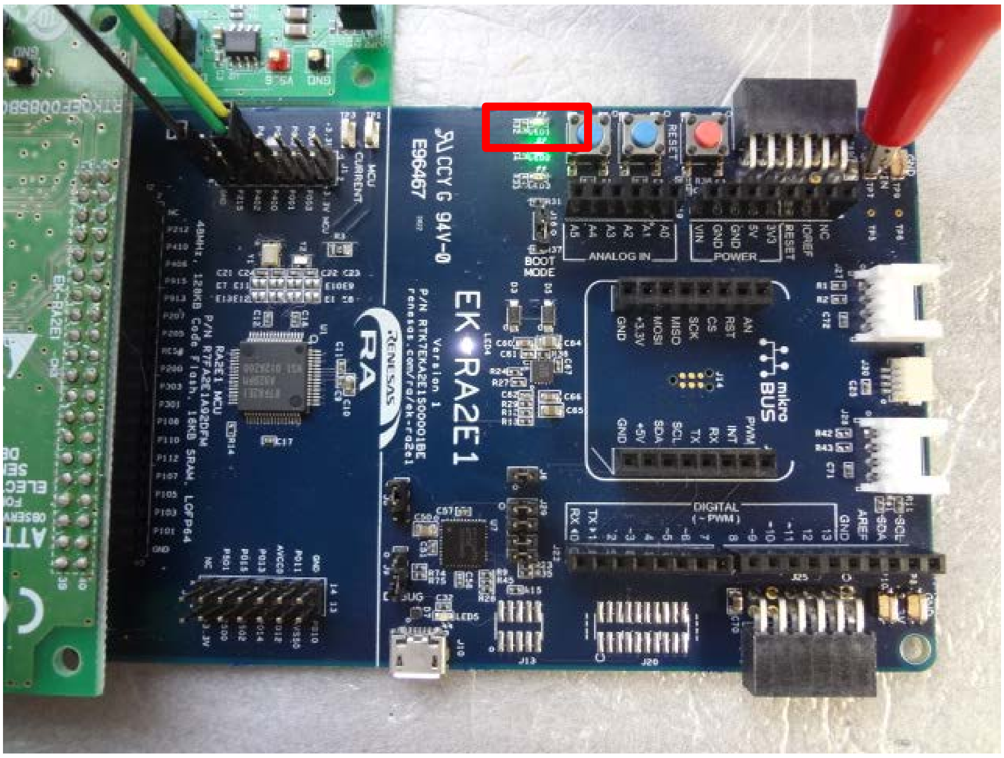


Figure 81. LED Status - SP1 Excess

7. Turn the SRB dial counterclockwise until the External Sensor value is less than SP2 (45000).

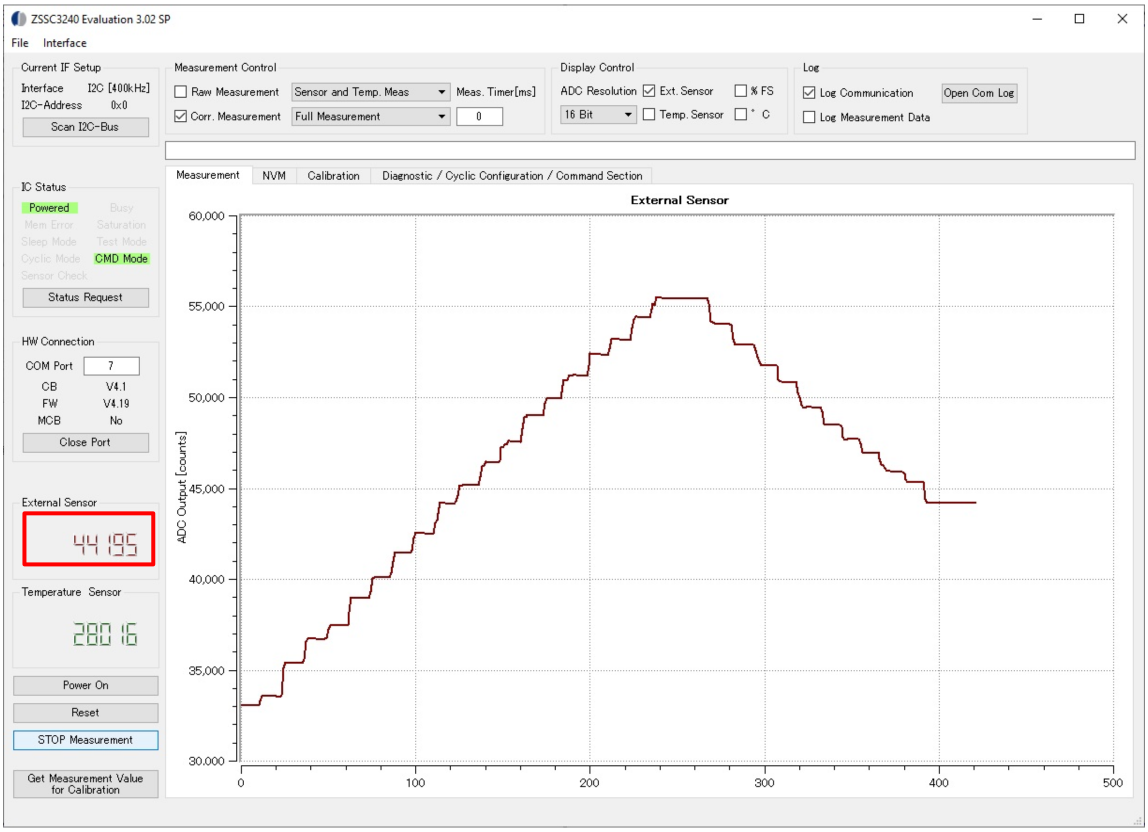


Figure 82. Measurement – Less than SP2

LED2 will turn off when the measured value less than SP2.

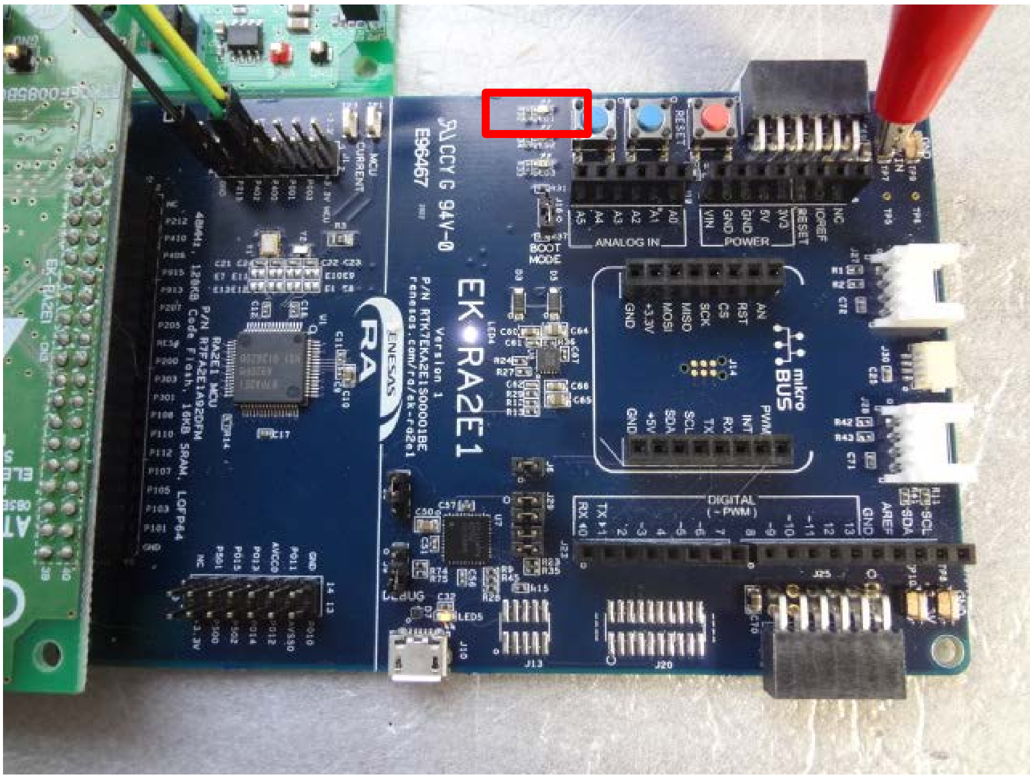


Figure 83. LED Status - Less than SP2

8.6 NVM Operation

This section explains how to operate the NVM (non-volatile memory) registers in the ZSSC.

8.6.1 Read NVM

Select the **NVM** tab. Initially, no values are loaded.

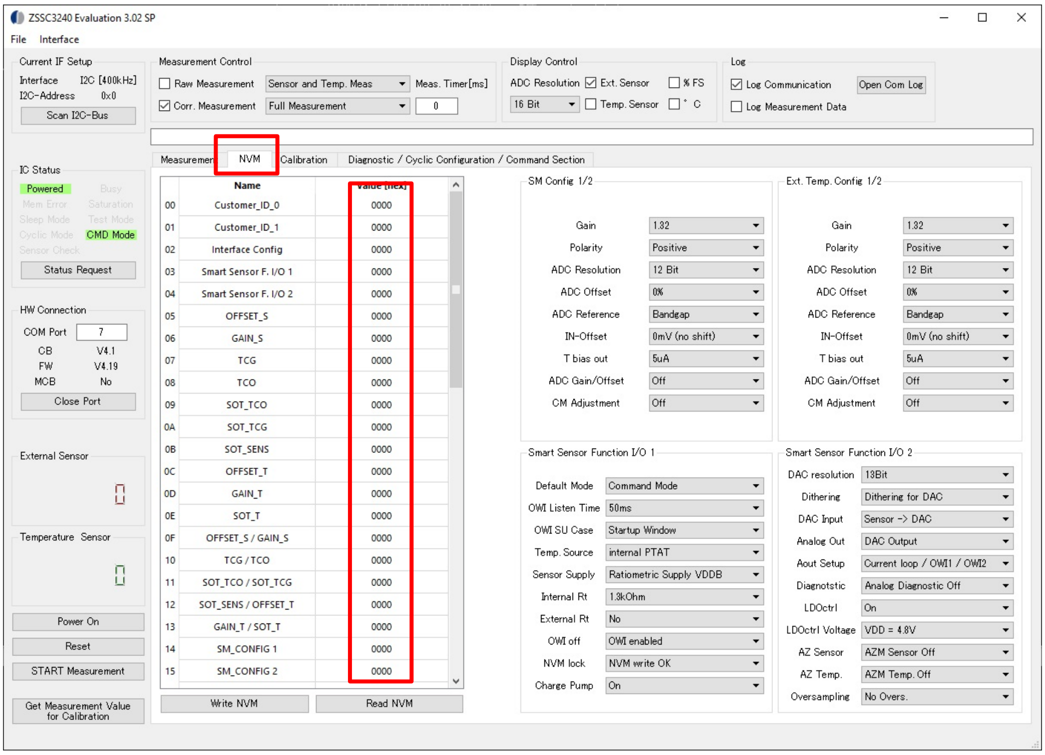


Figure 84. NVM Tab Initial State

Click **Read NVM** to communicate with ZSSC and read the values.

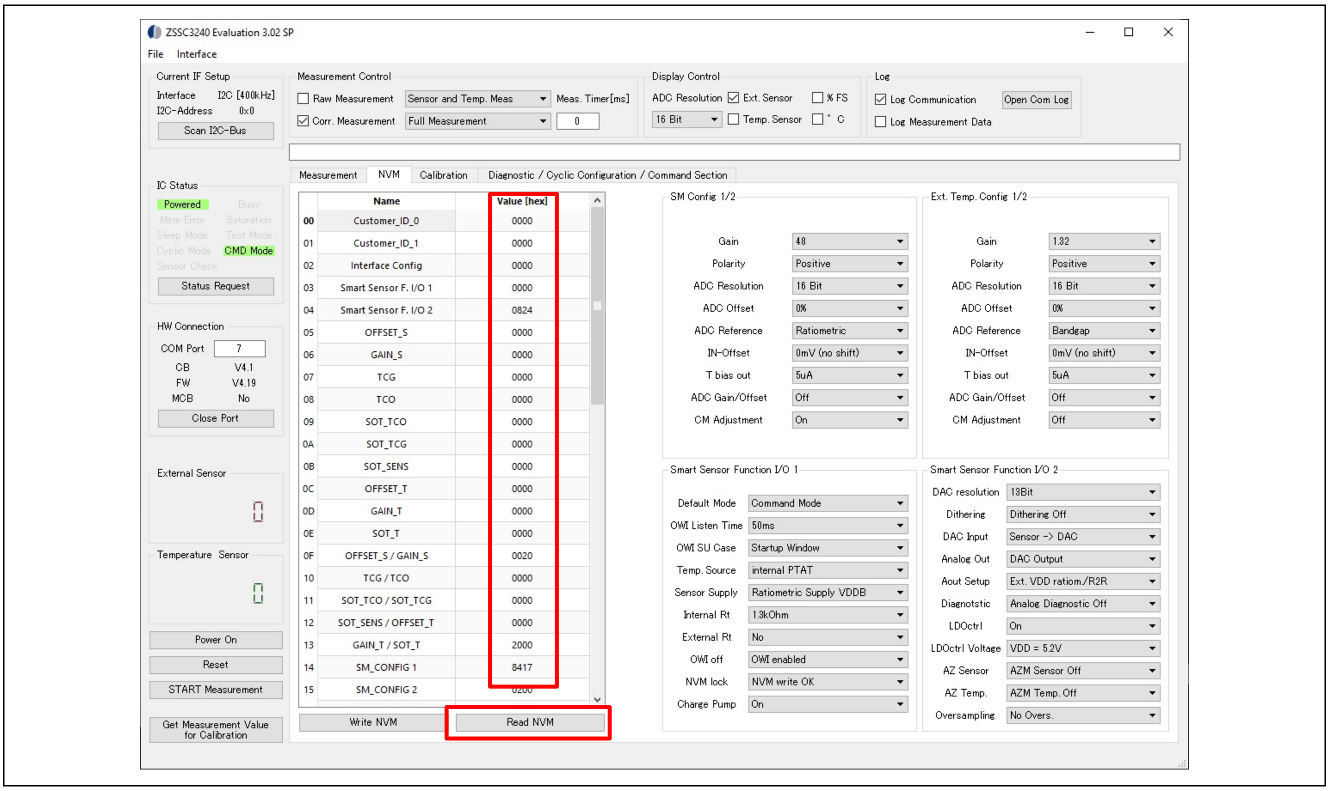


Figure 85. Read NVM Tab State after Execution

8.6.2 Write NVM

Change the setting value and execute Write NVM to transfer the setting to ZSSC. The following is an example of changing the ADC resolution.

Select 24-bit for **ADC Resolution** in **SM Config 1/2**. The changes are highlighted with a red background in the register list, as shown in Figure 86. ADC Resolution Change

.

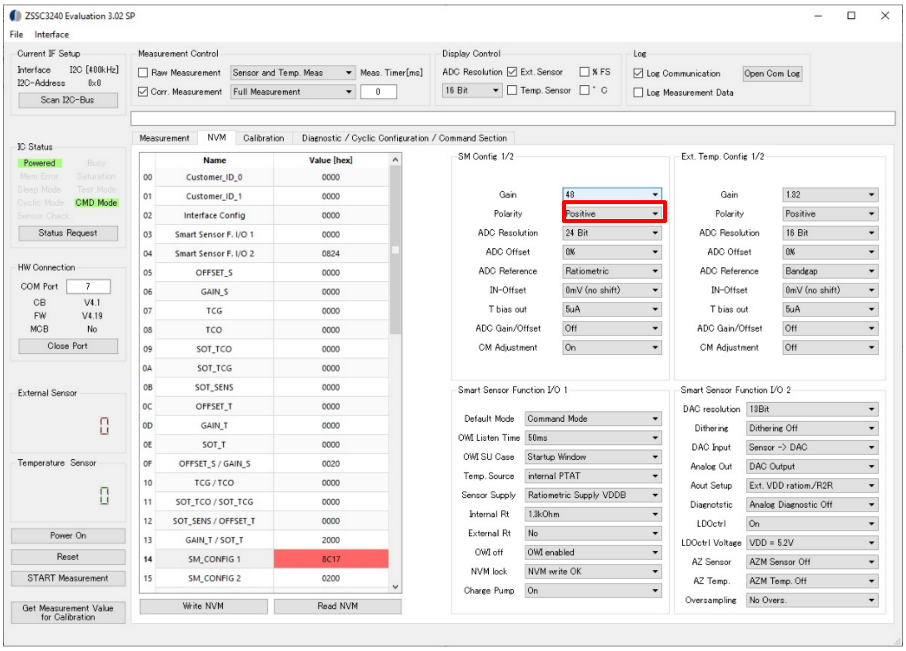


Figure 86. ADC Resolution Change

Click **Write NVM**. The changes are transferred to the NVM register of the ZSSC. When the transfer is complete, the red background of the highlighted changes will turn off.

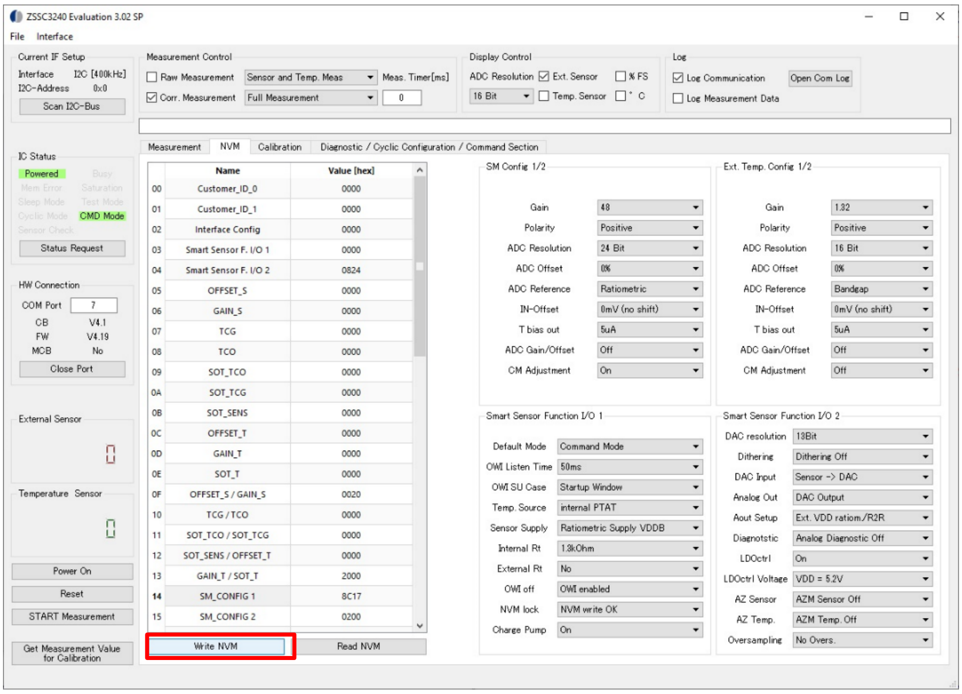


Figure 87. Write NVM

Note: If **Write NVM** is executed without having read the settings by **Read NVM**, the existing values may be unintentionally overwritten. Be sure to read the existing settings with **Read NVM** before executing **Write NVM**.

Click **Reset** to reset the ZSSC EVB. This operation reflects the NVM settings to the system.

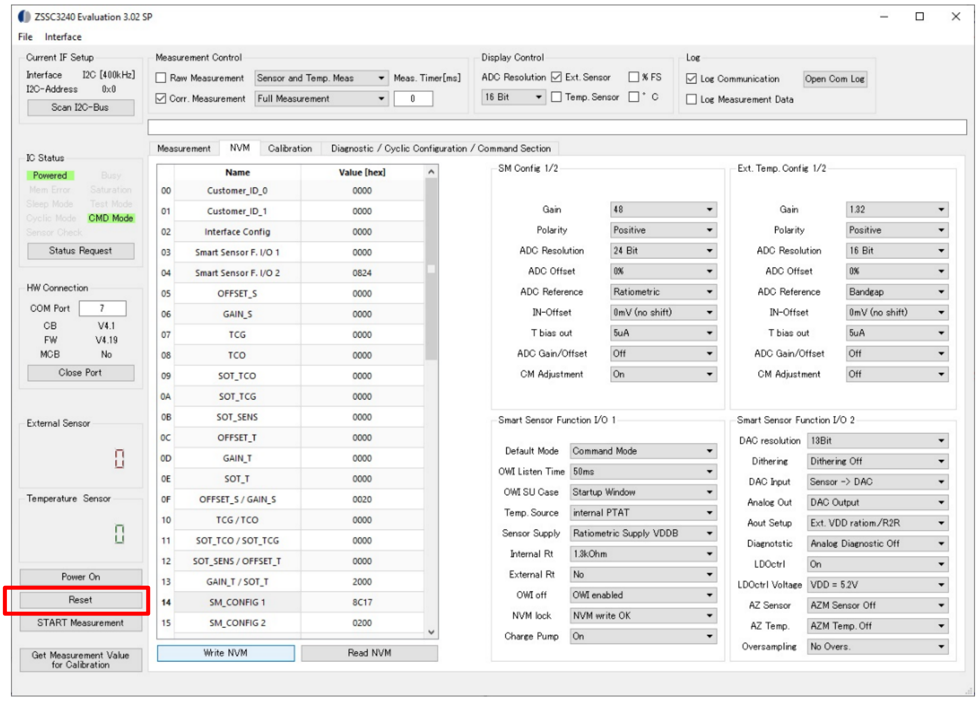


Figure 88. Reset

Note: The **Write NVM** execution alone does not make the setting effective. Be sure to reset the ZSSC EVB with the **Reset** button.

8.6.3 Write CRC

If CheckSum is inconsistent due to NVM change, the IC Status Mem Error lights up after the **Reset** button operation.

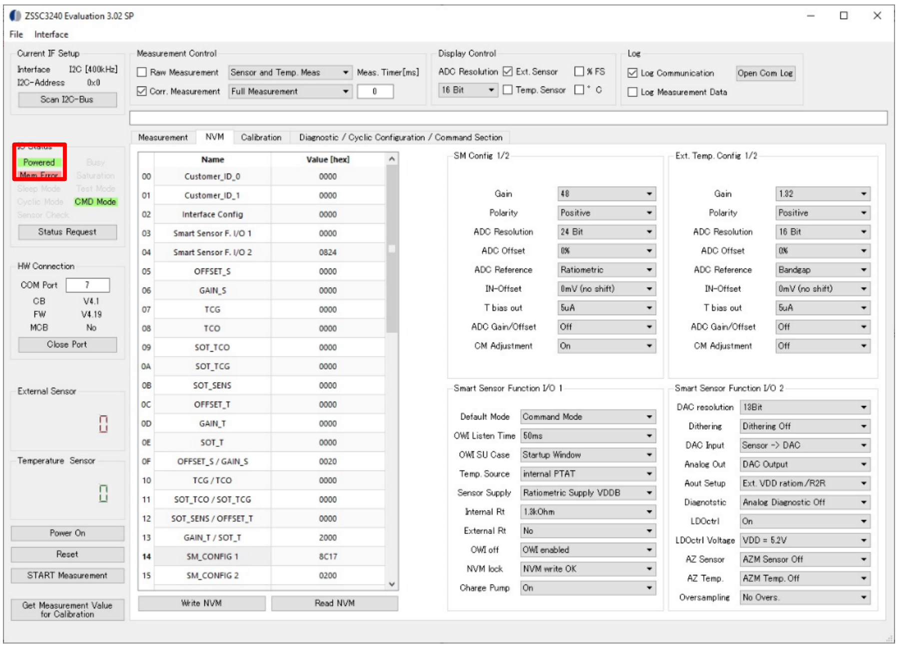


Figure 89. Mem Error

In this case, select the **Diagnostic / Cycle Configuration / Command Section** tab and click **Write CRC** to recalculate CheckSum and write it to NVM. If CheckSum is successfully reconfigured by NVM in ZSSC, MemError will turn off after the **Reset** button is clicked.

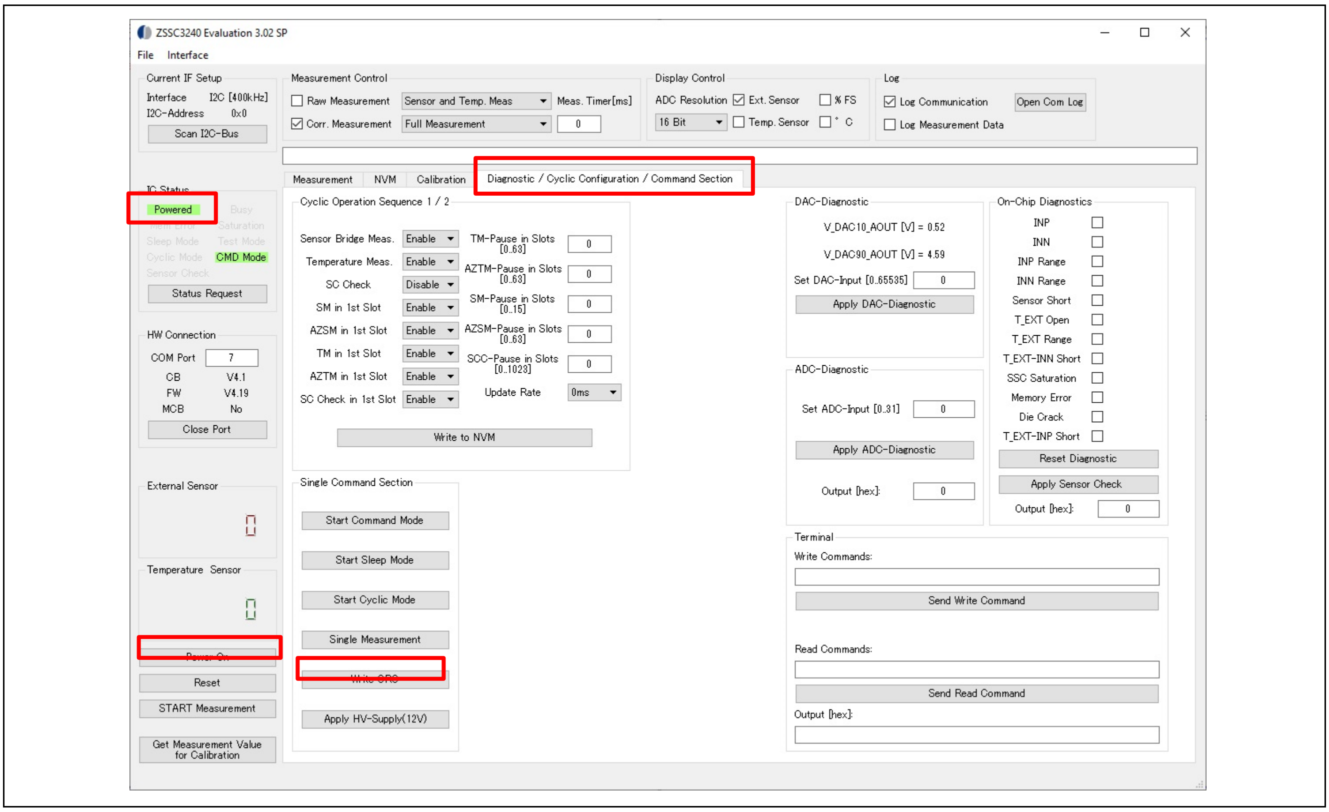


Figure 90. Write CRC

8.6.4 Return ADC resolution set value

In section, 8.6.2 Write NVM the operation to change the ADC resolution to 24-bit was explained as an example of NVM change.

For section, 7.2 Operating Methods, return to 16-bit before the change. Follow the procedure below to set the ADC resolution to 16-bit.

1. Change the ADC resolution to 16-bit according to the procedure in section 8.6.2 Write NVM.
2. Correct and Reset MemError according to the procedure in section, 8.6.3 Write CRC.

9. References

- R11QS0035JGxxxx RA2E1 Group Evaluation Kit for RA2E1 MCU Group EK-RA2E1 Quick Start Guide
- R20UT4825JGxxxx RA2E1 Group Evaluation Kit for RA2E1 MCU Group EK-RA2E1 v1 User's Manual
- R01UH0852JJxxxx Renesas RA2E1 Group User's Manual: Hardware
- R11UM0155EU0270 Renesas Flexible Software Package (FSP) v3.7.0 User’s Manual
- Renesas ZSSC3240 Evaluation Kit User Manual
- Renesas Datasheet ZSSC3240
- Renesas SSC Communication Board SSC-CB Datasheet

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Nov.01.22	-	First release

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

Trademarks

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/.