Renesas RA Family

EK-RA8T2 EtherCAT ETG5003 SubDevice Software

Introduction

This document describes how to set up the sample program for EtherCAT® SubDevice functionality using the adapted EtherCAT Stack Code for the RA8T2 series.

It also explains how to verify SubDevice behavior and stack features using TwinCAT® as the configuration tool for the MainDevice.

Target Device

EK-RA8T2

Contents

1. Overview 3
1.1 Abbreviations / Definitions 3
1.2 Reference 3
1.2.1 About RA8T2 3
2. Features 4
2.1 Folder structure relative to this application note 5
3. Requirements (Software and Hardware) 6
3.1 Requirements for this Sample Package 6
4. Hardware Setup 7
4.1 EK-RA8T2 Board 7
4.1.1 Jumper and Switch Configuration 7
5. Setting up the Host 8
5.1 Copying the ESI Files 8
5.2 Installing the Driver 8
5.3 Generating the EtherCAT SubDevice Stack Code 9
5.4 Setting Up the Renesas Device Partition Manager (RDPM) 12
5.5 Setting Up the Sample Code for e² studio 14
6. Demonstration of the Sample Application 19
6.1 Connecting to TwinCAT3 19
6.1.1 Scanning I/O Devices 19
6.1.2 Updating EEPROM Data 21
6.2 Checking the File over EtherCAT (FoE) Profile 23
6.2.1 Confirm the Initial Version 23
6.2.2 Firmware update 24

6.2.3 Check the Updated Version 25

7. Software Specifications 26
7.1 Details of Firmware Update 26
7.1.1 Procedure for Updating the Firmware 27
7.1.2 Hardware Configuration of the Sample Program 29
7.1.3 Functions 31

8. Appendix 32
8.1 Appendix A: Precautions When Using the SSC Tool 32
8.2 Appendix B: How to Install the Patch 34
8.3 Appendix C: Changing the EEPROM Capacity Setting 36
8.4 Appendix D: Semiconductor Device Profile [ETG.5003] 37
8.4.1 Common Device Profile [ETG.5003.1] 37
8.4.2 Semi Test Record [ETG.7000.2-Annex5003-0001] 38

9. Limitations 47

Revision History 48

1. Overview

This document describes how to set up the sample program for EtherCAT® SubDevice functionality using the adapted EtherCAT Stack Code for the RA8T2 series.

1.1 Abbreviations / Definitions

Table 1.1 Abbreviations / Definitions

Index	Abbreviations / Definitions	Description
1	CoE	CAN application protocol over EtherCAT
2	CiA	CAN in Automation
3	DC	Distributed Clock
4	EEPROM	Electrically Erasable Programmable Read-Only Memory
5	EoE	Ethernet over EtherCAT
6	ESC	EtherCAT SubDevice Controller
7	ESI	EtherCAT SubDevice Information
8	ESM	EtherCAT State Machine
9	ETG	EtherCAT Technology Group
10	FoE	File Access over EtherCAT
11	PDO	Process Data Object
12	SDO	Service Data Object
13	SSC	SubDevice Stack Code
14	FSP	Flexible Software Package
15	IDE	Integrated Development Environment
16	GCC	GNU Compiler Collection

1.2 Reference

1.2.1 About RA8T2

Technical information about EtherCAT is available on the ETG member site, and information about RA8T2 is available from Renesas.

Table 1.2 Technical reference for RA8T2

Document Type	Description	Document Title	Document No.
User's Manual	RA8T2 Group User's Manual Hardware	RA8T2 User's Manual Hardware	r01uh1067ej****
Datasheet	RA8T2 Group Datasheet	RA8T2 Group Datasheet	r01ds0436ej****
Quick Start Guide	EK-RA8T2 v1 Quick Start Guide	Demonstration for the EK-RA8T2 and instructions on how to customize it for application development.	r20qs0097eg****
User's Manual	EK-RA8T2 v1 User's Manual	EK-RA8T2 User's Manual Hardware	r20ut5714ej****

2. Features

This package includes firmware for the EtherCAT SubDevice stack generated by the SSC Tool for Renesas RA8T2 series processors.

This package includes the following features:

- ESM (EtherCAT State Machine)
- Mailbox protocols:
  - CoE (CAN application protocol over EtherCAT)
  - FoE (File Access over EtherCAT)
- Synchronization modes:
  - Free Run
  - Sync Manager Synchronization
  - DC Synchronization
- I/O function:
  - DIP switch input (I/O)
  - LED output (I/O)
- ETG.5003 semiconductor device profiles:
  - Common Device Profile (CDP) [ETG.5003.1]
  - Firmware update functionality [ETG.5003.2]

2.1 Folder structure relative to this application note

After extracting the package, the folder structure is as follows:

Table 2.1 Sample package overview

Item	Description
r01an8283xx0100-ek-ra8t2-ethercat-package	EK-RA8T2 EtherCAT Sample Package
├── RA8T2_EtherCAT_EK_rev0100	Archive for RA8T2 (EK board)
│ ├── common	Common resources for SSC Tool
│ │ ├── CiA402	For CiA402
│ │ │ ├── ESI	EtherCAT SubDevice Information
│ │ │ ├── Patch	Patch for this CiA402 project
│ │ │ └── SSCconfig	SSC Tool configuration file
│ │ ├── ETG5003	For ETG5003
│ │ │ ├── ESI	EtherCAT SubDevice Information
│ │ │ ├── Patch	Patch for this ETG5003 project
│ │ │ └── SSCconfig	SSC Tool configuration file
│ │ └── Replacement Files	Replacement Files
│ └── project	IDE project folder
│ ├── CiA402	For CiA402
│ │ └── e2studio	e^{2}studio project
│ └── ETG5003	For ETG5003
│ └── e2studio	e^{2}studio project
├── r01an8283ej0100-ek-ra8t2-ethercat-package.pdf	This release note
├── r01an8284ej0100-ek-ra8t2-ecat-cia402.pdf	Application note for CiA402 projects
└── r01an8285ej0100-ek-ra8t2-ecat-etg5003.pdf	Application note for ETG5003 projects

3. Requirements (Software and Hardware)

This project was developed and tested in the environment listed below using the following boards and tools.

3.1 Requirements for this Sample Package

Table 3.1 Requirements

Category	Name	Version	Description
Board	EK-RA8T2 evaluation kit	-	Renesas EK-RA8T2 - Evaluation Kit for RA8T2 MCU Group
IDE	e² studio	2026-04.2	Renesas RA Flexible Software Package (FSP)
Configurator	FSP Smart Configurator	2026-04.2	
Flexible Software Package	FSP for Renesas	v6.5.0	
GCC Compiler	GNU ARM Embedded Toolchain	13.3.Rel1	
	GNU ARM A-Profile (AArch64 bare-metal)	13.2.Rel1	
Emulator	J-Link ™	8.60	SEGGER SEGGER - The Embedded Experts - Downloads - J-Link / J-Trace
Software	SSC Tool	5.13	Beckhoff Automation ET9300 | EtherCAT Slave Stack Code | Beckhoff Worldwide
	TwinCAT3	4026.19	Beckhoff Automation TwinCAT 3.1 Build 4026 | Beckhoff Worldwide
	Tera Term	5.5.1	Tera Term Tera Term Open Source Project

4. Hardware Setup

This section describes the main hardware used in this application. For more information about the boards, refer to the respective evaluation board user manuals and schematics.

4.1 EK-RA8T2 Board

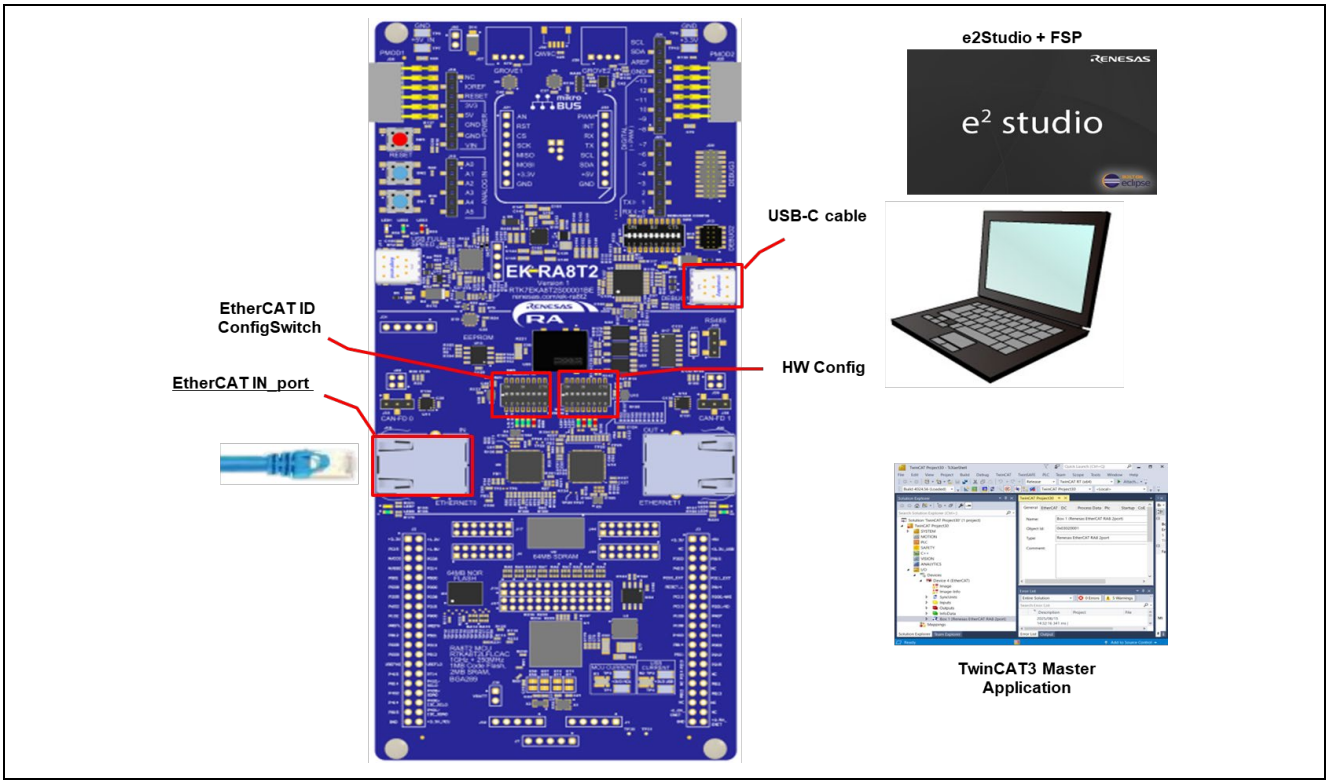


Figure 4.1. EK-RA8T2 board layout

4.1.1 Jumper and Switch Configuration

The tables below show the jumper settings.
Enable the following settings when using EtherCAT communication.

Table 4.1 Jumper configuration

Reference	Jumper Position	Description
E71	Short	Suppressing 1000Mbps communication (Eth0)
E72	Short	Suppressing 1000Mbps communication (Eth1)

Table 4.2 DIP Switch configuration

SW4	Setting	Description
SW4-1	ON	I2C mode setting
SW4-2	OFF	I2C mode setting

Table 4.3 DIP Switch configuration

SW5	Description
SW5-1,2,3,4	Lower 4 bits: input bit
SW5-5,6,7,8	Upper 4 bits: EtherCAT device ID

5. Setting up the Host

5.1 Copying the ESI Files

Before starting TwinCAT, copy the ESI file(.xml) included in the release folder to the TwinCAT destination folder.

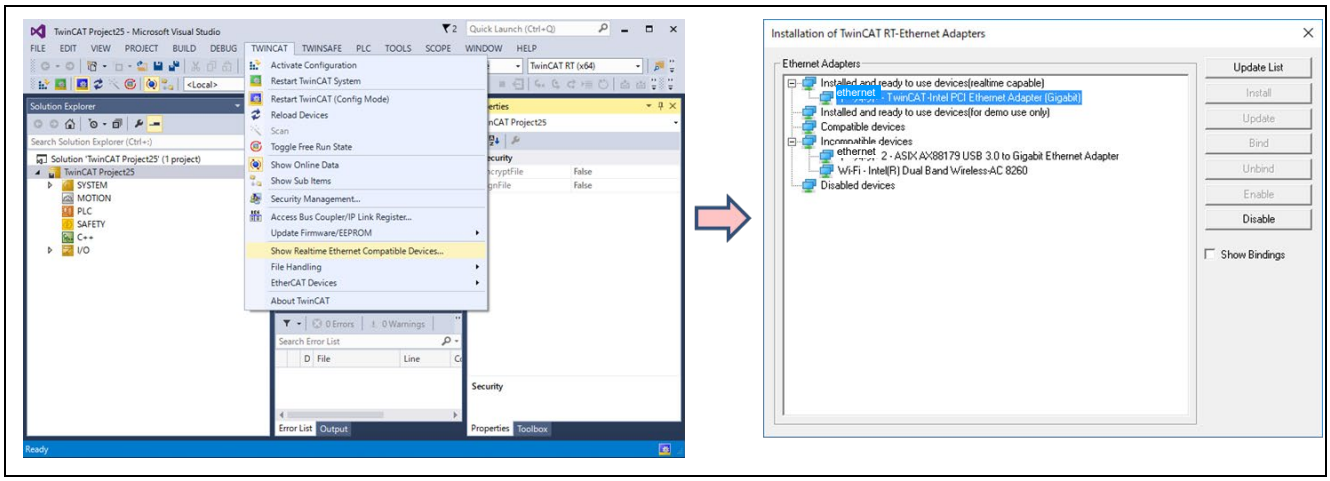
The release folder:
"RA8T2_EtherCAT_EK_rev0100\common\ETG5003\ESI"

The ESI file name:
"Renesas EtherCAT RA8 ETG5003.xml"

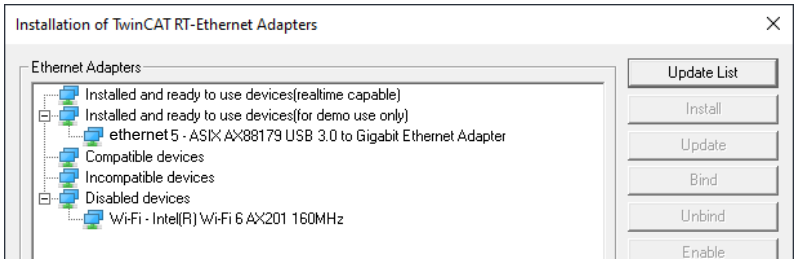
The TwinCAT destination folder:
"TwinCAT\3.x\Config\IO\EtherCAT"

5.2 Installing the Driver

Add the EtherCAT driver for TwinCAT (first time only).
From the Start menu, select [TwinCAT3] → [Show Realtime Ethernet Compatible Devices…].
Select the connected Ethernet port from the list of communication ports and install the driver.
If the selected Ethernet adapter appears under [Installed and ready to use devices (realtime capable)], the installation is complete.



Note: If you are not using an Intel NIC, the Ethernet adapter may appear under [Installed and ready to use devices (for demo use only)].



5.3 Generating the EtherCAT SubDevice Stack Code

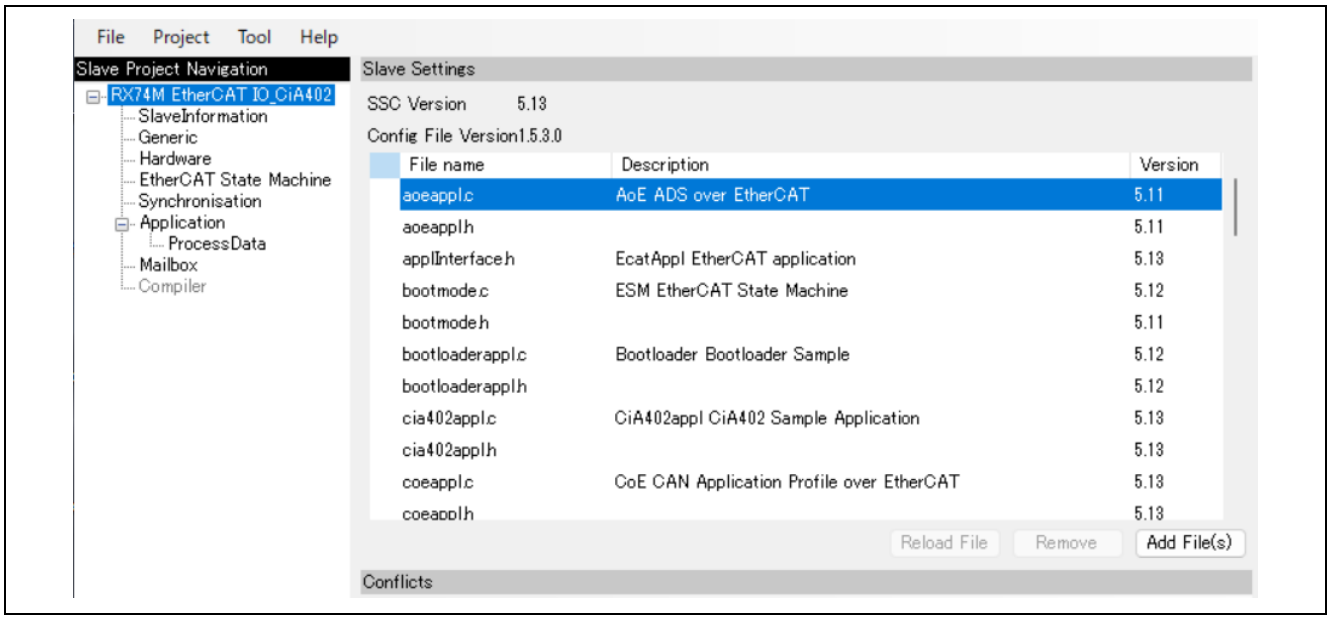
This chapter describes how to generate the EtherCAT SubDevice Stack Code by using the SSC tool.

* EtherCAT SubDevice stack code is not included in this sample project.
* "EtherCAT SubDevice Stack Code (SSC) Tool" is required to generate EtherCAT SubDevice stack code.
* SSC Tool is available from the ETG Association.

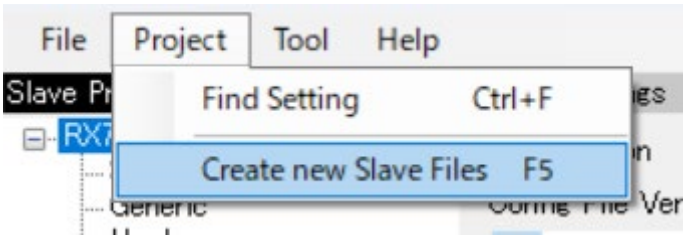
EtherCAT Technology Group | EtherCAT Slave Stack Code (SSC) ET9300

1. Start the SSC Tool and open the following folder.
   "RA8T2_EtherCAT_EK_rev0100\common\ETG5003\SSCconfig"

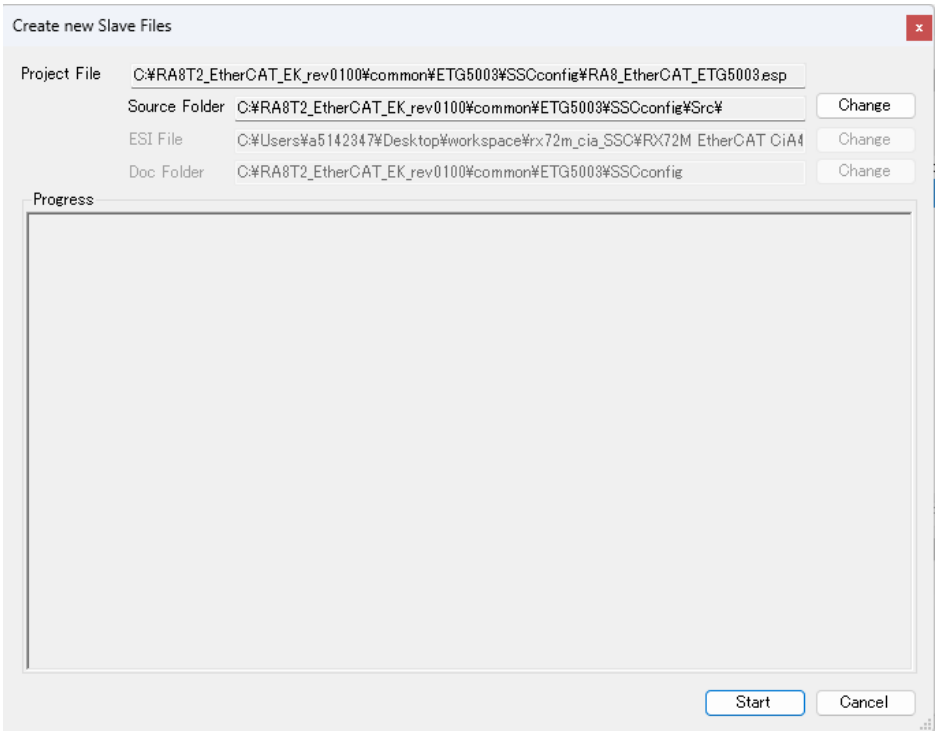
   The following window opens.



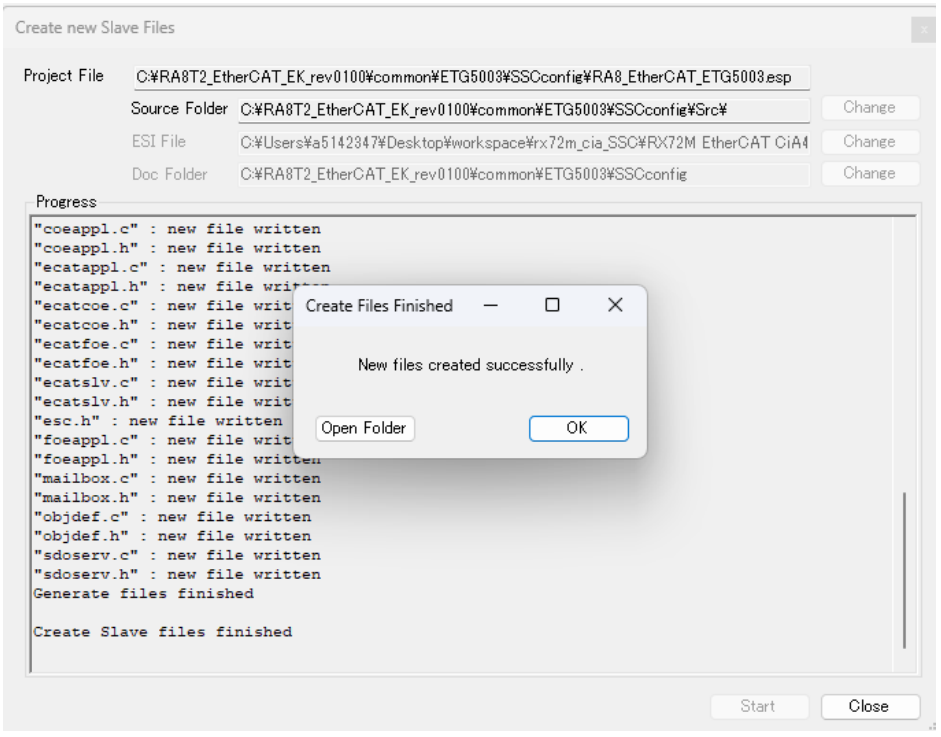
2. Select Project > Create new SubDevice Files.



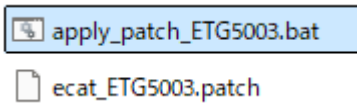
3. Click the [Start] button to start creating the EtherCAT SubDevice Stack Code.



4. When a message “New file created successfully” appears, file creation is complete, and the source files are located in the following folder.
“RA8T2_EtherCAT_EK_rev0100\common\ETG5003\SSCconfig\Src”



5. Run the patch file below to move the SSC code.
   "RA8T2_EtherCAT_EK_rev0100\common\ETG5003\Patch\apply_patch_ETG5003.bat"



6. The generated EtherCAT SubDevice stack code is moved to the EtherCAT application source folder.

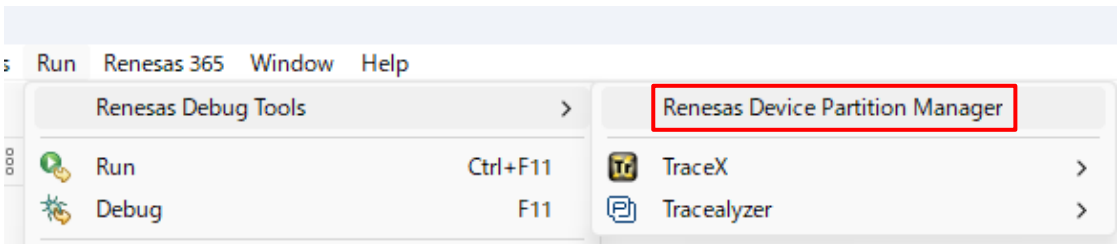
   Source folder: (code generated in the Src folder by SSC)
   "*RA8T2_EtherCAT_EK_rev0100\common\ETG5003\SSCconfig\Src*"

   Copy destination folder:
   "*RA8T2_EtherCAT_EK_rev0100\project\ETG5003\e2studio\src\ethercat\beckhoff\Src*"

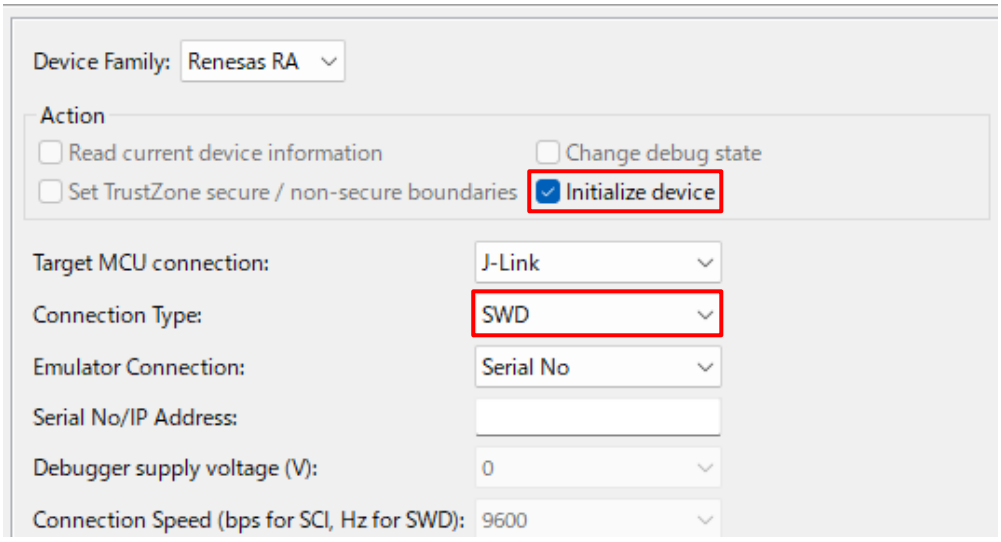
5.4 Setting Up the Renesas Device Partition Manager (RDPM)

To run an EtherCAT ETG5003 project on the EK-RA8T2 board, the entire code MRAM must be configured as a secure area in advance.
Use Renesas Device Partition Manager (RDPM) to configure the secure area.

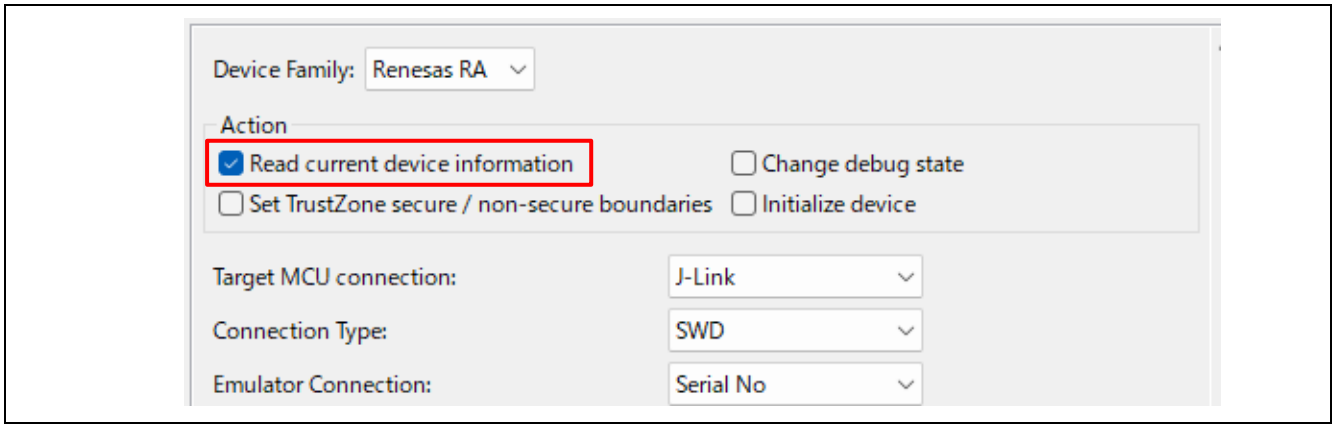
1. Launch Renesas Device Partition Manager by clicking [Run] → [Renesas Debug Tools] → [Renesas Device Partition Manager].

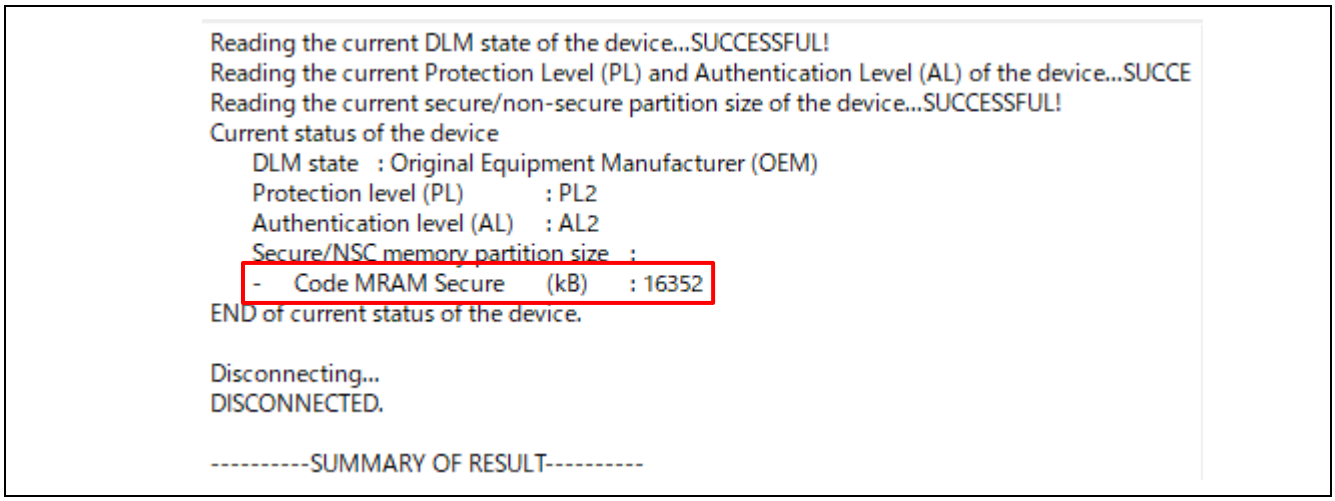


2. Configure the following settings in the RDPM window, and then click the [Run] button in the lower-right corner. The execution log will be displayed at the bottom of the screen; if "SUCCESSFUL!" is displayed, it was successful.
   Action: Select “**Initialize device**”.
   Connection Type: Select "**SWD**"



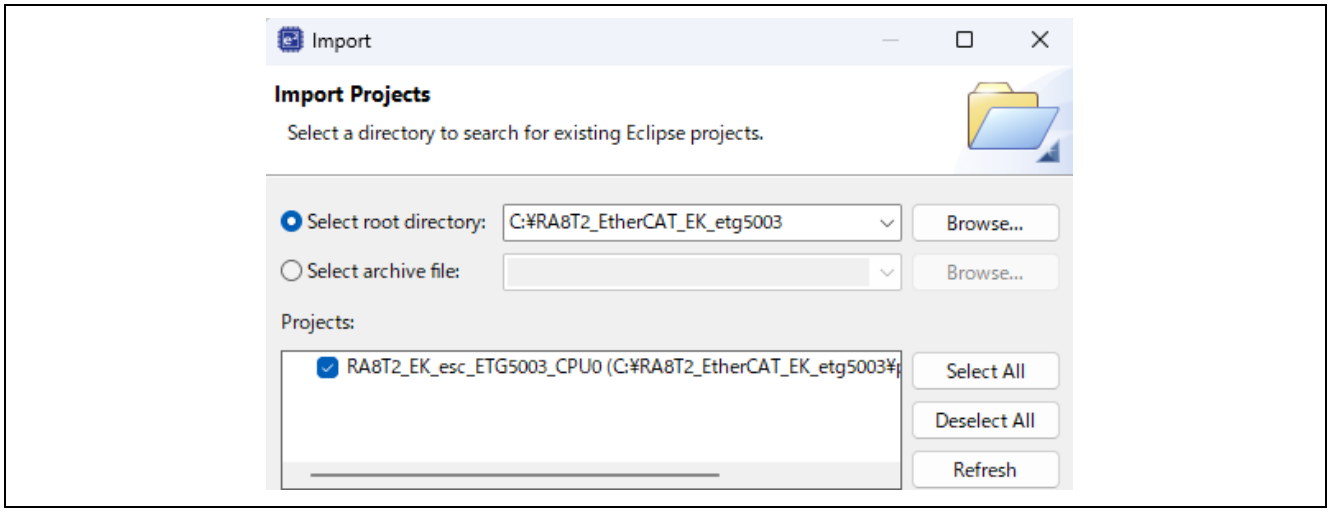
3. Select “Read current device information”, and then click [Run].
   If the log shows “Code MRAM Secure (kB): 16352”, the setting was applied successfully.



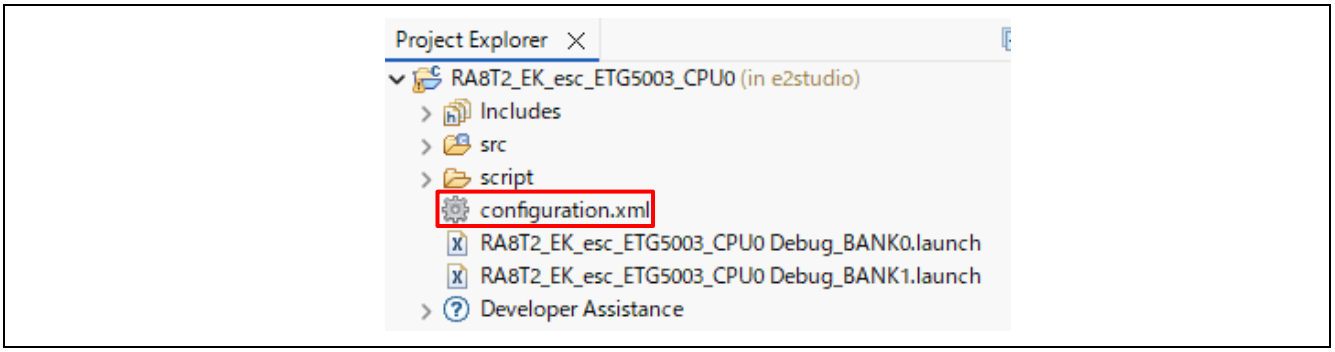


5.5 Setting Up the Sample Code for e² studio

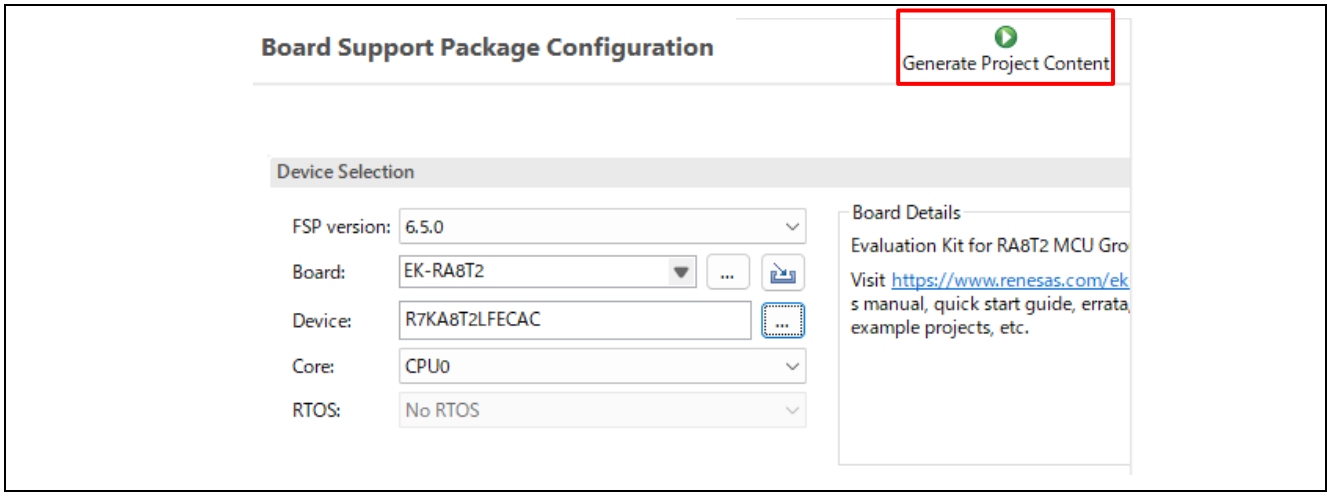
1. Import the sample project. After starting the program, select [File] → [Import] → [Existing Projects into Workspace]. Check "Select root directory", select the **"RA8T2_EtherCAT_EK_rev0100\project\ETG5003"** folder, select the project, and then click [Finish].



2. Open "configuration.xml" in the " **RA8T2_EK_esc_ETG5003_CPU0**" project.



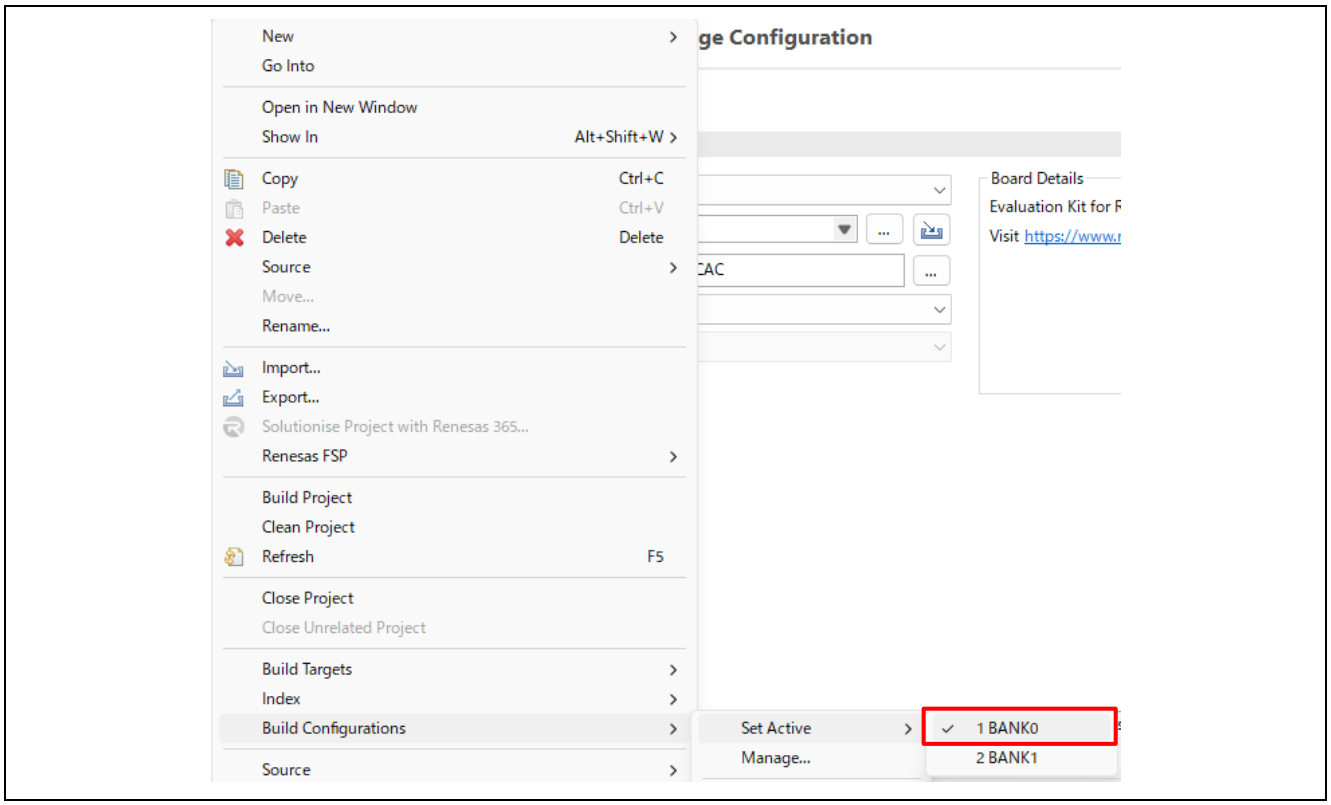
3. Generate the code by selecting "**Generate Project Content**".



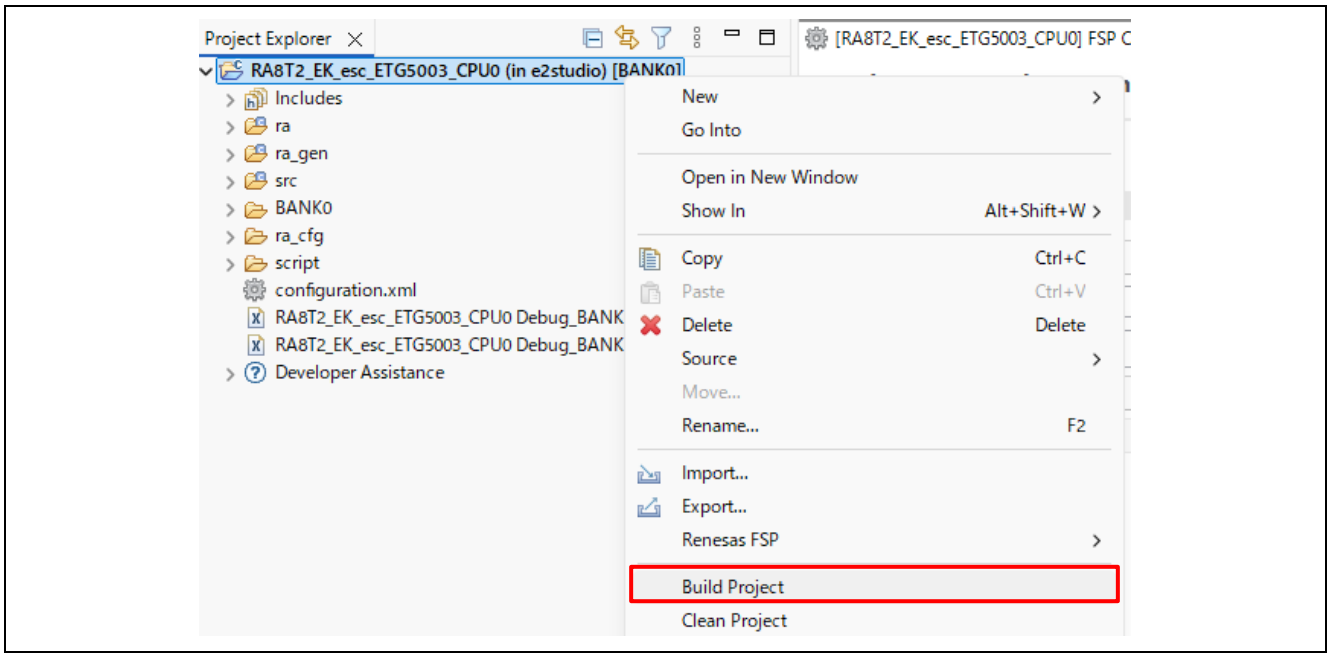
4. Replace the PHY configuration file located in the "ReplacementFiles" folder.

   “**RA8T2_EtherCAT_EK_rev0100\common\ReplacementFiles\ra\fsp\src\r_ethercat_phy**”
   copy to
   “**RA8T2_EtherCAT_EK_rev0100\project\ETG5003\e2studio\ra\fsp\src\r_ethercat_phy**”

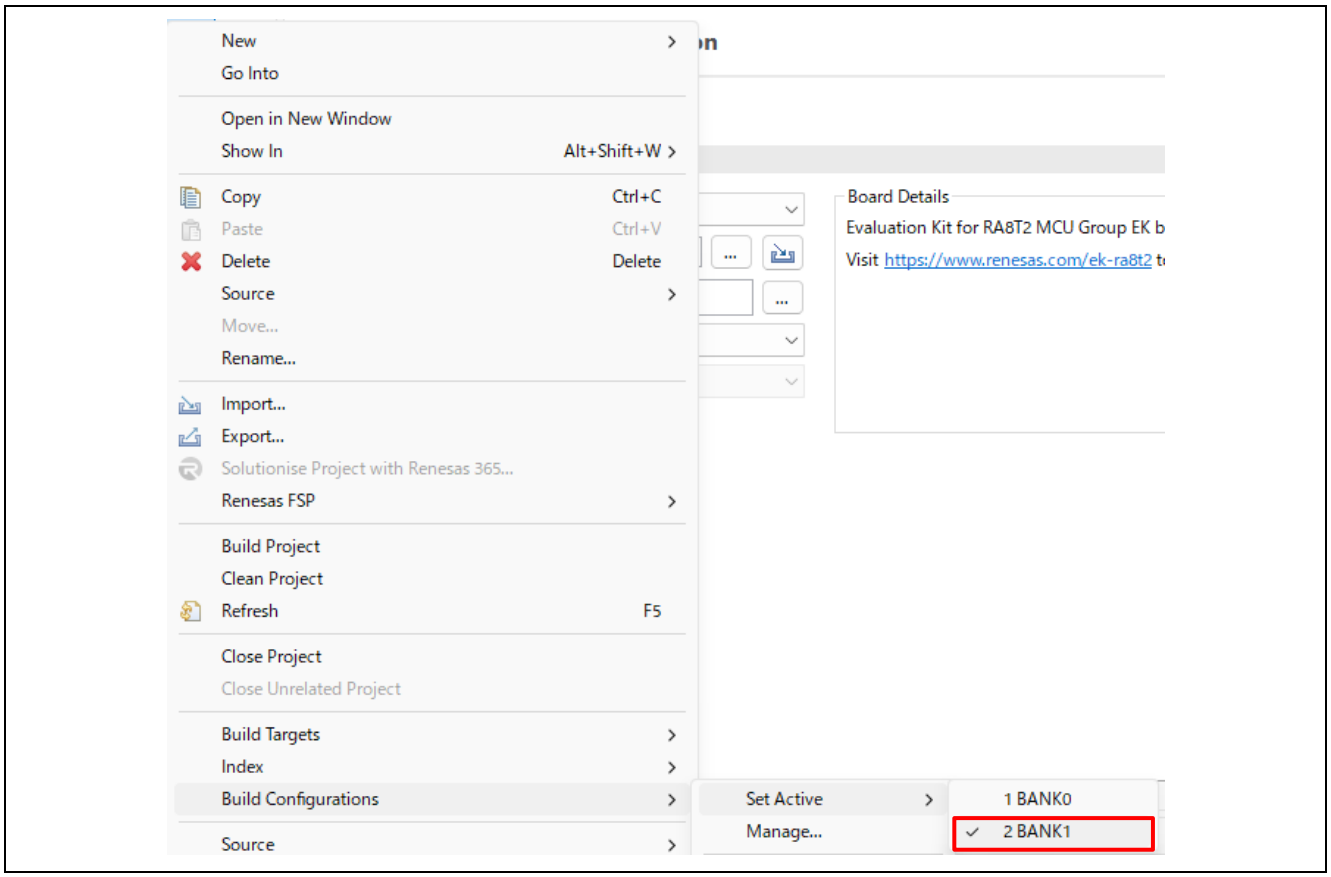
5. Select the project and confirm that “**BANK0**” is selected.



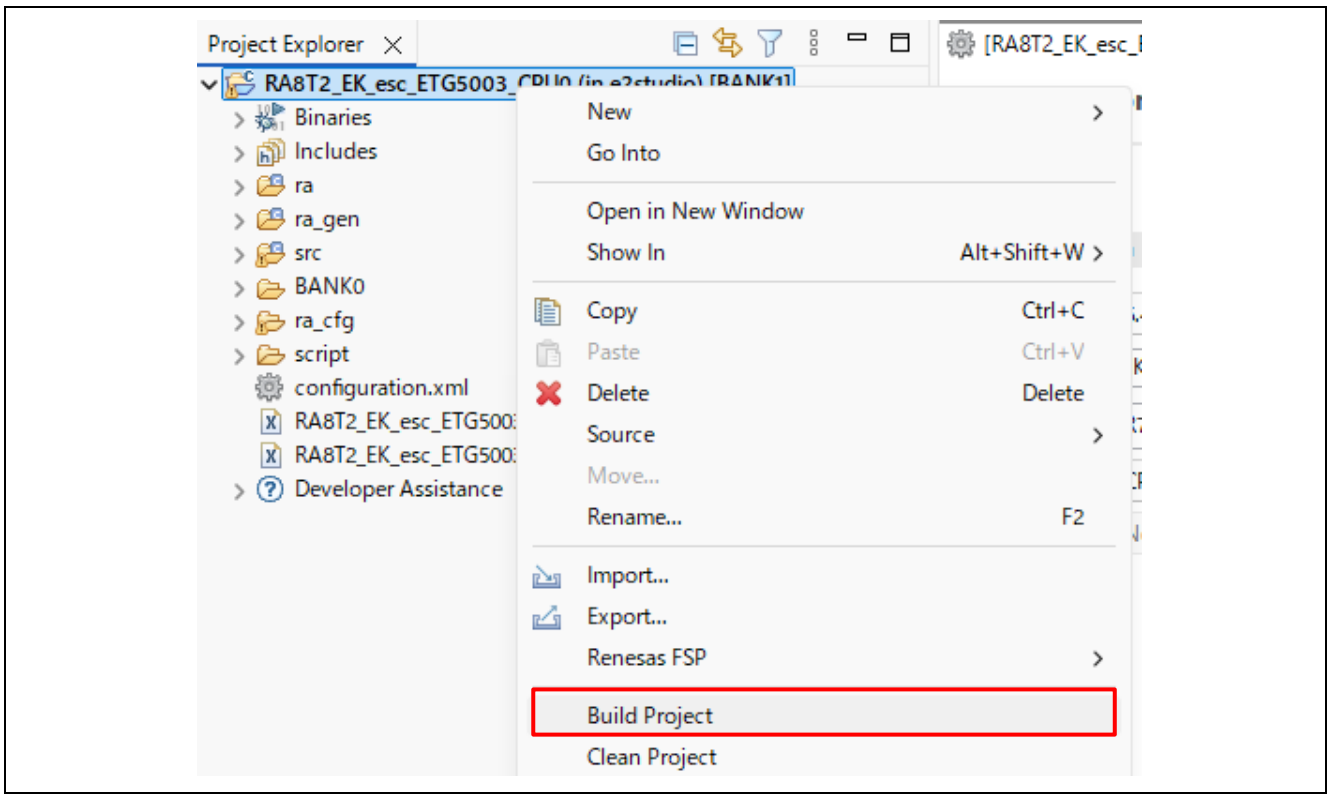
6. Build the **BANK0** project.



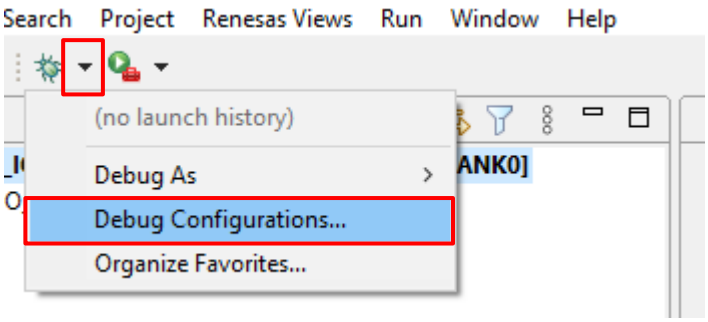
7. Select the project and change from “**BANK0**” to “**BANK1**”.



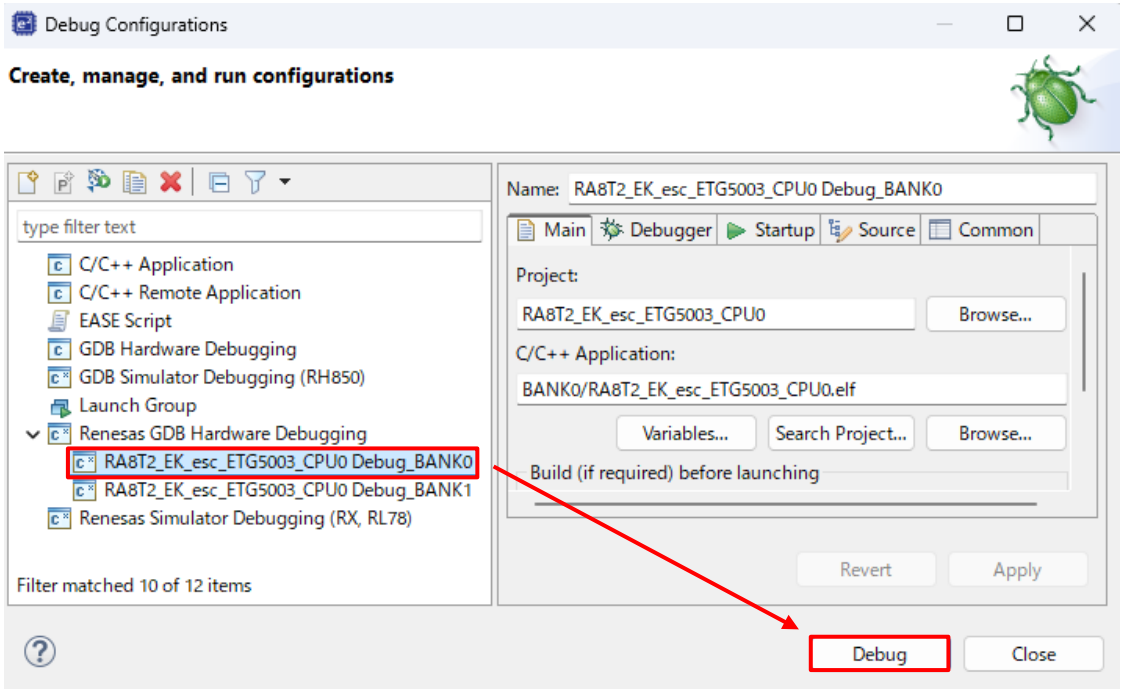
8. Build the **BANK1** project.



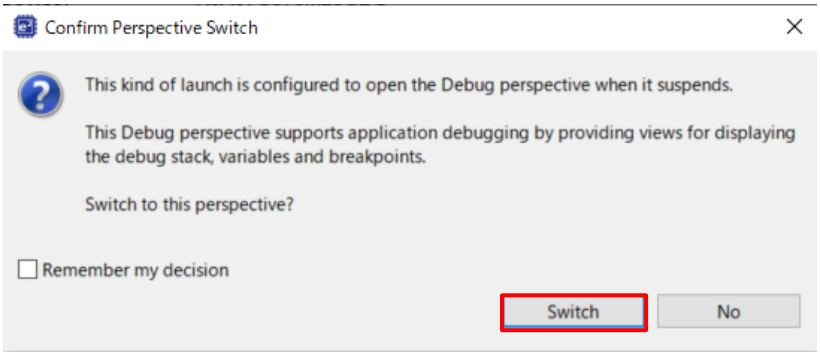
9. After connecting the board and J-Link, start debugging and downloading by following the procedure below.
   Push the triangle next to the [**Debug**] button, then push [**Debug Configurations**].



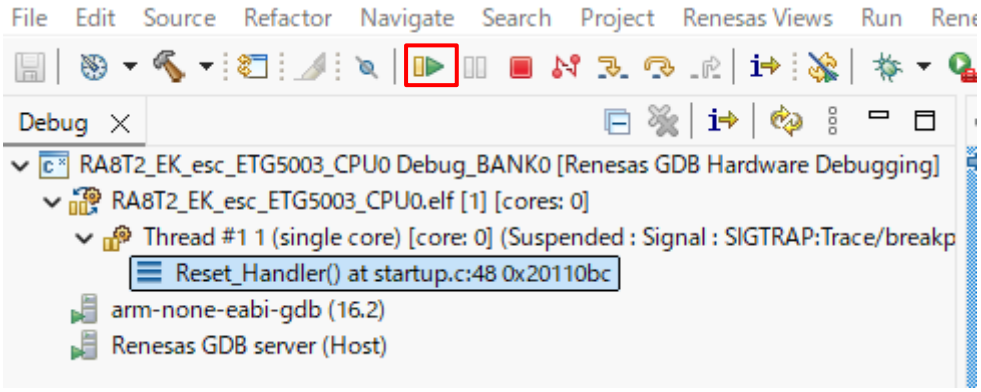
[**Renesas GDB Hardware Debugging**] → [**RA8T2_EK_esc_ETG5003_CPU0 Debug_BANK0**] item, then press [**Debug**].



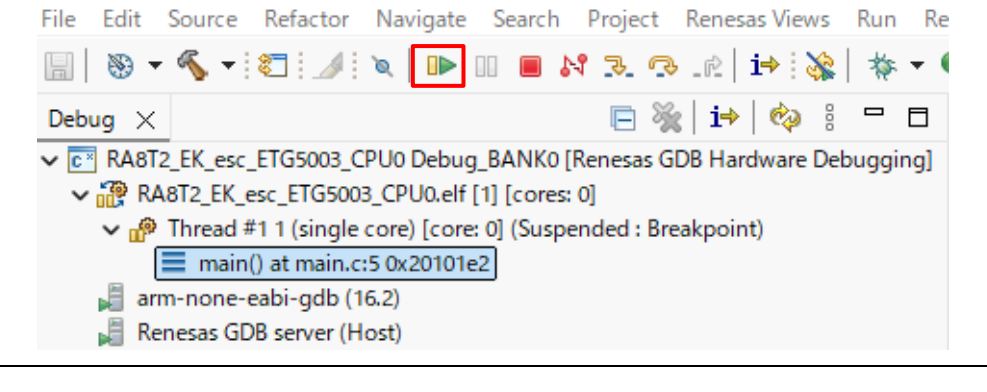
The following dialog appears. Switch to the Debug Perspective.



10. When debugging starts, the program is interrupted at "hal_entry();" in main.c.
    Press the "**Resume**" button.



11. Press the “Resume” button again to continue program execution.



6. Demonstration of the Sample Application

Before starting this chapter, power on the EK-RA8T2 board and connect the PC/PLC to the ETH0 connector. For details, refer to Chapter 4.

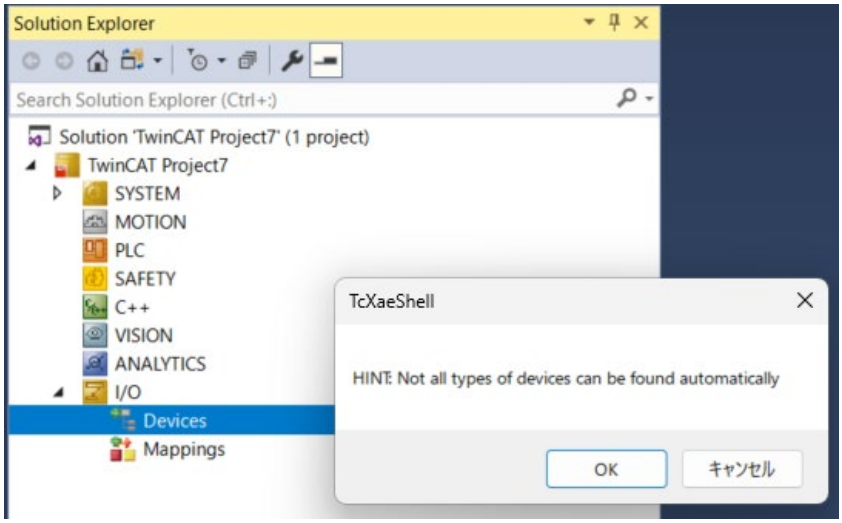
6.1 Connecting to TwinCAT3

Start TwinCAT3 using the procedure described below.
From the start menu, select [**Beckhoff**] → [**TwinCAT3**] → [**TwinCAT XAE**].

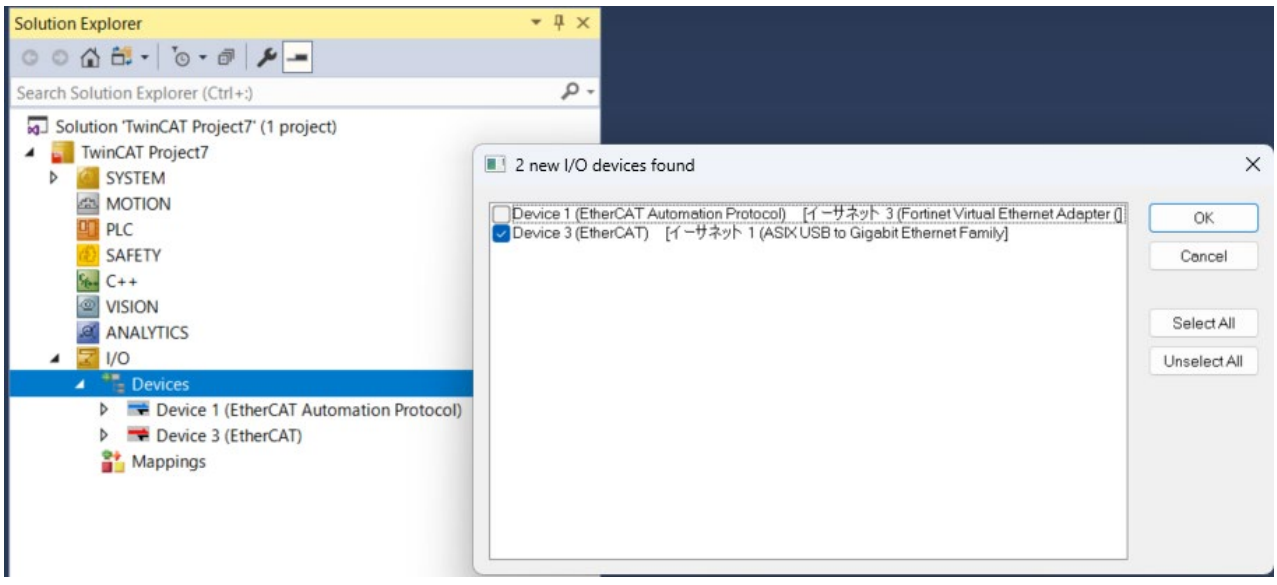
After starting the program, select [**File**] → [**New**] → [**Project**], create a new project of the TwinCAT XAE Project type. The subsequent procedure is described below.

6.1.1 Scanning I/O Devices

1. (Scan for devices): Under Solution Explorer → [**I/O**] → [**Devices**], select “Scan” as in the figure below.

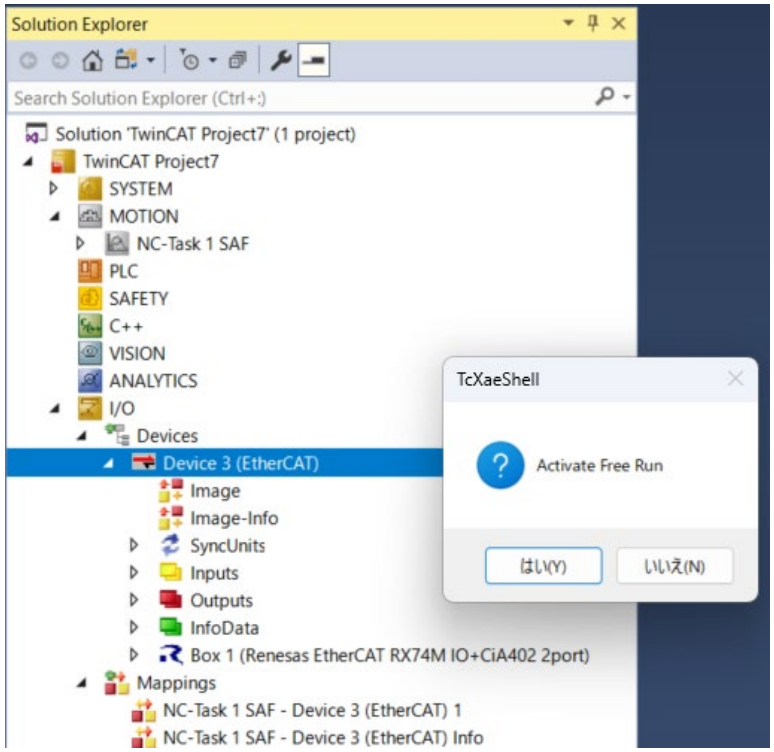


2. (Selecting port): The EtherCAT port will be displayed as shown below. Select it and click [OK].

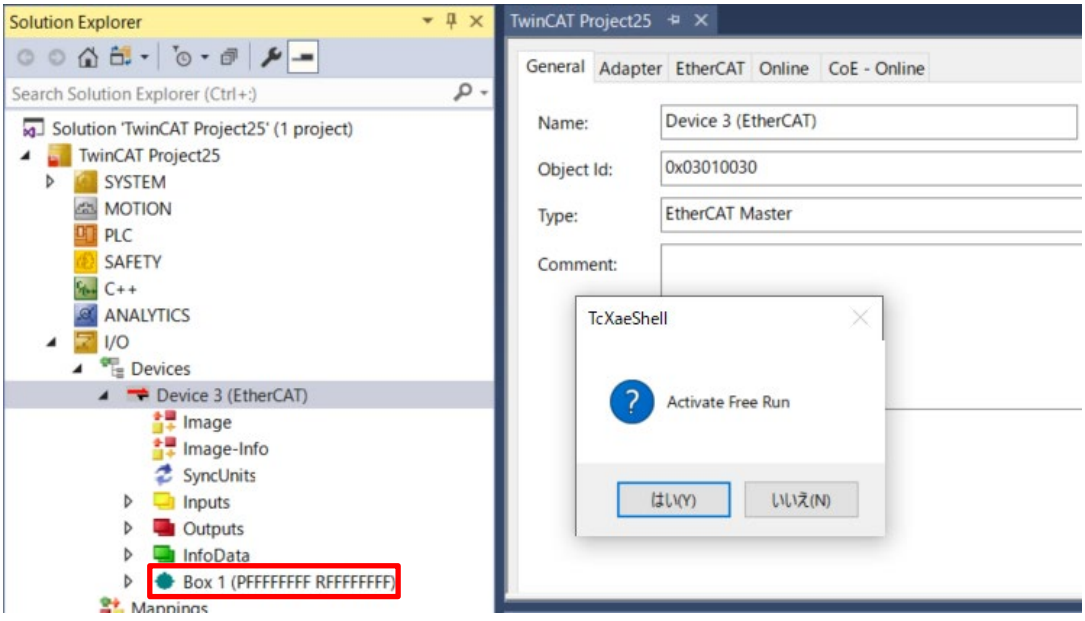


Note: If a valid SubDevice exists on the network, the EtherCAT controller will display the list.

3. (Activate SubDevice): The SubDevice is displayed in one of the boxes,. In our case, “Renesas EtherCAT” is in Box1Box 1, shown in Figurethe figure below. Click [Activate Free Run].



4. If the detected device is "**Box1 (PFFFFFFFF)**" as shown in the figure, the ESI file has not been downloaded, so proceed to 6.1.2, Updating EEPROM Data. If the detected device showsdisplays the desired ESI file namefilename, such as "Box1 (Renesas EtherCAT xxxx)", select Free run.



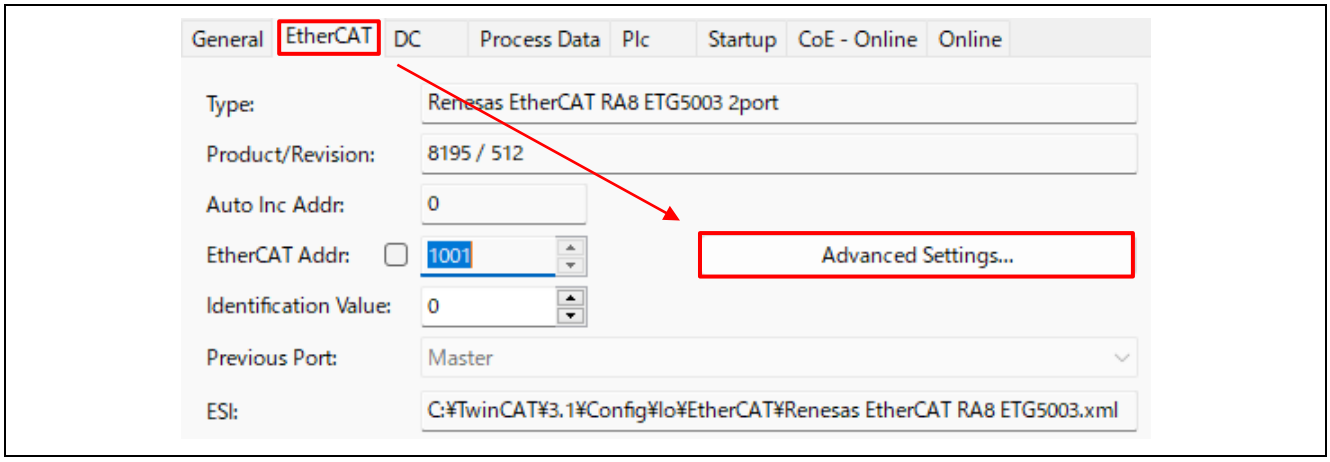
6.1.2 Updating EEPROM Data

If data from another application has already been written to the EEPROM, overwrite it.
The following shows the procedure for replacing the data on the EEPROM:

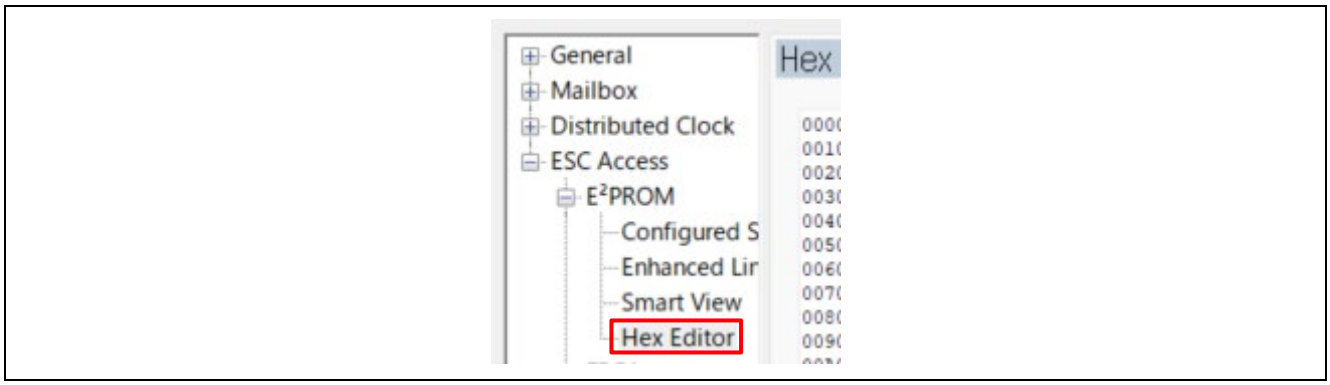
1. Double-click [**Box 1**] to display a panel on the right side of the window.

2. Select the [**EtherCAT**] tab.

3. Click the [**Advanced Setting**] button.

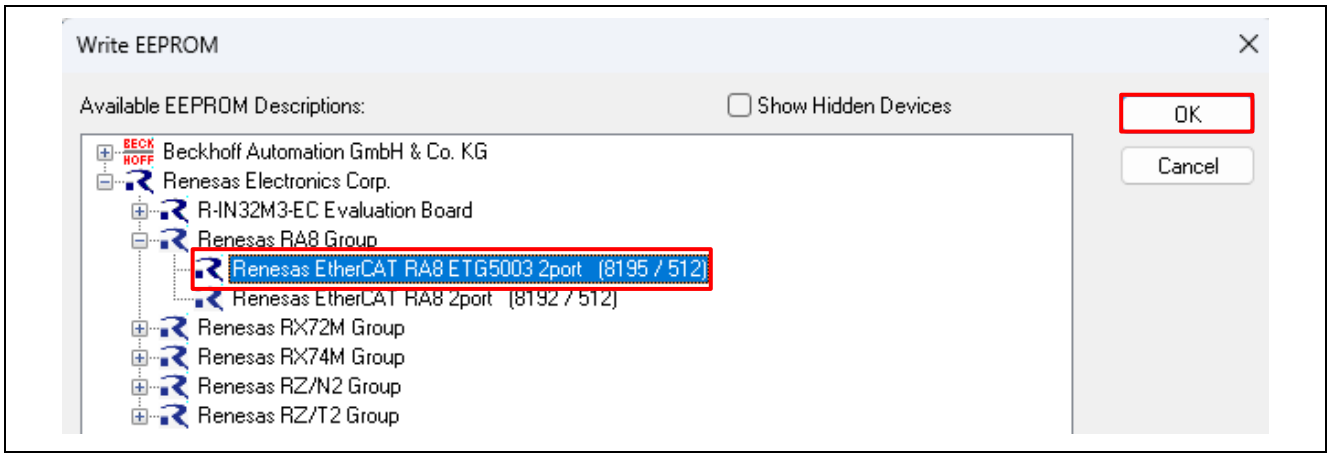


4. Select [**ESC Access**] → [**EEPROM**] → [**Hex Editor**].



5. Select [**Download from List**]

7. Select the ESI File
   "*RA8T2_EtherCAT_EK_rev0100\common\ETG5003\ESI\Renesas EtherCAT RA8 ETG5003.xml*"



8. Click OK and download the file.

9. Scan the device again to recognize the ESI file.

Option A - Create ESI binary file from ESI XML and download.

1. SSC Tool → [**Tool**] → [**EEPROM Programmer**].
2. [**FILE**] → [**OPEN**] →Browse and select the ESI file.
3. [**FILE**] → [**Save AS**] → Select type as binary.
4. A binary file will be generated in the specified folder.
5. [**Read from File**] Select the ESI binary file → [**Download**].
6. Confirm the write status using the [**Upload**] option.

After the data is replaced, restart the RA8T2 (by turning it off and on, or resetting it) so that the new data is applied to the microcomputer. Execute [Restart TwinCAT System].

6.2 Checking the File over EtherCAT (FoE) Profile

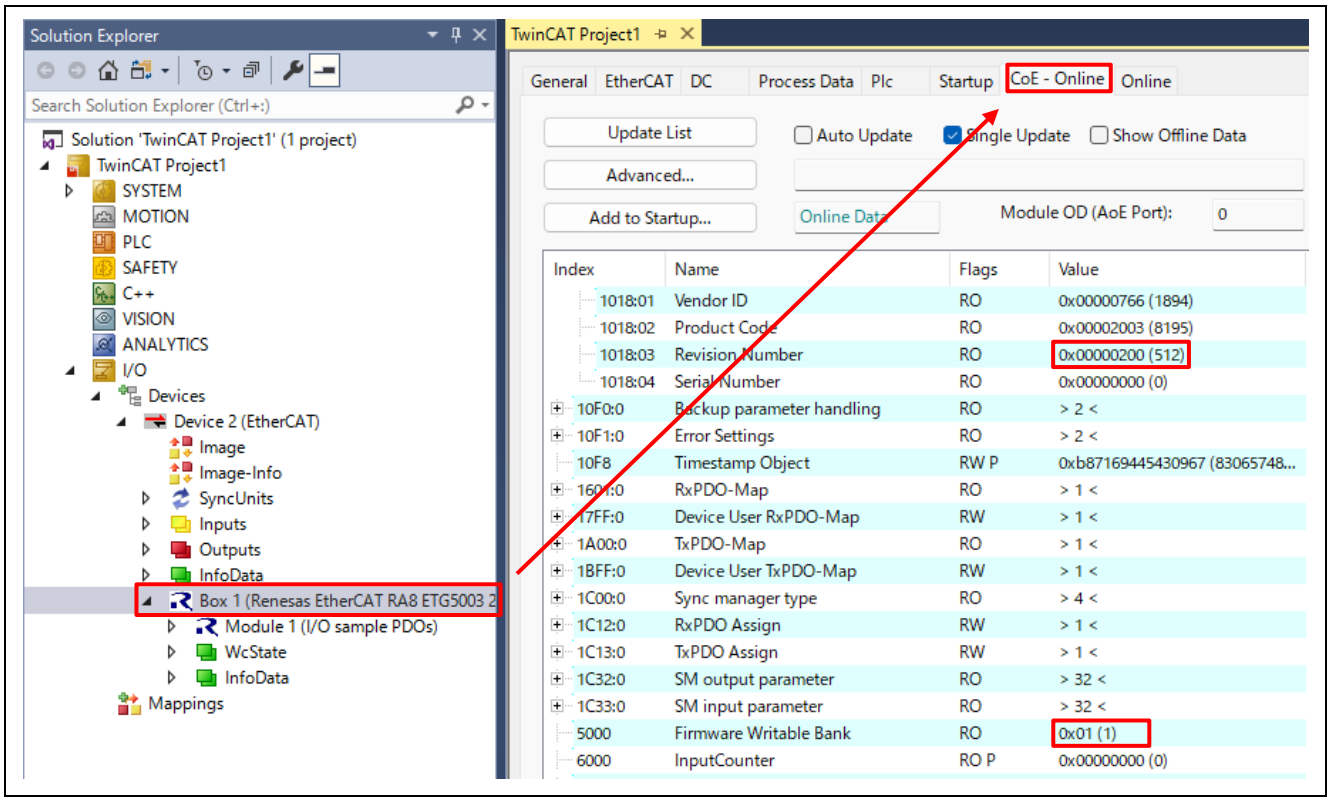
This chapter describes how to check the operation of the File over EtherCAT (FoE) profile.
Follow the steps below to rewrite the firmware.

Stop debugging, then press the reset button on the board to reset the RA8T2 microprocessor.

6.2.1 Confirm the Initial Version

1. Confirm [Index 1018:0 Identity Object] on the [CoE-Online] tab.

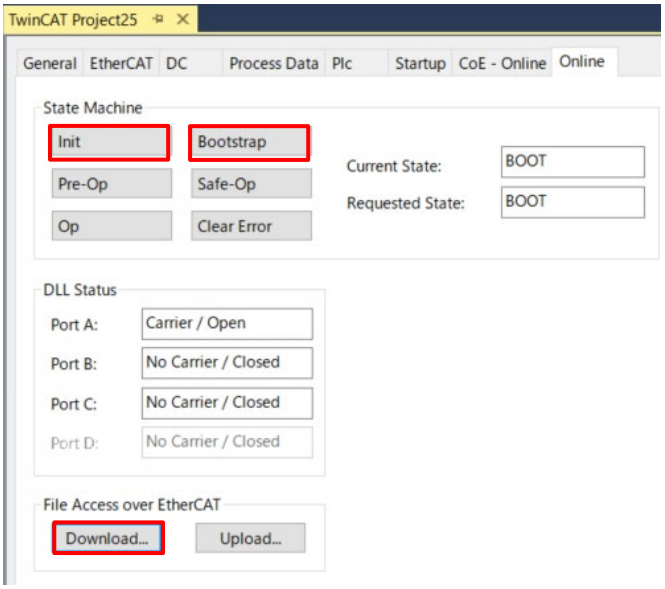
2. Confirm [1018:03 Revision Number] is **0x00000200.** Currently, Bank0 is running.
   Confirm [5000:00 Firmware Writable Bank] is **0x01.** At this time, you can update the firmware in BANK1.



6.2.2 Firmware update

1. Select the [**Online**] tab.

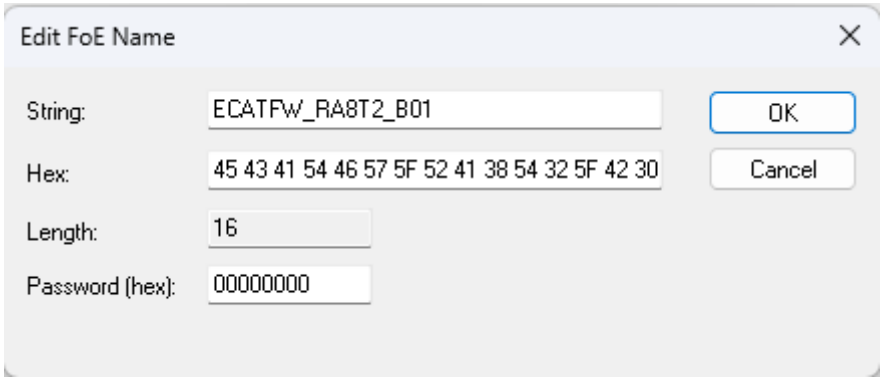
2. Press the [**Init**] → [**Bootstrap**] button to transition to boot mode.



3. Click the [**Download**] button and select the following download file for BANK1.

   "**RA8T2_EtherCAT_EK_rev0100\project\ETG5003\e2studio\BANK1\ECATFW_RA8T2_B01.efw**"

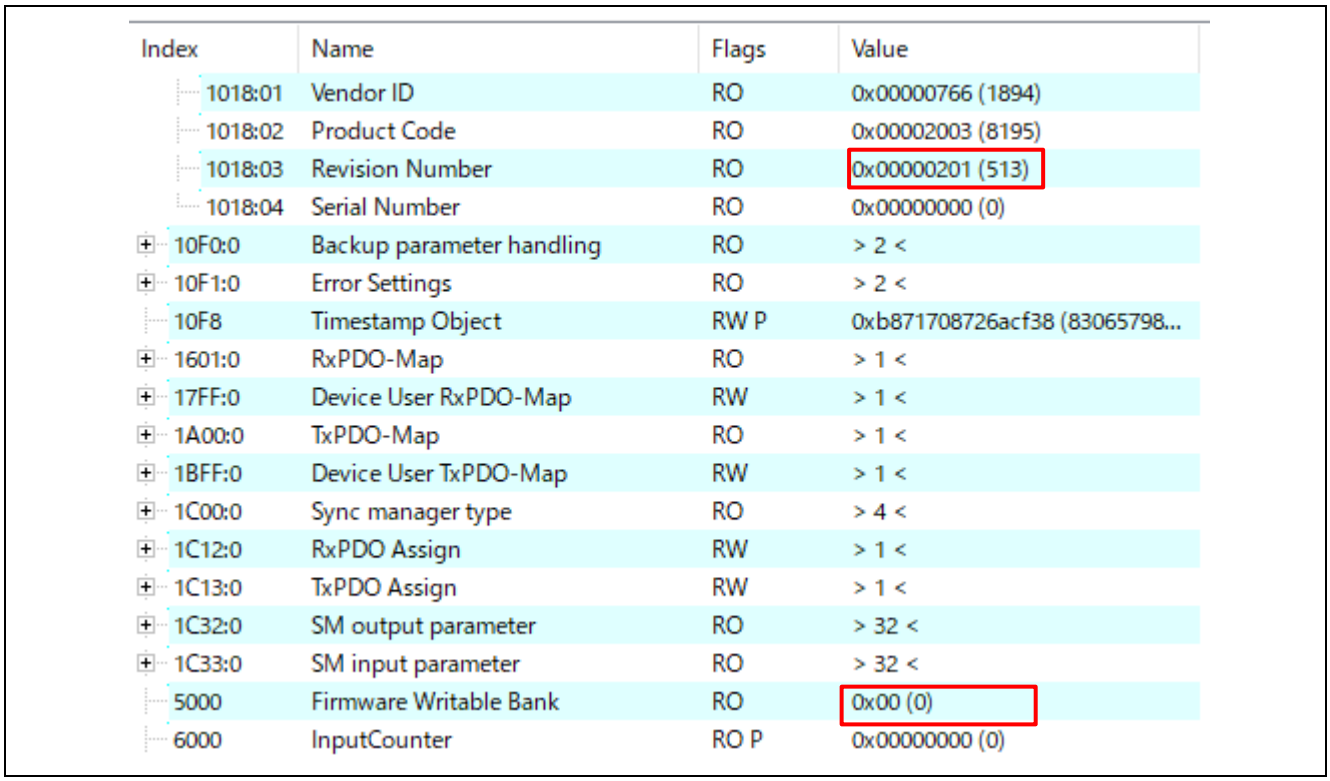
4. The attributes of the update file are displayed. If no changes are required, leave them as is.



5. After the download is complete, press the [**Init**] button.
   The EK board will be reset by software and the program will restart.

6.2.3 Check the Updated Version

1. Check the downloaded firmware.
   Confirm that the [1018:03 Revision Number] is **0x0000020**1 on the [CoE-Online] tab. Currently, bank1 is running.
   Confirm [5000:00 Firmware Writable Bank] is **0x00.** At this time, you can update the firmware in BANK0.



Index	Name	Flags	Value
1018:01	Vendor ID	RO	0x00000766 (1894)
1018:02	Product Code	RO	0x00002003 (8195)
1018:03	Revision Number	RO	0x00000201 (513)
1018:04	Serial Number	RO	0x00000000 (0)
10F0:0	Backup parameter handling	RO	> 2 <
10F1:0	Error Settings	RO	> 2 <
10F8	Timestamp Object	RW P	0xb871708726acf38 (83065798...
1601:0	RxPDO-Map	RO	> 1 <
17FF:0	Device User RxPDO-Map	RW	> 1 <
1A00:0	TxPDO-Map	RO	> 1 <
1BFF:0	Device User TxPDO-Map	RW	> 1 <
1C00:0	Sync manager type	RO	> 4 <
1C12:0	RxPDO Assign	RW	> 1 <
1C13:0	TxPDO Assign	RW	> 1 <
1C32:0	SM output parameter	RO	> 32 <
1C33:0	SM input parameter	RO	> 32 <
5000	Firmware Writable Bank	RO	0x00 (0)
6000	InputCounter	RO P	0x00000000 (0)

Note: When updating to bank0, use the following efw file.

“**RA8T2_EtherCAT_EK_rev0100\project\ETG5003\e2studio\BANK0\ ECATFW_RA8T2_B00.efw**”

7. Software Specifications

7.1 Details of Firmware Update

The sample program can be used to update the firmware of the SubDevice as described below.

A SubDevice vendor can provide an updated firmware file and password to a user, and the user can download the firmware to a SubDevice by using the FoE from a MainDevice such as TwinCAT.

The firmware update file includes a checksum, which allows checking the validity of received data.

The updated firmware is written to a different area within the code MRAM than the factory default firmware. After the update, the user application program for the updated firmware will start.

If the update fails, the device will boot up with the previous firmware.

7.1.1 Procedure for Updating the Firmware

This section describes the procedure for updating the firmware for a SubDevice and operations of the EtherCAT MainDevice and SubDevice during the procedure. "Function" column indicates which program implements each SubDevice operation.

Table 7.1 Procedure for Updating the Firmware

No	Master/User	Slave	Function		
			SSC	FW boot loader	FW updater
1	Requset BOOT	Confirm BOOT	✔		
2	Download new slave FW	Download new slave FW			
		(1) Check filename			✔
		(2) Check password			✔
		(3) Compare checksum of receiving data			
		(4) Write file data to serial flash			✔
3		Update SII			✔
4	Request INIT	Reboot			
		(1) download new firmware to internal RAM		✔	
		(2) Start new FW		✔	
5	Request PREOP	Check if SII and firmware match	✔		
6		Confirm PREOP	✔		
7	User: Check firmware version				
8	Request SAFEOP	Confirm SAFEOP	✔		
9	Request OP	Confirm OP	✔		

1. Request the BOOT state.
   Make the transition to the BOOT state for execution of the FoE.

2. Download new SubDevice firmware.

3. Download the updated firmware from the MainDevice.
   The SubDevice checks if (1) the filename and (2) the password are correct. If they are correct, it (3) compares the checksum of the received data with the calculated checksum and (4) writes data to the serial flash ROM.

4. Update the EEPROM.
   Write the revision number of the new firmware to the EEPROM.

5. Request the INIT state.
   After the transition from the BOOT to the INIT state, the SubDevice is rebooted, and (1) downloads the program code from the serial flash ROM to the internal SRAM and then (2) operates with the new firmware.

5. Request the PREOP state.
   Check whether the revision number in the EEPROM matches the firmware's revision number.

6. Confirm the PREOP state.
   Confirm the transition to the PREOP state.

7. User: Check firmware version.
   The user can check whether the firmware has been updated to the new version by reading the value at 0x100A through the CoE object.
   The user can check the revision number by reading the value at 0x1018:03.

8. Request the SAFEOP state.
   Make the transition to the SAFEOP state.

9. Request the OP state.
   Make the transition to the OP state.

7.1.2 Hardware Configuration of the Sample Program

7.1.2.1 Booting from the Serial Code MRAM

To start the boot loader in the code MRAM

7.1.2.2 Memory Map of Code MRAM

The use of the MRAM code is divided into three areas.

Table 7.2 Classification of the code MRAM Areas

Address Range	Name (Size)	Description
0200_0000H to 0200_1FFFH	Bank0 Startup area (8KB)	Startup program area for the BANK0
0200_2000H to 0200_3FFFH	Bank1 Startup area (8KB)	Startup program area for the BANK1
0200_4000H to 0200_4FFFH	Backup parameter area (4KB)	Backup parameter area for SEMI profile
0200_5000H to 0208_27FFH	BANK0 area (502KB)	Factory-default firmware area to be written by a serial flash ROM writer, ICE, etc.
0208_2800H to 020F_FFFFH	BANK1 area (502KB)	Updating firmware area to be written by the FoE

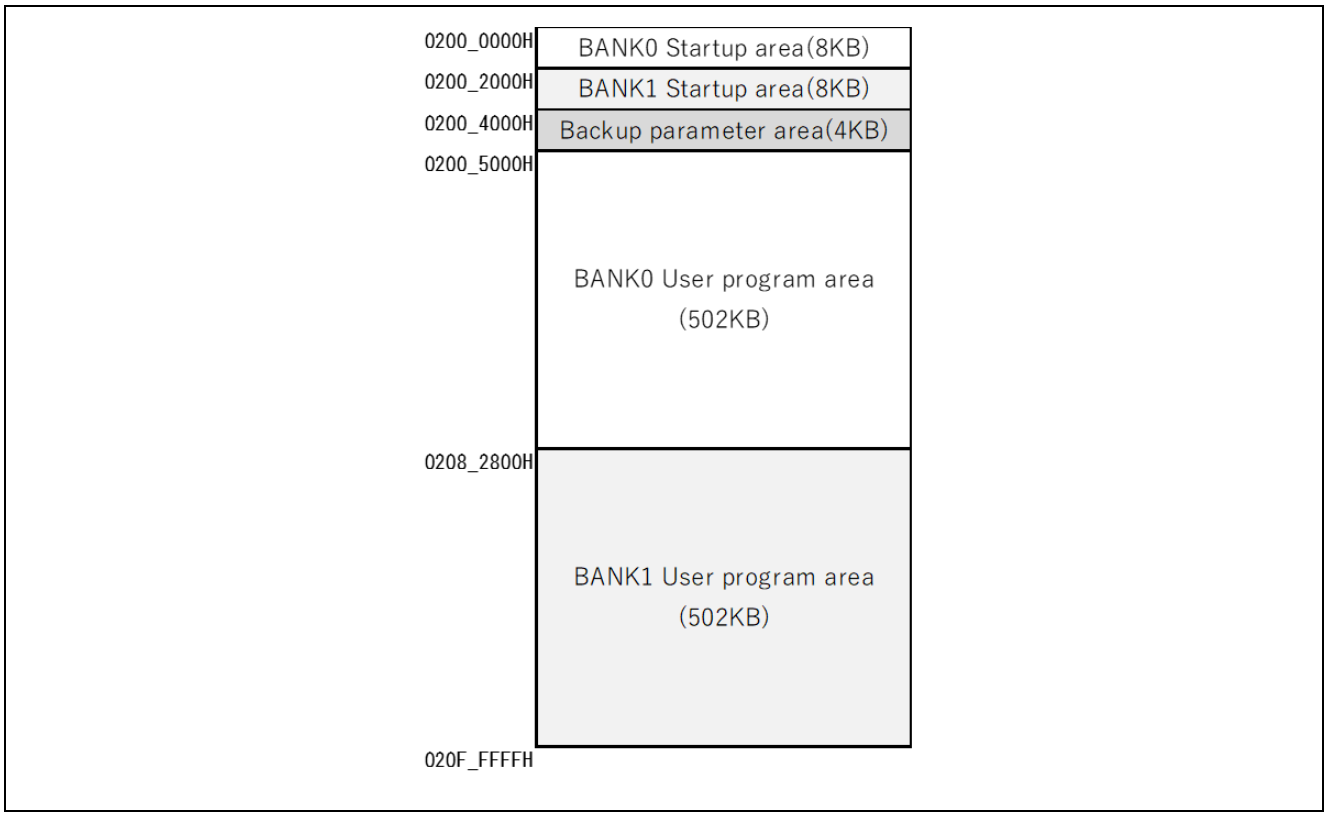


Figure 7.1. Memory Map

7.1.2.3 Overview of Bank0 Boot Operations

The following describes the boot operation for booting the factory-default firmware written to bank0 through the procedure illustrated in Figure 7.2. Boot Operations.

The following describes the RA8T2's boot function.

SAS.BTFLG=1 selects the BANK0 startup area. In the sample program, the vector table is placed in the BANK0 startup area. After booting, execution jumps to the reset handler address located in the BANK0 user program and performs the startup process. The startup process copies the user program to SRAM. To support rewriting the code in MRAM via firmware updates, the user program runs from SRAM.

To update BANK1 with new firmware, rewrite the BANK1 startup area and the BANK1 user program area. If the update completes successfully, set SAS.BTFLG=0. On reboot, the startup area is switched, so the BANK1 startup area is selected and the BANK1 user program starts. If the firmware update is interrupted for any reason, SAS.BTFLG is not changed, and the startup area is not switched. Therefore, even if an interruption occurs during the rewrite of BANK1, the BANK0 user program will run at the next boot.

For further firmware updates, you can alternatively use the areas in the order BANK0 → BANK1 → BANK0.

For details of the startup-area selection function using SAS.BTFLG, refer to the RA8T2 Group User's Manual: Hardware, "58.9.3 Startup Program Protection."

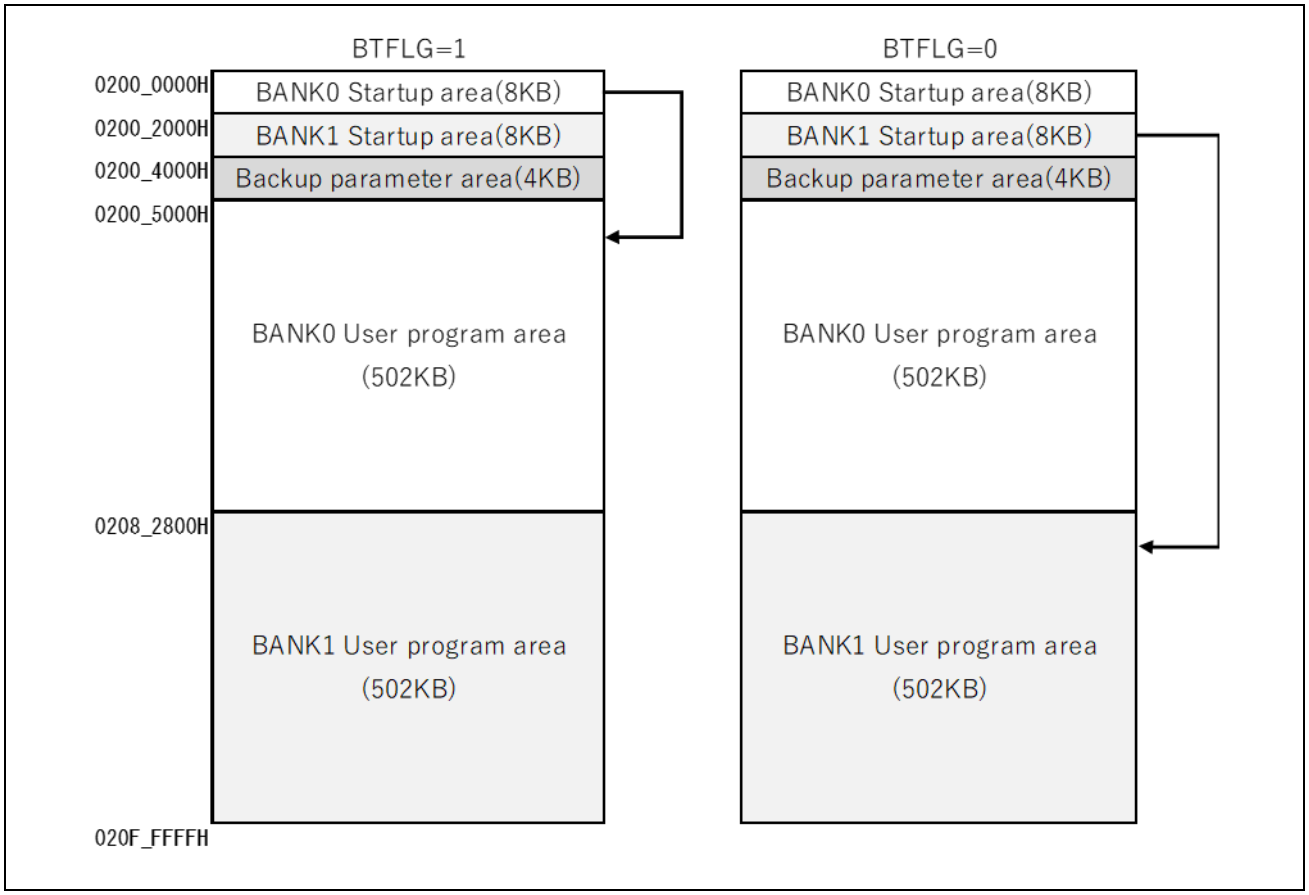


Figure 7.2. Boot Operations

7.1.3 Functions

The tables below list the functions related to the boot loader and updating of the FoE firmware.

Table 7.3 List of Functions Related to Updating of the FoE Firmware

Function Name	Outline
BL_Start	Handles processing to start the transition from the INIT to the BOOT state.
BL_StartDownload	Handles processing to start downloading the FoE file data.
BL_Data	Handles processing to receive the FoE file data.
BL_Data_write	Handles processing to write the file data to the serial flash ROM.
BL_SetRebootFlag	Sets the reboot flag.
BL_CheckRebootFlag	Checks the reboot flag.
BL_Reboot	Reboot processing (BOOT -> INIT)

8. Appendix

8.1 Appendix A: Precautions When Using the SSC Tool

※ When opening the SSC Tool

1. Run the SSC Tool as an administrator.
   Otherwise, generating SSC code may fail.

2. When opening the SSC Tool for the first time, the following window may be displayed.
   Please select [No] to skip the update check.

Figure 8.1. The update window

※ Regarding SSC Tool settings

3. Please go to [Tool] > [Options] to confirm SSC Tool settings.

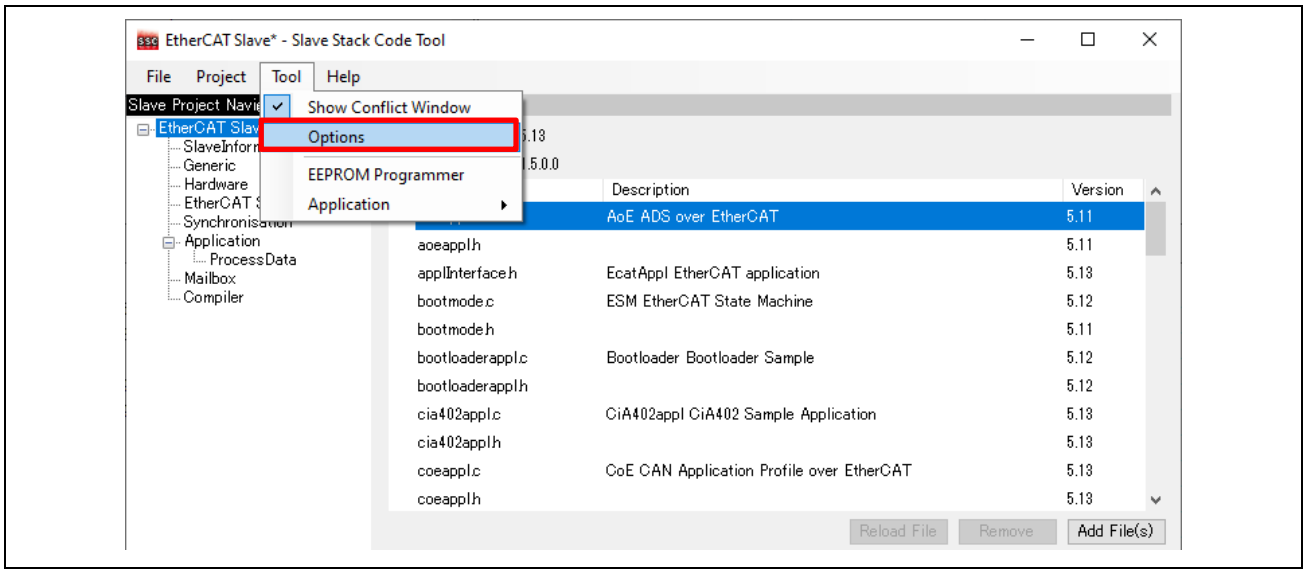


Figure 8.2. SSC tool main window

4. Under the [Generic] tab, uncheck [Open last project] and [Check for updates].

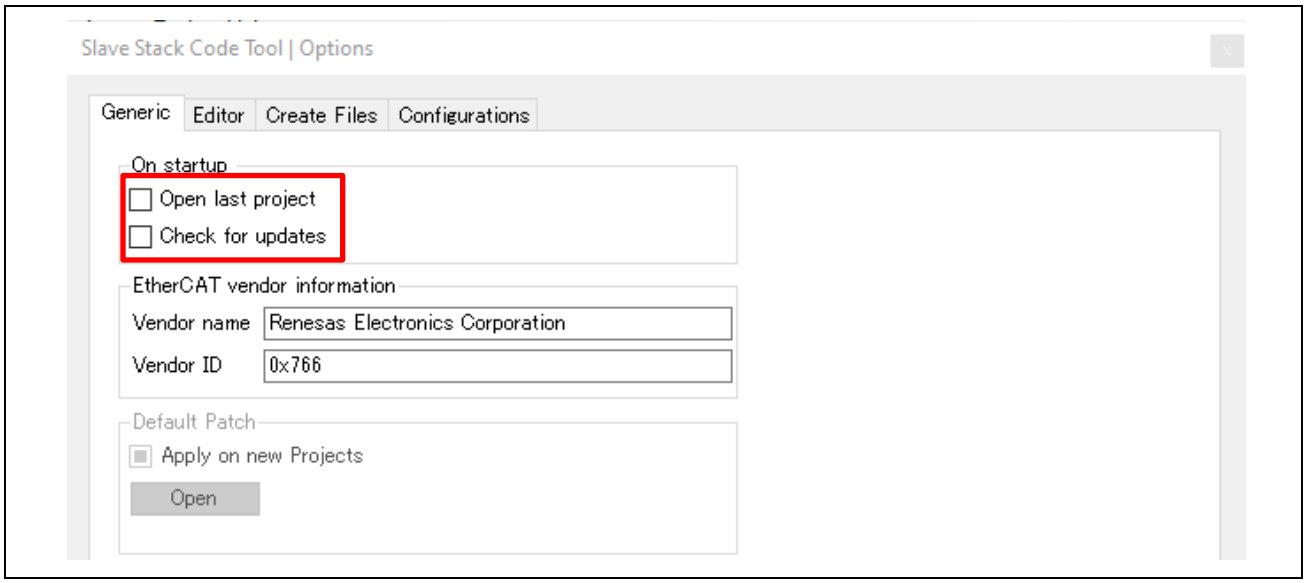


Figure 8.3 Option settings (1)

5. Under the [Create Files] tab, uncheck [Create device description (ESI)].

Figure 8.4 Option settings (2)

6. Click [OK] to apply the updated settings.
   This completes the SSC Tool setup for this sample program.

8.2 Appendix B: How to Install the Patch

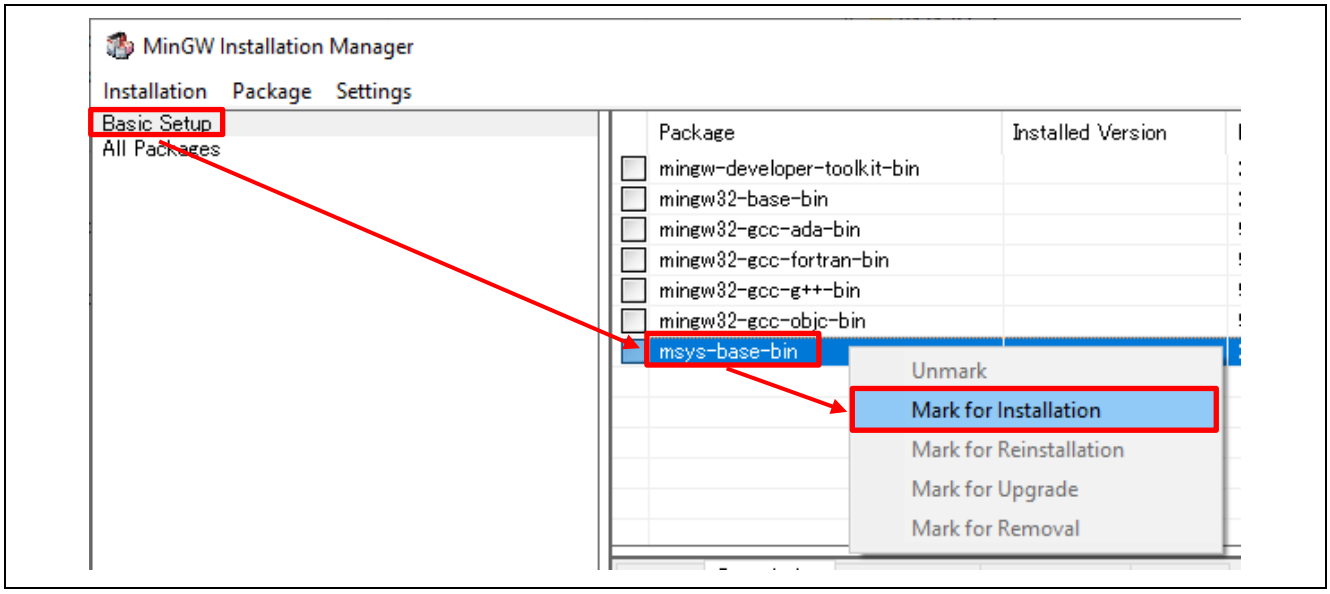
This section describes how to install the patch utility in MinGW.

1. Download “mingw-get-setup.exe” from the following URL.

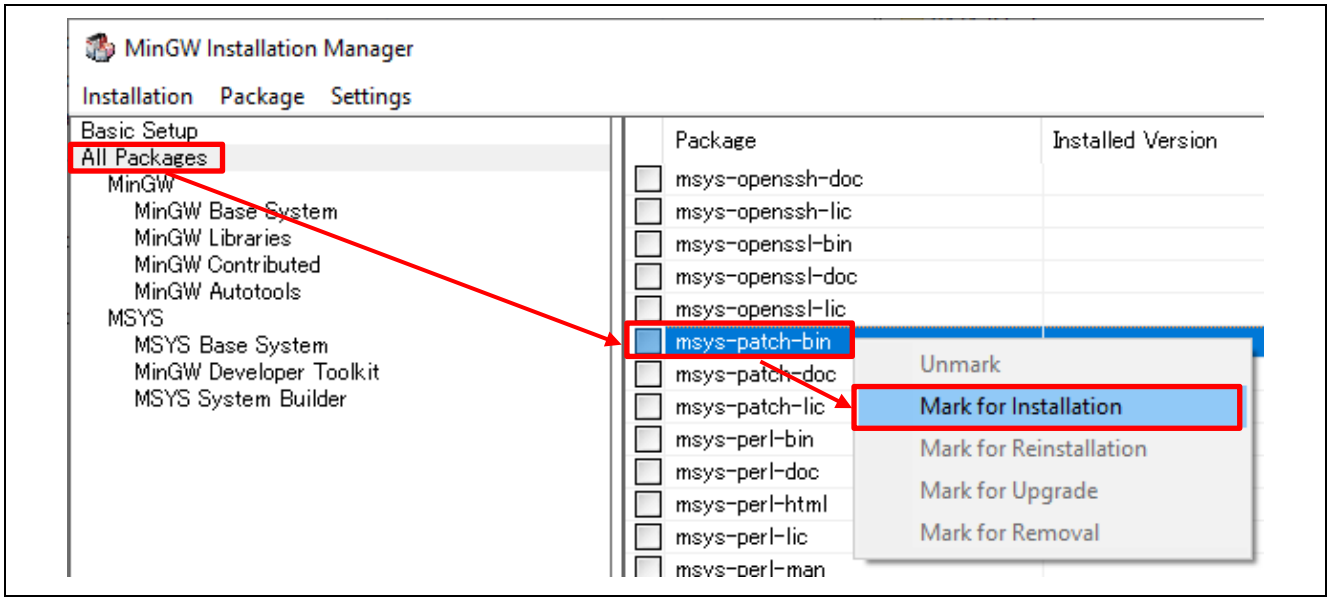
   https://sourceforge.net/projects/mingw/postdownload

2. Execute “mingw-get-setup.exe”, install “Mingw-installation-manager” according to the dialog.

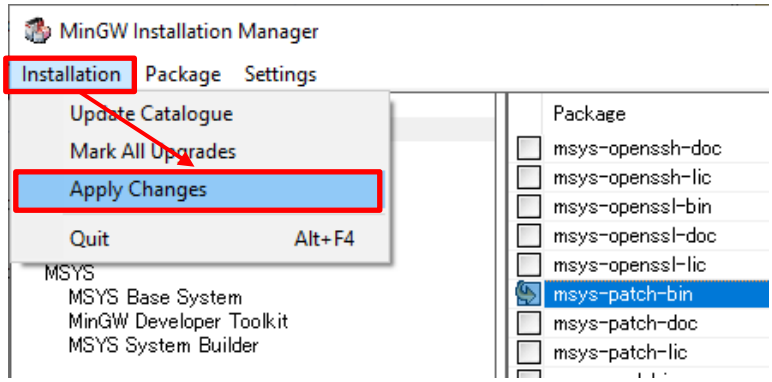
3. If it is completed and the Mingw-installation-manager window is displayed, select “Basic Setup” in the left window, right-click on “msys-base-bin” in the right window, and select “Mark for Installation”.



4. Select “All Packages” in the left window, right-click on “msys-patch-bin” in the right window and select “Mark for Installation”.



5. Select “Apply Changes” in “Installation” in the above menu bar.

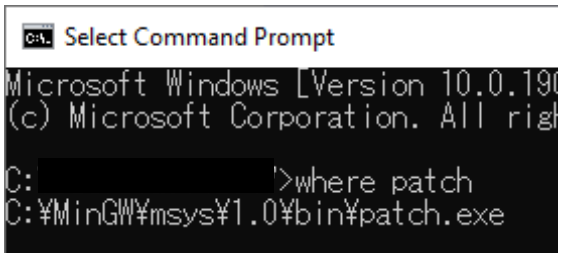


6. The “Schedule of Pending Actions” window is displayed; click the “Apply” button.

7. If “All changes were applied successfully; you may now close this dialogue.” is displayed, the installation of patch.exe is successful.

8. Register the path to the installed patch.exe into the system environment variables.
After registering, reboot your PC.

9. Start Command Prompt, enter “where patch”.
If the path to the installed patch.exe is displayed, there is no problem.

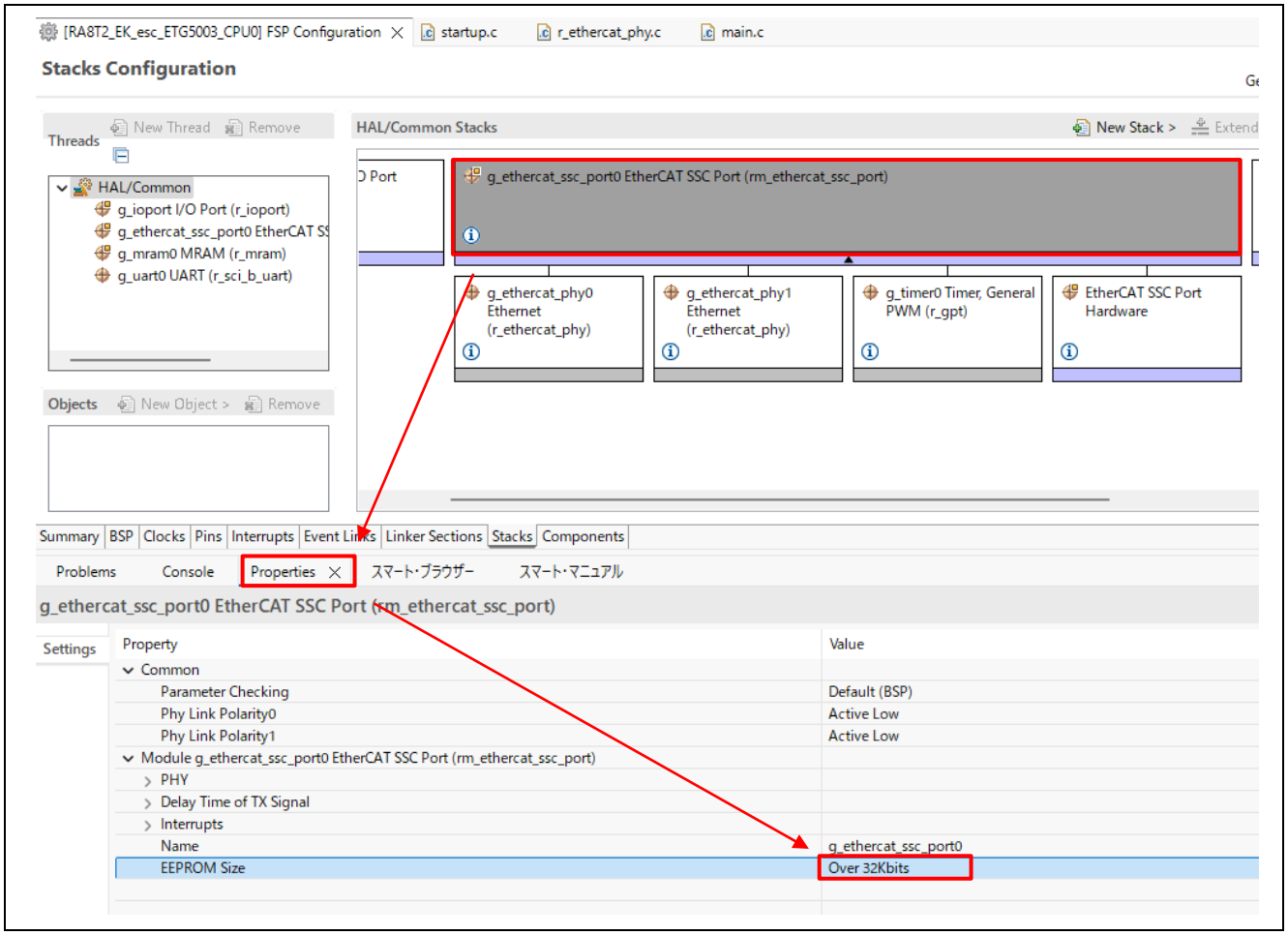


8.3 Appendix C: Changing the EEPROM Capacity Setting

Use the Smart Configurator to change the EEPROM capacity size and generate the code.
According to the EtherCAT specification, the setting must be changed based on the 32-Kbit boundary.

1. Open "configuration.xml" in the " **RA8T2_EK_esc_ETG5003_CPU0**" project.
   Select SSC_port stack. [**Properties**] → [**EEPROM Size**]

2. Select "**Under 32Kbits**" or "**Over 32Kbits**" depending on the capacity of the EEPROM.



3. Generate the code with "**Generate Project Content**".

8.4 Appendix D: Semiconductor Device Profile [ETG.5003]

8.4.1 Common Device Profile [ETG.5003.1]

When handling semiconductor devices with EtherCAT, it is necessary to support the device profile specified in the ETG5003 specifications.

The structure of ETG.5003 is as follows.

1. Common Device Profile (CDP) [ETG.5003.1]
2. Firmware update functionality [ETG.5003.2]
3. Specific Device Profile (SDP) [ETG.5003.2xxx]

Common Device Profile (CDP) specifies requirements that apply to all devices described in Specific Device Profile (SDP)

This sample program provides the object dictionary definition equivalent to Appendix A of the ETG document “ETG.5003 EtherCAT Semiconductor Device Profile - Part 1: Common Device Profile V1.1.0.”

This sample program provides only the framework for the object dictionary definition. Consider and implement the settings and necessary processing separately.

For Common Device Profile Ver1.1.0, please refer to the following ETG.5003.1 standard.

If you have any questions regarding CDP, please contact the ETG Association.

ETG5003.1 standard

ETG5003-1 S (R) V1.1.0

EtherCAT Semiconductor Device Profile

Part1 Common Device Profile

8.4.2 Semi Test Record [ETG.7000.2-Annex5003-0001]

In this sample program, the program is implemented assuming that it corresponds to the following test items of the Semi Test Record ETG.7000.2-Annex5003-0001.

1. Device Reset Command (Standard reset)
2. Dynamic PDO
3. Store Parameters

8.4.2.1 Device Reset Command (Standard reset)

When a specific value is entered in subindex 1 of the object 0xFBF0, the evaluation board restarts.

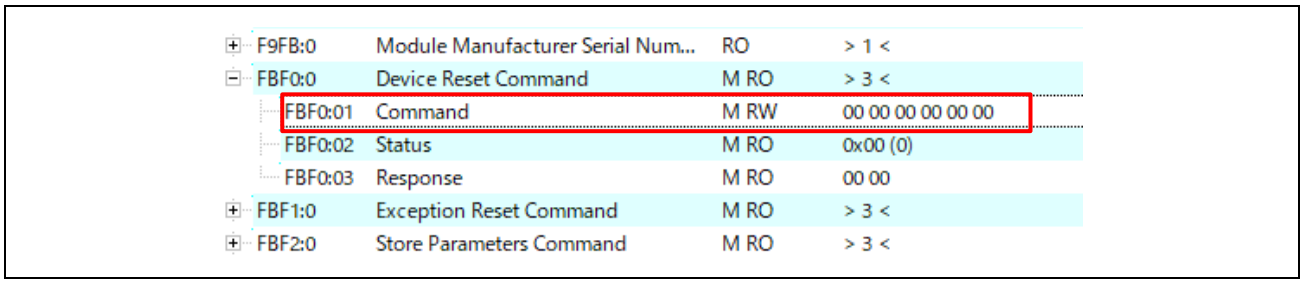
Note: When debugging, this operation will fail.
Stop debugging, and push the reset button on the EK board to reset RA8T2.

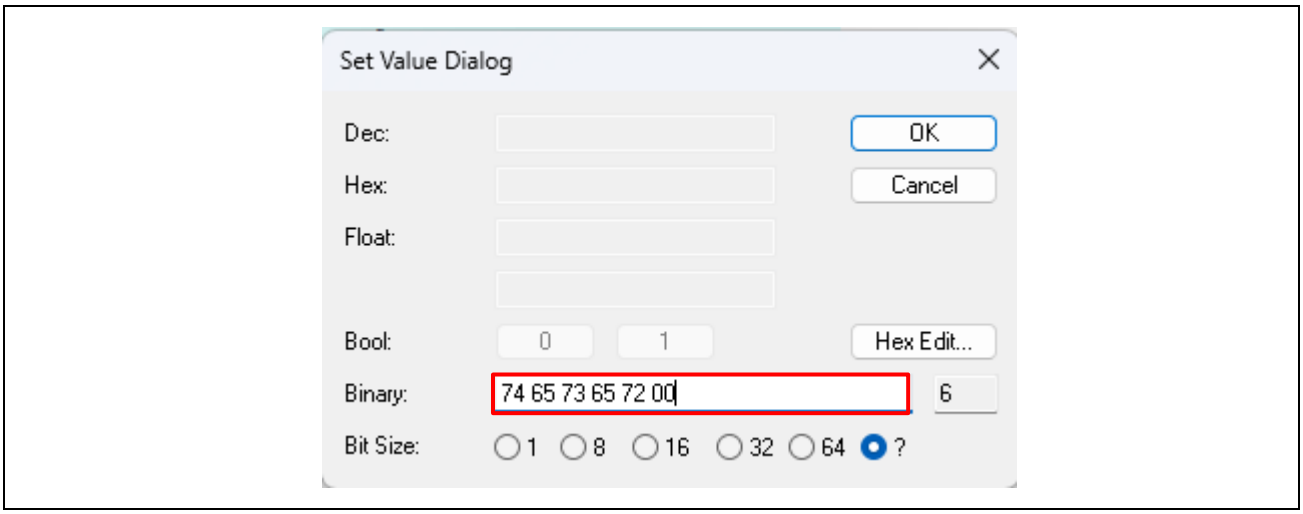
- How to check

(1) Connect the evaluation board to TwinCAT3 according to "6.1 Connecting to TwinCAT3".

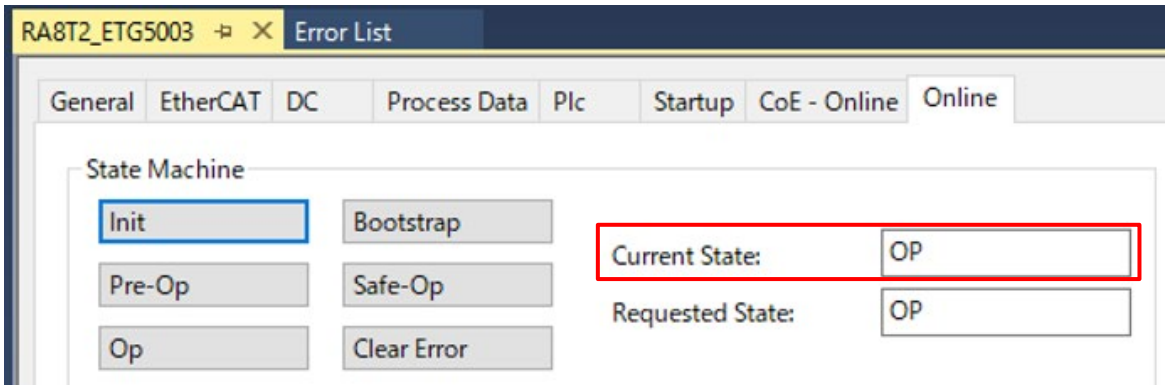
(2) Select the "**Online**" tab and make sure that the "**Current Status**" is set to "**OP**".

(3) Select the "**CoE - Online**" tab, double-click on Index **FBF0:01** "Command" and write "**74 65 73 65 72 00**" in "binary" to reset





(4) If the Current State in the TwinCAT3 Online tab transitions from INIT to PREOP to OP, the restart was successful, the evaluation board restarted normally, and the device reset command is working correctly.



8.4.2.2 Dynamic PDO

In this sample program, the settings related to PDO in the ESI file are shown in Table 8.1.

Table 8.1 PDO Settings for this Sample Program

PDO Items	setting
PdoAssign	true
PdoConfig	true

In addition, the PDO assignment/mapping object settings are shown in Table 8.2.

Table 8.2 PDO Assignment Mapping Object Settings

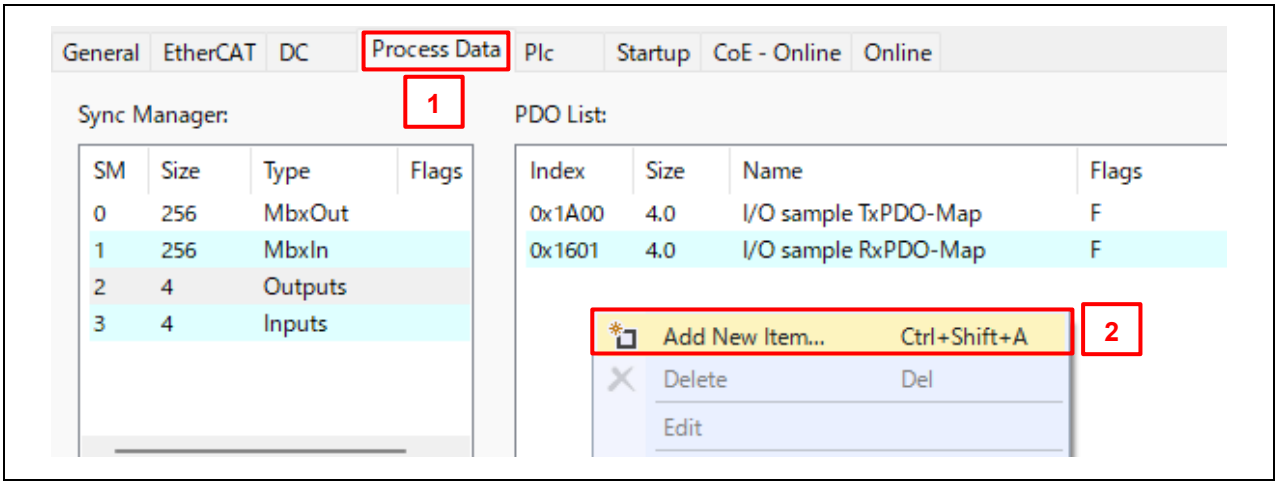
Object Name	Index	Access
RxPDO assign	0x1C12	RW only in the PreOP state
TxPDO assign	0x1C13	RW only in the PreOP state
Device User RxPDO-Map	0x17FF	RW only in the PreOP state
Device User TxPDO-Map	0x1BFF	RW only in the PreOP state

Therefore, objects 0x17FF "Device User RxPDO-Map" and 0x1BFF "Device User TxPDO-Map" that are not assigned by default can be assigned and mapped on the EtherCAT setting tool.

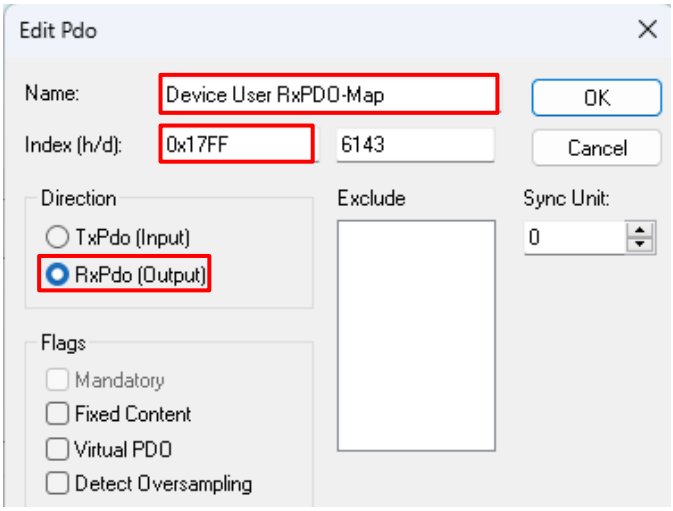
- How to set up

(1) Select the "Online" tab and make sure that the "**Current Status**" is set to "**OP**".

(2) Select the "**Process Data**" tab and right-click in the "PDO List" area, select "**Add New Item...**".



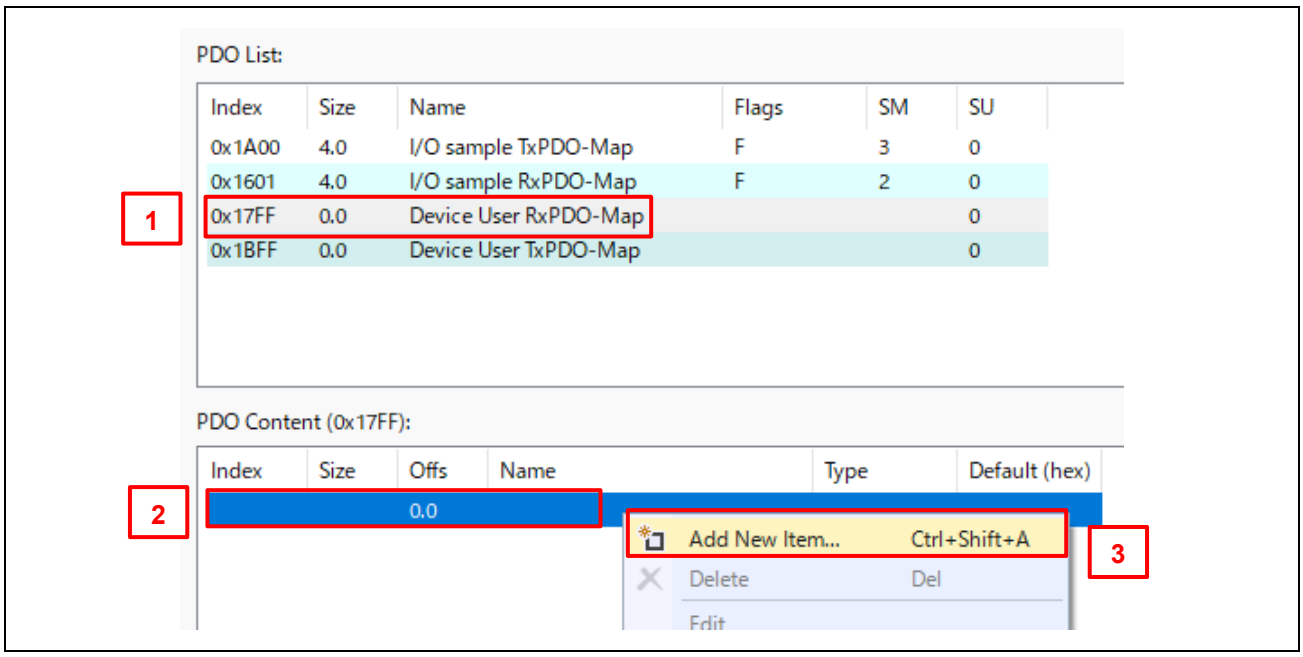
(3) In the "Edit Pdo" window, set the name to "**Device User RxPDO-Map**", the Index to "**0x17FF**", the Direction to "**RxPdo**" and press "OK".



(4) Right-click in the "PDO List" area again, select "Add New Item...", in the "Edit Pdo" window set the name to "**Device User TxPDO-Map**", the Index to "**0x1BFF**", the Direction to "**TxPdo**" and press "OK".

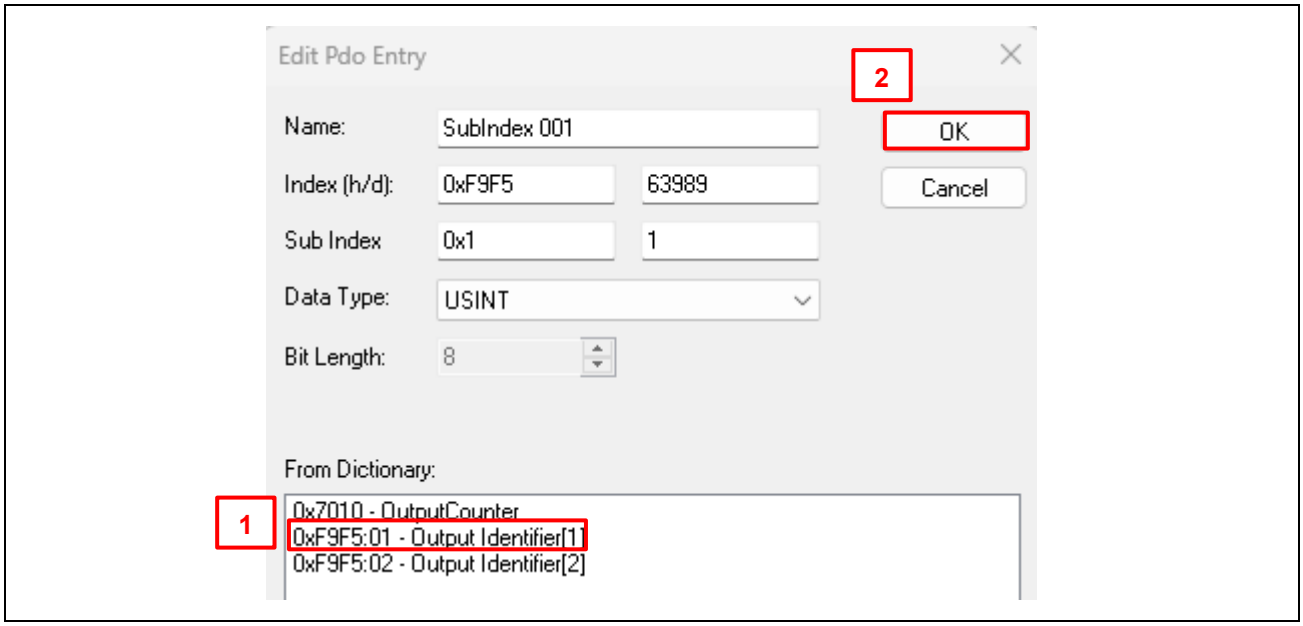
"Device User RxPDO-Map" and "Device User TxPDO-Map" have now been added to the "PDO List".

(5) Click "**Device User RxPDO-Map**" in the "PDO List" and right-click in the "**PDO Content (0x17FF):**" area, select "**Add New Item...**".



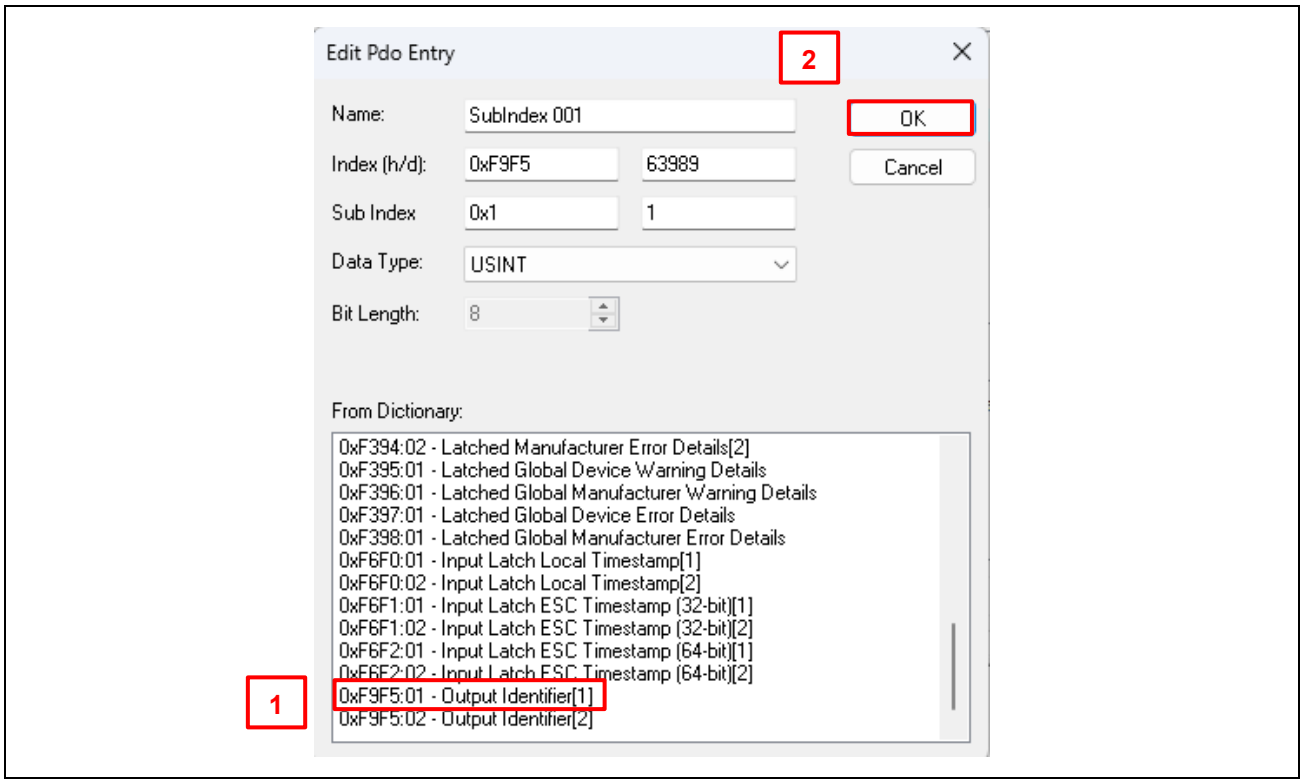
(6) In the "Edit Pdo Entry" window, click "**0xF9F5:01-Output Identifier[1]**" from "From Dictionary" and press OK.

The Index 0xF9F5 is now mapped to the Index 0x17FF.

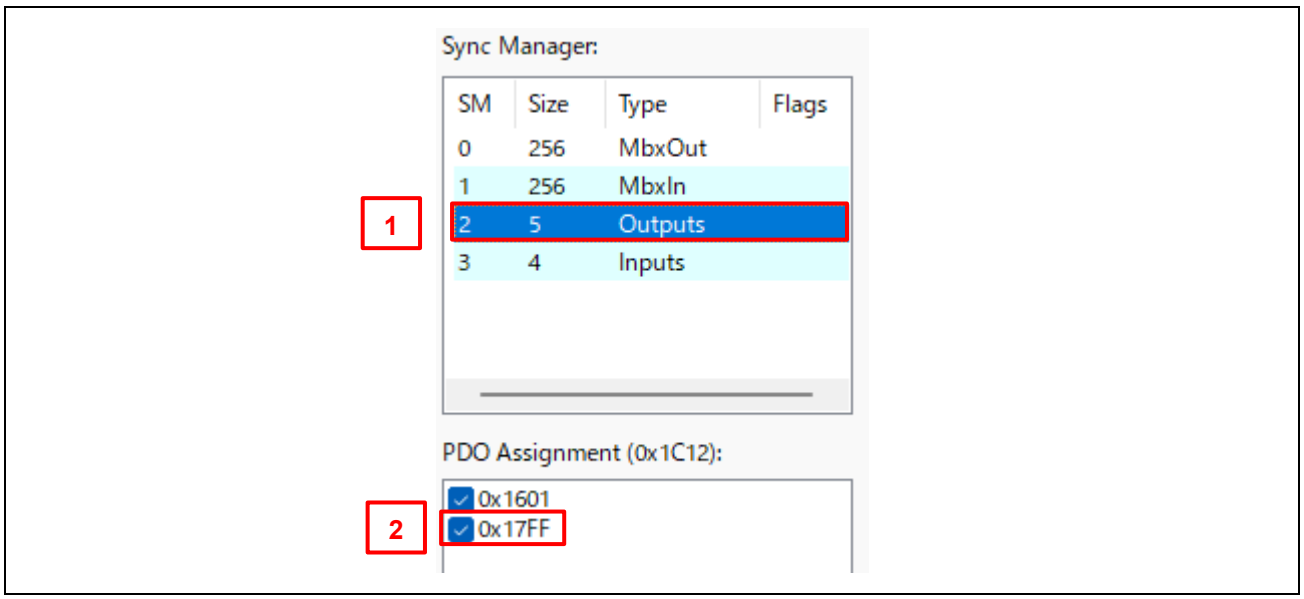


(7) Similarly, click on "**Device User TxPDO-Map**" in the "PDO List", right-click on the "**PDO Content (0x1BFF):**" area, and select "**Add New Item...**".

In the "Edit Pdo Entry" window, select "From Dictionary" to "**0xF9F5:01-Output Identifier[1]**" and press OK. The Index 0xF9F5 is now mapped to the Index 0x1BFF.

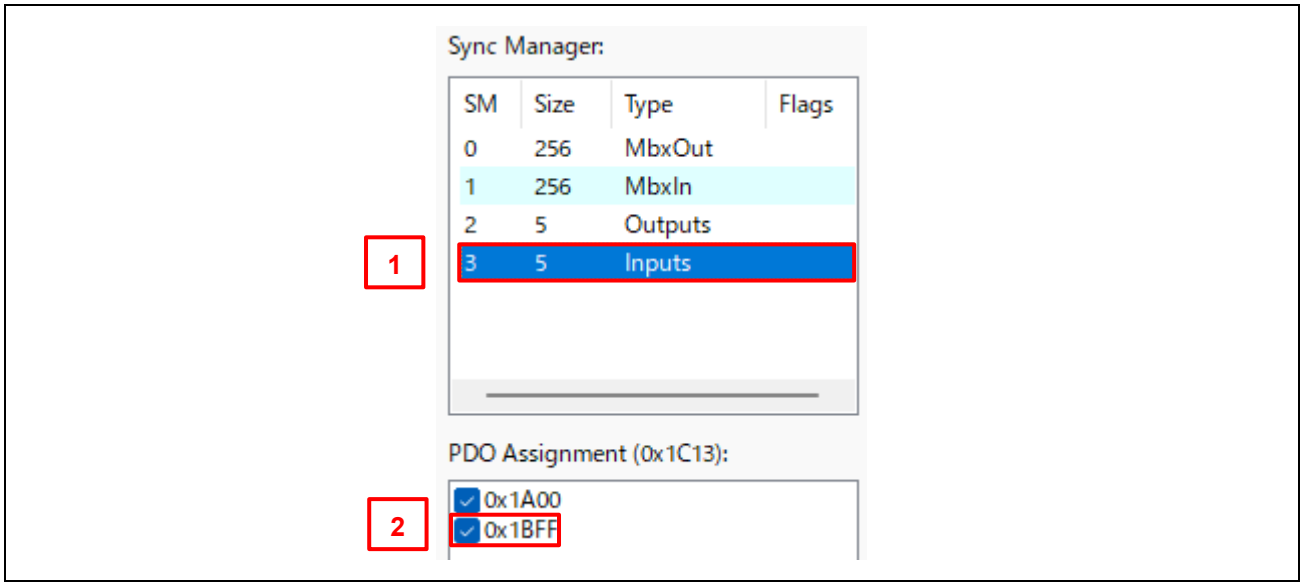


(8) In the "Sync Manager" area, click on the row with the "SM" column of "**2**" and check the "**0x17FF**" in the "PDO Assignment (0x1C12)" area.



(9) Similarly, in the "Sync Manager" area, click on the row with the "SM" column of "**3**" and check the "**0x1BFF**" in the "PDO Assignment (0x1C13)" area.

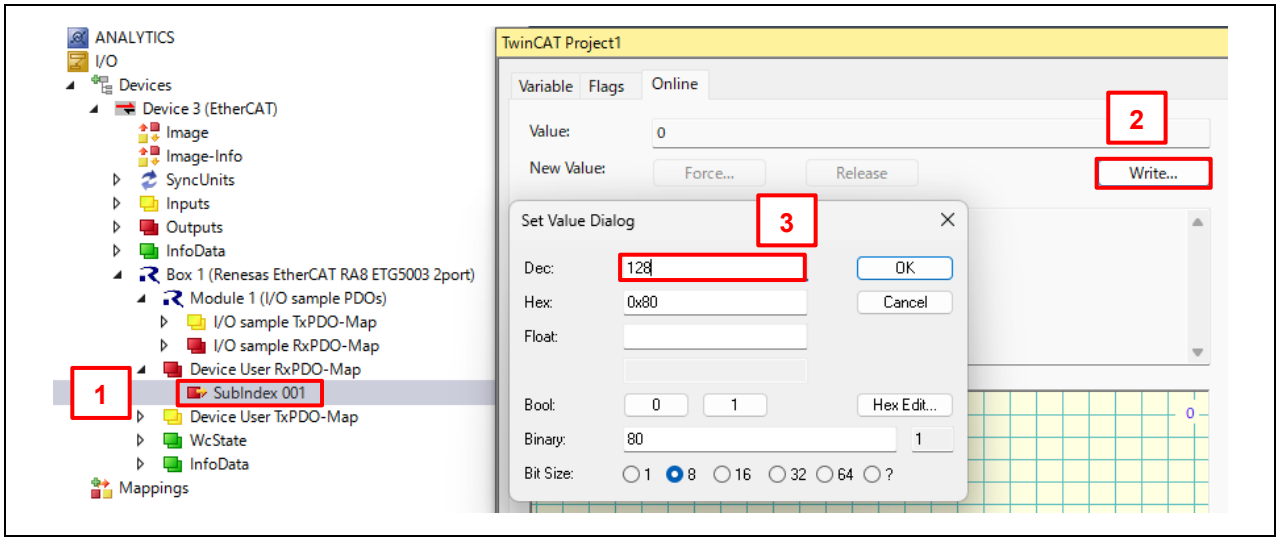
You have now assigned an additional 0x17FF to Index 0x1C12 and an additional 0x1BFF to 0x1C13.



(10) Restart TwinCAT by pressing [TwinCAT] → [**Restart TwinCAT (Config Mode)**] in the upper menu bar.

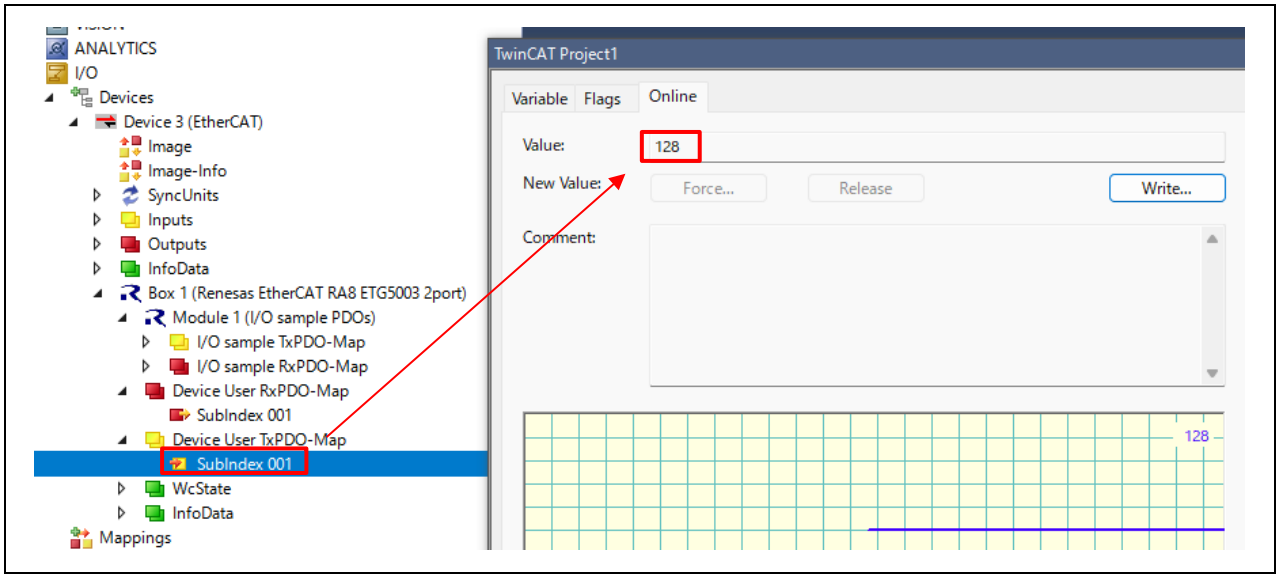
(11) Expand [**Device User RxPDO-Map**] in [Box 1] and click [**SubIndex 001**].

(12) Write an arbitrary value of "**1~255**" in Value.



(13) Expand [**Device User TxPDO-Map**] and click [**SubIndex 001**].

(14) If the value of Value is the same as the value written in (13), it is normal.



8.4.2.3 Store Parameters

In this sample program, when the object value with the backup flag is changed via SDO communication, its value is stored in the evaluation board's non-volatile memory.

Table 8.3 shows the objects with the backup flag in this sample program.

Table 8.3 Objects with Backup Flags

Index	Name
0x10F0	Backup parameter handling
0xF3A1	Device Warming Mask
0xF3A2	Manufacture Warming Mask
0xF3A3	Device Error Mask
0xF3A4	Manufacture Error Mask
0xF3A5	Global Device Warming Mask
0xF3A6	Global Manufacture Warming Mask
0xF3A7	Global Device Error Mask
0xF3A8	Global Manufacture Error Mask

Table 8.4 shows the code flash memory area used in this sample program as a non-volatile memory area in which the values of objects with the Backup flag are stored.

Table 8.4 Non-volatile Memory Areas for Backup Objects

First address	Last address	Capacity
0x60F0_0000	0x60F0_1000	4KB

Table 8.5 shows the constants used by programs that store in non-volatile memory.

Table 8.5 Constants used in programs storing non-volatile memory (samplesemi.h)

Constant Name	Setting Values	Details
BACKUP_MEMORY_START_ADDRESS	Table 8.4 First Addresses	The beginning address of the non-volatile memory area for the Backup objects.
BACKUP_BYTESIZE	4096	Amount of non-volatile memory area for Backup objects.
Default_Data_Is_Not_Initialized	(0xFFFF)	The non-volatile memory area for the Backup object has not been initialized.
Default_Data_Is_Initialized	(0x5555)	The non-volatile memory area for the Backup object has been initialized.
NonVolatileWordOffset_0x10F0	(0x0002)	Number of offset words from BACKUP_MEMORY_START_ADDRESS.
NonVolatileWordOffset_0xF3A1	(0x0008)	Number of offset words from BACKUP_MEMORY_START_ADDRESS.
NonVolatileWordOffset_0xF3A2	(0x000c)	Number of offset words from BACKUP_MEMORY_START_ADDRESS.
NonVolatileWordOffset_0xF3A3	(0x0010)	Number of offset words from BACKUP_MEMORY_START_ADDRESS.

NonVolatileWordOffset_0xF3A4	(0x0014)	Number of offset words from BACKUP_MEMORY_START_ADDRESS.
NonVolatileWordOffset_0xF3A5	(0x0018)	Number of offset words from BACKUP_MEMORY_START_ADDRESS.
NonVolatileWordOffset_0xF3A6	(0x001c)	Number of offset words from BACKUP_MEMORY_START_ADDRESS.
NonVolatileWordOffset_0xF3A7	(0x0020)	Number of offset words from BACKUP_MEMORY_START_ADDRESS.
NonVolatileWordOffset_0xF3A8	(0x0024)	Number of offset words from BACKUP_MEMORY_START_ADDRESS.

This sample program uses the functions defined in 6.2.4.1 "Backup Parameter Support" of Application Note ET9300 (EtherCAT SubDevice Stack Code).

For more information, see Application Note ET9300 (EtherCAT SubDevice Stack Code) and samplesemi.c.

- How to check

(1) Connect the evaluation board to TwinCAT3 by following “5. Connecting to TwinCAT3.”

(2) Select the "Online" tab and make sure that "Current Status" is set to "OP".

(3) Write the value to the Backup object. Here, we write "0xAAAAAAAA" at Index 0xF3A1:01.

(4) Reboot the RA8T2.

(5) It is normal if the Index 0xF3A1:01 written in (4) is "0xAAAAAAAA".
Objects with the backup flag are stored in non-volatile memory and are read at startup.

9. Limitations

There are no limitations.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun. 30, 2026	-	First issued

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.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

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/.