RZ/T2, RZ/N2

Quick Start Guide: EtherCAT ETG5003 SubDevice Software

Introduction

This application note explains sample program setup procedures for EtherCAT® SubDevice functionalities with the adapted EtherCAT Stack Code for RZ/T2, RZ/N2 series microprocessor.

This describes steps to confirm SubDevice behaviour and stack features using TwinCAT® MainDevice Configuration tool.

Target Device

RZ/T series: RZ/T2L, RZ/T2M, RZ/T2ME, RZ/T2H
RZ/N series: RZ/N2L, RZ/N2H

Contents

1. Overview

This application note explains sample program setup procedures for EtherCAT® SubDevice functionalities with the adapted EtherCAT Stack Code for RZ/T2, RZ/N2 series microprocessor.

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 RZ/T2, RZ/N2

Technical information about EtherCAT is available via ETG member site, and information about RZ/T2 and RZ/N2 is available via Renesas.

Table 1.2 Common Technical Inputs for RZ/T2, RZ/N2

Document Type	Description	Document Title	Document No.
Application Note	Describes how to use the Renesas Flexible Software Package (FSP) for writing applications for the RZ/T2, RZ/N2 microprocessor series.	RZ/T2, RZ/N2 Getting Started with Flexible Software Package	r01an6434ej****

Table 1.3 Technical Inputs for RZ/T2L

Document Type	Description	Document Title	Document No.
User's Manual	Describes the technical details of the RSK+RZT2L hardware.	Renesas Starter Kit+ for RZ/T2L User's Manual	r20ut5164eg****
User's Manual	Provides technical details of the RZ/T2L microprocessor.	RZ/T2L Group User's Manual Hardware	r01uh0985ej****

Table 1.4 Technical Inputs for RZ/N2L

Document Type	Description	Document Title	Document No.
User's Manual	Describes the technical details of the RSK+RZN2L hardware.	Renesas Starter Kit+ for RZ/N2L User's Manual	r20ut4984eg****
User's Manual	Provides technical details of the RZ/N2L microprocessor.	RZ/N2L Group User's Manual Hardware	r01uh0955eg****

Table 1.5 Technical Inputs for RZ/T2M and RZ/T2ME

Document Type	Description	Document Title	Document No.
User's Manual	Describes the technical details of the RSK+RZT2M and RSK+RZT2ME hardware.	Renesas Starter Kit+ for RZ/T2M, RZ/T2ME User's Manual	r20ut4939eg****
User's Manual	Provides technical details of the RZ/T2M microprocessor.	RZ/T2M Group User's Manual Hardware	r01uh0916eg****
User's Manual	Provides technical details of the RZ/T2ME microprocessor.	RZ/T2ME Group User's Manual Hardware	r01uh1062eg****

Table 1.6 Technical Inputs for RZ/T2H and RZ/N2H

Document Type	Description	Document Title	Document No.
User's Manual	Describes the technical details of the RZ/T2H EVB hardware.	RZ/T2H Evaluation Board User's Manual	r20ut5405ej****
User's Manual	Describes the technical details of the RZ/N2H EVB hardware.	RZ/N2H Evaluation Board User's Manual	r20ut5522ej****
User's Manual	Provides technical details of the RZ/T2H and RZ/N2H microprocessor.	RZ/T2H and RZ/N2H Groups User's Manual: Hardware	r01uh1039eg****

2. Features

This package includes the firmware for the EtherCAT SubDevice stack generated by SSC Tool on Renesas' RZ/T2 and RZ/N2 series processors.

This sample program 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:
  - I/O Input DIP SW
  - I/O Output LED
- ETG.5003 Semiconductor Device Profiles:
  - Common Device Profile (CDP) [ETG.5003.1]
  - Firmware update functionality [ETG.5003.2]

EtherCAT is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

2.1 Folder structure relative to this application note

After extracting the package, the following structure is obtained.

Table 2.1 Sample package overview

Item	Description
r01an8274xx0400-rzt2-rzn2-ethercat-package	RZ/T2, N2 EtherCAT Sample Package
├── RZT2M_RZT2ME_EtherCAT_RSK_rev0400.zip	Archive for RZ/T2M, RZ/T2ME (RSK board)
├── RZT2L_EtherCAT_RSK_rev0400.zip	Archive for RZ/T2L (RSK board)
├── RZT2H_EtherCAT_EVB_rev0400.zip	Archive for RZ/T2H (EVB board)
├── RZN2L_EtherCAT_RSK_rev0400.zip	Archive for RZ/N2L (RSK board)
├── RZN2H_EtherCAT_EVB_rev0400.zip	Archive for RZ/N2H (EVB board)
└── r01an8277ej0400-rzt2-rzn2-ecat-etg5003.pdf	This application note

After extracting the archive for multicore, the following structure is obtained.

Table 2.2 Multi-Core folder structure (RZ/T2M, RZ/T2ME)

Item	Description
RZT2M_RZT2ME_EtherCAT_RSK_rev0400	RZ/T2M, RZ/T2ME RSK folder
└── CR52_Dual	Cortex-R52 CPU0 and Cortex-R52 CPU1
├── common	Common resources for SSC Tool
│ └── ETG5003	For ETG5003
│ ├── ESI	EtherCAT SubDevice Information
│ ├── Patch	Patch for this ETG5003 project
│ └── SSCconfig	SSC Tool setting file
└── project	IDE project folder
└── ETG5003	For ETG5003
├── e2studio	The e^{2}studio projects
│ ├── RZT2M_RSK_ESC_ETG5003_CR52_Dual_Primary	Primary core project
│ └── RZT2M_RSK_ESC_ETG5003_CR52_Dual_Secondary	Secondary core project
└── ewarm	The EWARM projects
├── RZT2M_RSK_ESC_ETG5003_CR52_Dual_Primary	Primary core project
└── RZT2M_RSK_ESC_ETG5003_CR52_Dual_Secondary	Secondary core project

Table 2.3 Multi-Core folder structure (RZ/T2H)

Item	Description
RZT2H_EtherCAT_EVB_rev0400	RZ/T2H EVB folder
├── CR52_Dual	Cortex-R52 CPU0 and Cortex-R52 CPU1
│ ├── common	Common resources for SSC Tool
│ │ └── ETG5003	For ETG5003
│ │ ├── ESI	EtherCAT SubDevice Information
│ │ ├── Patch	Patch for this ETG5003 project
│ │ └── SSCconfig	SSC Tool setting file
│ └── project	IDE project folder
│ └── ETG5003	For ETG5003
│ ├── e2studio	The e^{2}studio projects
│ │ ├── RZT2H_EVB_ESC_ETG5003_CR52_Dual_Primary	Primary core project
│ │ └── RZT2H_EVB_ESC_ETG5003_CR52_Dual_Secondary	Secondary core project
│ └── ewarm	The EWARM projects
│ ├── RZT2H_EVB_ESC_ETG5003_CR52_Dual_Primary	Primary core project
│ └── RZT2H_EVB_ESC_ETG5003_CR52_Dual_Secondary	Secondary core project
└── CR52_CA55	Cortex-R52 CPU0 and Cortex-A55 Core2
├── common	Common resources for SSC Tool
│ └── ETG5003	For ETG5003

│ ├── ESI	EtherCAT SubDevice Information
│ ├── Patch	Patch for this ETG5003 project
│ └── SSCconfig	SSC Tool setting file
└── project	IDE project folder
└── ETG5003	For ETG5003
├── e2studio	The e²studio projects
│ ├── RZT2H_EVB_ESC_ETG5003_CR52_CA55_Primary	Primary core project
│ └── RZT2H_EVB_ESC_ETG5003_CR52_CA55_Secondary	Secondary core project
└── ewarm	The EWARM project
├── RZT2H_EVB_ESC_ETG5003_CR52_CA55_Primary	Primary core project
└── RZT2H_EVB_ESC_ETG5003_CR52_CA55_Secondary	Secondary core project

Table 2.4 Multi-Core folder structure (RZ/N2H)

Item	Description
RZN2H_EtherCAT_EVB_rev0400	RZ/N2H EVB folder
├── CR52_Dual	Cortex-R52 CPU0 and Cortex-R52 CPU1
│ ├── common	Common resources for SSC Tool
│ │ └── ETG5003	For ETG5003
│ │ ├── ESI	EtherCAT SubDevice Information
│ │ ├── Patch	Patch for this ETG5003 project
│ │ └── SSCconfig	SSC Tool setting file
│ └── project	IDE project folder
│ └── ETG5003	For ETG5003
│ ├── e2studio	The e²studio projects
│ │ ├── RZN2H_EVB_ESC_ETG5003_CR52_Dual_Primary	Primary core project
│ │ └── RZN2H_EVB_ESC_ETG5003_CR52_Dual_Secondary	Secondary core project
│ └── ewarm	The EWARM projects
│ ├── RZN2H_EVB_ESC_ETG5003_CR52_Dual_Primary	Primary core project
│ └── RZN2H_EVB_ESC_ETG5003_CR52_Dual_Secondary	Secondary core project
├── CR52_CA55	Cortex-R52 CPU0 and Cortex-A55 Core2
│ ├── common	Common resources for SSC Tool
│ │ └── ETG5003	For ETG5003
│ │ ├── ESI	EtherCAT SubDevice Information
│ │ ├── Patch	Patch for this ETG5003 project
│ │ └── SSCconfig	SSC Tool setting file
│ └── project	IDE project folder
│ └── ETG5003	For ETG5003
│ ├── e2studio	The e²studio projects
│ │ ├── RZN2H_EVB_ESC_ETG5003_CR52_CA55_Primary	Primary core project
│ │ └── RZN2H_EVB_ESC_ETG5003_CR52_CA55_Secondary	Secondary core project
│ └── ewarm	The EWARM projects
│ ├── RZN2H_EVB_ESC_ETG5003_CR52_CA55_Primary	Primary core project
│ └── RZN2H_EVB_ESC_ETG5003_CR52_CA55_Secondary	Secondary core project

After extracting the archive for single core, the following structure is obtained.

Table 2.5 Single-Core folder structure (RZ/T2L)

Item	Description
RZT2L_EtherCAT_RSK_rev0400	RZ/T2L RSK folder
├─── common	Common resources for SSC Tool
│ └─── ETG5003	For ETG5003
│ ├─── ESI	EtherCAT SubDevice Information
│ ├─── Patch	Patch for this ETG5003 project
│ └─── SSCconfig	SSC Tool setting file
└─── project	IDE project folder
└─── ETG5003	For ETG5003
├─── e2studio	The e²studio projects
│ └─── RZT2L_RSK_ESC_ETG5003	Single core project
└─── ewarm	The EWARM projects
└─── RZT2L_RSK_ESC_ETG5003	Single core project

Table 2.6 Single-Core folder structure (RZ/N2L)

Item	Description
RZN2L_EtherCAT_RSK_rev0400	RZ/N2L RSK folder
├─── common	Common resources for SSC Tool
│ └─── ETG5003	For ETG5003
│ ├─── ESI	EtherCAT SubDevice Information
│ ├─── Patch	Patch for this ETG5003 project
│ └─── SSCconfig	SSC Tool setting file
└─── project	IDE project folder
└─── ETG5003	For ETG5003
├─── e2studio	The e²studio projects
│ └─── RZN2L_RSK_ESC_ETG5003	Single core project
└─── ewarm	The EWARM projects
└─── RZN2L_RSK_ESC_ETG5003	Single core project

3. Requirements (Software and Hardware)

This project has been developed and tested on these environments using the following boards and tools.

3.1 Requirements for this sample package

Table 3.1 Requirements

Category	Name	Version	Description
Board	Renesas Starter Kit+ for RZ/T2M	-	Renesas RZ/T2M-RSK - Renesas Starter Kit Plus for RZ/T2M | Renesas
	Renesas Starter Kit+ for RZ/T2ME	-	Renesas RZ/T2ME-RSK - Renesas Starter Kit+ for RZ/T2ME | Renesas
	Renesas Starter Kit+ for RZ/T2L	-	Renesas RZ/T2L-RSK - Renesas Starter Kit+ for RZ/T2L | Renesas
	RZ/T2H Evaluation Board	-	Renesas RZ/T2H-EVKIT - Evaluation Board Kit for RZ/T2H | Renesas
	Renesas Starter Kit+ for RZ/N2L	-	Renesas RZ/N2L-RSK - Renesas Starter Kit+ for RZ/N2L | Renesas
	RZ/N2H Evaluation Board	-	Renesas RZ/N2H-EVKIT - Evaluation Board Kit for RZ/N2H | Renesas
IDE	EWARM	9.60.3	IAR Systems IAR Embedded Workbench for Arm | IAR
	e^2 studio	2025-12	Renesas Release v4.0.0 · renesas/rz-fsp · GitHub
Configurator	FSP Smart Configurator	2025-12	
Flexible Software Package	FSP for Renesas RZ	4.0.0	
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
	I-jet	-	IAR Systems IAR debug probes | IAR
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 document describes the major hardware. For more information about the boards, refer to the respective evaluation board user manuals and schematics.

4.1 RZ/T2M and RZ/T2ME RSK Board

Note) All descriptions are based on the RSK+RZT2M. When using the RSK+RZT2ME, substitute RZ/T2M with RZ/T2ME throughout this document.

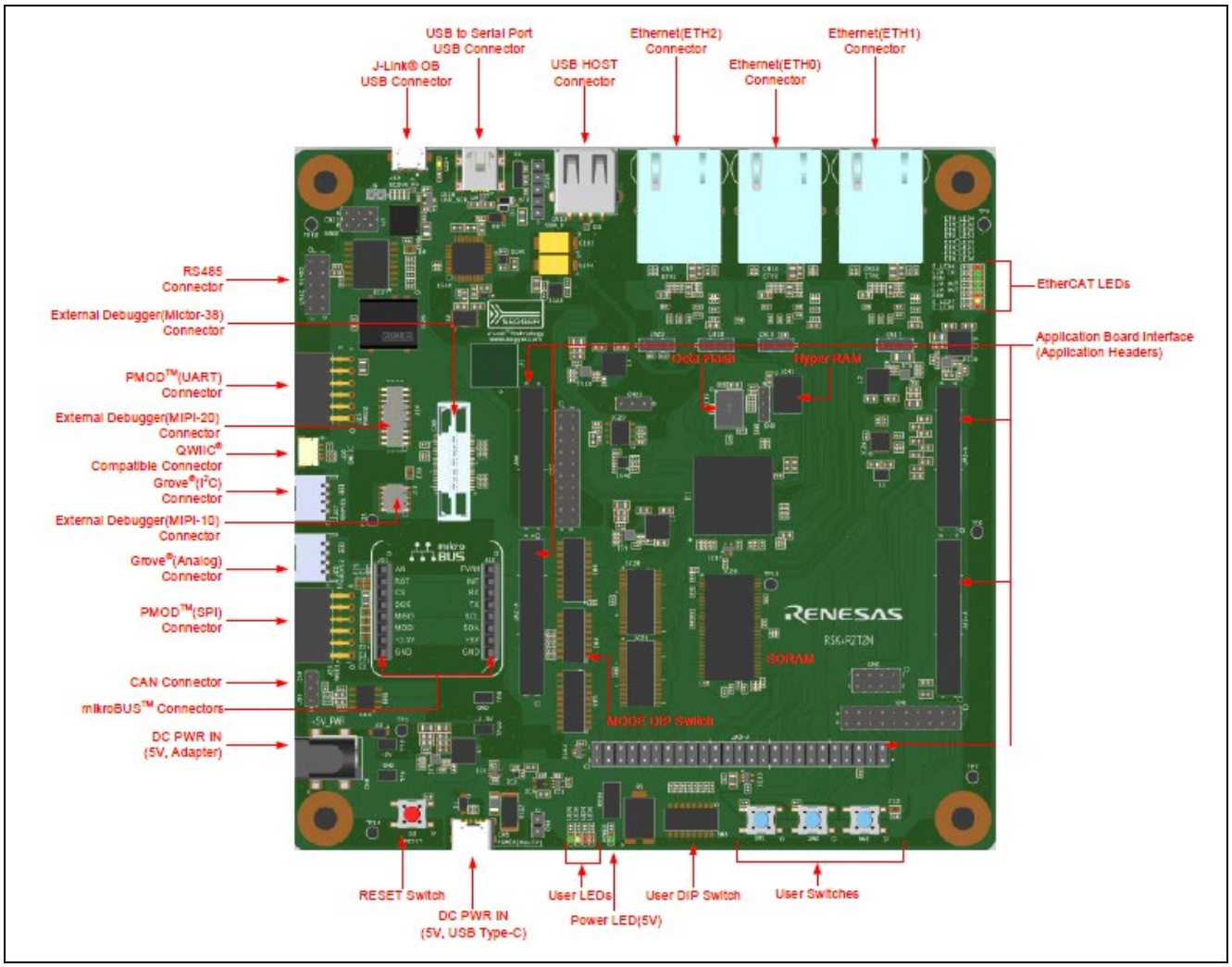


Figure 4.1 RZT2M RSK board layout

4.1.1 Jumper and Switch Configuration

The tables below show the jumper pin and switch settings.
Text in bold red type in the tables indicates the settings changed from the time of shipment of the board.

Table 4.1 RZ/T2M-RSK Jumper configuration

Reference	Jumper Position	Description
CN8	Short 2-3	Use QSPI Serial Flash
CN17	Short 2-3	VCC1833_2 Use power supply at 1.8V
CN18	Short 1-2	Use 3 ports in same PHY mode
CN19	Short 1-2	Use 3 ports in same PHY mode
CN20	Short 1-2	Use 3 ports in same PHY mode
CN21	Short 1-2	Use RS485 transmission method in full duplex
CN22	Short 1-2	Use RS485 transmission method in full duplex
J9	Open	Use J-Link OB

Table 1.2 RZ/T2M-RSK SW4 Settings

SW4	Setting	Description
SW4-1	ON	xSPI0 x1 boot mode * Refer to the Document No.”r20ut4939” and set according to the mode to be used.
SW4-2	ON	
SW4-3	ON	
SW4-4	ON	MDD=0, JTAG Authentication by Hash is disabled.
SW4-5	OFF	MDW=1, ACTM 1 wait, should be set when TCM is used with CPU operating frequencies above 400MHz
SW4-6	OFF	-
SW4-7	OFF	-
SW4-8	OFF	-

Table 4.3 RZ/T2M-RSK SW5 Setting.

SW5	Setting	Description
SW5-1	OFF	-
SW5-2	OFF	-
SW5-3	ON	Enable the "SCI_RTS" signal.
SW5-4	OFF	
SW5-5	ON	Enable the "SCI_RXD" signal.
SW5-6	OFF	
SW5-7	OFF	
SW5-8	OFF	Enable the "SCK3" signal.
SW5-9	ON	
SW5-10	OFF	

Table 4.4 RZ/T2M-RSK SW6 Setting

SW6	Setting	Description
SW6-1	OFF	Enable the "ETH2_MDIO" and "ETH2_MDC" signal.
SW6-2	OFF	-
SW6-3	ON	TRACE_CTL signal is enabled
SW6-4	OFF	
SW6-5	OFF	SCI_TXD signal is enabled
SW6-6	ON	
SW6-7	OFF	MB_RST# signal is enabled
SW6-8	ON	
SW6-9	OFF	CAN_RX_OB signal is enabled
SW6-10	ON	

4.1.2 Board Setup

Setting the board for running sample program is shown below.
Build and run the sample code on the RZ/T2M RSK board by following the steps below.
Both loading into RAM and flash can be done using IAR Embedded Workbench or e^2 studio.

1. Connect the decoder to the header "J20" on the RZ/T2M RSK board.

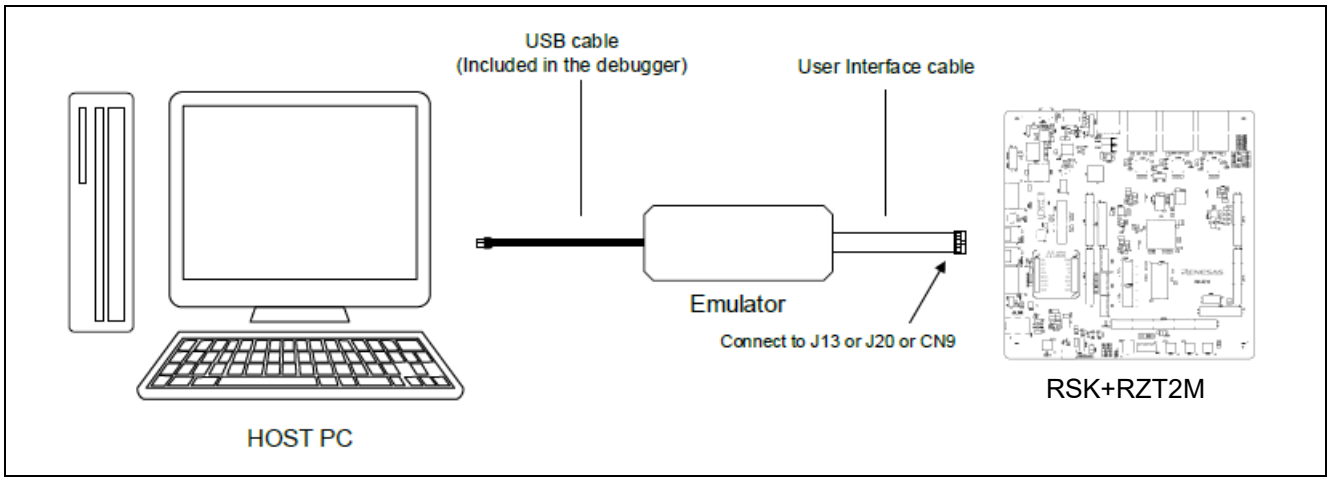


Figure 4.2 : RZ/T2M RSK board debug connection diagram

When using J-Link OB, connect the USB cable to the header "J10" and set "J9" to open.

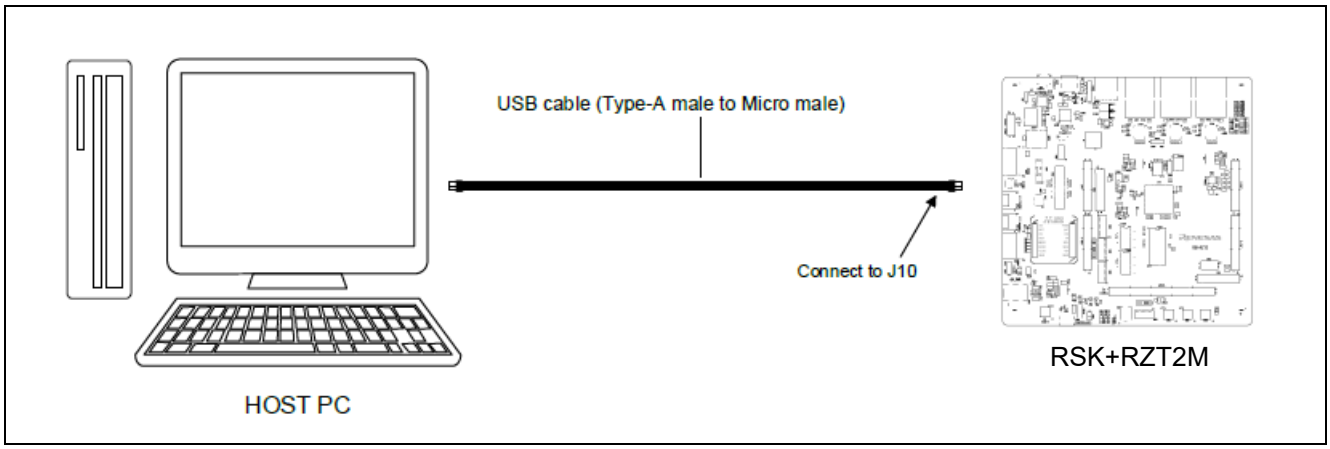


Figure 4.3 : RZ/T2M RSK board debug connection diagram (J-Link OB)

2. Power is supplied using a USB cable (Type-C) or an AC / DC adapter. When using a USB cable (Type-C), connect it to the USB connector "CN5" of the RZ/T2M RSK board. When connecting the AC/DC adapter, connect it to the “CN6” connector of the RZ/T2M RSK board.

3. Connect host PC and RZ/T2M RSK board using an ethernet cable. When using an ethernet cable, connect the ethernet cable into “ETH0” connector of the RZ/T2M RSK board.
Depending on the protocol specifications, it may be possible to connect to "ETH1".

4.2 RZ/N2L RSK Board

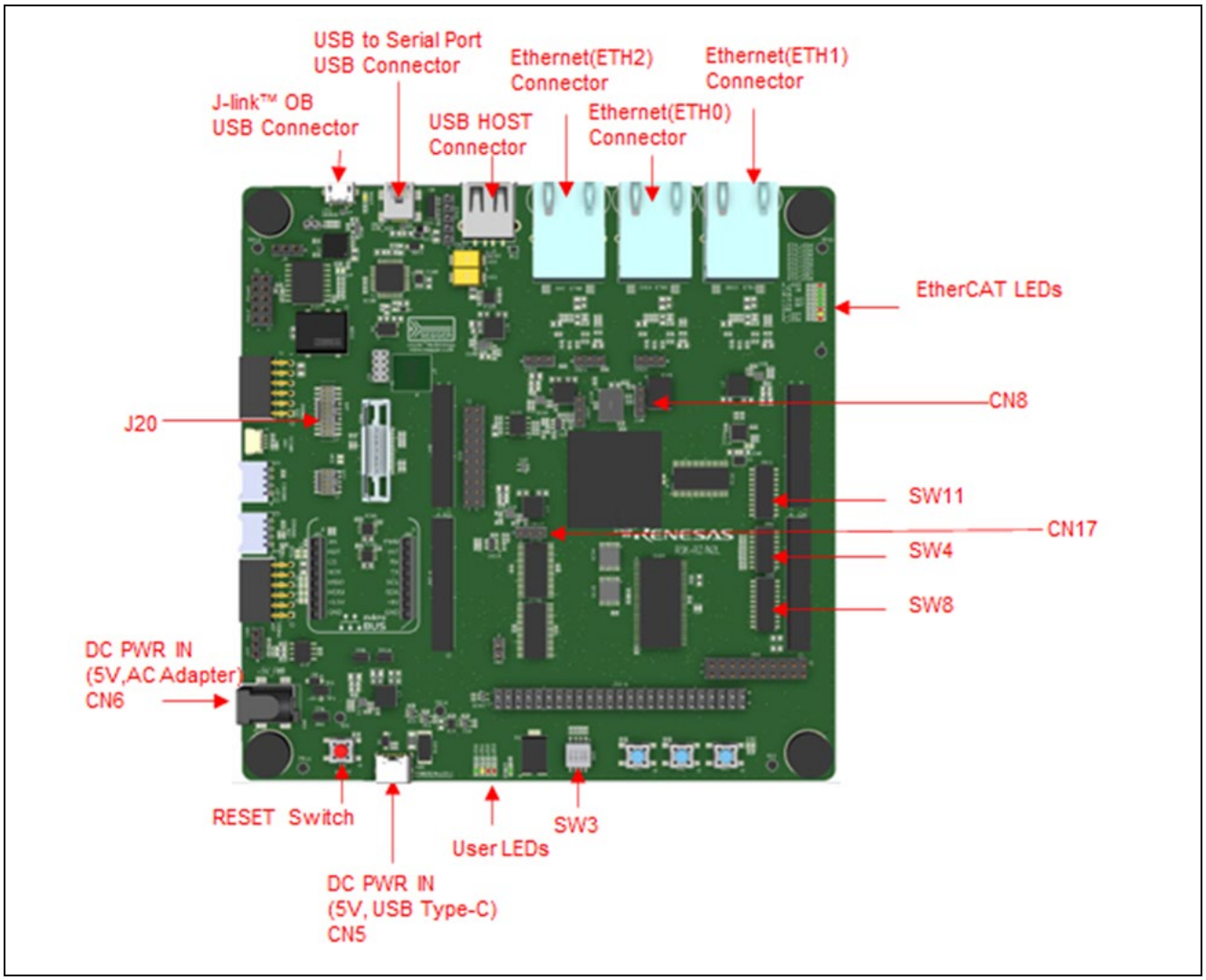


Figure 4.4 RZN2L RSK board layout

4.2.1 Jumper and Switch Configuration

The tables below show the jumper pin and switch settings.
Text in bold red type in the tables indicates the settings changed from the time of shipment of the board.

Table 4.5 RZ/N2L-RSK Jumper configuration

Reference	Jumper Position	Description
CN8	Short 2-3	Use QSPI Serial Flash
CN17	Short 2-3	VCC1833_2 Use power supply at 1.8V
CN20	Short 1-2	Use 3 ports in same PHY mode
CN21	Short 1-2	Use 3 ports in same PHY mode
CN22	Short 1-2	Use 3 ports in same PHY mode
CN24	Short 2-3	VCC1833_3 Use power supply at 1.8V
CN25	Short 1-2	When using other than the SHOST interface
CN27	Short 1-2	When using the HyperRAM
CN29	Short 1-2	When using the USB Serial
CN31	Short 1-2	Use RS485 transmission method in full duplex
CN32	Short 1-2	Use RS485 transmission method in full duplex
J9	Open	Use J-Link OB

Table 4.6 RZ/N2L-RSK SW4 Settings

SW4	Setting	Description
SW4-1	ON	xSPI0 x1 boot mode
SW4-2	ON	* Refer to" r20ut4984egxxxx-rskplus-rzn2l-v1-um.pdf" and set according to the mode to be used.
SW4-3	ON	
SW4-4	ON	MDD=0, JTAG Authentication by Hash is disabled.
SW4-5	OFF	-
SW4-6	OFF	Enables signals other than the trace signal. (Motor, RS485, etc)
SW4-7	ON	Enables signals other than the external bus. (CAN, Emulator, I2C, etc.)
SW4-8	OFF	Enable SW3.

Table 4.7 RZ/N2L-RSK SW8 Setting.

SW8	Setting	Description
SW8-1	OFF	Enable the "LED_GREEN" signal.
SW8-2	ON	
SW8-3	OFF	
SW8-4	ON	Enable the "LED5" signal.
SW8-5	OFF	
SW8-6	OFF	RS485_DE & M2_VN Configuration Switch Setting
SW8-7	ON	
SW8-8	OFF	CAN_TX & IRQ4 & P02_2 Configuration Switch Setting
SW8-9	OFF	
SW8-10	ON	

Table 4.8 RZ/N2L-RSK SW11 Setting

SW11	Setting	Description
SW11-1	ON	Enable the "LED_RED2" signal.
SW11-2	OFF	
SW11-3	OFF	
SW11-4	OFF	RS485_RX & M2_UP Configuration Switch Setting
SW11-5	ON	
SW11-6	OFF	P21_5 & M2_VP Configuration Switch Setting
SW11-7	ON	
SW11-8	OFF	CAN_RX & ADTRG & P01_7 Configuration Switch Setting
SW11-9	OFF	
SW11-10	ON	

4.2.2 Board Setup

Setting the board for running sample program is shown below.
Build and run the sample code on the RZ/N2L RSK board by following the steps below.
Both loading into RAM and flash can be done using IAR Embedded Workbench or e^2 studio.

1. Connect the decoder to the header "J20" on the RZ/N2L RSK board.

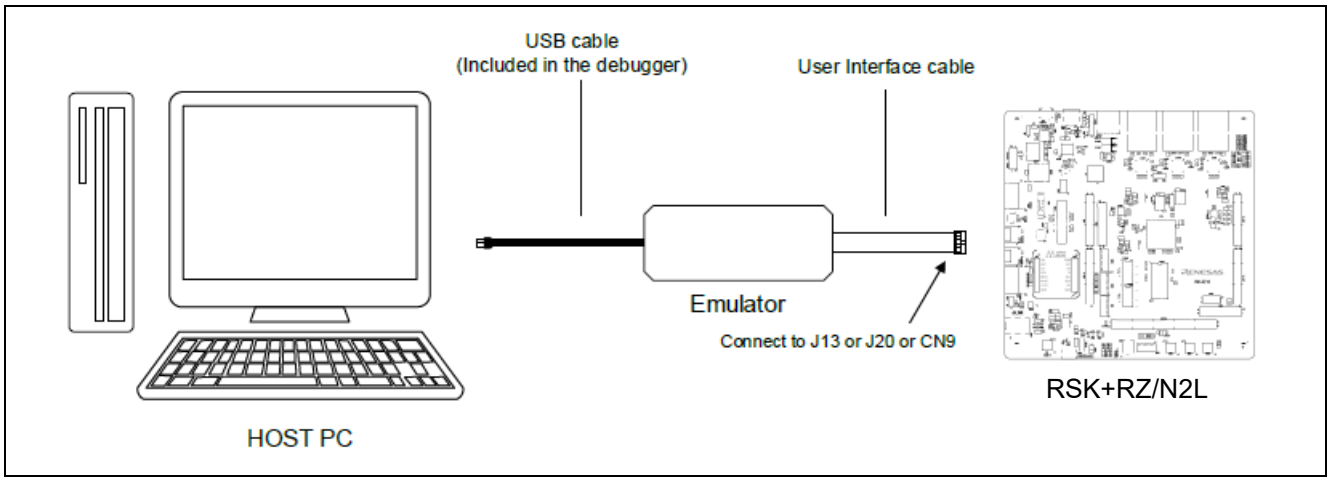


Figure 4.5 RZ/N2L RSK board debug connection diagram

When using J-Link OB, connect the USB cable to the header "J10" and set "J9" to open.

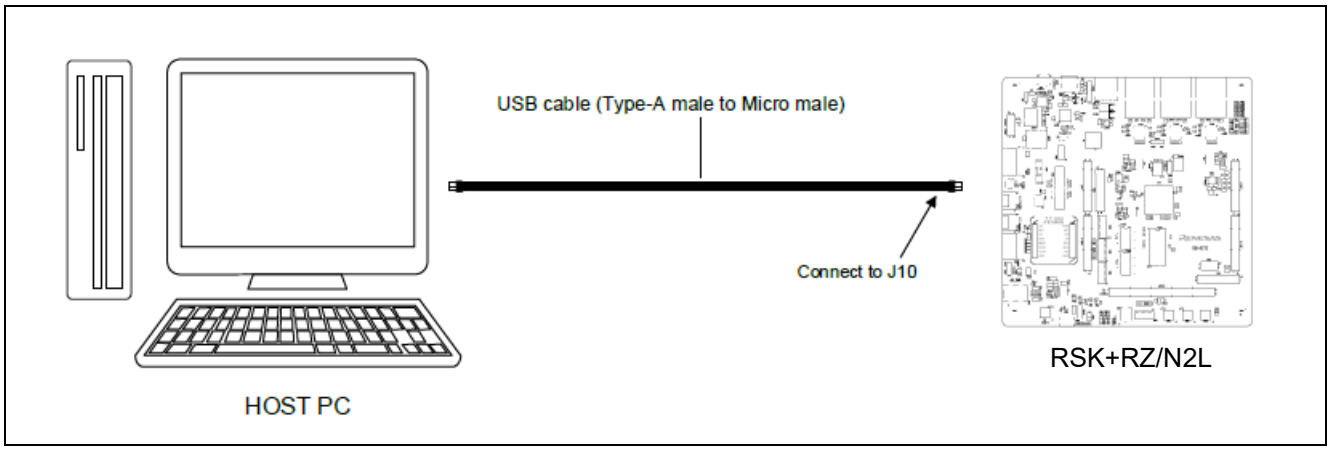


Figure 4.6 RZ/N2L RSK board debug connection diagram (J-Link OB)

2. Power is supplied using a USB cable (Type-C) or an AC / DC adapter. When using a USB cable (Type-C), connect it to the USB connector "CN5" of the RZ/N2L RSK board. When connecting the AC/DC adapter, connect it to the “CN6” connector of the RZ/N2L RSK board.

3. Connect host PC and RZ/N2L RSK board using an ethernet cable. When using an ethernet cable, connect the ethernet cable into “ETH0” connector of the RZ/N2L RSK board.
Depending on the protocol specifications, it may be possible to connect to "ETH1".

4.3 RZ/T2L RSK Board

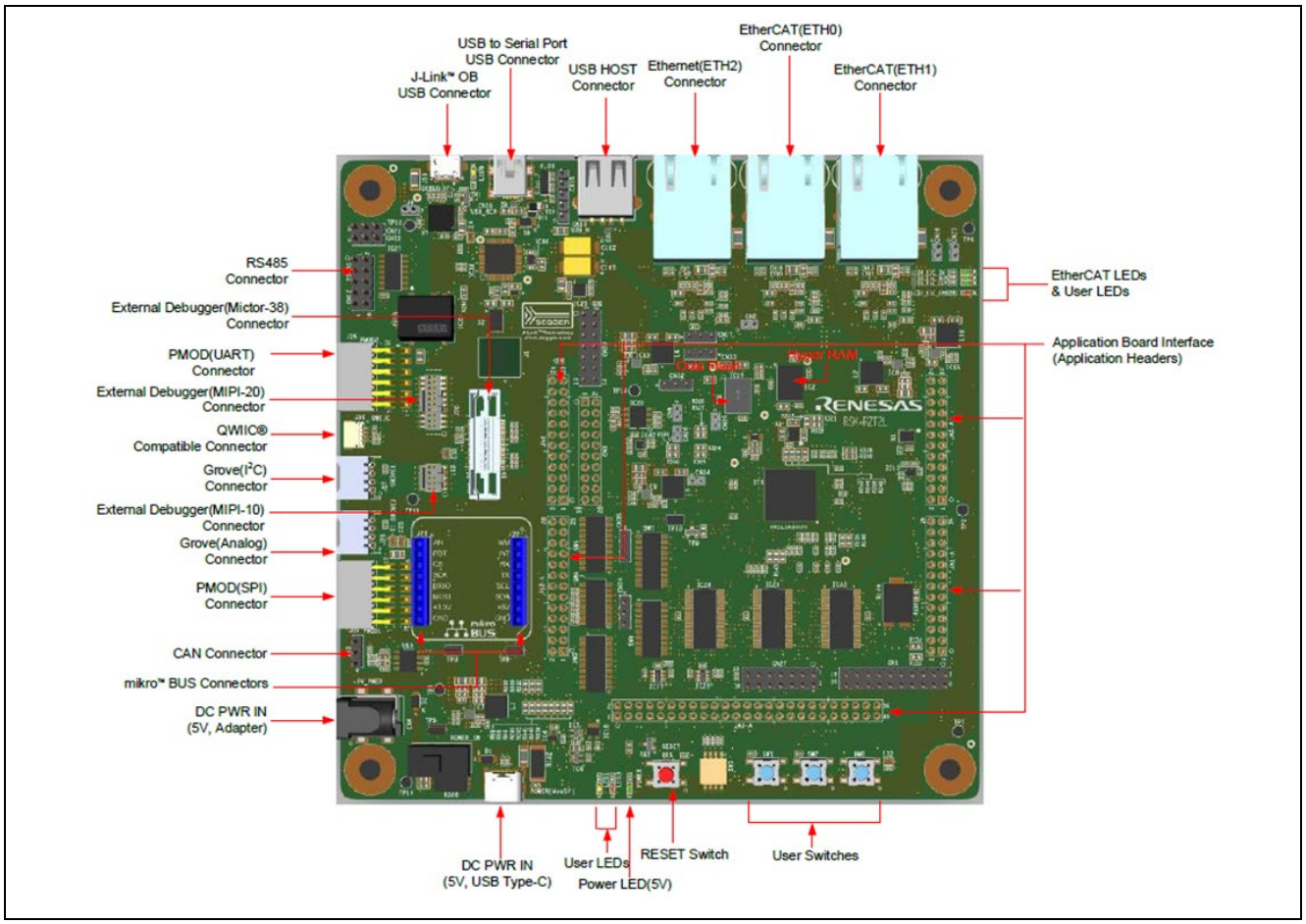


Figure 4.7 RZT2L RSK board layout

4.3.1 Jumper and Switch Configuration

The tables below show the jumper pin and switch settings.
Text in bold red type in the tables indicates the settings changed from the time of shipment of the board.

Table 4.9 RZ/T2L-RSK Jumper configuration

Reference	Jumper Position	Description
J9	**Open**	**The on-board debugging function J-Link® OB is enabled.**
CN2	Short	VCC1833_2 current measurement
CN4	Short	VCC1833_3 current measurement
CN24	Short	VCC11_RVCORE current measurement
CN25	Short	CPU1V8 current measurement
CN26	Short	CPU3V3 current measurement
CN17	Short 2-3	VCC1833_2 Use power supply at 1.8V
CN32	Short 1-2	VCC1833_3 Use power supply at 3.3V
CN21	Short 2-3	Half-duplex communication
CN22		
CN34	Short 1-2	Use TXD0_P16_0_JA1 as transmit data signal and RXD0_P16_1_JA1 as receive data signal (use SCI ch0)
CN35		
CN33	Short 1-2	Connect XSPI1_CS0# to CS# of QuadSPI FLASH (IC3)
CN18	Short 1-2	Connect P21_3 to LED0_ESC_RUN
CN23	Short 1-2	Connect P21_6 to LED4_ESC_IN

Table 4.10 RZ/T2L-RSK SW4 Settings

SW4	Setting	Description
SW4-1	ON	xSPI1 x1 boot mode
SW4-2	ON	* Refer to" r20ut5164ejxxxx-rskrzt2l.pdf" and set according to the mode to be used.
SW4-3	OFF	
SW4-4	OFF	MDD=0, JTAG Authentication by Hash is disabled.
SW4-5	ON	JTAG mode = Normal mode
SW4-6	ON	VCC1833_2 = 1.8V
SW4-7	OFF	VCC1833_3 = 3.3V
SW4-8	OFF	-

Table 4.11 RZ/T2L-RSK SW5 Setting.

SW5	Setting	Description
SW5-1	ON	P01_7 is used as CAN_RX_OB of the CAN interface.
SW5-2	OFF	
SW5-3	OFF	
SW5-4	OFF	
SW5-5	OFF	P02_0 is used as CAN_TX1_JA5 of JA5-A.
SW5-6	OFF	
SW5-7	OFF	
SW5-8	ON	
SW5-9	OFF	-
SW5-10	OFF	-

Table 4.12 RZ/T2L-RSK SW6 Setting.

SW6	Setting	Description
SW6-1	ON	P02_2 is used as CAN_TX_OB of the CAN interface.
SW6-2	OFF	
SW6-3	OFF	
SW6-4	OFF	
SW6-5	OFF	P02_3 is used as CAN_RX1_JA5 of JA5-A.
SW6-6	OFF	
SW6-7	OFF	
SW6-8	ON	
SW6-9	OFF	-
SW6-10	OFF	-

Table 4.13 RZ/T2L-RSK SW7 Setting.

SW7	Setting	Description
SW7-1	ON	**Use P05_5 as ETH1_LINK for EtherCAT Port1**
SW7-2	**OFF**	
SW7-3	OFF	
SW7-4	OFF	Use P17_6 as LED1 for user LED control
SW7-5	OFF	
SW7-6	ON	
SW7-7	OFF	Use P18_1 as LED3 for user LED control
SW7-8	OFF	
SW7-9	OFF	
SW7-10	ON	

Table 4.14 RZ/T2L-RSK SW8 Setting.

SW8	Setting	Description
SW8-1	ON	Use P22_3 as Ethernet Port GMAC_RESETOUT#
SW8-2	OFF	
SW8-3	OFF	
SW8-4	ON	Use P22_1 as serial host interface, HSPI_IO7_M2POE_BSC_D08 for JA5-A and JA3-A (When SW8-10=OFF:TRACE_OPTION_SEL=H)
SW8-5	OFF	
SW8-6	OFF	–
SW8-7	**ON**	**ECAT0_OPTION_SEL = 'L'** **Select ETH0-related signals with bus switch IC33**
SW8-8	**ON**	**ECAT1_OPTION_SEL = 'L'** **Select ETH1-related signals with bus switch IC41**
SW8-9	**ON**	**XSPI1_OPTION_SEL = 'H'** **Select signals other than XSPI1 related signals with bus switch IC37**
SW8-10	OFF	TRACE_OPTION_SEL = 'H' Select signals other than TRACE-related signals with bus switch IC12

4.3.2 Board Setup

Setting the board for running sample program is shown below.
Build and run the sample code on the RZ/T2L RSK board by following the steps below.
Both loading into RAM and flash can be done using IAR Embedded Workbench or e^2 studio.

1. Connect the decoder to the header "J20" on the RZ/T2L RSK board.

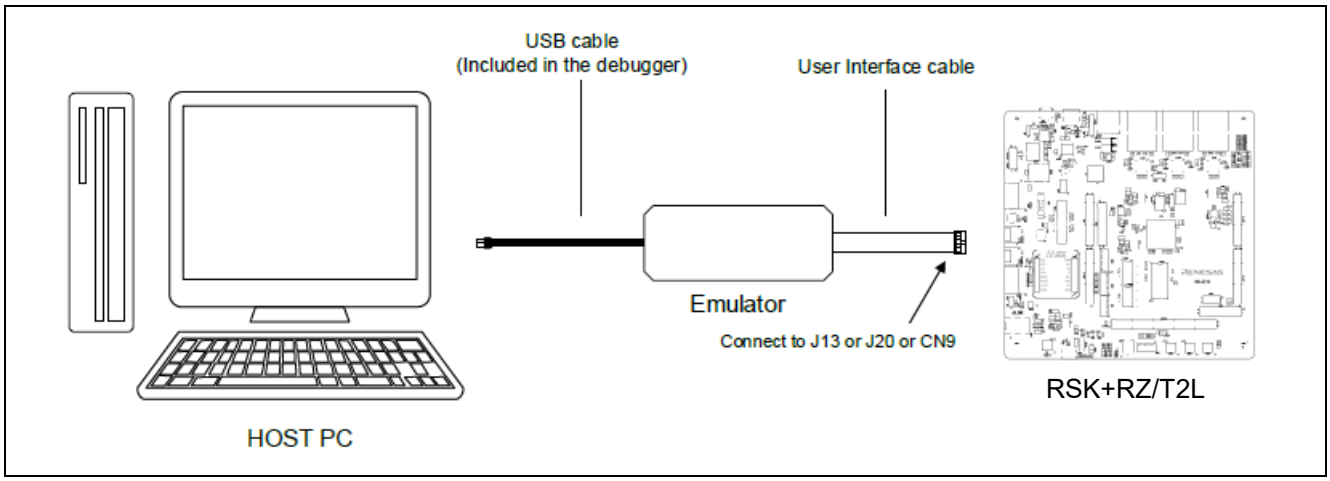


Figure 4.8: RZ/T2L RSK board debug connection diagram

When using J-Link OB, connect the USB cable to the header "J10" and set "J9" to open.

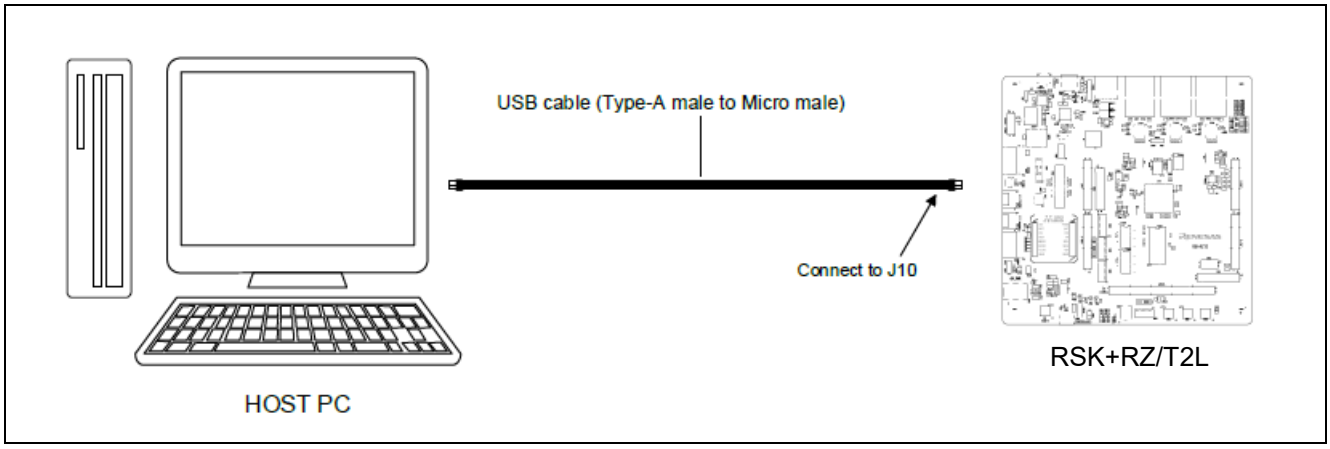


Figure 4.9: RZ/T2L RSK board debug connection diagram (J-Link OB)

2. Power is supplied using a USB cable (Type-C) or an AC / DC adapter. When using a USB cable (Type-C), connect it to the USB connector "CN5" of the RZ/T2L RSK board. When connecting to the AC/DC adapter, connect it to the connector "CN6" of the RZ/T2L RSK board.

3. Connect host PC and RZ/T2L RSK board using an ethernet cable. When using an ethernet cable, connect the ethernet cable into “ETH0” connector of the RZ/T2L RSK board.
Depending on the protocol specifications, it may be possible to connect to "ETH1".

4.4 RZ/T2H Evaluation Board

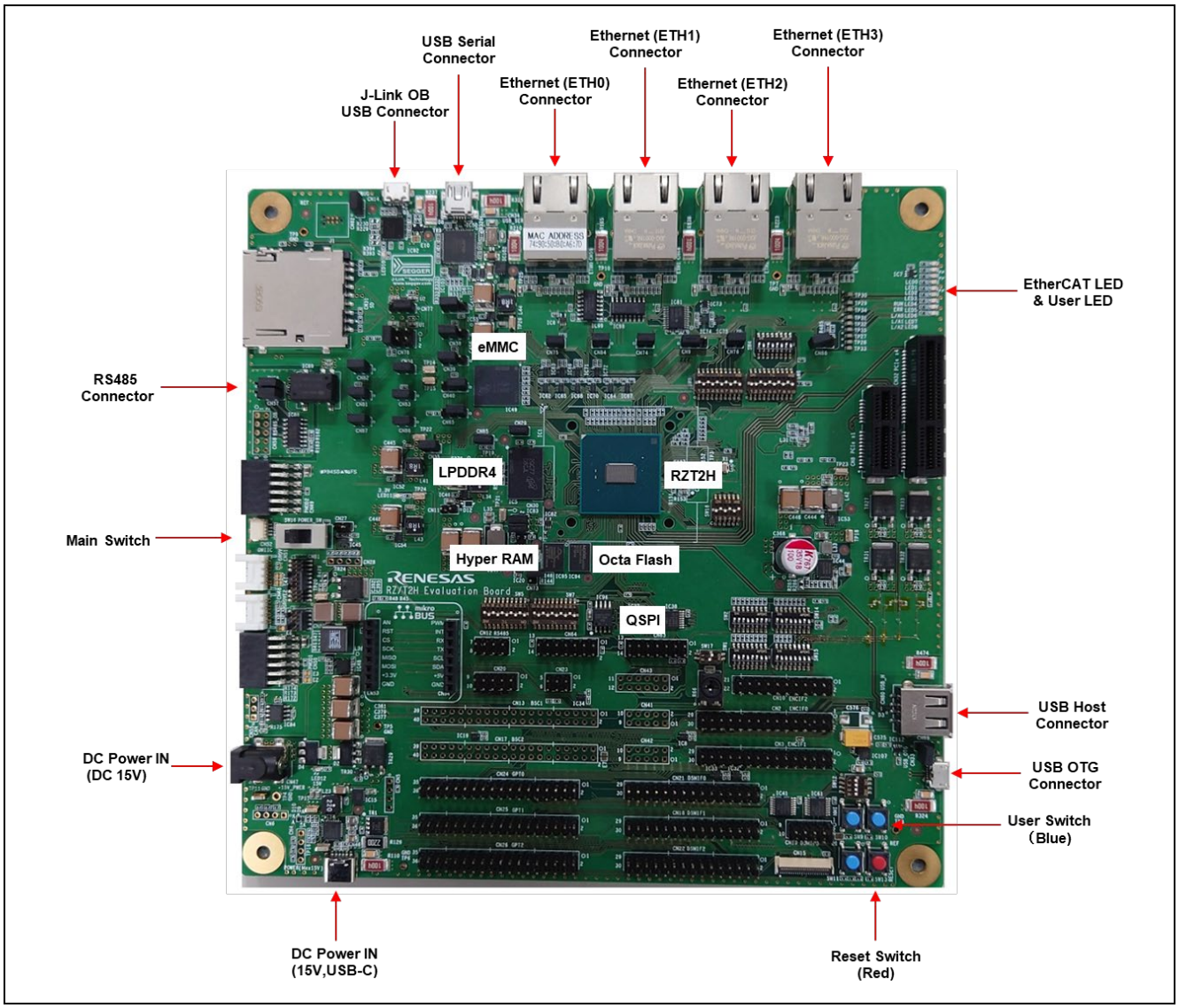


Figure 4.10: RZT2H Evaluation board layout

4.4.1 Jumper and Switch Configuration

The tables below show the jumper pin and switch settings.
Text in bold red type in the tables indicates the settings changed from the time of shipment of the board.

Table 4.15 RZ/T2H EVB Jumper configuration

Reference	Jumper Position	Description
CN62	**Open**	**The on-board debugging function J-Link™ OB is enabled.**
CN73	Open	PCIe reset is not included in the system reset factors.
CN9	Short 1-2	VCC1833_0 is supplied to VCC_ETH2_MDIO (SW2-6 is ON: When P21_4 and P21_5 are selected for MDIO).
CN37	Short 2-3	3.3-V power is supplied to VCC1833_0 (for Ethernet port 0).
CN38	Short 2-3	3.3-V power is supplied to VCC1833_1 (for Ethernet port 1).
CN39	Short 1-2	1.8-V power is supplied to VCC1833_2 (for Ethernet port 2).
CN40	Short 1-2	1.8-V power is supplied to VCC1833_3 (for Ethernet port 3).
CN77	Short 2-3	The output from the power-supply control IC for SD1 is supplied to VCC1833_7 (for SD1).
CN78	Short 5-6	1.8-V power is supplied to VCC1833_6 (for SD0).
CN56 CN57	Short 2-3	Half-duplex communication

Table 4.16 RZ/T2H EVB SW14 Settings

SW14	Setting	Description
SW14-1	ON	xSPI1 boot mode (x1 boot serial flash) * Refer to" r20ut5405ejxxxx-rzt2hevb.pdf" and set according to the mode to be used.
SW14-2	OFF	
SW14-3	ON	
SW14-4	OFF	CPU0 ATCM wait cycle = 1 wait cycle
SW14-5	OFF	CPU1 ATCM wait cycle = 1 wait cycle
SW14-6	OFF	Supply voltage of boot peripheral is 3.3 V
SW14-7	ON	JTAG mode = Normal mode
SW14-8	OFF	-

Table 4.17 RZ/T2H EVB SW1 Settings

SW1	Setting	Description
SW1-1	ON	XTALSEL = 'L' Select oscillator for RZ/T2H clock input.
SW1-2	OFF	-
SW1-3	ON	P35_3,4,5,6 are connected to SW12 and used as user DIPSW inputs.
SW1-4	OFF	P13_4, P13_5, and P14_0 are used as RXD3, TXD3, and DE3 of the RS485.
SW1-5	ON	P00_0,1,2 are used as USB power supply IC control signals.
SW1-6	ON	P01_0,1,2,4,5,6,7 and P02_0,1,2,3 are used as XSPI1 signals.
SW1-7	OFF	-
SW1-8	OFF	-

Table 4.18 RZ/T2H EVB SW2 Settings

SW2	Setting	Description
SW2-1	ON	P12_0 to 7, P13_0 to 2 are connected to eMMC.
SW2-2	ON	
SW2-3	**OFF**	**P17_4, P08_5, and P08_6 are connected to BSC1 (CN13), BSC2 (CN17), DSMIF0 (CN21), SEI (SW9), and LED3.**
SW2-4	OFF	-
SW2-5	OFF	-
SW2-6	ON	GMAC0 (P21_4, P21_5) are connected to MDC/MDIO of Ethernet Port2.
SW2-7	ON	P29_1 to 7, P30_0 to 4, P31_2 to 5 are used as Ethernet Port2 control signals.
SW2-8	ON	P27_2, P33_2 to P33_7, P34_0 to P34_5, P34_7, and P35_0 to P35_2 are used as control signals for Ethernet port 3.

Table 4.19 RZ/T2H EVB SW4 Settings

SW4	Setting	Description
SW4-1	ON	P27_0 is used as ETH1_CRS.
SW4-2	OFF	
SW4-3	ON	P27_1 is used as ETH1_COL.
SW4-4	OFF	
SW4-5	ON	P27_4 is used as RXD0 of USB-to-serial conversion.
SW4-6	OFF	
SW4-7	ON	P27_5 is used as TXD0 of USB-to-serial conversion.
SW4-8	OFF	

Table 4.20 RZ/T2H EVB SW5 Settings

SW5	Setting	Description
SW5-1	OFF	P32_2 is used as USER_LED1.
SW5-2	ON	
SW5-3	OFF	When SW2-3 = ON, P08-6 is used as SD1_IOVS.
SW5-4	ON	
SW5-5	OFF	P07_5 is used as XSPI0_ECS# for OctaFlash.
SW5-6	ON	
SW5-7	OFF	P23_0 is used as ESC_LINKACT1.
SW5-8	ON	
SW5-9	OFF	P22_7 is used as ESC_LINKACT0.
SW5-10	ON	

Table 4.21 RZ/T2H EVB SW6 Settings

SW6	Setting	Description
SW6-1	OFF	P11_0 is used as ESC_RESETOUT2#.
SW6-2	OFF	
SW6-3	ON	
SW6-4	OFF	P11_0 is used as ESC_RESETOUT01#.
SW6-5	ON	
SW6-6	OFF	-
SW6-7	ON	P23_3 is used as ESC_I2CCLK.
SW6-8	OFF	
SW6-9	ON	P23_4 is used as ESC_I2CDATA.
SW6-10	OFF	

Table 4.22 RZ/T2H EVB SW7 Settings

SW7	Setting	Description
SW7-1	OFF	P24_4 is used as CAN_TX.
SW7-2	ON	
SW7-3	OFF	P24_3 is used as CAN_RX.
SW7-4	ON	
SW7-5	OFF	P23_5 is used as ESC_LINKACT2.
SW7-6	ON	
SW7-7	OFF	VUBUSIN is used as USB_Function.
SW7-8	ON	
SW7-9	OFF	P00_0 is used as USB_HF_VBUSEN.
SW7-10	ON	

Table 4.23 RZ/T2H EVB SW8 Settings

SW8	Setting	Description
SW8-1	ON	P18_1 is used as ESC_LED_ERR.
SW8-2	OFF	
SW8-3	ON	P18_0 is used as ESC_LED_RUN.
SW8-4	OFF	
SW8-5	ON	P16_3 is used as RXD5 of USB-to-serial conversion.
SW8-6	OFF	
SW8-7	ON	P16_4 is used as TXD5 of USB-to-serial conversion.
SW8-8	OFF	
SW8-9	ON	P23_1 is used as USER_LED0.
SW8-10	OFF	

Table 4.24 RZ/T2H EVB SW15 Settings

SW15	Setting	Description
SW15-1	ON	PCIe functions is used as Root Complex.
SW15-2	ON	PCIe L1 is used as a root complex.
SW15-3	OFF	The PCIe function is used in a configuration of 2 lanes × 1 port.
SW15-4	OFF	-
SW15-5	OFF	The 12-V power supply of the PCIe x4 connector CN32 is OFF.
SW15-6	OFF	The 3.3-V power supply of the PCIe x4 connector CN32 is OFF.
SW15-7	OFF	The 3.3-V power supply of the PCIe x1 connector CN8 is OFF.
SW15-8	OFF	The 12-V power supply of the PCIe x1 connector CN8 is OFF.

Table 4.25 RZ/T2H EVB SW17 Settings

SW17	Setting	Description
SW17-1	ON	AN000 is connected to potentiometer.
SW17-2	OFF	

Table 4.26 RZ/T2H EVB SW18 Settings

SW18	Setting	Description
SW18-1	OFF	AN100 is connected to mikroBUS™.
SW18-2	ON	
SW18-3	OFF	AN101 is connected to Grove2.
SW18-4	ON	
SW18-5	OFF	AN102 is connected to Grove2.
SW18-6	ON	

4.4.2 Board Setup

Setting the board for running sample program is shown below.
Build and run the sample code on the RZ/T2H Evaluation board by following the steps below.
Both loading into RAM and flash can be done using IAR Embedded Workbench or e^2 studio.

1. Connect the decoder to the header "CN61" on the RZ/T2H evaluation board.

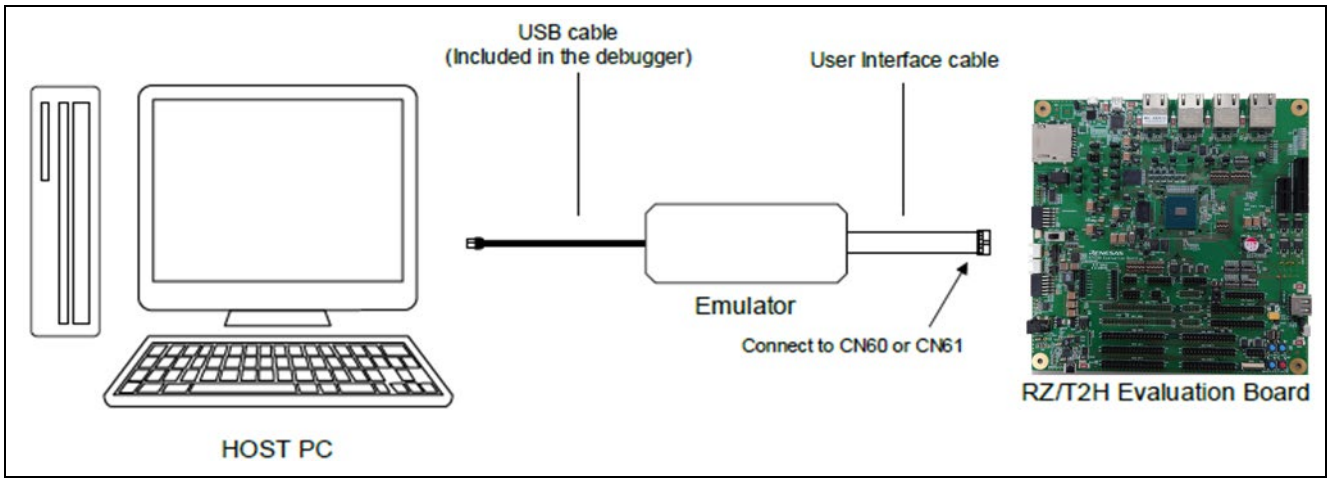


Figure 4.11: RZ/T2H evaluation board debug connection diagram

When using J-Link OB, connect the USB cable to the header "CN14" and set "CN62" to open.

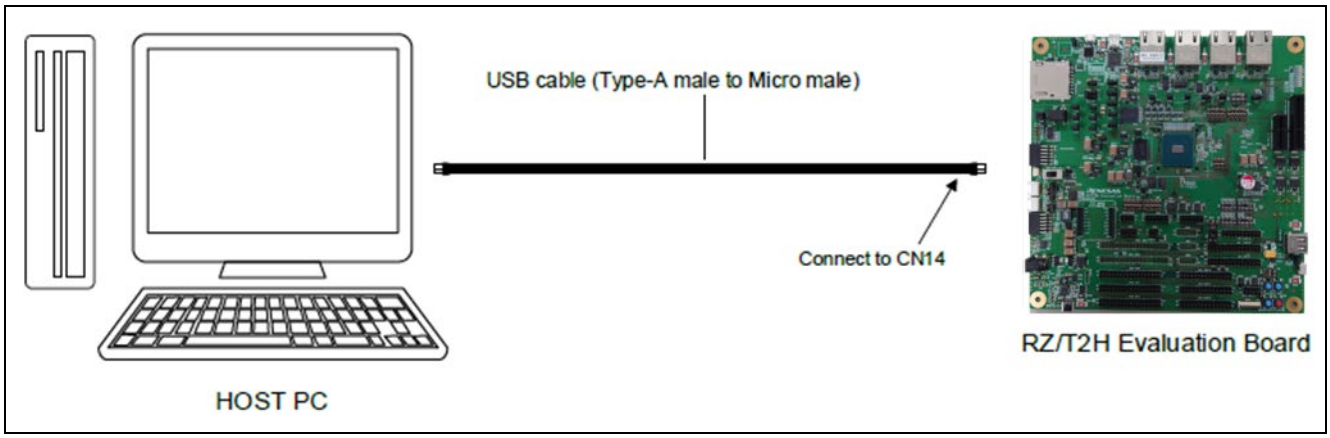


Figure 4.12: RZ/T2H evaluation board debug connection diagram (J-Link OB)

2. Power is supplied using a USB cable (Type-C) or an AC / DC adapter. When using a USB cable (Type-C), connect it to the USB connector "CN46" of the RZ/T2H evaluation board. When connecting to the AC/DC adapter, connect it to the connector "CN47" of the RZ/T2H evaluation board.

3. Connect host PC and RZ/T2H evaluation board using an ethernet cable. When using an ethernet cable, connect the ethernet cable into “ETH0” connector of the RZ/T2H evaluation board.
Depending on the protocol specifications, it may be possible to connect to "ETH1".

4.5 RZ/N2H Evaluation Board

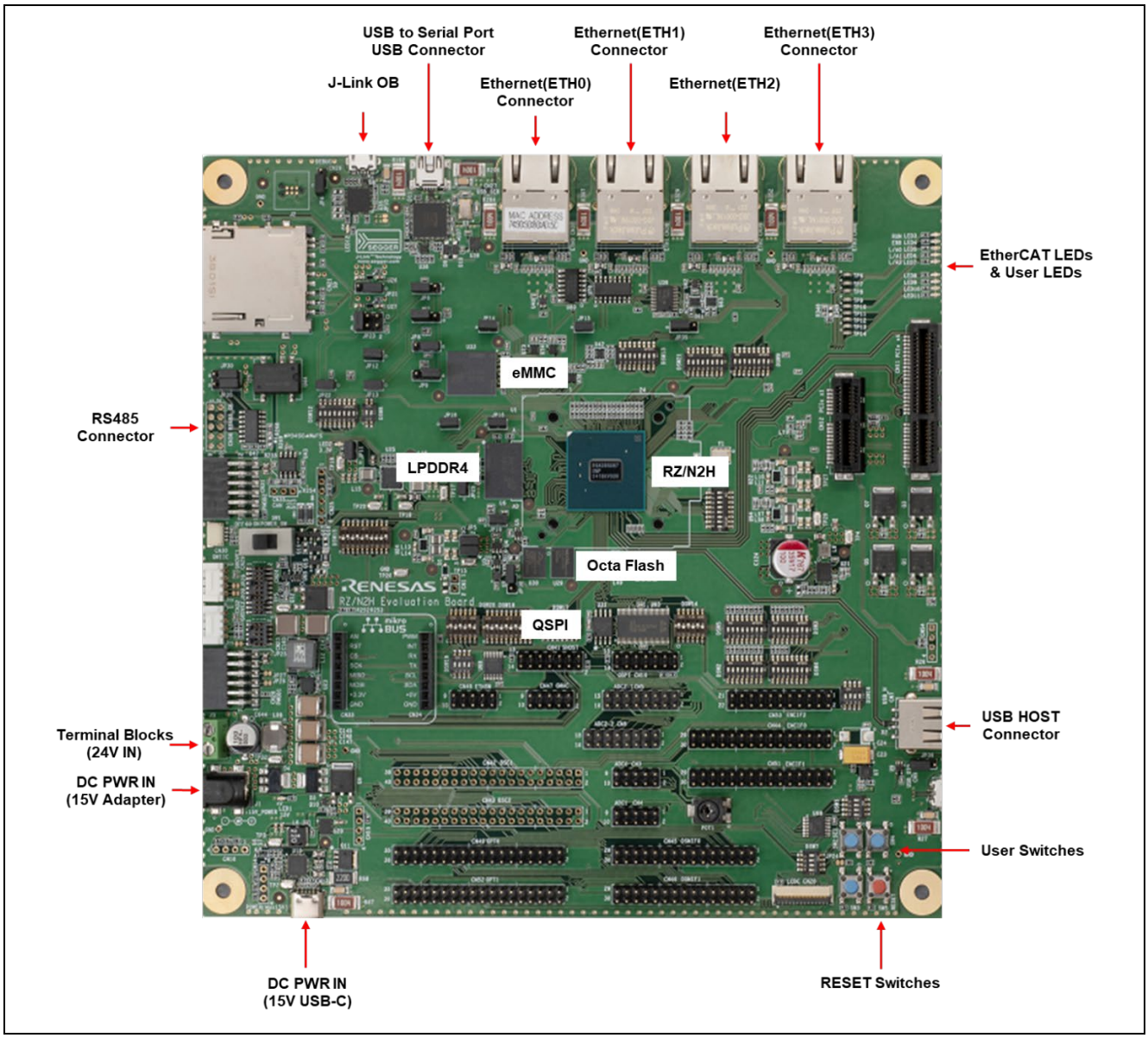


Figure 4.13: RZN2H Evaluation board layout

4.5.1 Jumper and Switch Configuration

The tables below show the jumper pin and switch settings.
Text in bold red type in the tables indicates the settings changed from the time of shipment of the board.

Table 4.27 RZ/N2H EVB Jumper configuration

Reference	Jumper Position	Description
JP6	Short 2-3	3.3-V power is supplied to VCC1833_0. (for Ethernet Port 0)
JP7	Short 2-3	3.3-V power is supplied to VCC1833_1. (for Ethernet Port 1)
JP8	Short 1-2	1.8-V power is supplied to VCC1833_2. (for Ethernet Port 2)
JP9	Short 1-2	1.8-V power is supplied to VCC1833_3. (for Ethernet Port 3)
JP21	Short 2-3	Power control IC output for SD1 is supplied to VCC1833_7. (for SD1)
JP23	Short 5-6	1.8-V power is supplied to VCC1833_6. (for SD0)
JP35	Short 1-2	VCC1833_0 is supplied to VCC_ETH2_MDIO (DSW5-6 is ON: When P21_4 and P21_5 are selected for MDIO).
JP40	**Open**	**The on-board debugging function J-Link OB is enabled.**
JP30	Short 2-3	Half-duplex communication
JP31		
JP38	Open	A PCIe reset is not included in the system reset factors.

Table 4.28 RZ/N2H EVB SW3 Settings

SW3	Setting	Description
SW3-1	ON	xSPI1 boot mode (x1 boot serial flash)
SW3-2	OFF	* Refer to" r20ut5522ejxxxx-rzn2hevb.pdf" and set according to the mode to be used.
SW3-3	ON	
SW3-4	OFF	The number of ATCM wait cycles of CPU0 in the Cortex-R52 is 1 wait cycle.
SW3-5	OFF	The number of ATCM wait cycles of CPU1 in the Cortex-R52 is 1 wait cycle.
SW3-6	OFF	The power-supply voltage of the boot peripheral is 3.3 V.
SW3-7	ON	JTAG mode = Normal mode
SW3-8	OFF	-

Table 4.29 RZ/N2H EVB SW2 Settings

SW2	Setting	Description
SW2-1	ON	XTALSEL = 'L' Select oscillator for RZ/N2H clock input.
SW2-2	OFF	-
SW2-3	ON	P27_2, P27_3, P27_6, and P31_3 are used as inputs to user DIP switches.
SW2-4	OFF	P13_4, P13_5, and P14_0 are used as RXD3, TXD3, and DE3 of the RS485.
SW2-5	ON	P00_0 to P00_2 are used as control signals for the USB power-supply IC. In this case, set DSW14-1, DSW14-3, and DSW14-5 to ON and DSW14-2, DSW14-4, and DSW14-6 to OFF.
SW2-6	ON	P01_0, P01_2, P01_4 to P01_7, and P02_0 to P02_3 are used as XSPI1 signals.
SW2-7	OFF	-
SW2-8	OFF	-

Table 4.30 RZ/N2H EVB SW4 Settings

SW4	Setting	Description
SW4-1	ON	PCIe L0 functions is used as Root Complex.
SW4-2	ON	PCIe L1functions is used as Root Complex.
SW4-3	OFF	The PCIe function is used in a configuration of 2 lanes × 1 port.
SW4-4	OFF	-
SW4-5	OFF	The 12-V power supply of the PCIe x1 connector CN12 is OFF.
SW4-6	OFF	The 3.3-V power supply of the PCIe x1 connector CN12 is OFF.
SW4-7	OFF	The 12-V power supply of the PCIe x4 connector CN11 is OFF.
SW4-8	OFF	The 3.3-V power supply of the PCIe x4 connector CN11 is OFF.

Table 4.31 RZ/N2H EVB SW5 Settings

SW5	Setting	Description
SW5-1	ON	P12_0 to 7, P13_0 to 2 are connected to eMMC.
SW5-2	ON	
SW5-3	ON	P08_6 and P17_4 are used as SD1 control signal.
SW5-4	OFF	-
SW5-5	OFF	-
SW5-6	ON	MDC and MDIO of Ethernet port 2 are connected to GMAC0 (P21_4 and P21_5).
SW5-7	ON	P29_1 to P29_7, P30_0 to P30_4, P30_7, P31_2, P31_4, and P31_5 are used as control signals for Ethernet Port 2.
SW5-8	ON	P00_0 to P00_2, P33_2 to P33_7, and P34_0 to P34_6 are used as control signals for Ethernet Port 3.

Table 4.32 RZ/N2H EVB SW6 Settings

SW6	Setting	Description
SW6-1	OFF	AN000 is connected to the potentiometer.
SW6-2	ON	
SW6-3	OFF	AN100 is connected to mikroBUS™(CN33).
SW6-4	ON	
SW6-5	OFF	AN101 is connected to Grove(Analog)(CN32).
SW6-6	ON	
SW6-7	OFF	AN102 is connected to Grove(Analog)(CN32).
SW6-8	ON	

Table 4.33 RZ/N2H EVB SW7 Settings

SW7	Setting	Description
SW7-1	ON	P03_3 is used as I2C_SCL of U11, LCDC(CN20), Grove(I2C)(CN29), QWIIC(CN30), and mikroBUS™(CN34).
SW7-2	OFF	
SW7-3	ON	P03_4 is used as I2C_SDA of U11, LCDC(CN20), Grove(I2C)(CN29), QWIIC(CN30), and mikroBUS™(CN34).
SW7-4	OFF	

Table 4.34 RZ/N2H EVB SW8 Settings

SW8	Setting	Description
SW8-1	**OFF**	**P11_0_ESC_RESETOUT# is used as RESET for Ethernet Port 2 as same as Ethernet Port 0 and 1.**
SW8-2	**ON**	

Table 4.35 RZ/N2H EVB SW9 Settings

SW9	Settings	Description
SW9-1	ON	P27_4 is used as RXD0 of the USB-to-serial conversion.
SW9-2	OFF	
SW9-3	ON	P27_5 is used as TXD0 of the USB-to-serial conversion.
SW9-4	OFF	
SW9-5	ON	P33_3 is used as RXD1 of the USB-to-serial conversion.
SW9-6	OFF	
SW9-7	ON	P33_4 is used as TXD1 of the USB-to-serial conversion.
SW9-8	OFF	

Table 4.36 RZ/N2H EVB SW12 Settings

SW12	Settings	Description
SW12-1	OFF	P00_3 is used as P00_3_ETH3_COL of Ethernet Port 3.
SW12-2	ON	
SW12-3	OFF	P11_0 is used as P11_0_ESC_RESETOUT# of Ethernet Port 0 and 1.
SW12-4	ON	
SW12-5	OFF	P03_2 is used as P03_2_GMAC_RESETOUT3# of Ethernet Port 3.
SW12-6	ON	
SW12-7	OFF	P03_1 is used as P03_1_GMAC_RESETOUT2# of Ethernet Port 2.
SW12-8	ON	In this case, set SW8-1 to ON and SW8-2 to OFF.

Table 4.37 RZ/N2H EVB SW13 Settings

SW13	Settings	Description
SW13-1	ON	P26_7 is used as P26_7_ETH1_RXER of Ethernet Port 1.
SW13-2	OFF	
SW13-3	ON	P27_0 is used as P27_0_ETH1_CRS of Ethernet Port 1.
SW13-4	OFF	
SW13-5	ON	P27_1 is used as P27_1_ETH1_COL of Ethernet Port 1.
SW13-6	OFF	
SW13-7	OFF	P13_7 is used as MDINT of Ethernet Port 2.
SW13-8	ON	

Table 4.38 RZ/N2H EVB SW14 Settings

SW14	Settings	Description
SW14-1	ON	P00_1 is used as USB_OVRCUR.
SW14-2	OFF	In this case, set SW2-5 to ON.
SW14-3	ON	Setting fixed
SW14-4	OFF	
SW14-5	ON	P00_0 is used as VBUSEN.
SW14-6	OFF	In this case, set SW2-5 to ON.

Table 4.39 RZ/N2H EVB SW15 Settings

SW15	Settings	Description
SW15-1	OFF	P22_6 is used as P22_6_SD0_WP.
SW15-2	ON	
SW15-3	OFF	P22_5 is used as P22_5_SD0_CD.
SW15-4	ON	
SW15-5	**ON**	**P14_7 is used as SDA of I2C for EEPROM access.**
SW15-6	**OFF**	
SW15-7	OFF	-
SW15-8	**OFF**	**P14_6 is used as SCK of I2C for EEPROM access.**
DSW15-9	**ON**	
SW15-10	**OFF**	

Table 4.40 RZ/N2H EVB SW16 Settings

SW16	Settings	Description
SW16-1	OFF	USB_VUBUSIN is connected to VBUS of CN8 for Function.
SW16-2	ON	
SW16-3	OFF	USB_VBUSEN is used as USB_HF_VBUSEN.
SW16-4	ON	

Table 4.41 RZ/N2H EVB SW17 Settings

SW17	Settings	Description
SW17-1	OFF	P03_0 is used as USER_LED3(LED11).
SW17-2	ON	
SW17-3	OFF	P02_7 is used as USER_LED2(LED10).
SW17-4	ON	
SW17-5	OFF	P02_6 is used as P02_6_SD0_IOVS.
SW17-6	ON	
SW17-7	OFF	P02_5 is used as P02_5_SD0_PWEN.
SW17-8	ON	

Table 4.42 RZ/N2H EVB SW18 Settings

SW18	Settings	Description
SW18-1	ON	P22_7 is used as ESC_LINKACT0(LED5).
SW18-2	OFF	
SW18-3	ON	P23_0 is used as ESC_LINKACT1(LED6).
SW18-4	OFF	
SW18-5	ON	P14_3 is used as ESC_LINKACT2(LED7).
SW18-6	OFF	
SW18-7	ON	P31_6 is used as ESC_LEDRUN(LED3).
SW18-8	OFF	
SW18-9	ON	P18_1 is used as ESC_LEDERR(LED4).
SW18-10	OFF	

Table 4.43 RZ/N2H EVB SW19 Settings

SW19	Settings	Description
SW19-1	OFF	P17_4 is used as SD1_CD.
SW19-2	ON	
SW19-3	OFF	P14_3 is used as DATG3 of LCDC(CN20), INT of PMOD1(CN28), TX of mikroBUS™(CN34), DREQ of BSC1(CN42), ENCIFOE00 of ENCIF0(CN44), or LED7 of LINKACT2.
SW19-4	ON	

Table 4.44 RZ/N2H EVB SW20 Settings

SW20	Settings	Description
SW20-1	ON	P14_3 is used as DREQ of BSC1(CN42) In this case, set SW18-5 to OFF, SW18-6 to ON, SW19-3 to OFF, and SW19-4 to ON.
SW20-2	OFF	
SW20-3	ON	P14_4 is used as DACK of BSC1(CN42).
SW20-4	OFF	
SW20-5	ON	P14_5 is used as TEND of BSC1(CN42).
SW20-6	OFF	

Table 4.45 RZ/N2H EVB SW21 Settings

SW21	Settings	Description
SW21-1	ON	P26_6 is used as P26_6_ETH1_TXER of Ethernet Port1 and P34_4 is used as CS2# of BSC1(CN42). In this case, set SW5-8 to OFF.
SW21-2	OFF	
SW21-3	ON	
SW21-4	OFF	P34_5 is used as CS3# of BSC1(CN42). In this case, set SW5-8 to OFF.
SW21-5	ON	
SW21-6	OFF	P34_6 is used as CS5# of BSC1(CN42). In this case, set SW5-8 to OFF.
SW21-7	ON	
SW21-8	OFF	-

4.5.2 Board Setup

Setting the board for running sample program is shown below.
Build and run the sample code on the RZ/N2H Evaluation board by following the steps below.
Both loading into RAM and flash can be done using IAR Embedded Workbench or e^2 studio.

1. Connect the decoder to the header "CN24" on the RZ/N2H evaluation board.

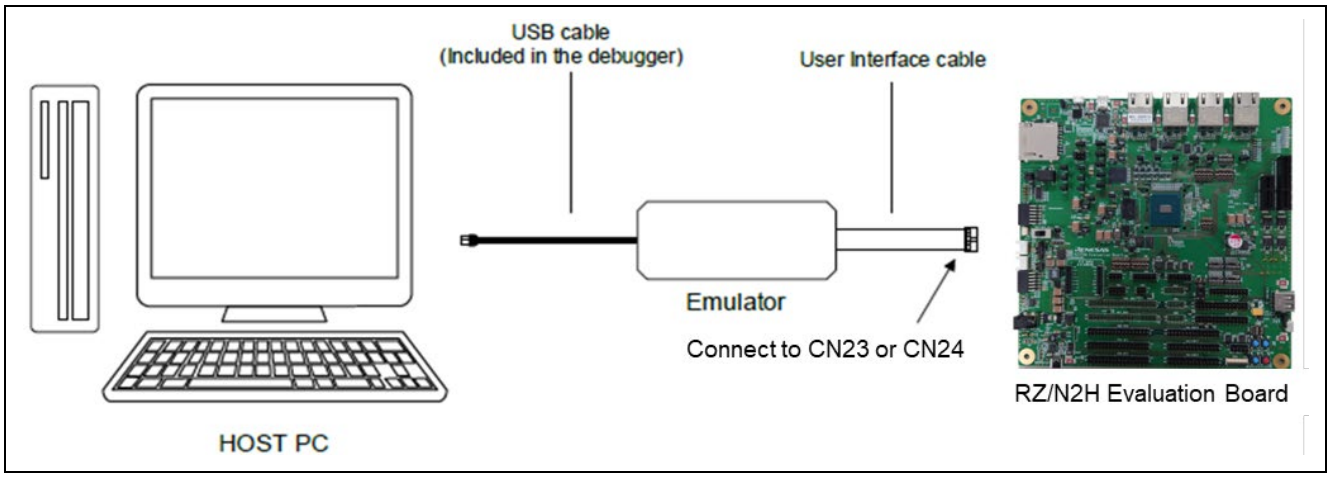


Figure 4.14: RZ/N2H evaluation board debug connection diagram

When using J-Link OB, connect the USB cable to the header "CN26" and set "JP40" to open.

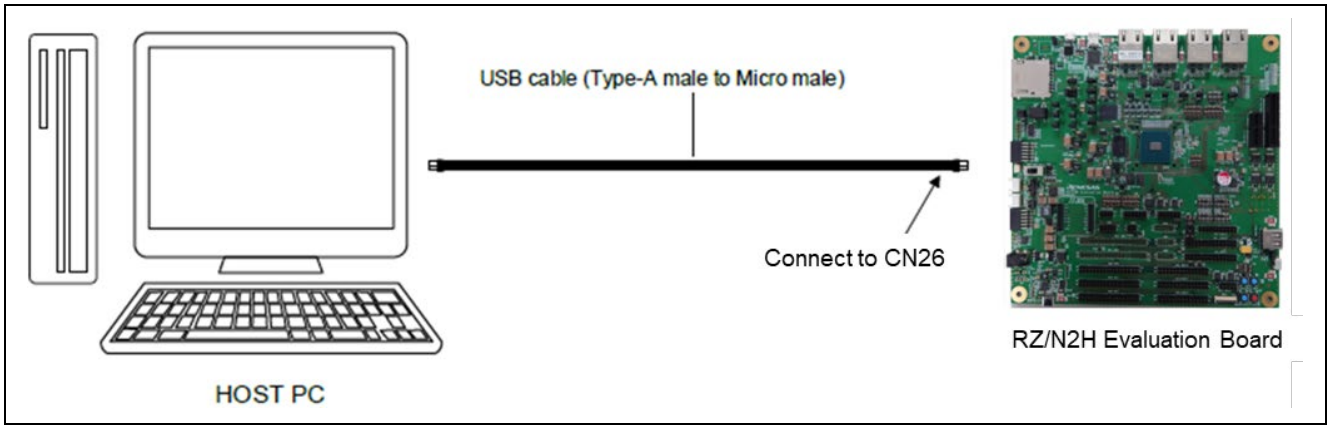


Figure 4.15: RZ/N2H evaluation board debug connection diagram (J-Link OB)

2. Power is supplied using a USB cable (Type-C) or an AC / DC adapter. When using a USB cable (Type-C), connect it to the USB connector "CN13" of the RZ/N2H evaluation board. When connecting to the AC/DC adapter, connect it to the connector "J1" of the RZ/N2H evaluation board.

3. Connect host PC and RZ/N2H evaluation board using an ethernet cable. When using an ethernet cable, connect the ethernet cable into “ETH0” connector of the RZ/N2H evaluation board.
   Depending on the protocol specifications, it may be possible to connect to "ETH1".

5. Set up the Host

5.1 Setting up a TwinCAT3

5.1.1 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 varies depending on the target device:

- For single-core projects (RZ/T2L, RZ/N2L):
  RZT2L_EtherCAT_RSK_rev0400\common\ETG5003\ESI

  Note) If you are using RZ/N2L, replace “**RZT2L”** with “**RZN2L”**.

- For multi-core projects (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/N2H):
  RZT2H_EtherCAT_EVB_rev0400\CR52_Dual\common\ETG5003\ESI

  Note) If you are using a different device or core configuration,
  replace the folder names according to your environment as shown below:

  - RZ/T2M, RZ/T2ME (CR52_Dual only):
    RZT2M_RZT2ME_EtherCAT_RSK_rev0400\CR52_Dual\common\ETG5003\ESI

  - RZ/N2H (CR52_Dual):
    RZN2H_EtherCAT_EVB_rev0400\CR52_Dual\common\ETG5003\ESI

  - RZ/N2H (CR52_CA55):
    RZN2H_EtherCAT_EVB_rev0400\CR52_CA55\common\ETG5003\ESI

The ESI file:

- For RZ/T2L, RZ/T2M, RZ/T2ME, RZ/T2H
  Renesas EtherCAT RZT2 ETG5003.xml

- For RZ/N2L, RZ/N2H
  Renesas EtherCAT RZN2 ETG5003.xml

The TwinCAT destination folder:

\TwinCAT\3.x\Config\IO\EtherCAT

5.1.2 Add Driver

Add the Ether driver for TwinCAT. (First time only)

From the start menu, select [TwinCAT3] → [Show Realtime Ethernet Compatible Devices…].

Select the connected Ether port from the communication ports and install it.

If the selected ethernet adapter is moved in [Installed and ready to use devices(realtime capable)], there is no problem.

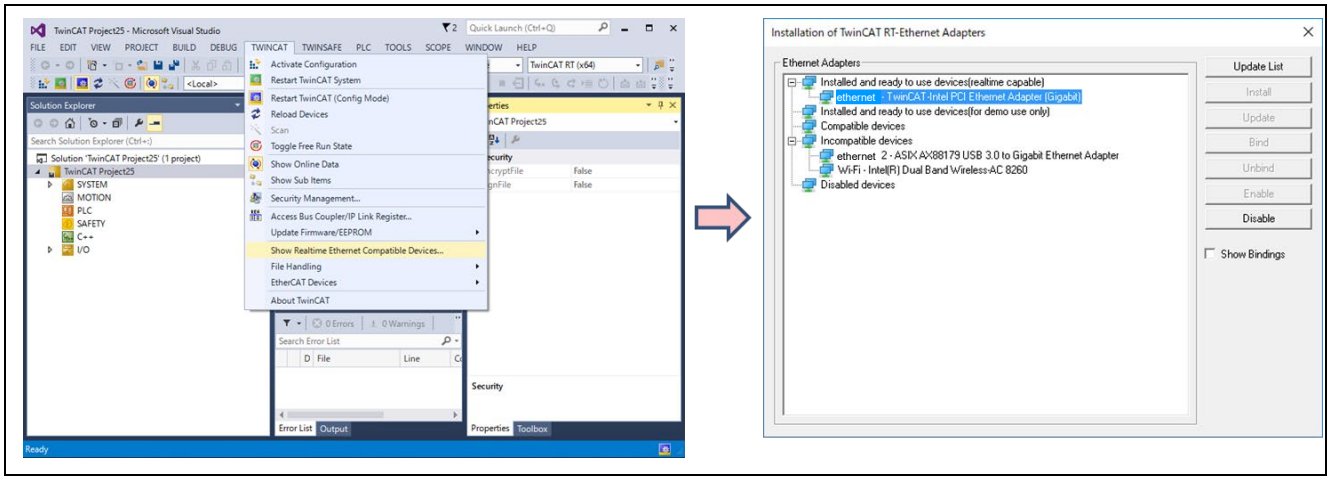


Figure 5.1 Add driver

Note). If you don't use the NIC made by Intel, ethernet adapter is moved in [Installed and ready to use devices(for demo use only)].

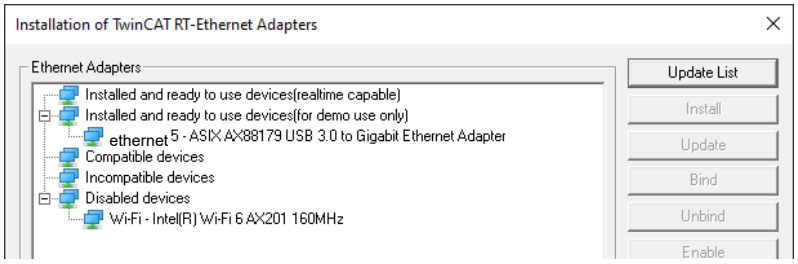


Figure 5.2 In the case of the NIC made by Intel

6. Running the Sample Application

This chapter describes how to generate the EtherCAT SubDevice Stack Code (SSC), build the sample project, and start debugging using either EWARM or e² studio.

The procedures in this chapter use **RZ/T2L** as the representative example for single-core projects, and **RZ/T2H (CR52_Dual)** as the representative example for multi-core projects.

If you are using a different device or core configuration, replace the folder names and project names, and core configuration in the procedures with the values shown in Table 6.1.

Table 6.1 Folder Names and Project Names by Target Device

Target device	Folder Name	Core	Project name (Primary)	Project name (Secondary)
RZ/T2L	RZT2L_EtherCAT_RSK_rev0400	Single (Cortex-R52)	RZT2L_RSK_ESC_ETG5003	(None)
RZ/N2L	RZN2L_EtherCAT_RSK_rev0400	Single (Cortex-R52)	RZN2L_RSK_ESC_ETG5003	(None)
RZ/T2M, RZ/T2ME	RZT2M_RZT2ME_EtherCAT_RSK_rev0400	CR52_Dual (Cortex-R52 CPU0 and Cortex-R52 CPU1)	RZT2M_RSK_ESC_ETG5003_CR52_Dual_Primary	RZT2M_RSK_ESC_ETG5003_CR52_Dual_Secondary
RZ/T2H	RZT2H_EtherCAT_EVB_rev0400	CR52_Dual (Cortex-R52 CPU0 and Cortex-R52 CPU1)	RZT2H_EVB_ESC_ETG5003_CR52_Dual_Primary	RZT2H_EVB_ESC_ETG5003_CR52_Dual_Secondary
		CR52_CA55 (Cortex-R52 CPU0 and Cortex-A55 Core2)	RZT2H_EVB_ESC_ETG5003_CR52_CA55_Primary	RZT2H_EVB_ESC_ETG5003_CR52_CA55_Secondary
RZ/N2H	RZN2H_EtherCAT_EVB_rev0400	CR52_Dual (Cortex-R52 CPU0 and Cortex-R52 CPU1)	RZN2H_EVB_ESC_ETG5003_CR52_Dual_Primary	RZN2H_EVB_ESC_ETG5003_CR52_Dual_Secondary
		CR52_CA55 (Cortex-R52 CPU0 and Cortex-A55 Core2)	RZN2H_EVB_ESC_ETG5003_CR52_CA55_Primary	RZN2H_EVB_ESC_ETG5003_CR52_CA55_Secondary

6.1 Generating the SubDevice Stack Code

The SSC Tool is used to generate the EtherCAT SubDevice stack code before building the project.

Caution) There is a point of caution when using the SSC Tool.
Please refer to the section "Appendix A : A point of caution when using SSC Tool" in advance.

Caution) If the patch command is not installed on your PC, you will need to install the 64-bit patch.exe.
Refer to the section "Appendix B : How to install patch" for instructions.
If it is already installed, skip this step.

1. Open the .esp file corresponding to your target device and core configuration:

   - For **single-core projects** (RZ/T2L, RZ/N2L):

     RZT2L_EtherCAT_RSK_rev0400\common\ETG5003\SSCconfig
     RZT2_EtherCAT_ETG5003_CR52.esp

   - For **multi-core projects** (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/N2H):

     RZT2H_EtherCAT_EVB_rev0400\CR52_Dual\common\ETG5003\SSCconfig
     RZT2_EtherCAT_CR52.esp

Note) If you are using a different device or core configuration, replace the folder name and project name, according to Table 6.1.

Figure 6.1 The .esp file (This is the example of RZ/T2L)

The following window opens.

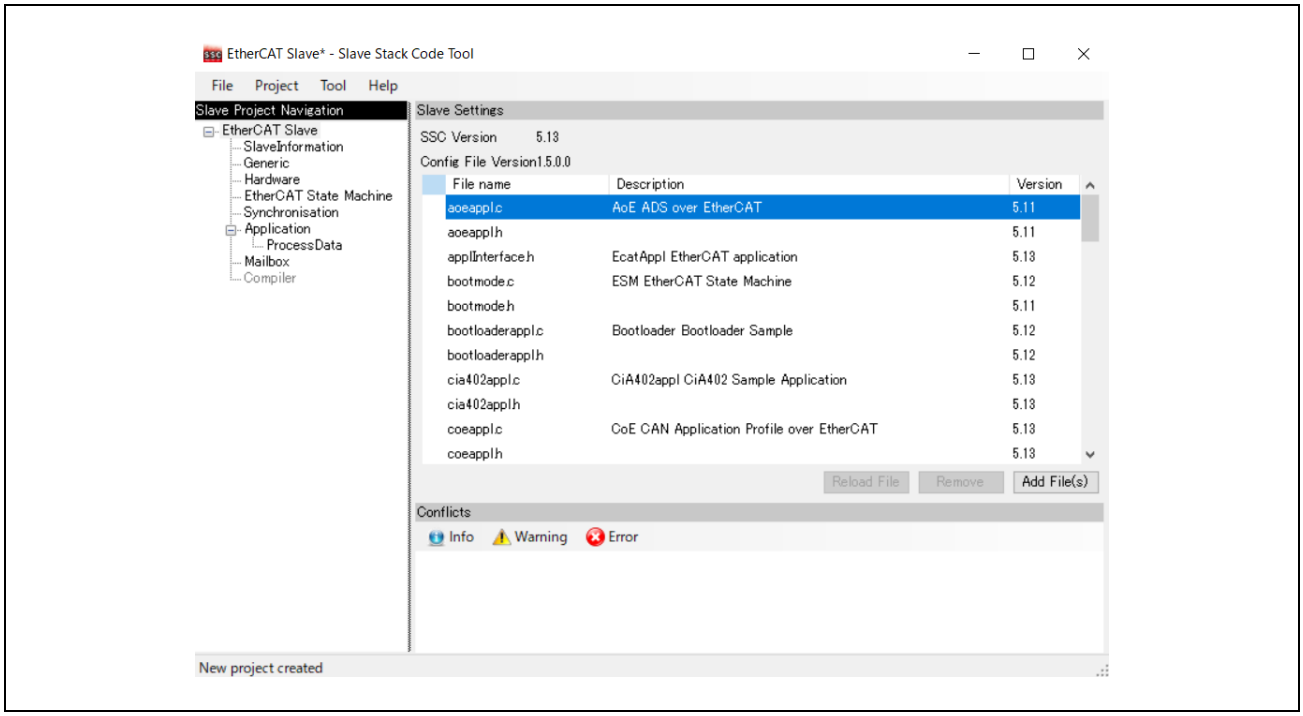


Figure 6.2 SSC tool main window

2. Select **[Project]** → **[Create new Slave Files]**.

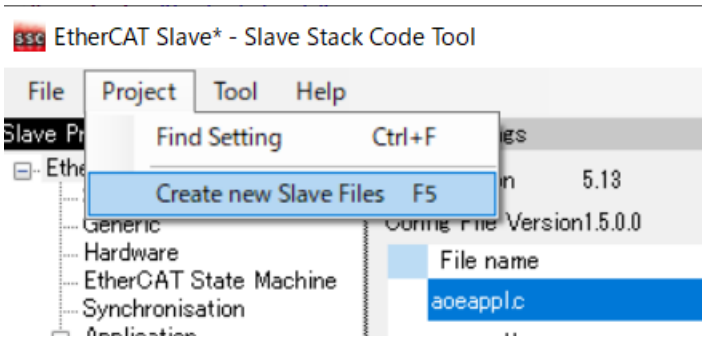


Figure 6.3 Generate EtherCAT SubDevice Stack (1)

3. Click the **[Start]** button to begin generating the EtherCAT SubDevice Stack Code.

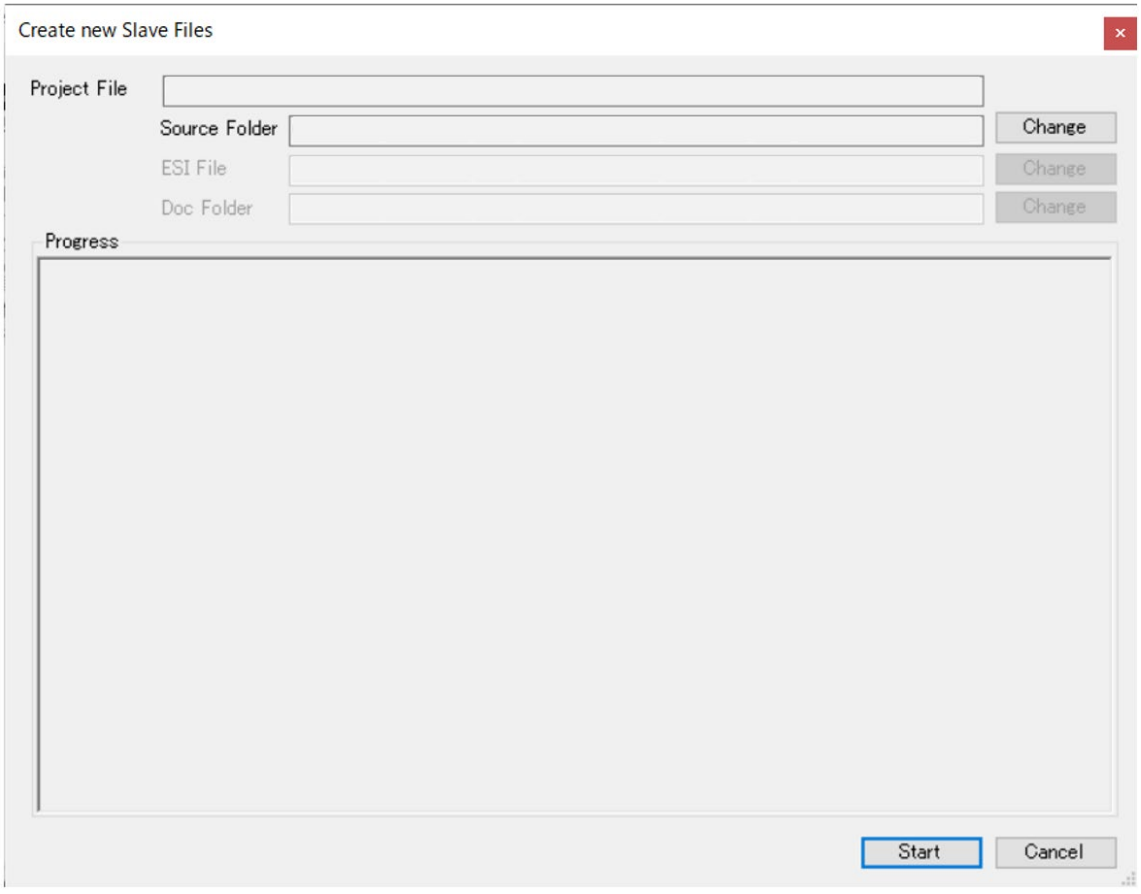


Figure 6.4 Generate EtherCAT SubDevice Stack (2)

4. When the message **"New file created successfully"** appears, the generation is complete.
   The source files are located in:
   - For **single-core projects** (RZ/T2L, RZ/N2L):
     RZT2L_EtherCAT_RSK_rev0400\common\ETG5003\SSCconfig
   - For **multi-core projects** (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/N2H):
     RZT2H_EtherCAT_EVB_rev0400\CR52_Dual\common\ETG5003\SSCconfig

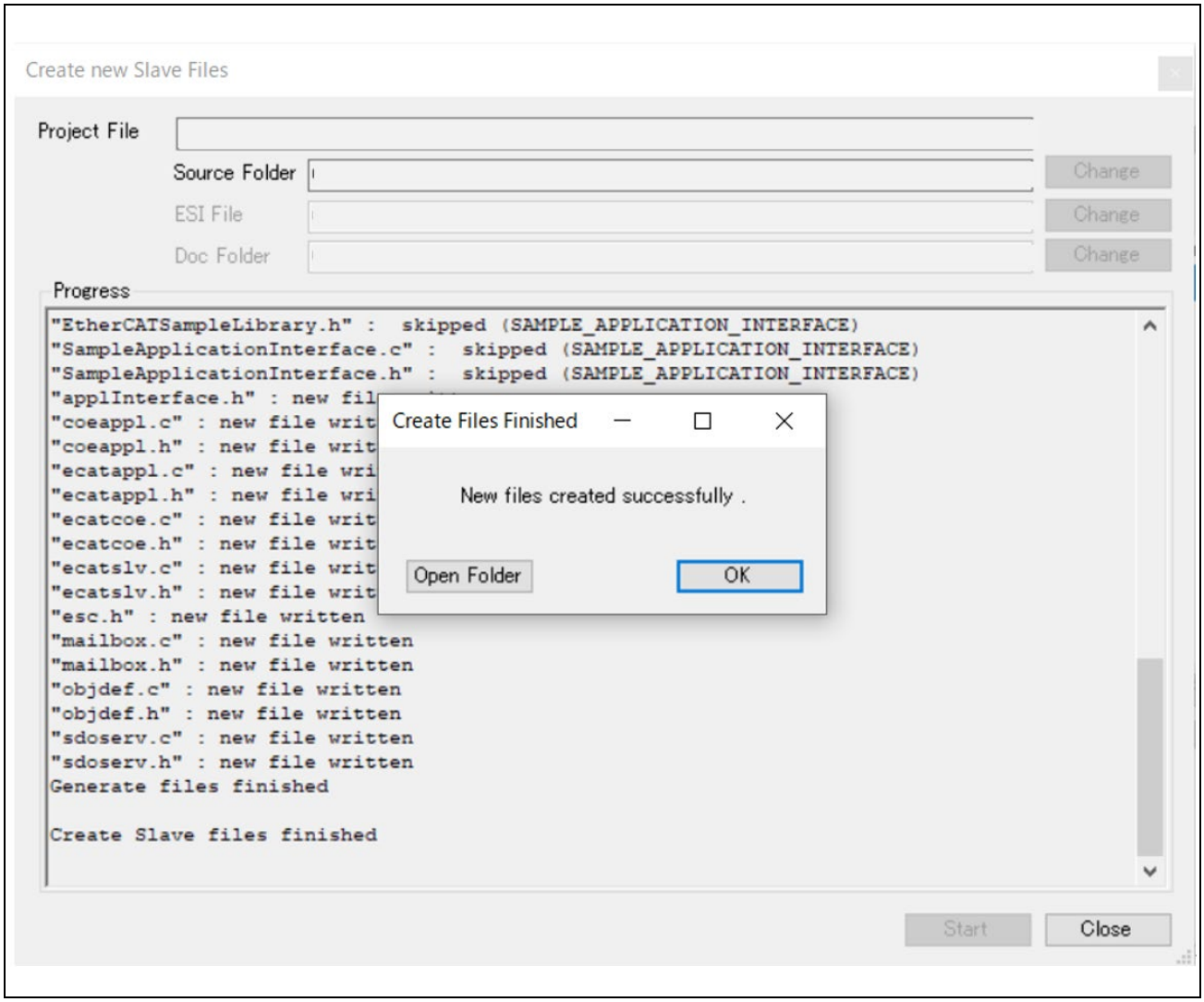


Figure 6.5 Generate EtherCAT SubDevice Stack (3)

5. Execute the batch file to apply the patch and copy the generated stack code to the project folder:

   - For **single-core projects** (RZ/T2L, RZ/N2L):

     RZT2L_EtherCAT_RSK_rev0400\common\ETG5003\Patch\apply_patch_ETG5003.bat

   - For **multi-core projects** (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/N2H):

     RZT2H_EtherCAT_EVB_rev0400\CR52_Dual\common\ETG5003\Patch\apply_patch_ETG5003.bat

   The batch file performs the following operations:

   A) Modifies the EtherCAT SubDevice stack code generated by the SSC Tool.

   The modifications are described in the patch file (.patch).

   B) Copies the Src folder containing the modified stack code to the following project directories:

   - For **single-core projects** (RZ/T2L, RZ/N2L):

     (1) RZT2L_EtherCAT_RSK_rev0400\project\ETG5003\e2studio\RZT2L_RSK_ESC_ETG5003\src\ethercat\beckhoff

     (2) RZT2L_EtherCAT_RSK_rev0400\project\ETG5003\ewarm\RZT2L_RSK_ESC_ETG5003\src\ethercat\beckhoff

   - For **multi-core projects** (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/N2H):

     (1) RZT2H_EtherCAT_EVB_rev0400\CR52_Dual\project\ETG5003\e2studio\RZT2H_EVB_ESC_ETG5003_CR52_Dual_Secondary\src\ethercat\beckhoff

     (2) RZT2H_EtherCAT_EVB_rev0400\CR52_Dual\project\ETG5003\ewarm\RZT2H_EVB_ESC_ETG5003_CR52_Dual_Secondary\src\ethercat\beckhoff

Note) For multi-core projects, the Src folder is copied to the Secondary core project folder, not the Primary.

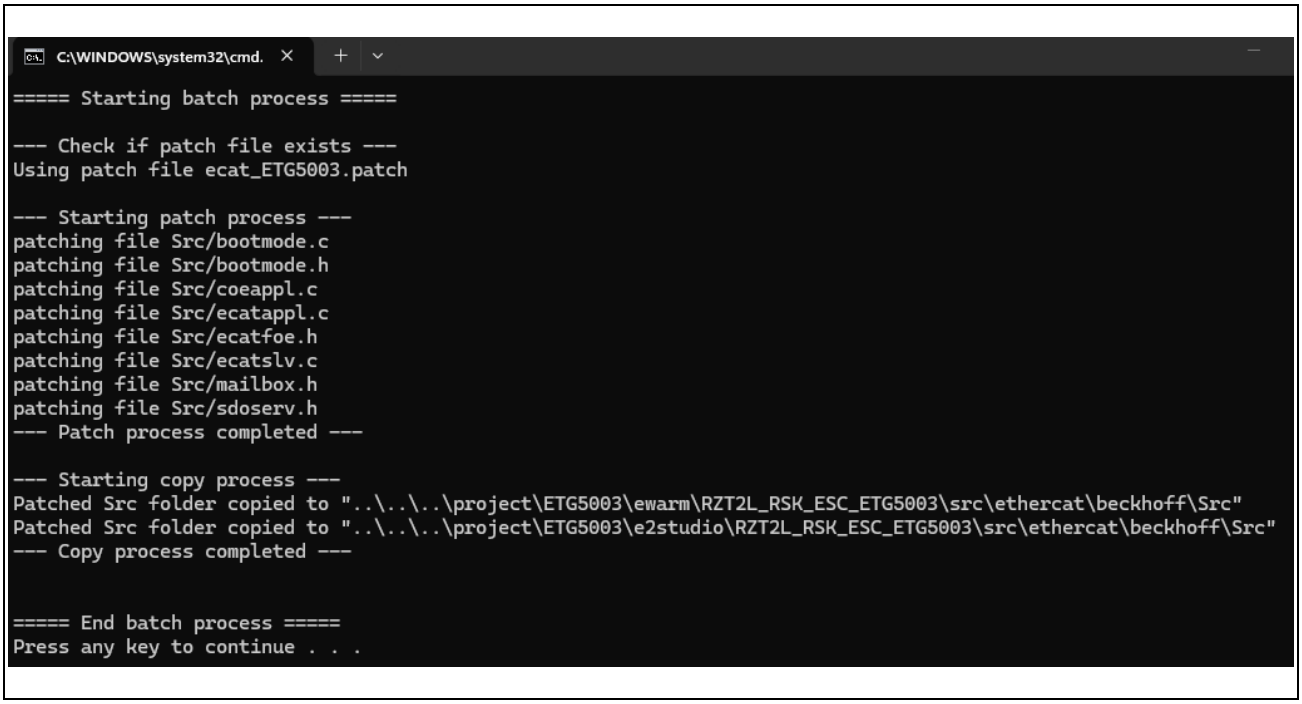


Figure 6.6 Execute the batch file (This is the example of RZ/T2L)

6.2 Debugging with EWARM

6.2.1 Single-Core Project (RZ/T2L, RZ/N2L)

1. Open the sample project (.eww).
   RZT2L_EtherCAT_RSK_rev0400\project\ETG5003\ewarm\RZT2L_RSK_ESC_ETG5003\RZT2L_RSK_ESC_ETG5003.eww

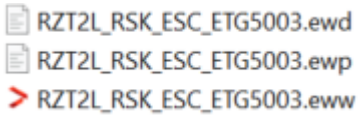


Figure 6.7 Open EWARM workspace file

Note) If you are using a different device or core configuration, replace the folder name and project name, according to Table 6.1.

2. Open [Tools] > [Configure Tools…], register the following contents.

Table 6.2 Register the [Configure Tools…] settings

Setting item	Setting example
Menu Text	RZ Smart Configurator
Command	$RASC_EXE_PATH$
Argument	--compiler IAR configuration.xml
Initial Directory	$PROJ_DIR$

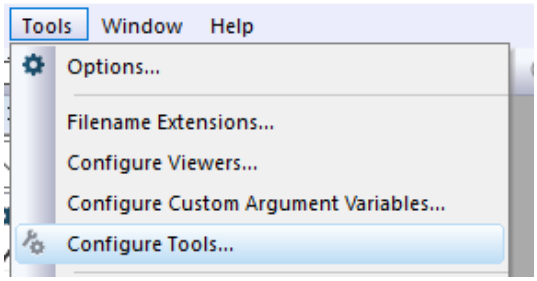


Figure 6.8 Open [Configure Tools…]

3. Open the buildinfo.ipcf file, enter your FSP Smart Configurator path in “RASC_EXE_PATH”.
   (FSP Smart Configurator is required to support RZ FSP v4.0.0 or later.)

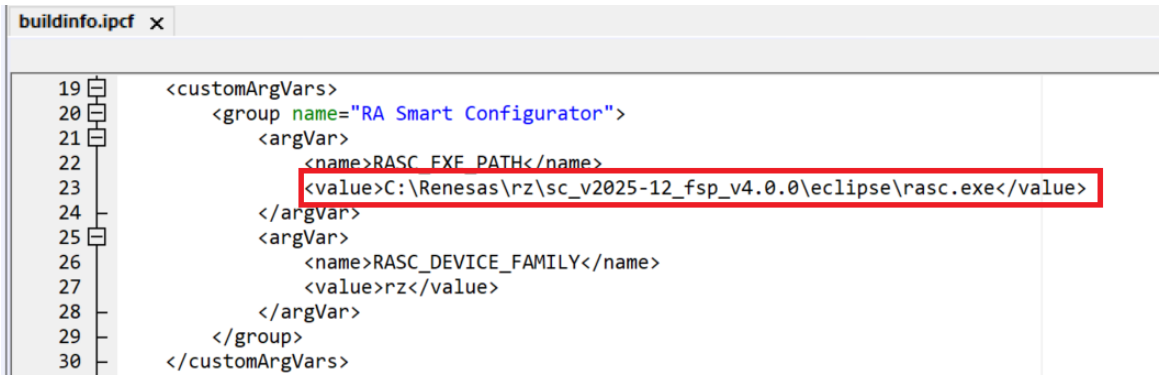


Figure 6.9 RASC_EXE_PATH in buildinfo.ipcf

4. Launch the Smart Configurator by selecting the registered menu entry (e.g., [RZ Smart Configurator]).

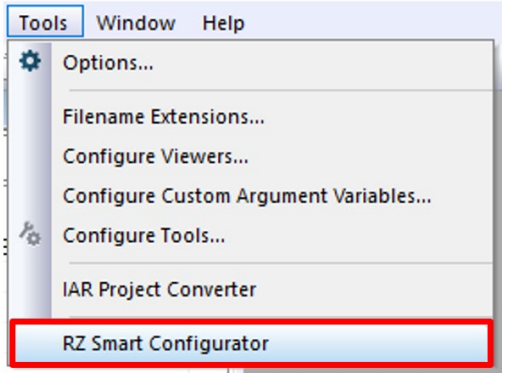


Figure 6.10 Open the Smart Configurator

5. Select the BSP tab and open [Properties].
   Navigate to the following list to change the value of [LDR_ADDR_NML] for BANK1.

Table 6.3 Setting LDR_ADDR_NML for BANK1

Target device	The value of [LDR_ADDR_NML]
RZ/T2L RSK	0x68200060
RZ/N2L RSK	0x60200060

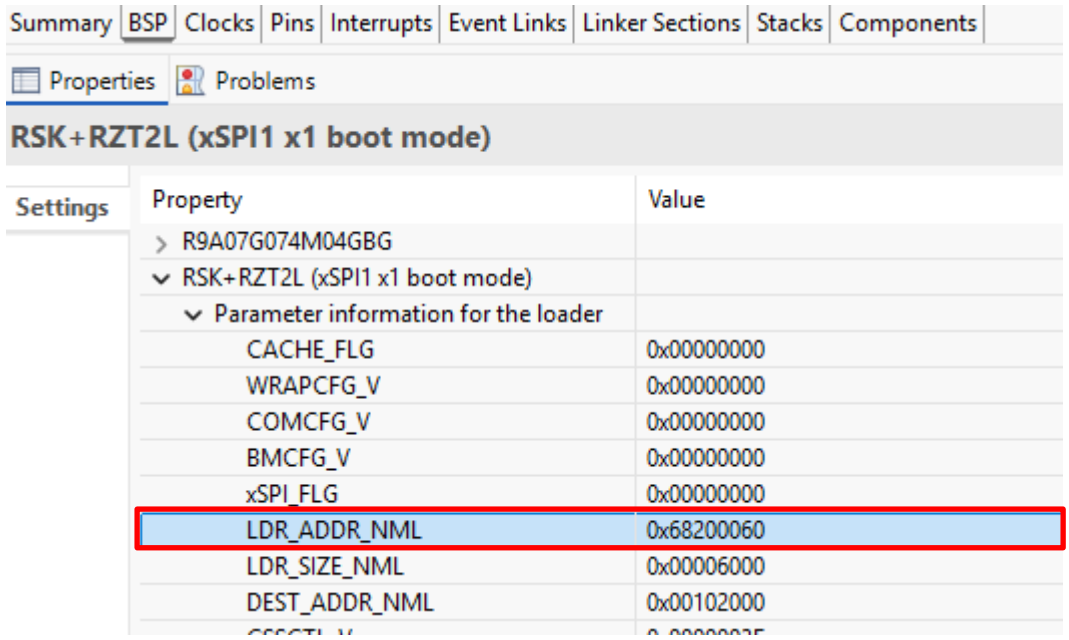


Figure 6.11 Setting for BANK1 (This is the example of RZ/T2L RSK)

6. Click "Generate Project Content" to generate the FSP code.

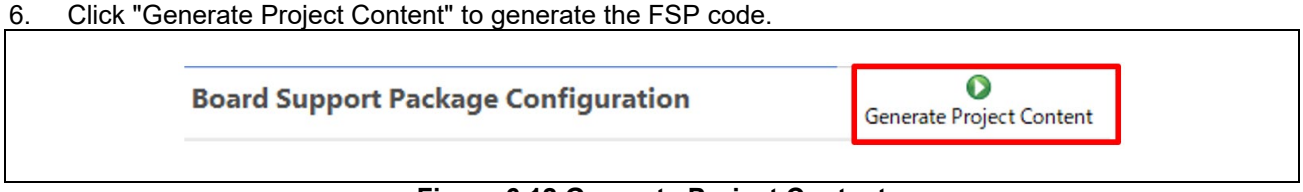


Figure 6.12 Generate Project Content

7. Select "**BANK1**" of the project.

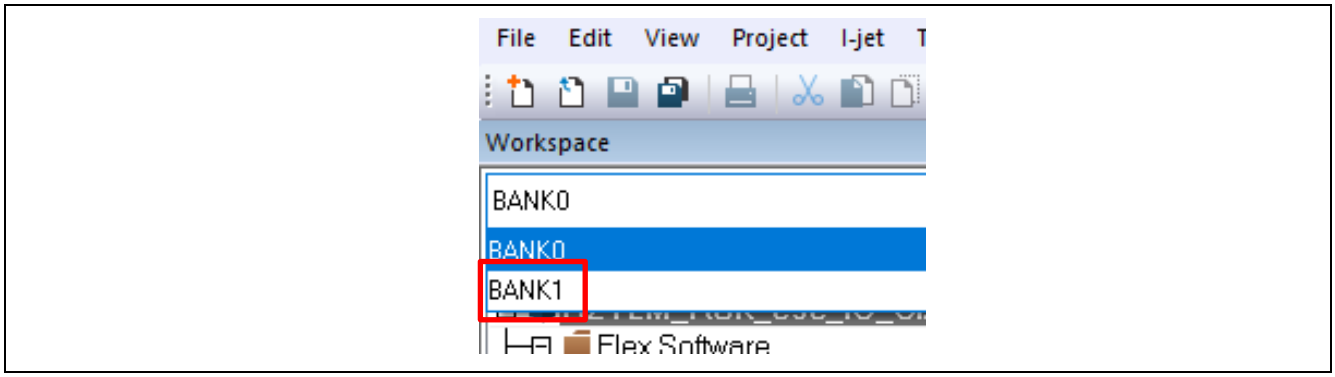


Figure 6.13 Change the configuration to BANK1

8. Select the "Options…" item in the "Project" menu, open the "Config" tab in the "Linker" category. Change "Linker configuration file" to the following linker script path shown in the table below.

Table 6.4 Setting linker script path for BANK1

Target device	Linker script path
RZ/T2L RSK	$PROJ_DIR$/script/fsp_xspi1_boot_bank1.icf
RZ/N2L RSK	$PROJ_DIR$/script/fsp_xspi0_boot_bank1.icf

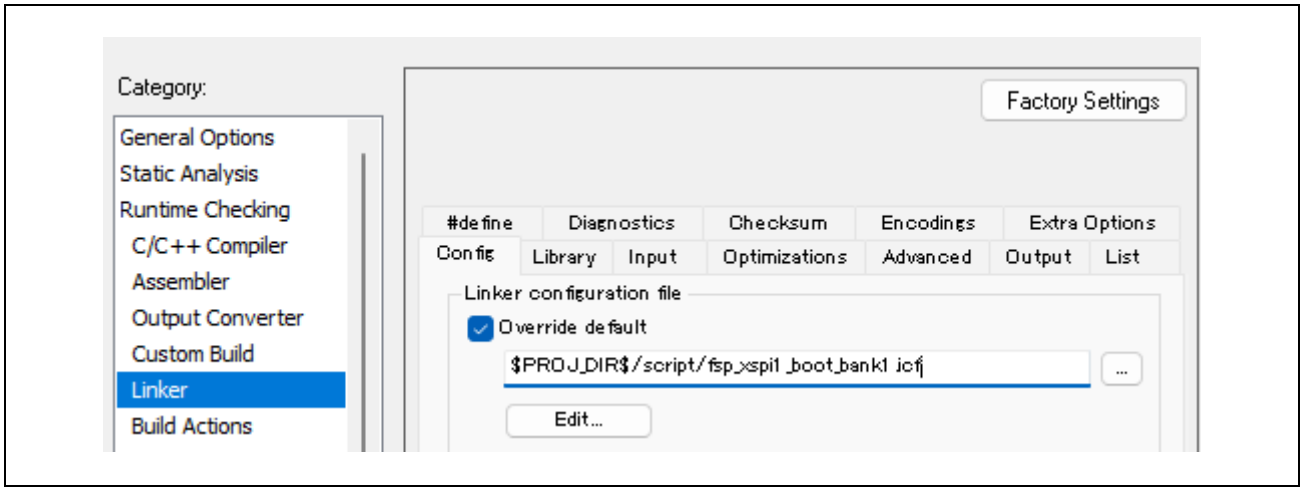


Figure 6.14 Setting the linker script for BANK1 (this is the example of RZ/T2L-RSK)

9. Click the "Make" button to build the project for BANK1.

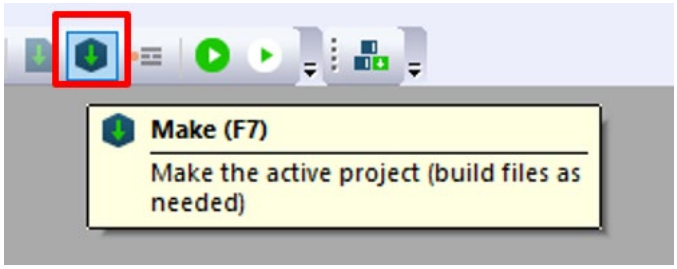


Figure 6.15 Make the project for BANK1

10. Launch the Smart Configurator again.

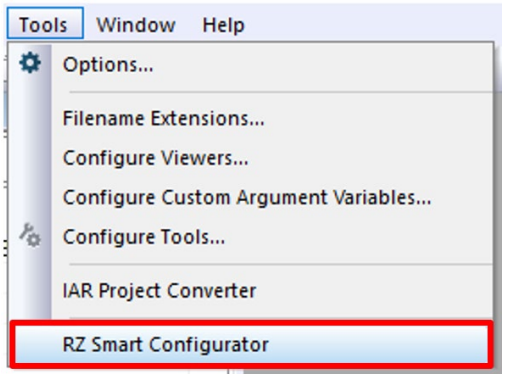


Figure 6.16 Open the Smart Configurator

11. Select the BSP tab and open [Properties].
Navigate to the following list to change the value of [LDR_ADDR_NML] for BANK0.

Table 6.5 Setting LDR_ADDR_NML for BANK0

Target device	The value of [LDR_ADDR_NML]
RZ/T2L RSK	0x68100060
RZ/N2L RSK	0x60100060

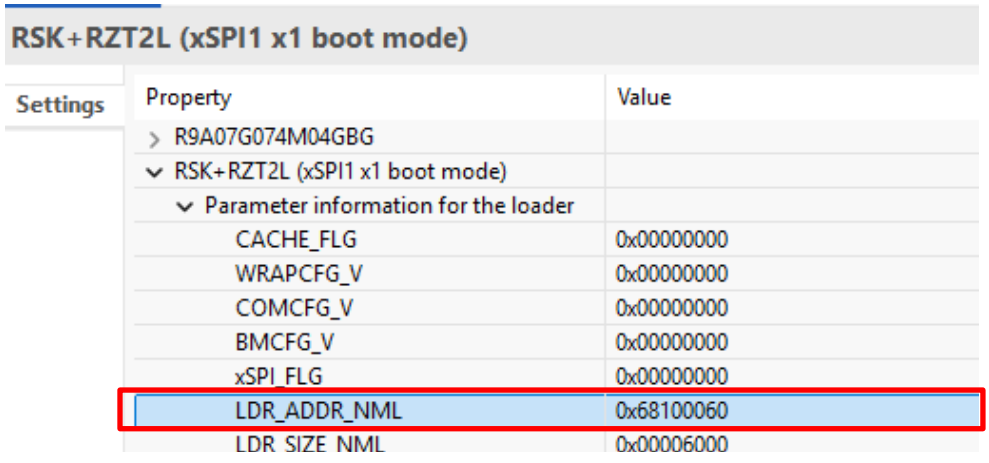


Figure 6.17 Setting for BANK0 (This is the example of RZ/T2L RSK)

12. Click "Generate Project Content" to generate the FSP code again.

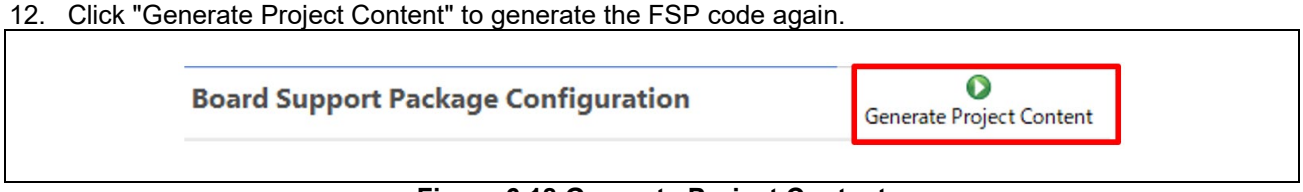


Figure 6.18 Generate Project Content

13. Select “**BANK0**” of the project.

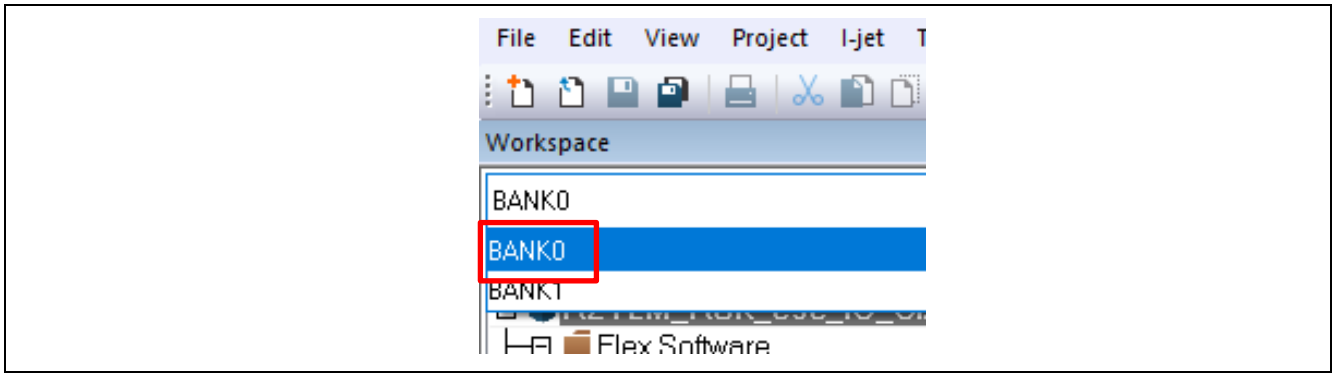


Figure 6.19 Change the configuration to BANK0

14. Select the “Options…” item in the “Project” menu, open the “Config” tab in the “Linker” category. Change “Linker configuration file” to the following linker script path shown in the table below.

Table 6.6 Setting linker script path for BANK0

Target device	Linker script path
RZ/T2L RSK	$PROJ_DIR$/script/fsp_xspi1_boot_bank0.icf
RZ/N2L RSK	$PROJ_DIR$/script/fsp_xspi0_boot_bank0.icf

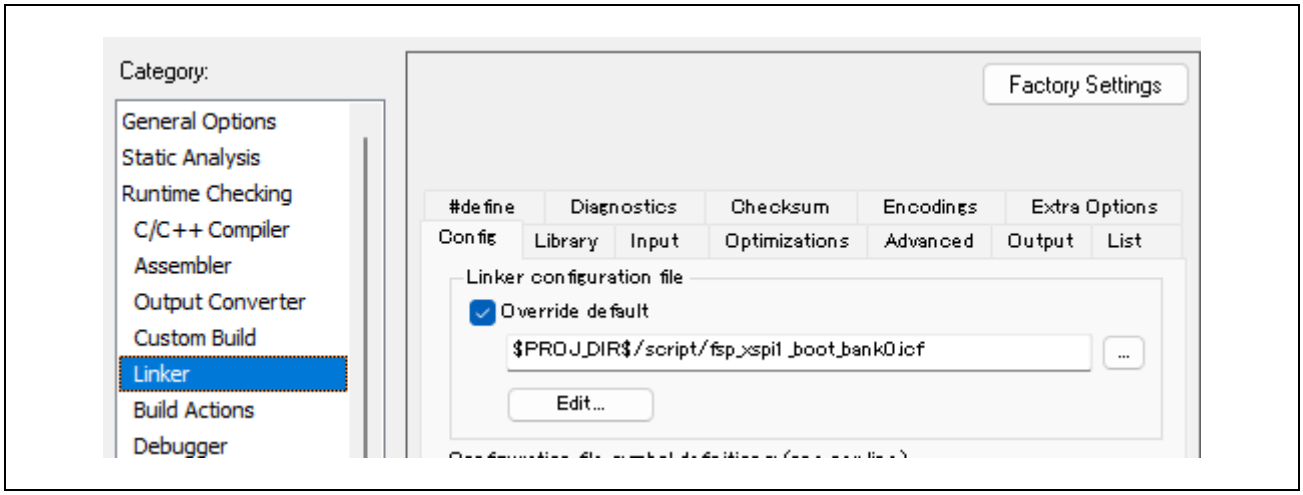


Figure 6.20 Setting the linker script for BANK0 (this is the example of RZ/T2L-RSK)

15. Click the "Make" button to build the project for BANK0.

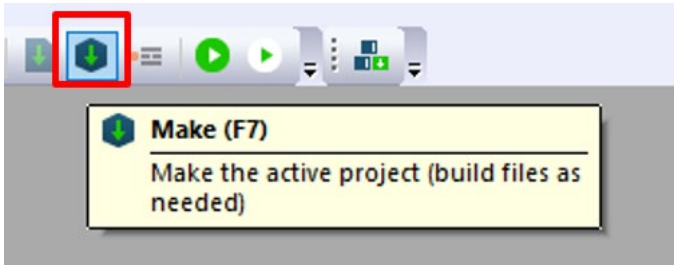


Figure 6.21 Make the project for BANK0

16. Click the "Download and Debug" button to download the binary to the flash memory on the board.

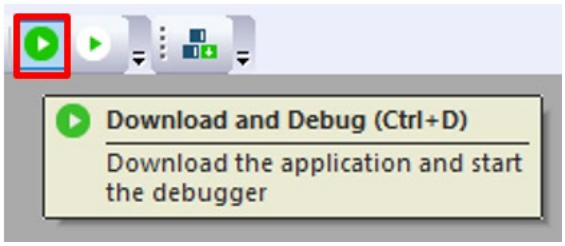


Figure 6.22 Download and Debug

17. Click the "Go" button to start program execution.

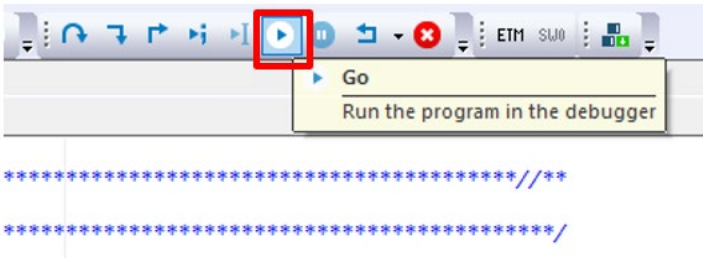


Figure 6.23 Run the program

6.2.2 Multi-Core Project (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/N2H)

6.2.2.1 Primary Core First Building

1. Open the Primary core project (.eww):

RZT2H_EtherCAT_EVB_rev0400\CR52_Dual\project\ETG5003\ewarm\RZT2H_EVB_ESC_ETG5003_CR52_Dual_Primary\RZT2H_EVB_ESC_ETG5003_CR52_Dual_Primary.eww

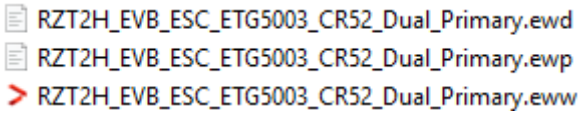


Figure 6.24 Open Primary EWARM workspace file

Note) If you are using a different device or core configuration, replace the folder name and project name, according to **Table 6.1**.

2. Open [Tools] > [Configure Tools…], register the following contents.

Table 6.7 Register the [Configure Tools…] settings

Setting item	Setting example
Menu Text	RZ Smart Configurator
Command	$RASC_EXE_PATH$
Argument	--compiler IAR configuration.xml
Initial Directory	$PROJ_DIR$

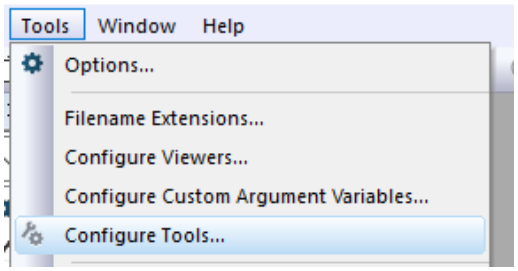


Figure 6.25 Open [Configure Tools…]

3. Open the buildinfo.ipcf file, enter your FSP Smart Configurator path in “RASC_EXE_PATH”. (FSP Smart Configurator is required to support RZ FSP v4.0.0 or later.)

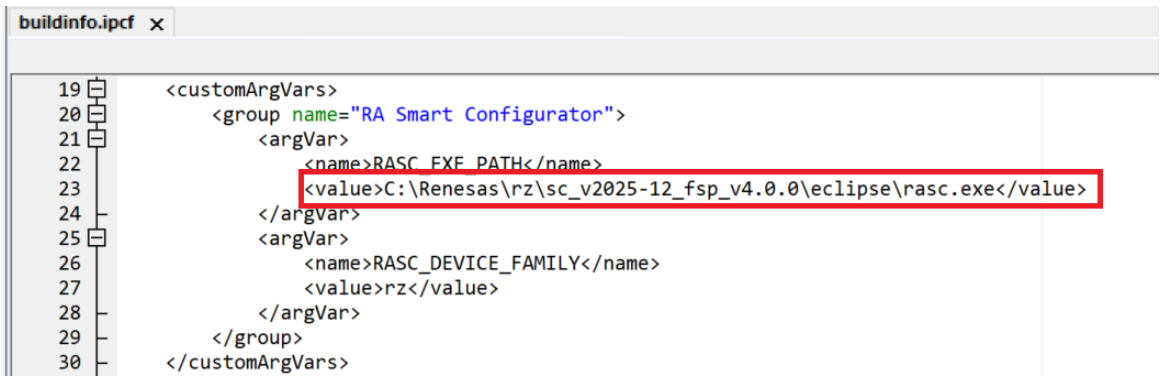


Figure 6.26 RASC_EXE_PATH in buildinfo.ipcf

4. Launch the Smart Configurator by selecting the registered menu entry (e.g., [RZ Smart Configurator]).

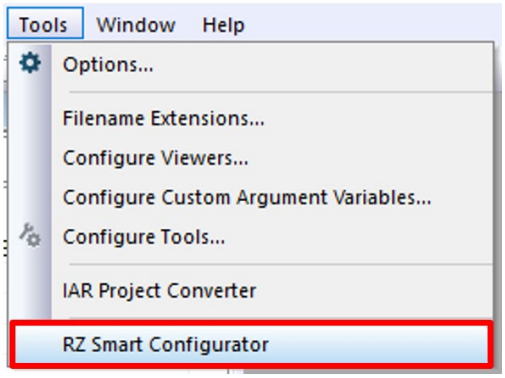


Figure 6.27 Open the Smart Configurator

Note) When using RZ/T2ME-RSK,
refer to the section Appendix C : How to Convert FSP Configuration from RZ/T2M to RZ/T2ME to change the primary core FSP configuration.

5. Select the BSP tab and open [Properties].
Navigate to the following list to change the value of [LDR_ADDR_NML] for BANK0.

Table 6.8 Setting LDR_ADDR_NML for BANK0

Target device	The value of [LDR_ADDR_NML]
RZ/T2M RSK, RZ/T2ME RSK	0x60100060
RZ/T2H EVB, RZ/N2H EVB	0x50100060

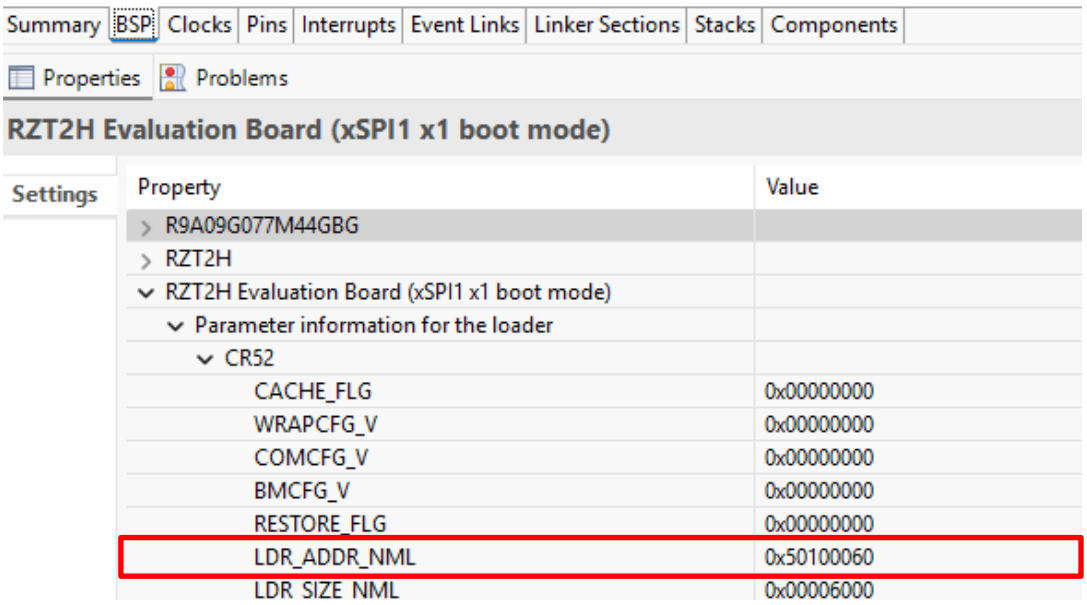


Figure 6.28 Setting for BANK0 (This is the example of RZ/T2H EVB)

6. Click "Generate Project Content" to generate the FSP code.

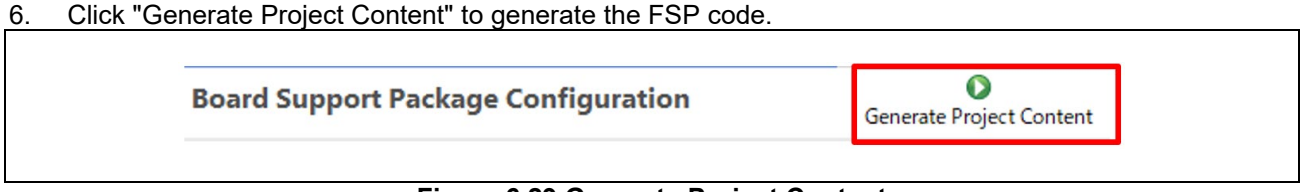


Figure 6.29 Generate Project Content

7. Select “**BANK0**” of the project.

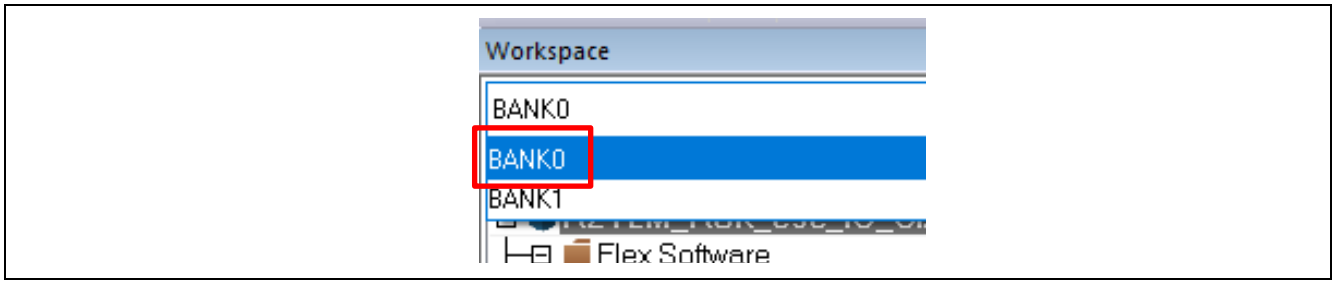


Figure 6.30 Change the configuration to BANK0

8. Select the “Options…” item in the “Project” menu, open the “Config” tab in the “Linker” category. Change “Linker configuration file” to the following linker script path shown in the table below.

Table 6.9 Setting linker script path for BANK0

Target device	Linker script path
RZ/T2M RSK, RZ/T2ME RSK	$PROJ_DIR$/script/fsp_xspi0_boot_bank0.icf
RZ/T2H EVB, RZ/N2H EVB	$PROJ_DIR$/script/fsp_xspi1_boot_bank0.icf

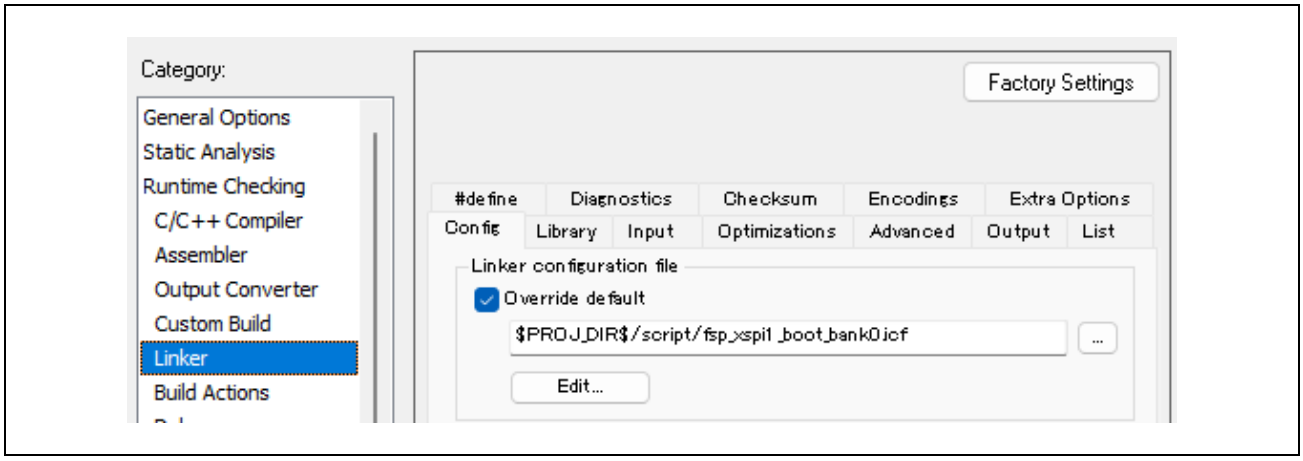


Figure 6.31 Setting the linker script for BANK0 (this is the example of RZ/T2H EVB)

9. Click the "Make" button to build the Primary core project for BANK0.

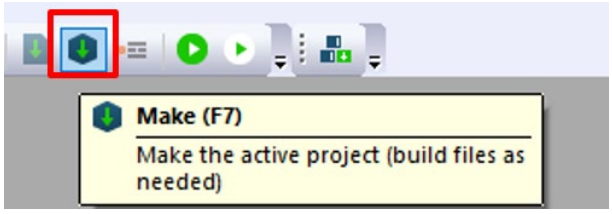


Figure 6.32 Make the Primary project for BANK0

10. Launch the Smart Configurator by selecting the registered menu entry again.

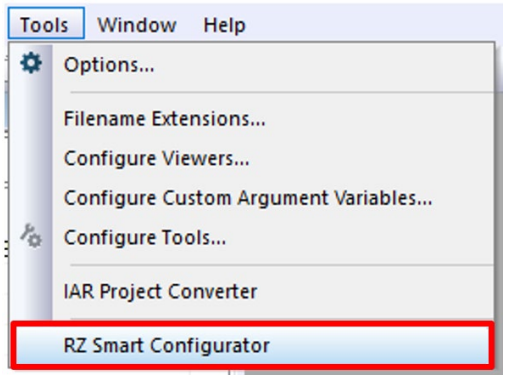


Figure 6.33 Open the Smart Configurator

11. Select the BSP tab and open [Properties].
    Navigate to the following list to change the value of [LDR_ADDR_NML] for BANK1.

Table 6.10 Setting LDR_ADDR_NML for BANK1

Target device	The value of [LDR_ADDR_NML]
RZ/T2M RSK, RZ/T2ME RSK	0x60200060
RZ/T2H EVB, RZ/N2H EVB	0x50200060

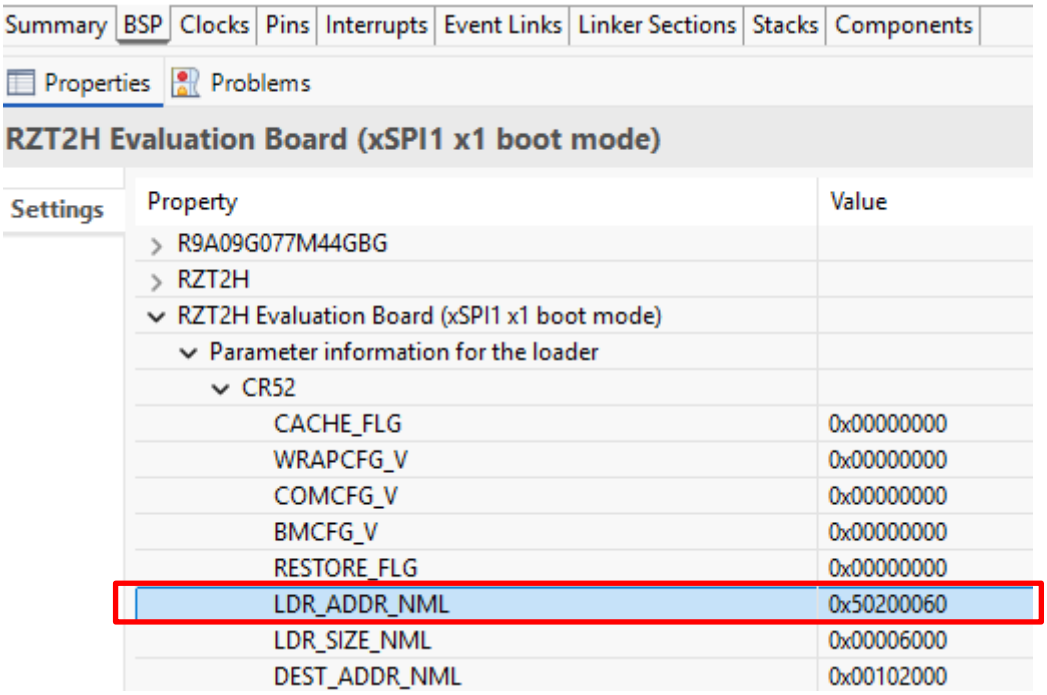


Figure 6.34 Setting for BANK1 (This is the example of RZ/T2H EVB)

12. Click "Generate Project Content" to generate the FSP code.

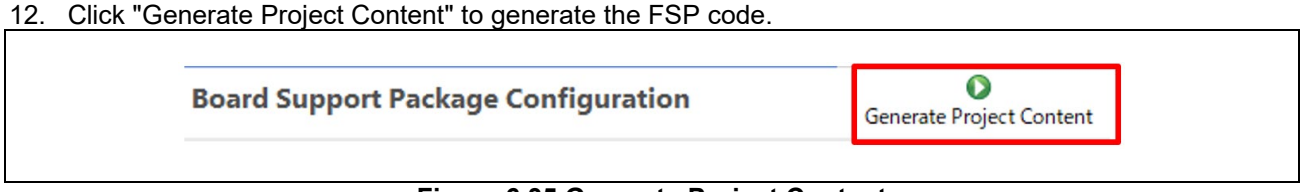


Figure 6.35 Generate Project Content

13. Select "**BANK1**" of the project.

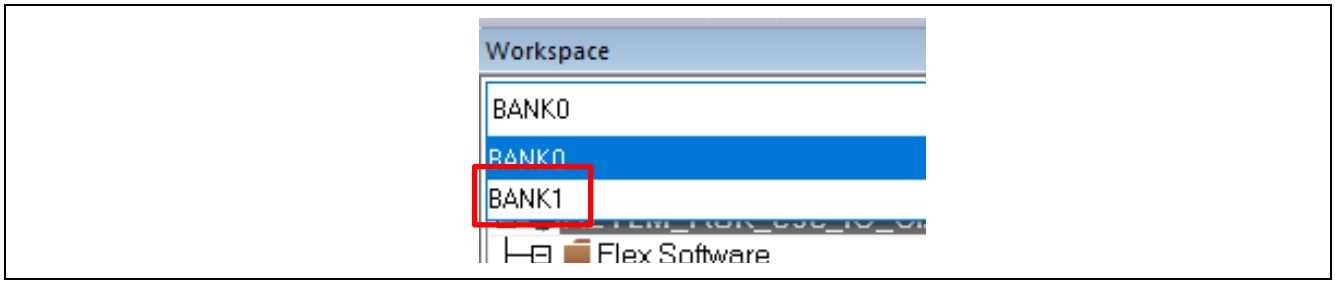


Figure 6.36 Change the configuration to BANK1

14. Select the "Options…" item in the "Project" menu, open the "Config" tab in the "Linker" category. Change "Linker configuration file" to the following linker script path shown in the table below.

Table 6.11 Setting linker script path for BANK1

Target device	Linker script path
RZ/T2M RSK, RZ/T2ME RSK	$PROJ_DIR$/script/fsp_xspi0_boot_bank1.icf
RZ/T2H EVB, RZ/N2H EVB	$PROJ_DIR$/script/fsp_xspi1_boot_bank1.icf

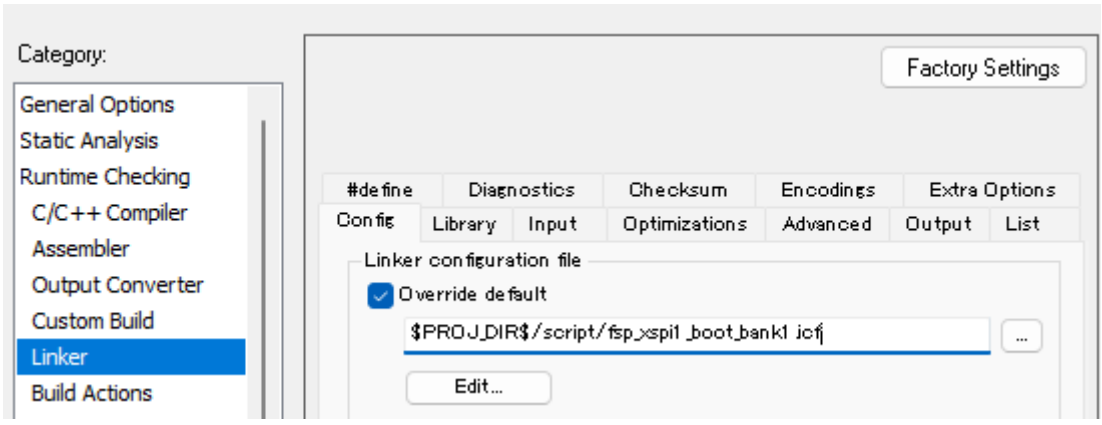


Figure 6.37 Setting the linker script for BANK1 (this is the example of RZ/T2H EVB)

15. Click the "Make" button to build the Primary core project for BANK1.

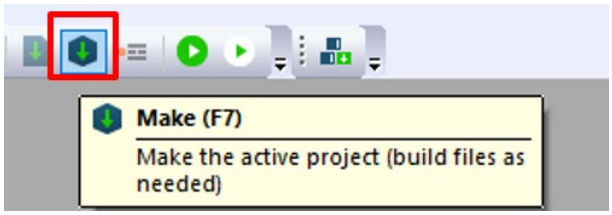


Figure 6.38 Make the Primary project for BANK1

6.2.2.2 Secondary Core Building

1. Open the Secondary project (.eww):

RZT2H_EtherCAT_EVB_rev0400\CR52_Dual\project\ETG5003\ewarm\RZT2H_EVB_ESC_ETG5003_CR52_Dual_Secondary\RZT2H_EVB_ESC_ETG5003_CR52_Dual_Secondary.eww

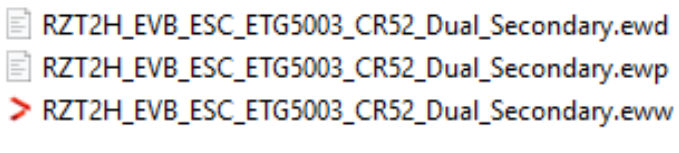


Figure 6.39 Open Secondary EWARM workspace file

2. Open the buildinfo.ipcf file, enter your FSP Smart Configurator path in “RASC_EXE_PATH”. (FSP Smart Configurator is required to support RZ FSP v4.0.0 or later.)

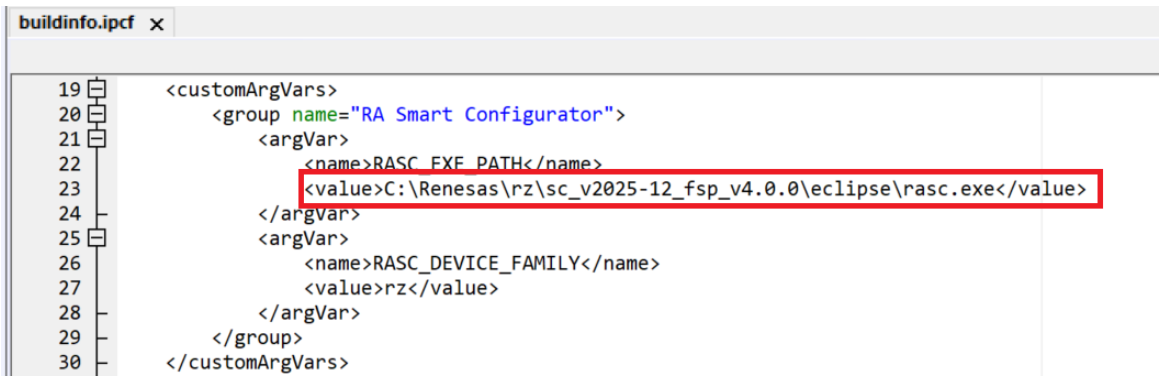


Figure 6.40 RASC_EXE_PATH in buildinfo.ipcf

3. Launch the Smart Configurator by selecting the registered menu entry (e.g., [RZ Smart Configurator]).

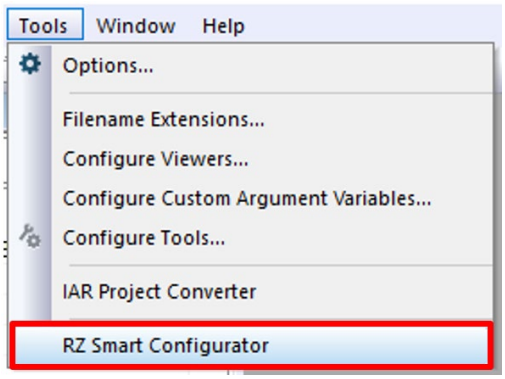


Figure 6.41 Open the Smart Configurator

Note) When using RZ/T2ME-RSK,
if you followed Appendix C : How to Convert FSP Configuration from RZ/T2M to RZ/T2ME
the secondary core FSP configuration have been automatically changed.

4. Click "Generate Project Content" to generate the FSP code.

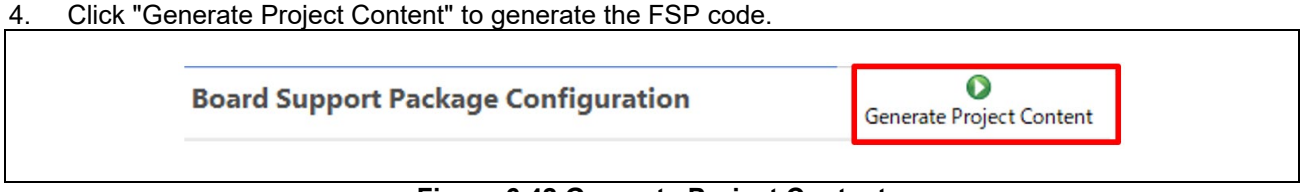


Figure 6.42 Generate Project Content

5. Select “**BANK1**” of the Secondary project.

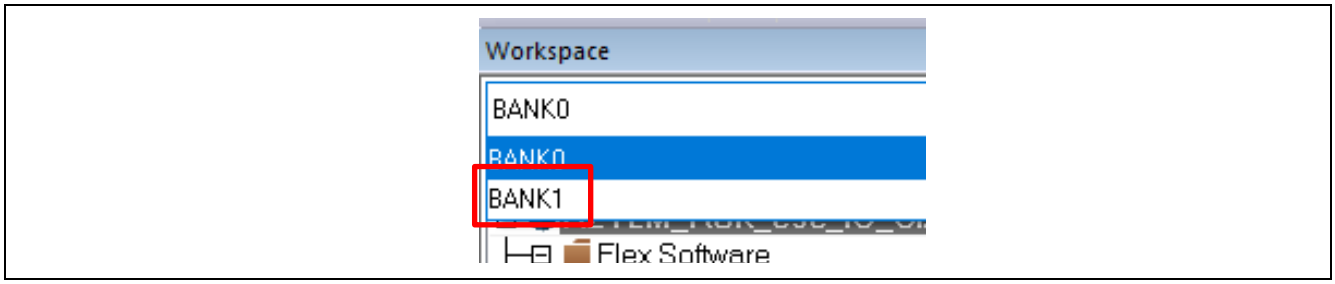


Figure 6.43 Change the configuration to BANK1

6. Select the “Options…” item in the “Project” menu, open the “Config” tab in the “Linker” category. Change “Linker configuration file” to the following linker script path shown in the table below.

Table 6.12 Setting linker script path for BANK1

Target device	Linker script path
RZ/T2M RSK, RZ/T2ME RSK	$PROJ_DIR$\script\fsp_xspi0_boot_bankx.icf
RZ/T2H EVB, RZ/N2H EVB	$PROJ_DIR$\script\fsp_xspi1_boot_bankx.icf

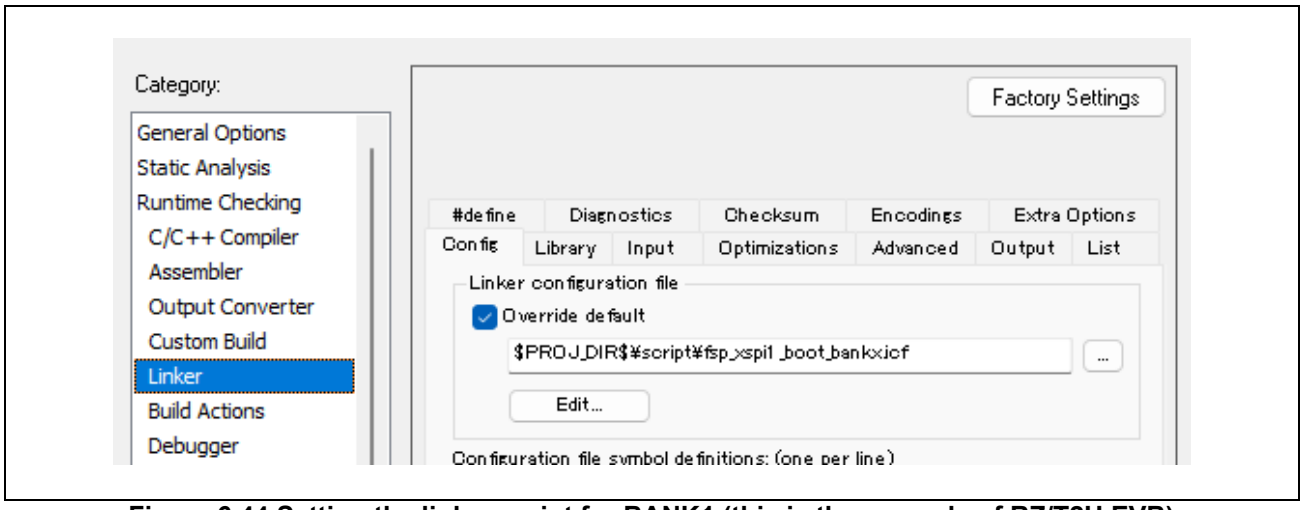


Figure 6.44 Setting the linker script for BANK1 (this is the example of RZ/T2H EVB)

16. Click the "Make" button to build the Secondary core project for BANK1.

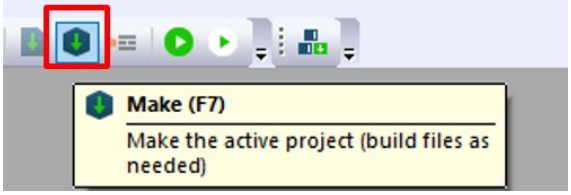


Figure 6.45 Make the Secondary project for BANK1

7. Select “**BANK0**” of the Secondary project.

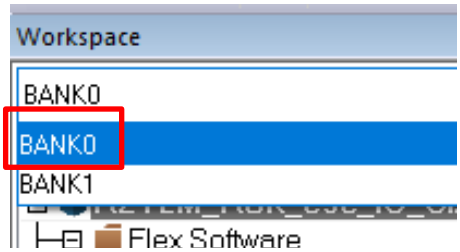


Figure 6.46 Change the configuration to BANK0

8. Select the “Options…” item in the “Project” menu, open the “Config” tab in the “Linker” category. Change “Linker configuration file” to the following linker script path shown in the table below.

Table 6.13 Setting linker script path for BANK0

Target device	Linker script path
RZ/T2M RSK, RZ/T2ME RSK	$PROJ_DIR$\script\fsp_xspi0_boot_bankx.icf
RZ/T2H EVB, RZ/N2H EVB	$PROJ_DIR$\script\fsp_xspi1_boot_bankx.icf

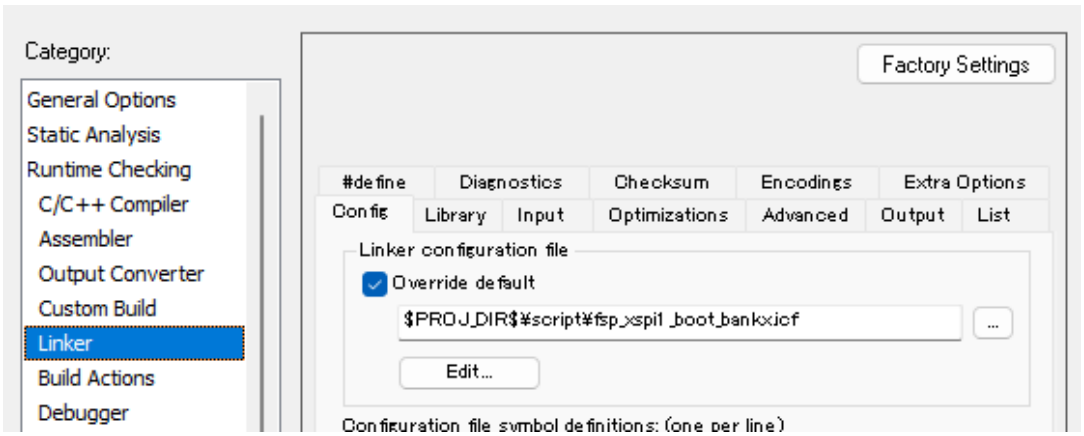


Figure 6.47 Setting the linker script for BANK0 (this is the example of RZ/T2H EVB)

17. Click the "Make" button to build the Secondary core project for BANK0.

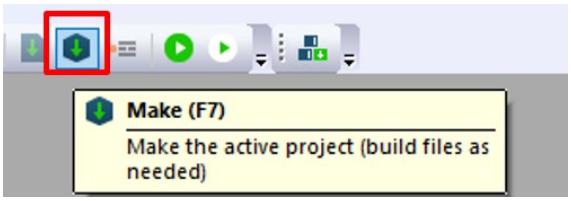


Figure 6.48 Make the Secondary project for BANK0

6.2.2.3 Primary Core Second Building

1. Close the Secondary core project and reopen the Primary core project.

2. Launch the Smart Configurator by selecting the registered menu entry again.

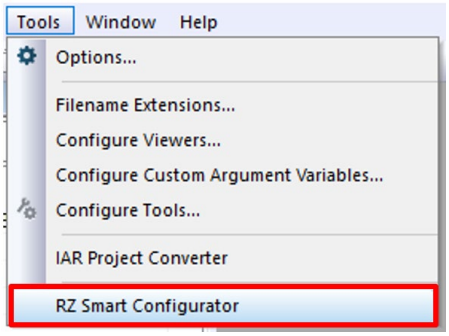


Figure 6.49 Open the Smart Configurator again

3. Select the BSP tab and open [Properties].
   Navigate to the following list to confirm the value of [LDR_ADDR_NML] for BANK1.

Table 6.14 Setting LDR_ADDR_NML for BANK1

Target device	The value of [LDR_ADDR_NML]
RZ/T2M RSK, RZ/T2ME RSK	0x60200060
RZ/T2H EVB, RZ/N2H EVB	0x50200060

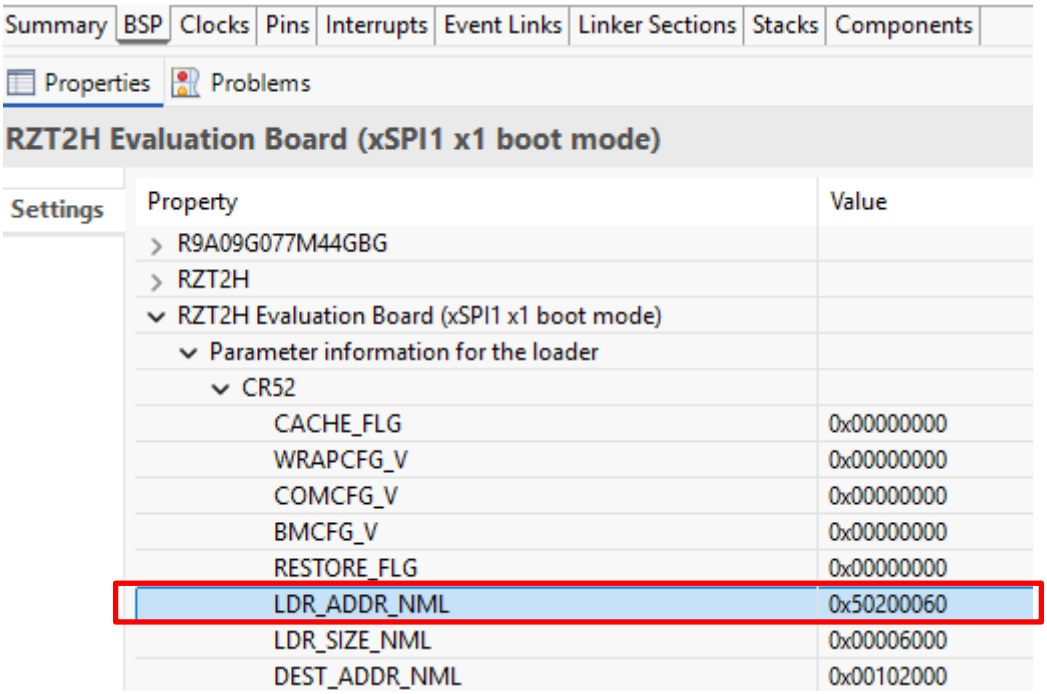


Figure 6.50 Setting for BANK1 (This is the example of RZ/T2H EVB)

4. Click "Generate Project Content" to generate the FSP code.

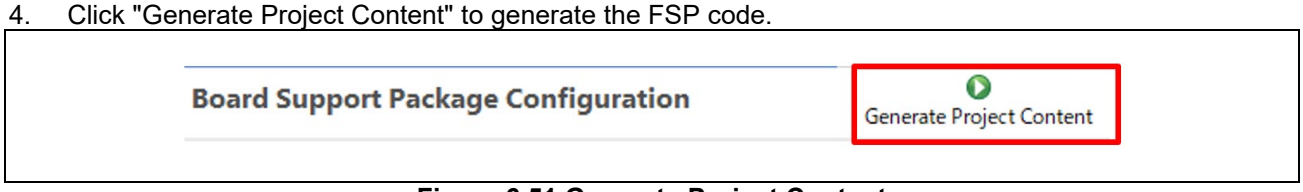


Figure 6.51 Generate Project Content

5. Select "**BANK1**" of the project.

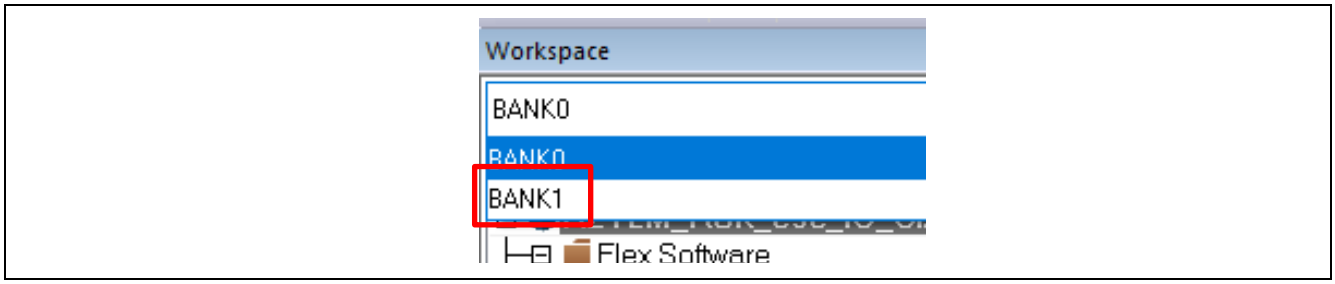


Figure 6.52 Change the configuration to BANK1

6. Select the "Options…" item in the "Project" menu, open the "Config" tab in the "Linker" category. Change "Linker configuration file" to the following linker script path shown in the table below.

Table 6.15 Setting linker script path for BANK1

Target device	Linker script path
RZ/T2M RSK, RZ/T2ME RSK	$PROJ_DIR$/script/fsp_xspi0_boot_bank1.icf
RZ/T2H EVB, RZ/N2H EVB	$PROJ_DIR$/script/fsp_xspi1_boot_bank1.icf

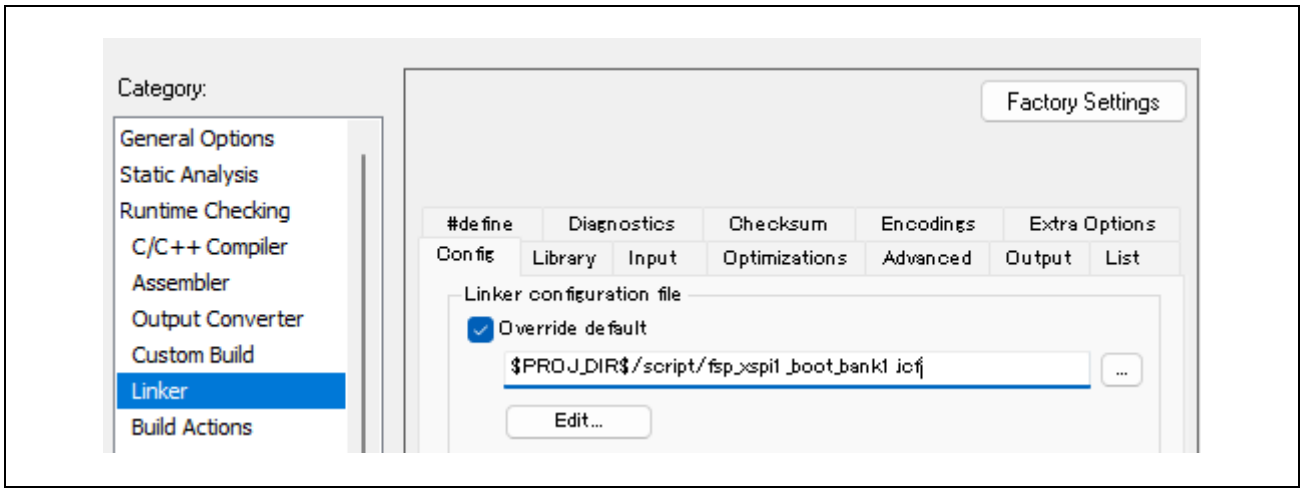


Figure 6.53 Setting the linker script for BANK1 (this is the example of RZ/T2H EVB)

7. Click the "Make" button to build the Primary core project for BANK1 again.

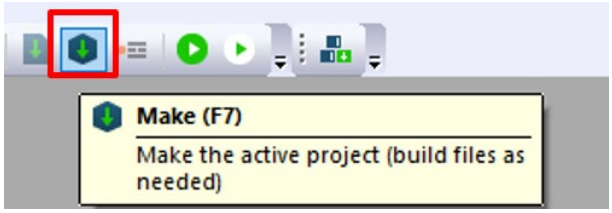


Figure 6.54 Make the Primary project for BANK1 again

8. Launch the Smart Configurator by selecting the registered menu entry again.

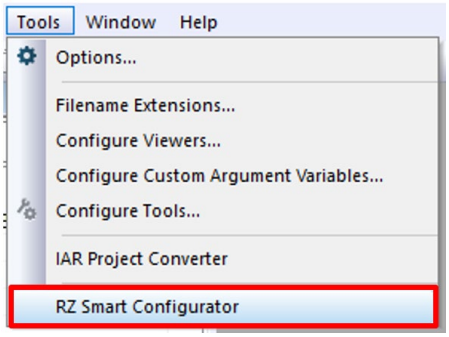


Figure 6.55 Open the Smart Configurator again

9. Select the BSP tab and open [Properties].
Navigate to the following list to confirm the value of [LDR_ADDR_NML] for BANK0.

Table 6.16 Setting LDR_ADDR_NML for BANK0

Target device	The value of [LDR_ADDR_NML]
RZ/T2M RSK, RZ/T2ME RSK	0x60100060
RZ/T2H EVB, RZ/N2H EVB	0x50100060

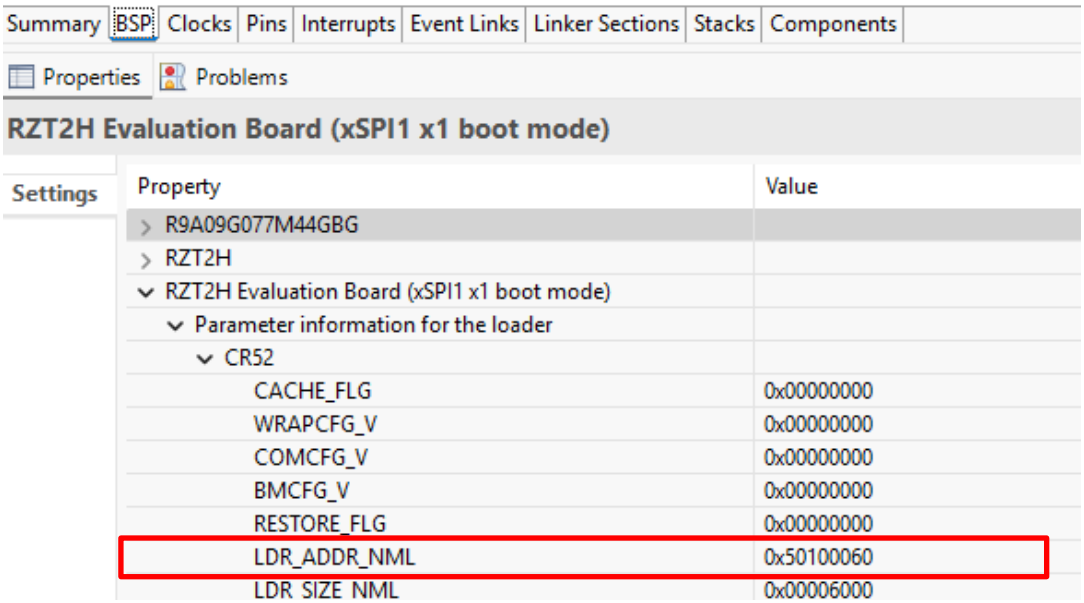


Figure 6.56 Setting for BANK0 (This is the example of RZ/T2H EVB)

10. Click "Generate Project Content" to generate the FSP code.

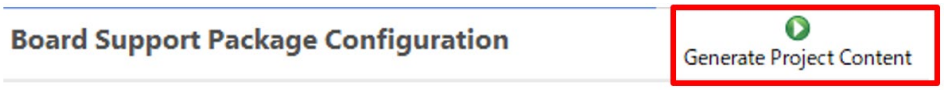


Figure 6.57 Generate Project Content

11. Select “**BANK0**” of the project.

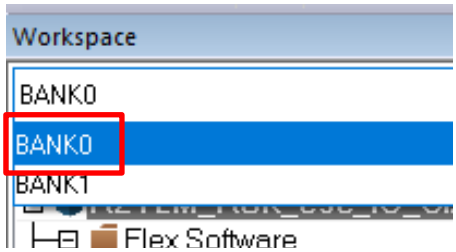


Figure 6.58 Change the configuration to BANK0

12. Select the “Options…” item in the “Project” menu, open the “Config” tab in the “Linker” category. Change “Linker configuration file” to the following linker script path shown in the table below.

Table 6.17 Setting linker script path for BANK0

Target device	Linker script path
RZ/T2M RSK, RZ/T2ME RSK	$PROJ_DIR$/script/fsp_xspi0_boot_bank0.icf
RZ/T2H EVB, RZ/N2H EVB	$PROJ_DIR$/script/fsp_xspi1_boot_bank0.icf

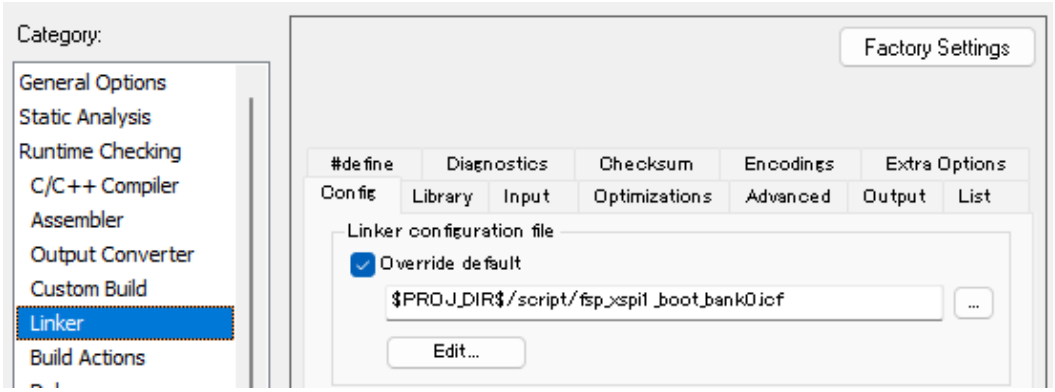


Figure 6.59 Setting the linker script for BANK0 (this is the example of RZ/T2H EVB)

13. Click the "Make" button to build the Primary core project for BANK0 again.

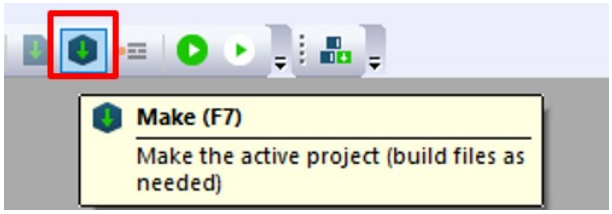


Figure 6.60 Make the Primary project for BANK0 again

6.2.2.4 Debug and Download

1. Push the “Debug and Download” button on the primary project to start dual core debugging.

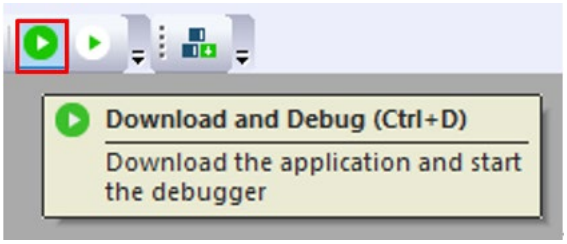


Figure 6.61 Download and Debug

The secondary project is automatically opened.

Press the "Resume" button on the primary and secondary project to start debugging the source code.

6.3 Debugging with e² studio

6.3.1 Single-Core Project (RZ/T2L, RZ/N2L)

1. Launch e² studio and import the sample project by selecting [File] → [Import] → [Existing Projects into Workspace].

   Check the "select root directory" and push “Browse” button to select the following folder:

   RZT2L_EtherCAT_RSK_rev0400\project\ETG5003

   Check the following project, push “Finish” button to import project.

   RZT2L_RSK_ESC_ETG5003

Note) If you are using a different device or core configuration, replace the folder name and project name, according to **Table 6.1**.

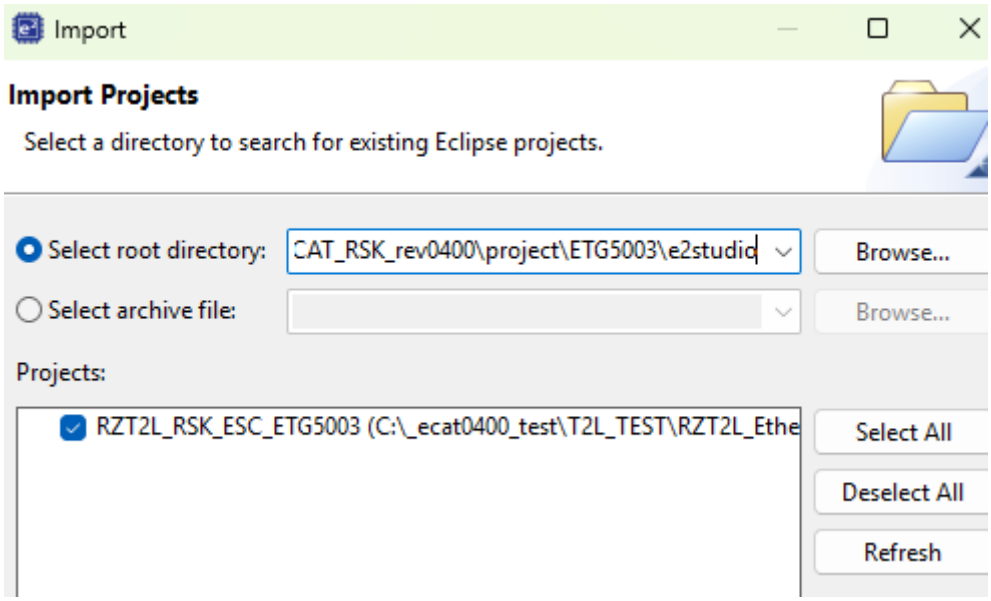


Figure 6.62 Import the project

2. Open configuration.xml in the project to launch the FSP Smart Configurator.

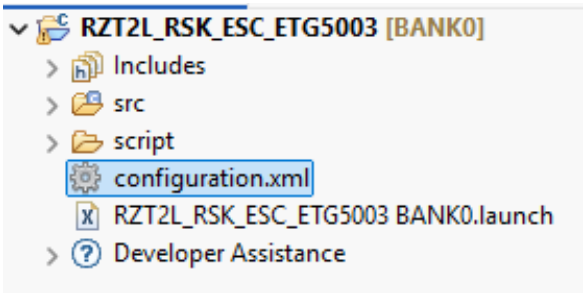


Figure 6.63 Open configuration (1)

3. Select the BSP tab and open [Properties].

   Navigate to the following list to change the value of [LDR_ADDR_NML] for BANK1.

Table 6.18 Setting LDR_ADDR_NML for BANK1

Target device	The value of [LDR_ADDR_NML]
RZ/T2L RSK	0x68200060
RZ/N2L RSK	0x60200060

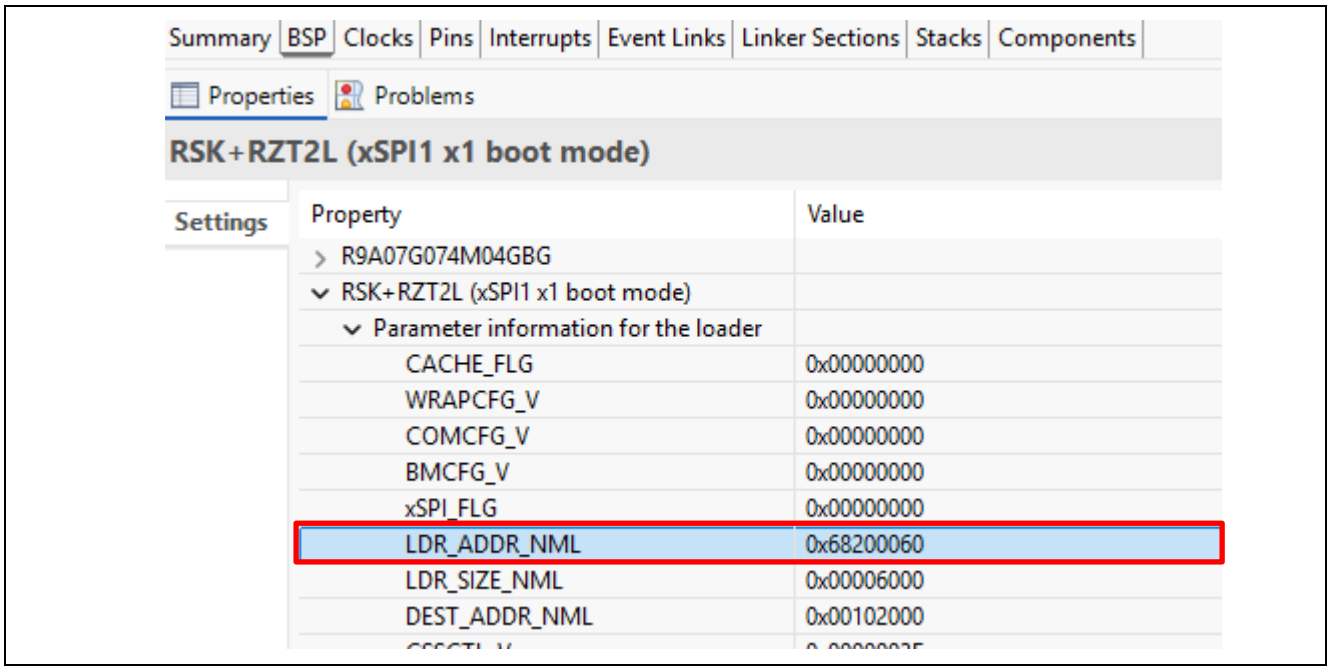


Figure 6.64 Setting for BANK1 (This is the example of RZ/T2L RSK)

4. Click **"Generate Project Content"** to generate the FSP code.

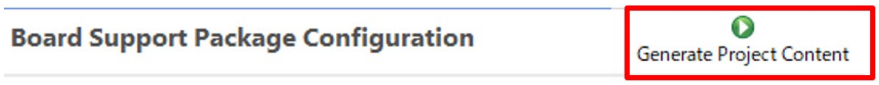


Figure 6.65 Generate Project Content

5. Select the drop-down menu next to the debug icon and choose [BANK1].

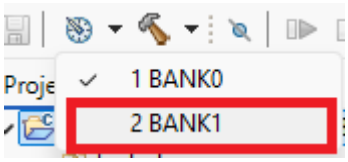


Figure 6.66 Change the configuration to BANK1

6. Click the build button (hammer icon) to build the project for BANK1.

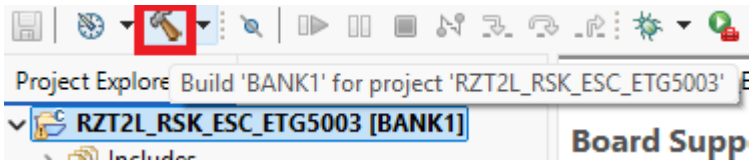


Figure 6.67 Build the project for BANK1

7. Open configuration.xml in the project to launch the FSP Smart Configurator again.

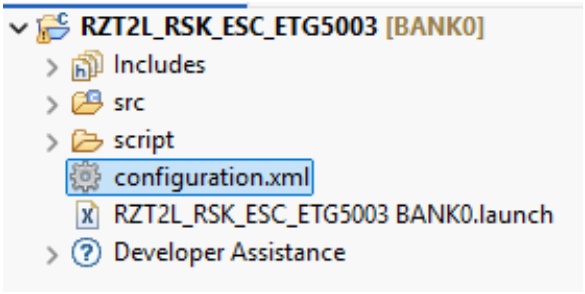


Figure 6.68 Open configuration (2)

8. Select the BSP tab and open [Properties].

   Navigate to the following list to change the value of [LDR_ADDR_NML] for BANK0.

Table 6.19 Setting LDR_ADDR_NML for BANK0

Target device	The value of [LDR_ADDR_NML]
RZ/T2L RSK	0x68100060
RZ/N2L RSK	0x60100060

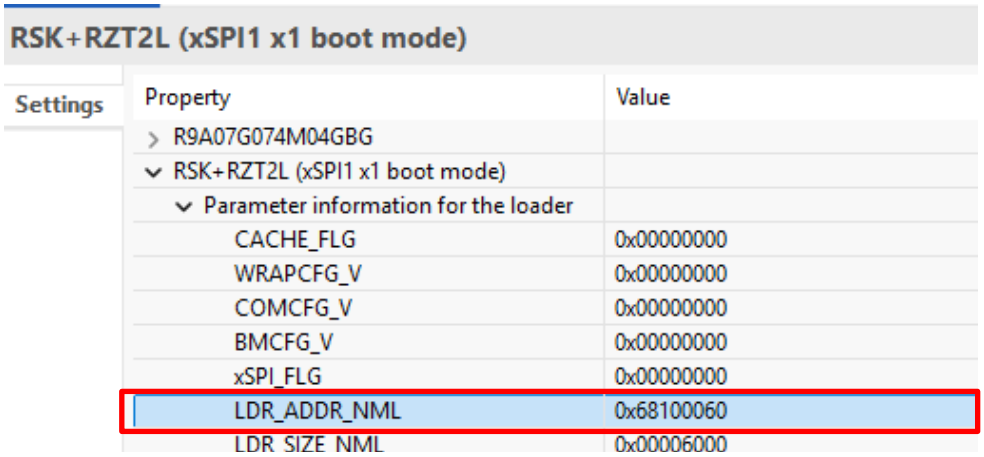


Figure 6.69 Setting for BANK0 (This is the example of RZ/T2L RSK)

9. Click **"Generate Project Content"** to generate the FSP code.

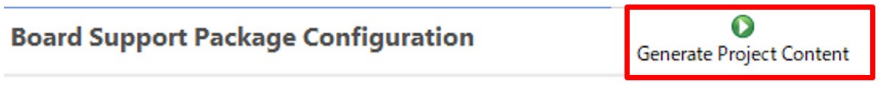


Figure 6.70 Generate Project Content

10. Select the drop-down menu next to the debug icon and choose [BANK0].

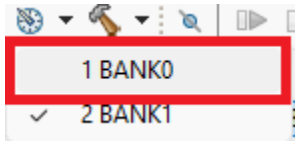


Figure 6.71 Change the configuration to BANK0

11. Click the build button (hammer icon) to build the project for BANK0.

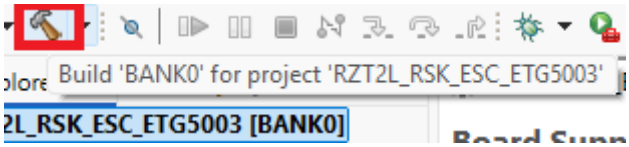


Figure 6.72 Build the project for BANK0

12. Select the drop-down menu next to the debug icon and choose [Debug Configurations…].

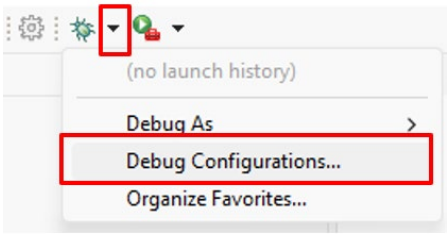


Figure 6.73 Select the drop-down menu

13. Under [Renesas DBG Hardware Debugging],
select RZT2L_RSK_ESC_ETG5003 BANK0 and click [Debug].

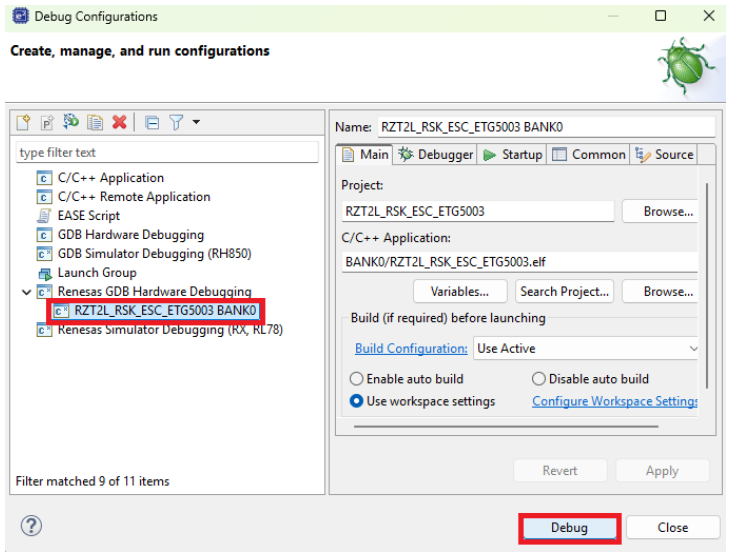


Figure 6.74 Debug Configuration window

Note) A perspective switch dialog will appear. Switch to the debug perspective.

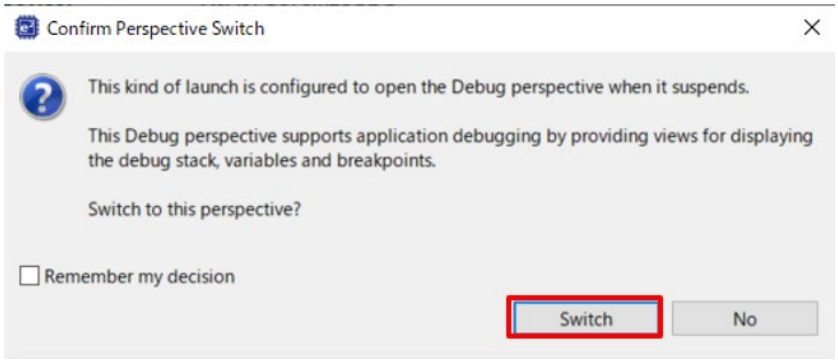


Figure 6.75 Confirm perspective switch dialog

14. The program will be downloaded to flash memory and debugging will start.
Press the "Resume" button.
When the program pauses at hal_entry() in main.c, press "Resume" again to start execution.

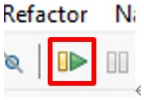


Figure 6.76 Resume button

6.3.2 Multi-Core Project (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/N2H)

6.3.2.1 Primary Core First Building

1. Launch e² studio and import the sample project by selecting [File] → [Import] → [Existing Projects into Workspace].

   Check the "select root directory" and push “Browse” button to select the following folder:

   RZT2H_EtherCAT_EVB_rev0400\CR52_Dual\project\ETG5003

   Check the following project, push “Finish” button to import project.

   RZT2H_EVB_ESC_ETG5003_CR52_Dual_Primary

   RZT2H_EVB_ESC_ETG5003_CR52_Dual_Secondary

Note) If you are using a different device or core configuration, replace the folder name and project name, according to Table 6.1.

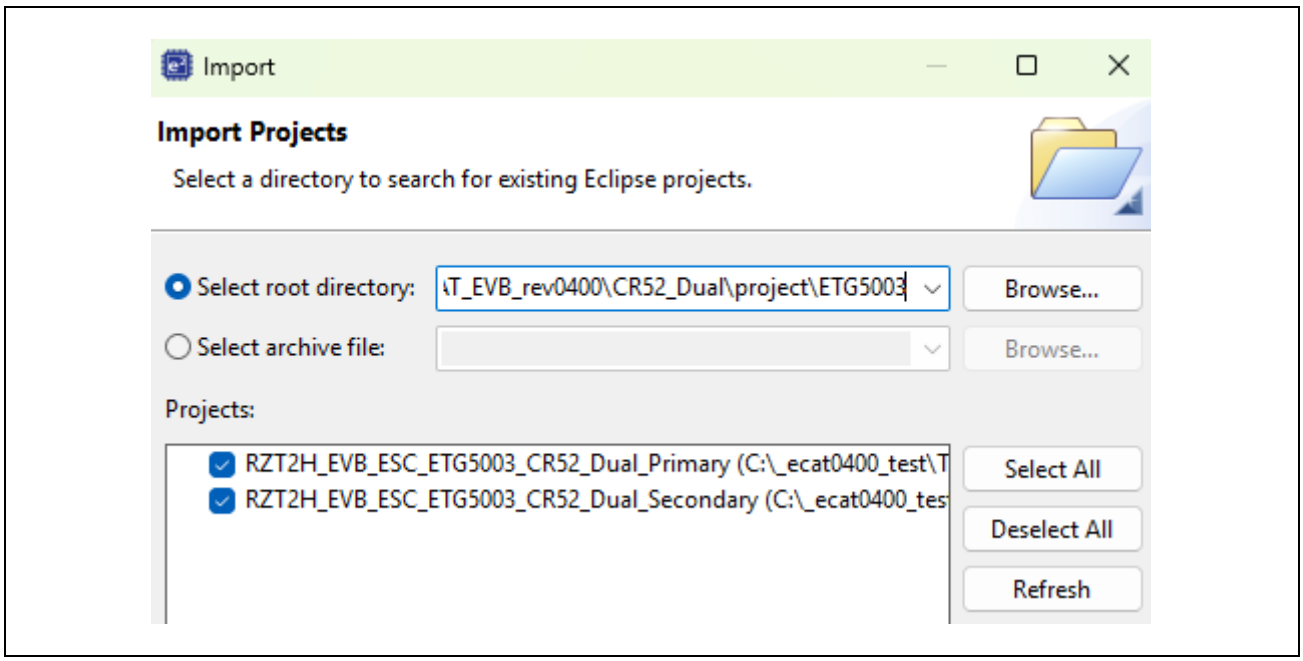


Figure 6.77 Import the projects

2. Open configuration.xml in the Primary project to launch the FSP Smart Configurator.

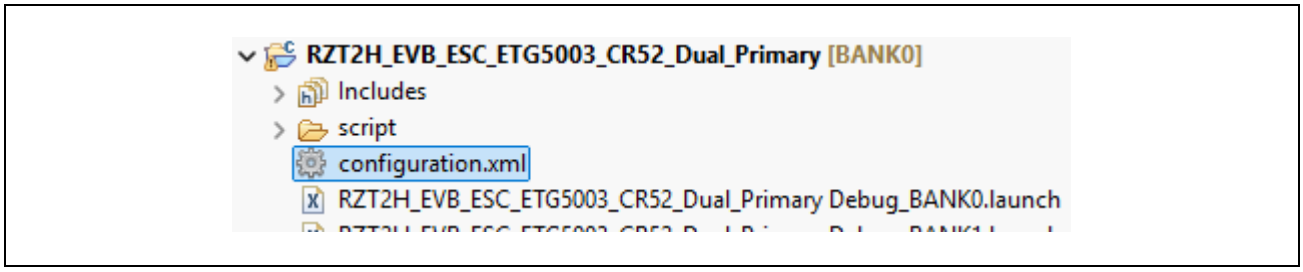


Figure 6.78 Open Primary configuration

Note) When using RZ/T2ME-RSK,

refer to the section Appendix C : How to Convert FSP Configuration from RZ/T2M to RZ/T2ME to change the primary core FSP configuration.

3. Select the BSP tab and open [Properties].

   Navigate to the following list to change the value of [LDR_ADDR_NML] for BANK0.

Table 6.20 Setting LDR_ADDR_NML for BANK0

Target device	The value of [LDR_ADDR_NML]
RZ/T2M RSK, RZ/T2ME RSK	0x60100060
RZ/T2H EVB, RZ/N2H EVB	0x50100060

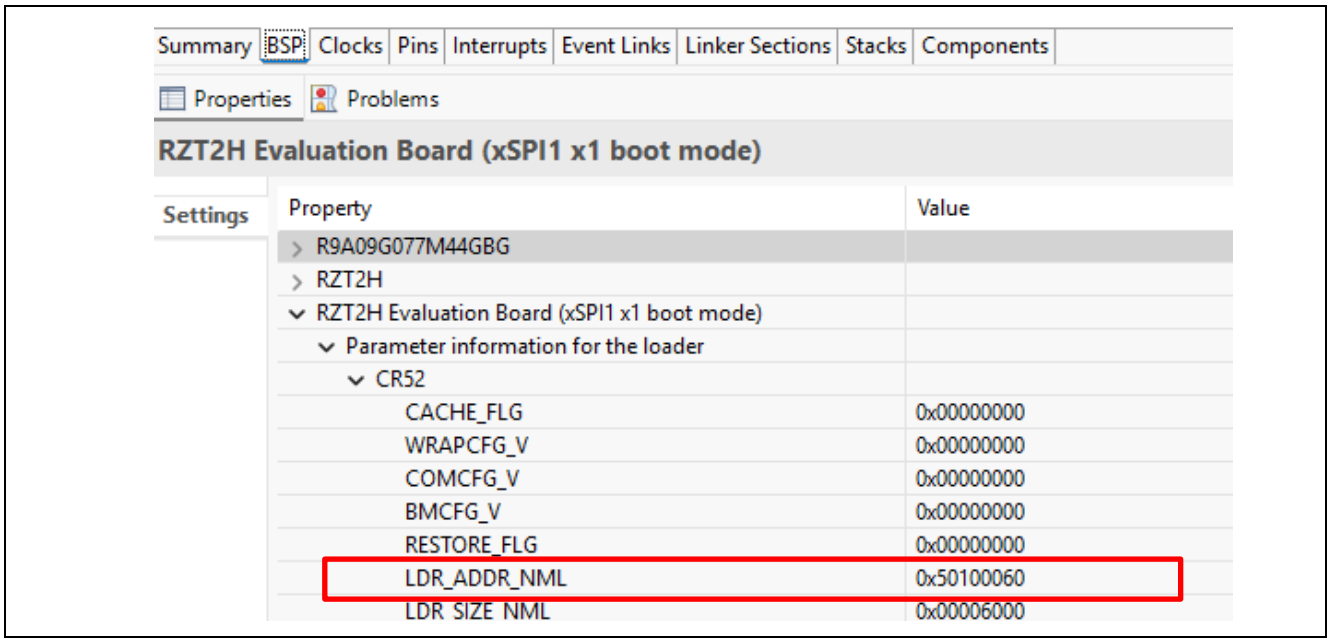


Figure 6.79 Setting for BANK0 (This is the example of RZ/T2H EVB)

4. Click **"Generate Project Content"** to generate the FSP code.

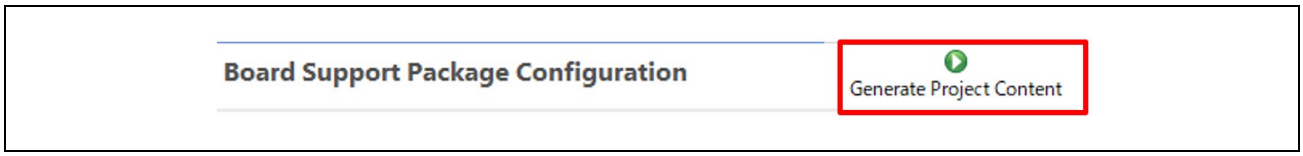


Figure 6.80 Generate Project Content

5. Select the drop-down menu next to the clock icon and choose [BANK0].

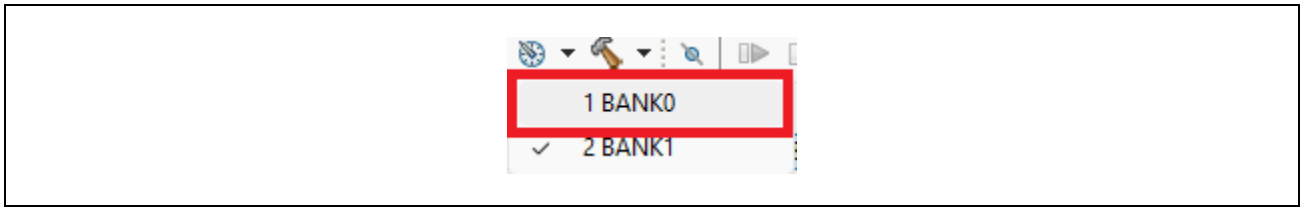


Figure 6.81 Change the Primary project configuration to BANK0

6. Click the build button (hammer icon) to build the Primary project for BANK0.

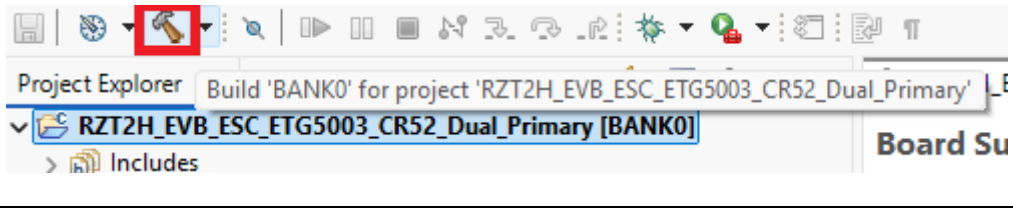


Figure 6.82 Build the Primary project for BANK0

7. Open configuration.xml in the Primary project to launch the FSP Smart Configurator again.

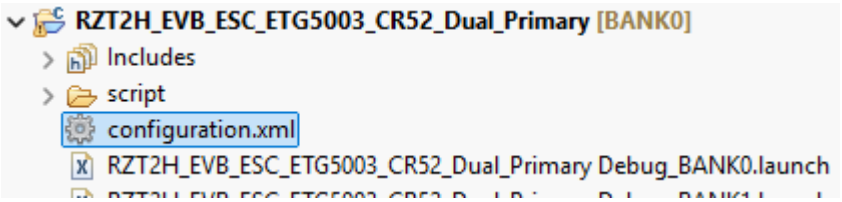


Figure 6.83 Open Primary configuration

8. Select the BSP tab and open [Properties].
   Navigate to the following list to change the value of [LDR_ADDR_NML] for BANK1.

Table 6.21 Setting LDR_ADDR_NML for BANK1

Target device	The value of [LDR_ADDR_NML]
RZ/T2M RSK, RZ/T2ME RSK	0x60200060
RZ/T2H EVB, RZ/N2H EVB	0x50200060

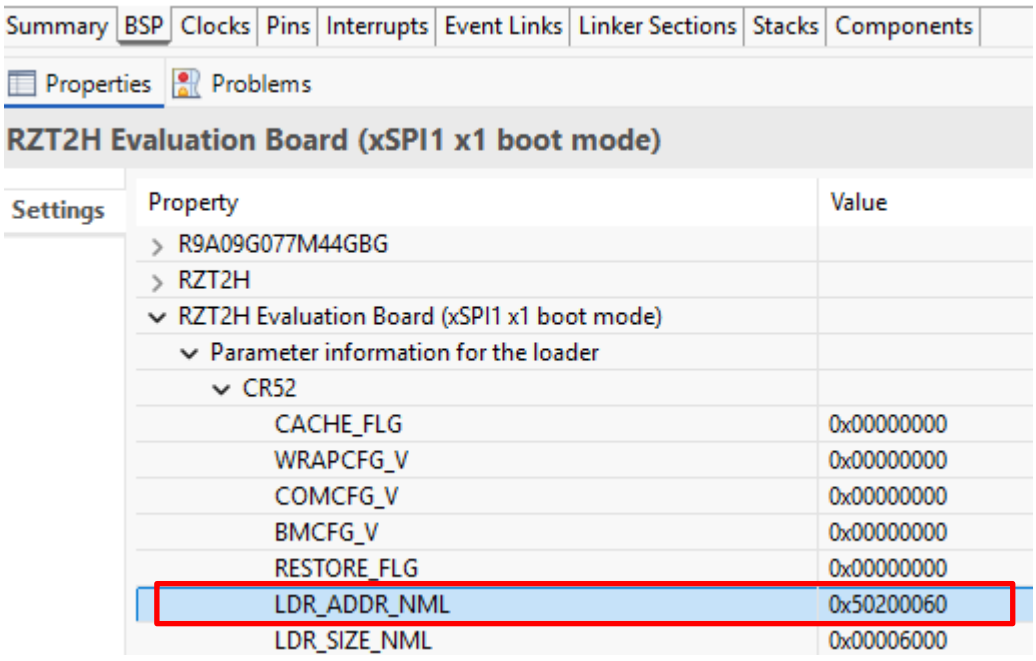


Figure 6.84 Setting for BANK1 (This is the example of RZ/T2H EVB)

9. Click **"Generate Project Content"** to generate the FSP code.

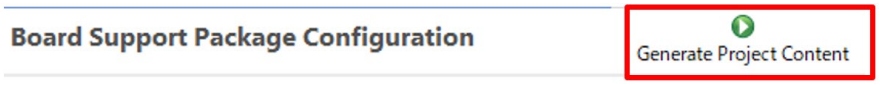


Figure 6.85 Generate Project Content

10. Select the drop-down menu next to the clock icon and choose [BANK1].

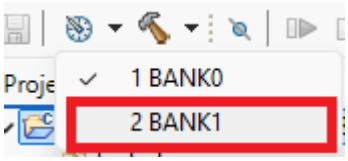


Figure 6.86 Change the Primary project configuration to BANK1

11. Click the build button (hammer icon) to build the Primary project for BANK1.

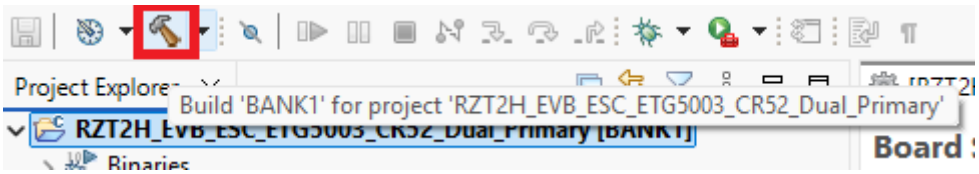


Figure 6.87 Build the Primary project for BANK1

6.3.2.2 Secondary Core Building

1. Open configuration.xml in the Secondary project to launch the FSP Smart Configurator.

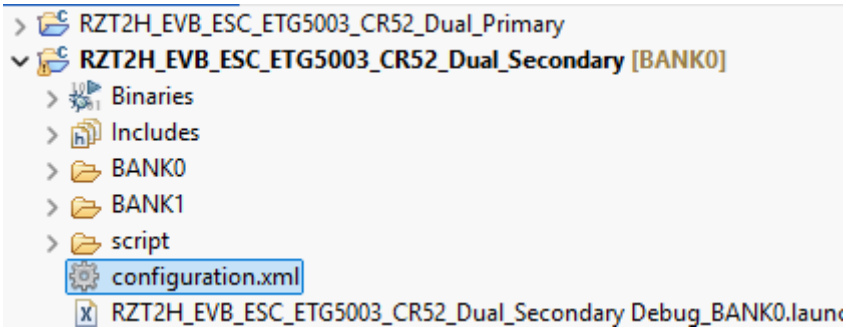


Figure 6.88 Open Secondary configuration

Note) When using RZ/T2ME-RSK,
if you followed Appendix C : How to Convert FSP Configuration from RZ/T2M to RZ/T2ME
the secondary core FSP configuration have been automatically changed.

2. Click **"Generate Project Content"** to generate the FSP code.

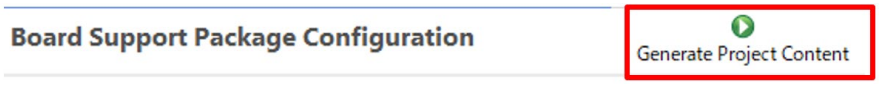


Figure 6.89 Generate Project Content

3. Select the drop-down menu next to the clock icon and choose [BANK1].

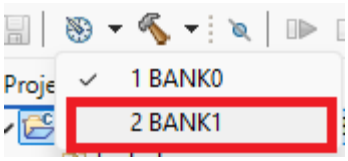


Figure 6.90 Change the Secondary project configuration to BANK1

4. Click the build button (hammer icon) to build the Secondary project for BANK1.

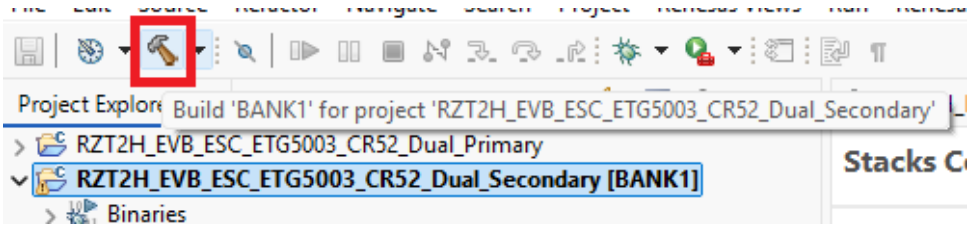


Figure 6.91 Build the Secondary project for BANK1

5. Select the drop-down menu next to the clock icon and choose [BANK0].

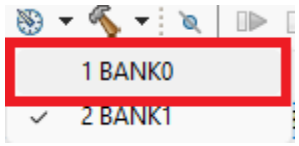


Figure 6.92 Change the Secondary project configuration to BANK0

6. Click the build button (hammer icon) to build the Secondary project for BANK0.

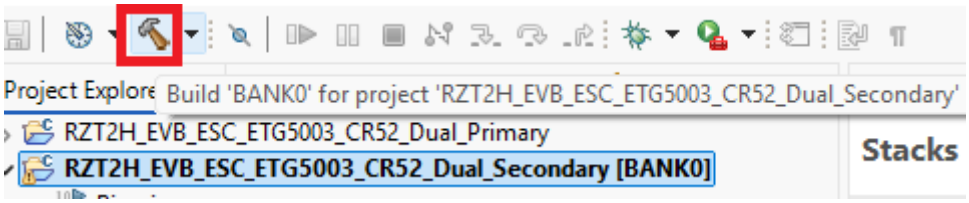


Figure 6.93 Build the Secondary project for BANK0

6.3.2.3 Primary Core Second Building

1. Open configuration.xml in the Primary project to launch the FSP Smart Configurator.

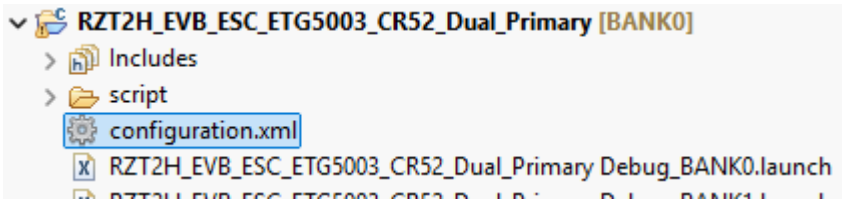


Figure 6.94 Open Primary configuration

2. Select the BSP tab and open [Properties].

   Navigate to the following list to change the value of [LDR_ADDR_NML] for BANK1.

Table 6.22 Setting LDR_ADDR_NML for BANK1

Target device	The value of [LDR_ADDR_NML]
RZ/T2M RSK, RZ/T2ME RSK	0x60200060
RZ/T2H EVB, RZ/N2H EVB	0x50200060

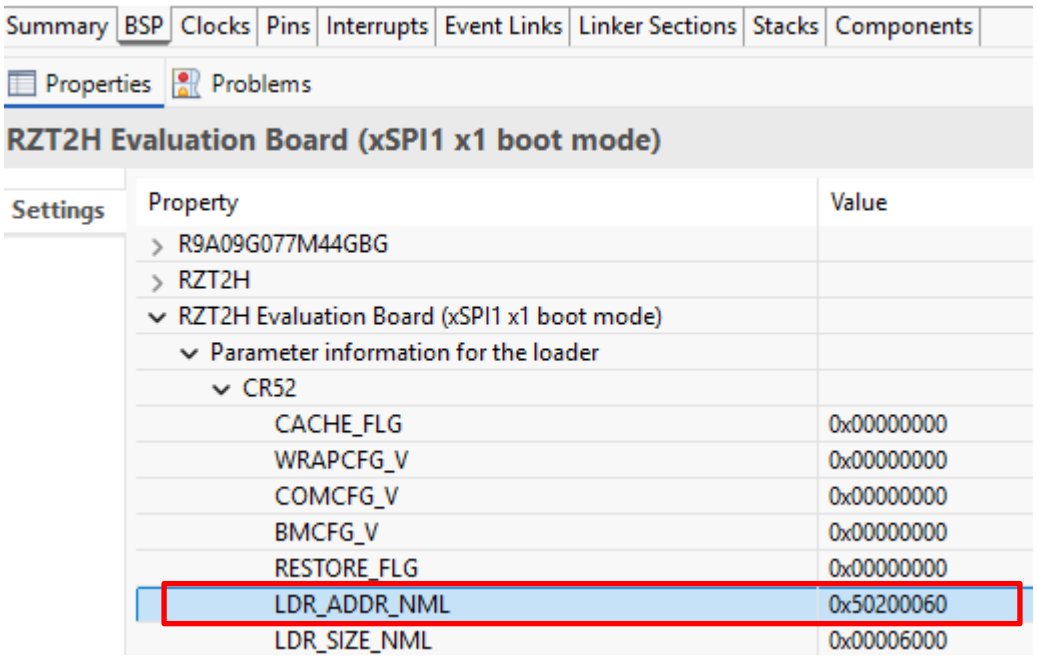


Figure 6.95 Setting for BANK1 (This is the example of RZ/T2H EVB)

3. Click **"Generate Project Content"** to generate the FSP code.

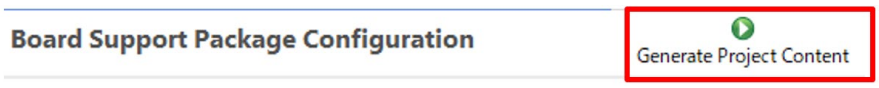


Figure 6.96 Generate Project Content

4. Select the drop-down menu next to the clock icon and choose [BANK0].

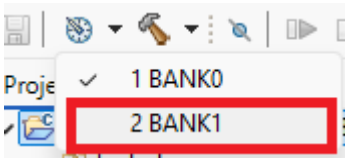


Figure 6.97 Change the Primary project configuration to BANK1

5. Please clean the BANK1 Primary project.

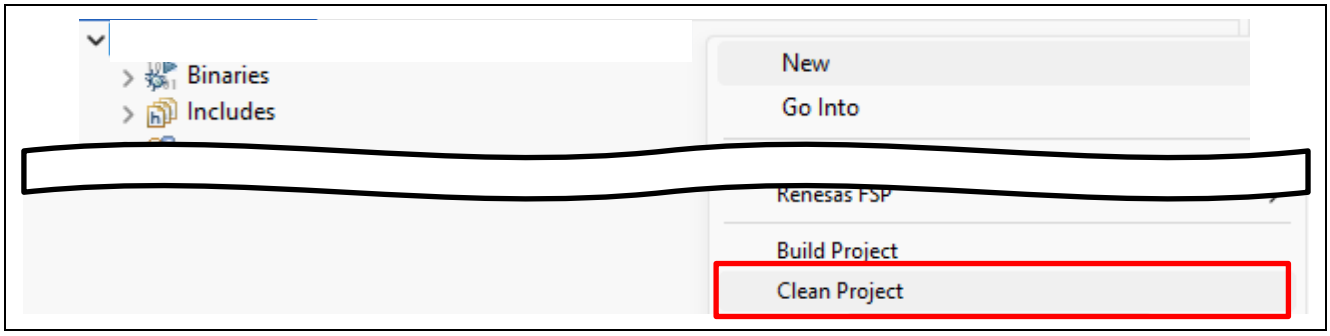


Figure 6.98 Clean the Primary project to BANK1

6. Click the build button (hammer icon) to build the Primary project for BANK1.

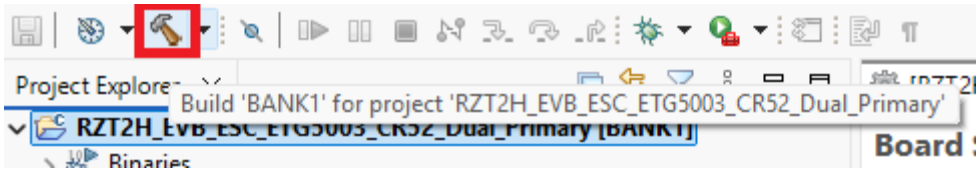


Figure 6.99 Build the Primary project for BANK1

7. Open configuration.xml in the Primary project to launch the FSP Smart Configurator again.

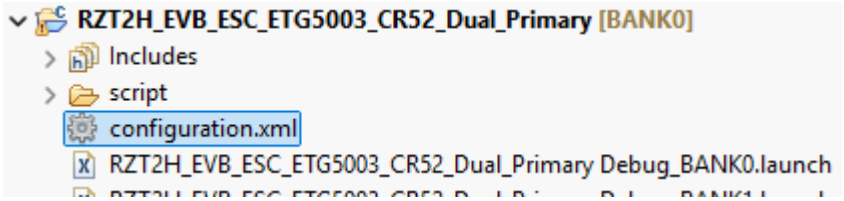


Figure 6.100 Open Primary configuration

8. Select the BSP tab and open [Properties].

   Navigate to the following list to change the value of [LDR_ADDR_NML] for BANK0.

Table 6.23 Setting LDR_ADDR_NML for BANK0

Target device	The value of [LDR_ADDR_NML]
RZ/T2M RSK, RZ/T2ME RSK	0x60100060
RZ/T2H EVB, RZ/N2H EVB	0x50100060

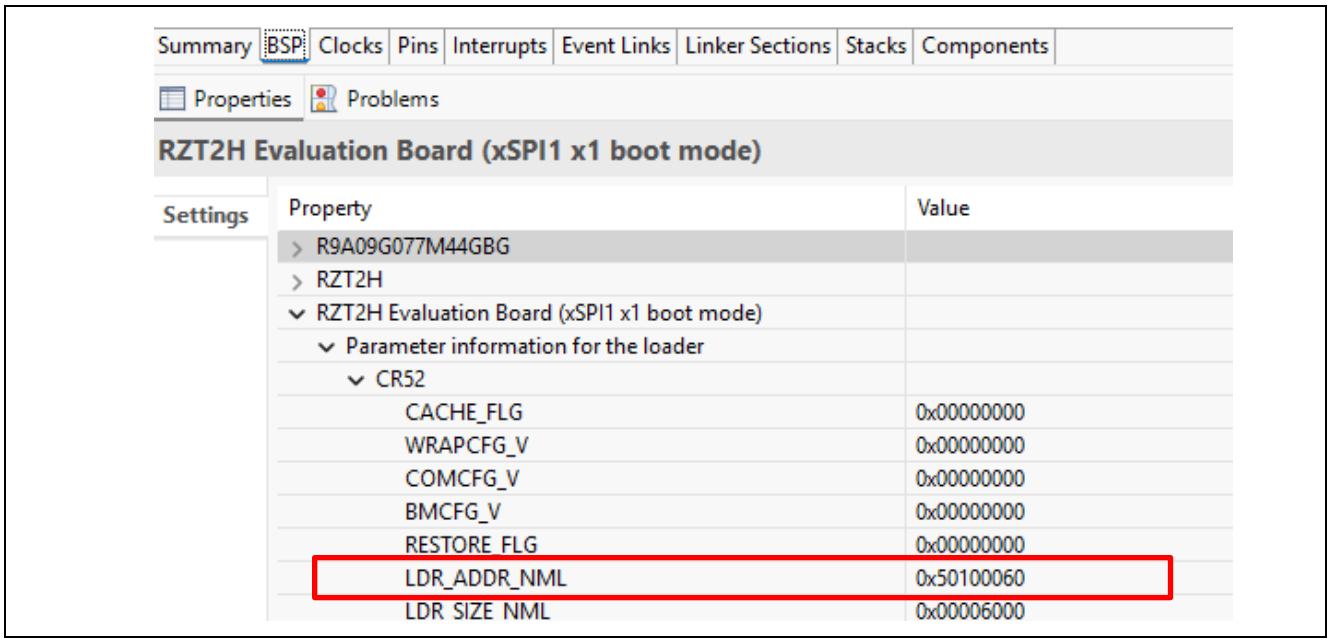


Figure 6.101 Setting for BANK0 (This is the example of RZ/T2H EVB)

9. Click **"Generate Project Content"** to generate the FSP code.

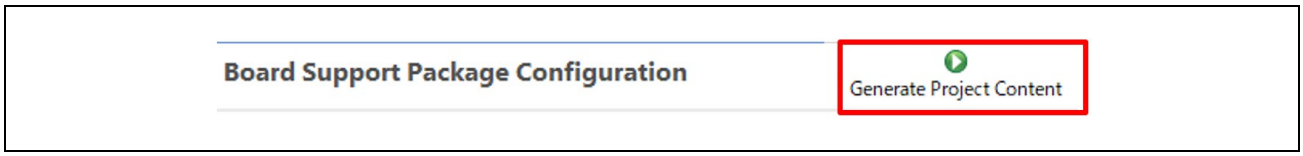


Figure 6.102 Generate Project Content

10. Select the drop-down menu next to the clock icon and choose [BANK0].

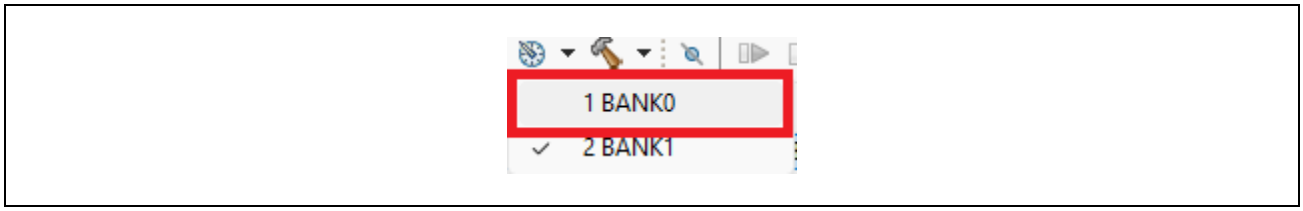


Figure 6.103 Change the Primary project configuration to BANK0

11. Please clean the BANK0 Primary project.

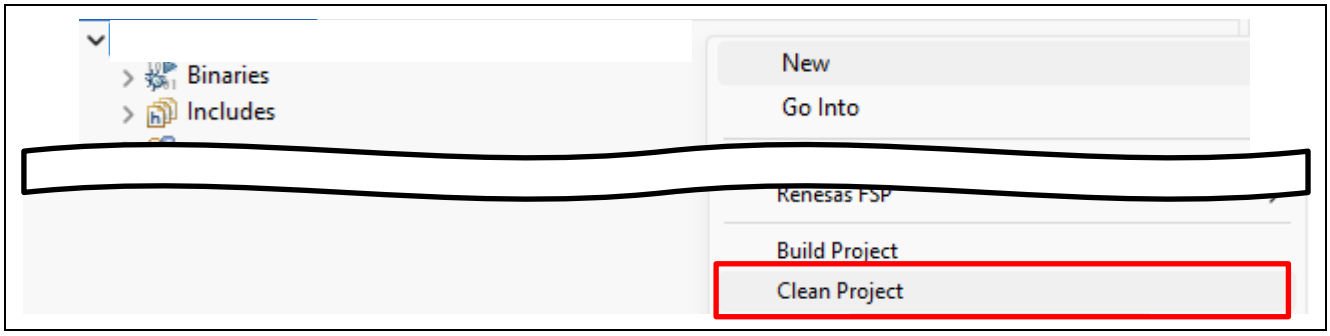


Figure 6.104 Clean the Primary project to BANK0

12. Click the build button (hammer icon) to build the Primary project for BANK0.

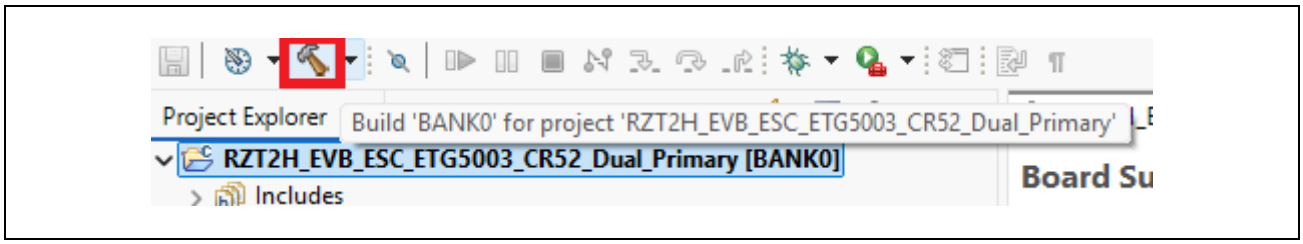


Figure 6.105 Build the Primary project for BANK0

6.3.2.4 Debug and Download for Dual Core

1. Select the drop-down menu next to the debug icon and choose [Debug Configurations…].

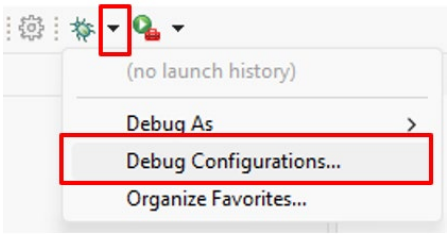


Figure 6.106 Select the drop-down menu

2. Under [Renesas DBG Hardware Debugging],
select [RZT2H_EVB_ESC_ETG5003_CR52_Dual_Primary BANK0] and click [Debug].

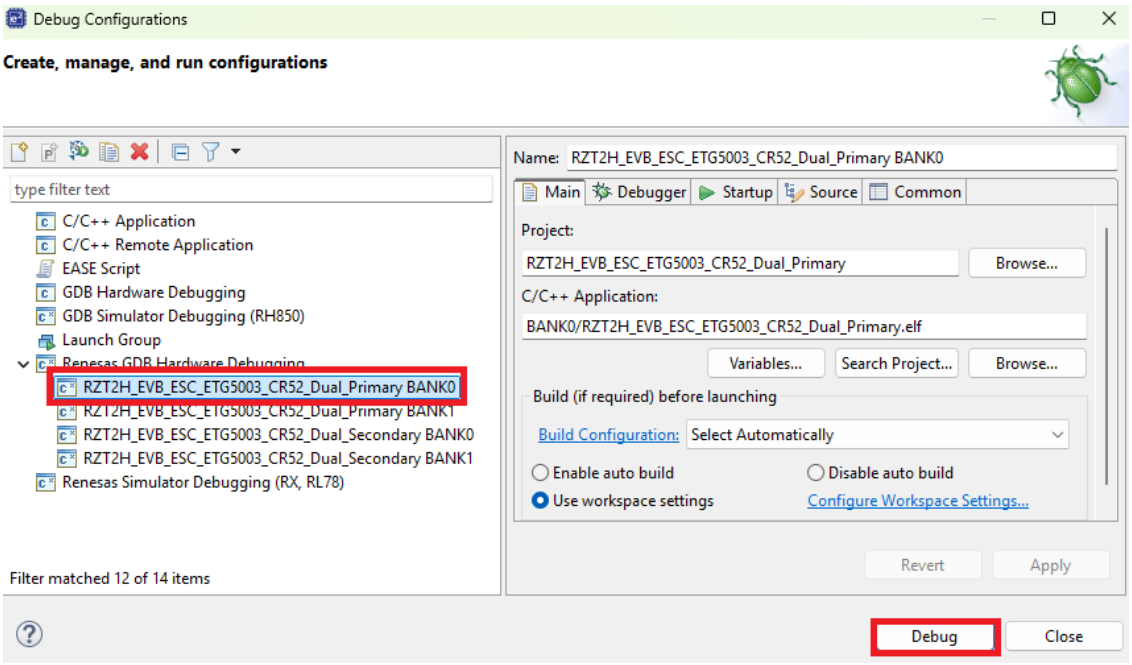


Figure 6.107 Select Primary Debug Configuration

Note) A perspective switch dialog will appear. Switch to the debug perspective.

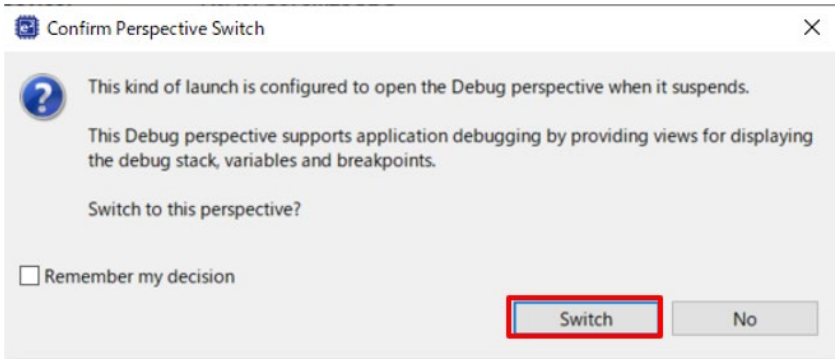


Figure 6.108 Select the drop-down menu

3. The project program will be downloaded to flash memory, and debugging will be started.
Push again the triangle next to the [Debug] button, and [Debug Configurations].

Select [Renesas DBG Hardware Debugging] →
[RZT2H_EVB_ESC_ETG5003_CR52_Dual_Secondary BANK0] item, then press [Debug].

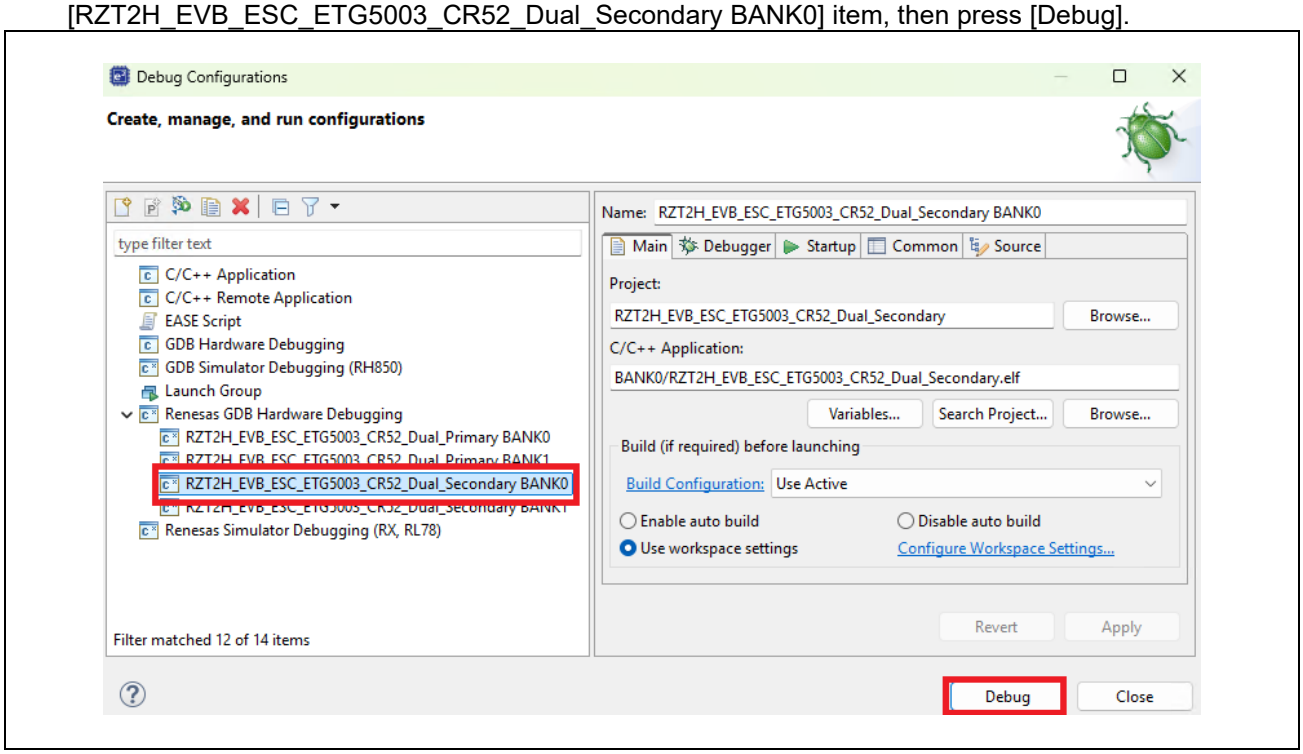


Figure 6.109 Select Secondary Debug Configuration

Note). The following dialog will appear, so click [No].

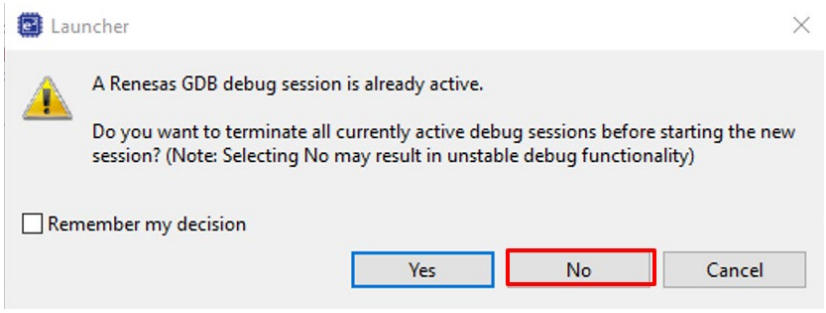


Figure 6.110 Already active dialog

Note). The following dialog will appear again, so click [Yes].

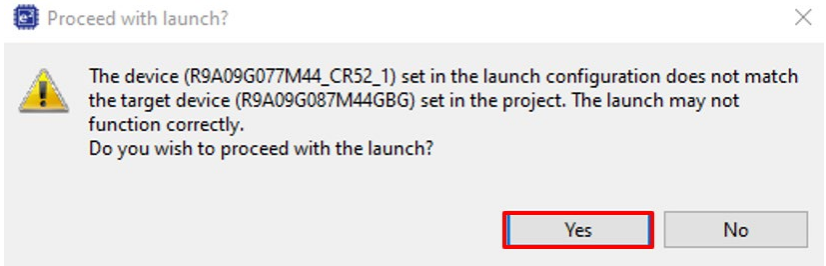


Figure 6.111 Proceed with launch dialog

4. The Secondary project program will be downloaded to RAM, and debugging will be started.
   Press the "Resume" button on the primary and secondary project to start debugging the source code.

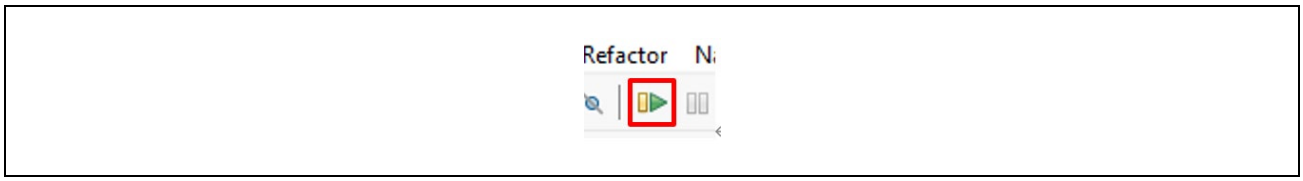


Figure 6.112 Resume button

7. Demonstration of the Sample Application

Start TwinCAT3 by using the procedure described below,
From the start menu, select [Beckhoff] → [TwinCAT3] → [TwinCAT XAE (VS2013)].

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

7.1 Scanning I/O Devices

➢ (Scan for devices): Under solution explorer -> I/O -> Devices, select 'Scan' as in Figure below.

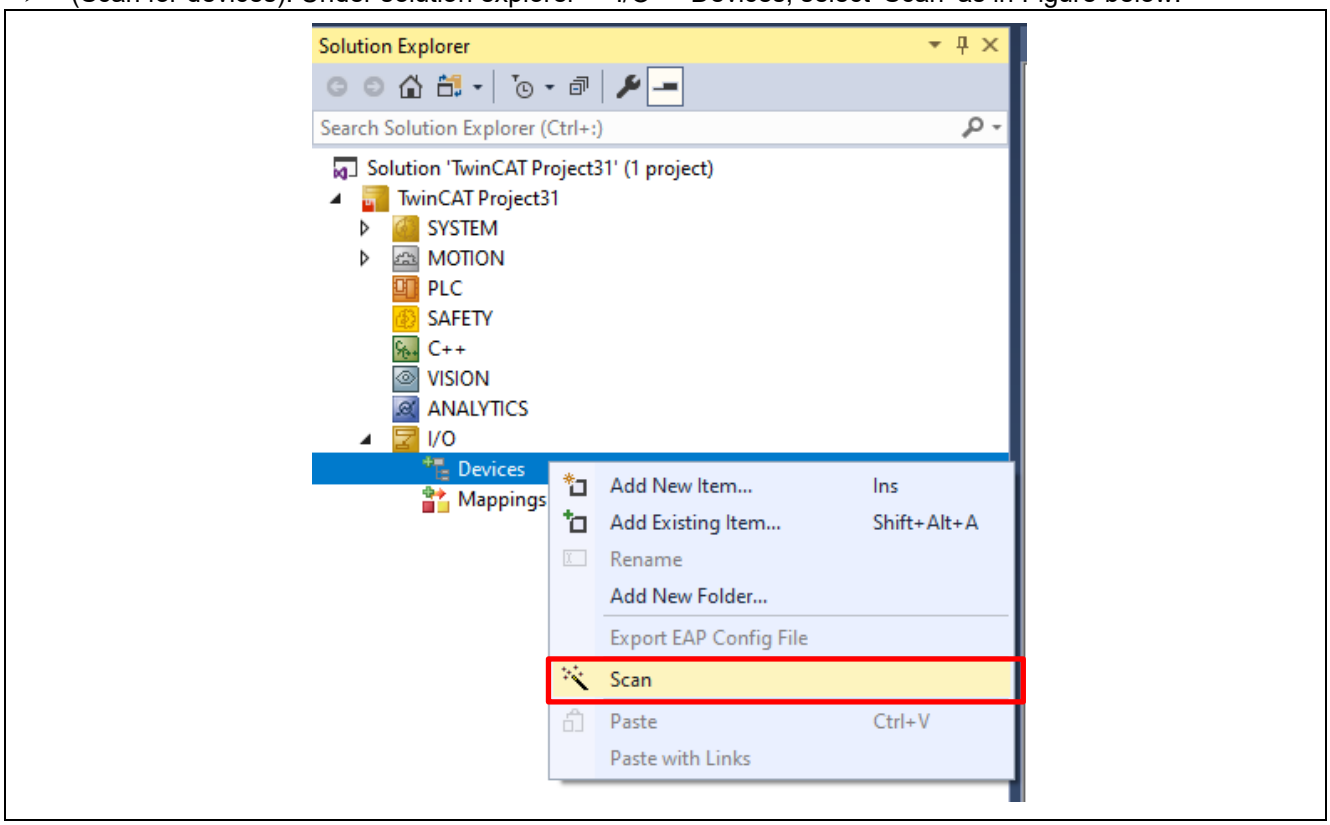


Figure 7.1 Scan devices

2. (Selecting port): EtherCAT port will be displayed as below. Select and press OK.

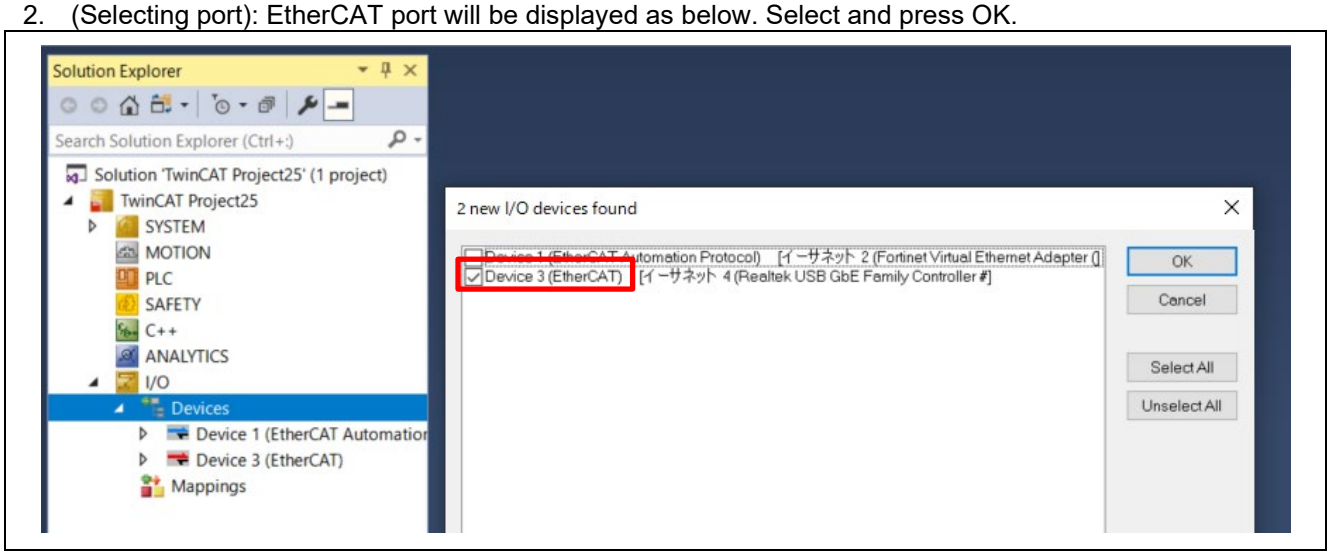


Figure 7.2 EtherCAT port is displayed

Note). If a valid EtherCAT SubDevice exists in the network, TwinCAT will display the candidate with a checkbox.

3. Start scanning the SubDevice.

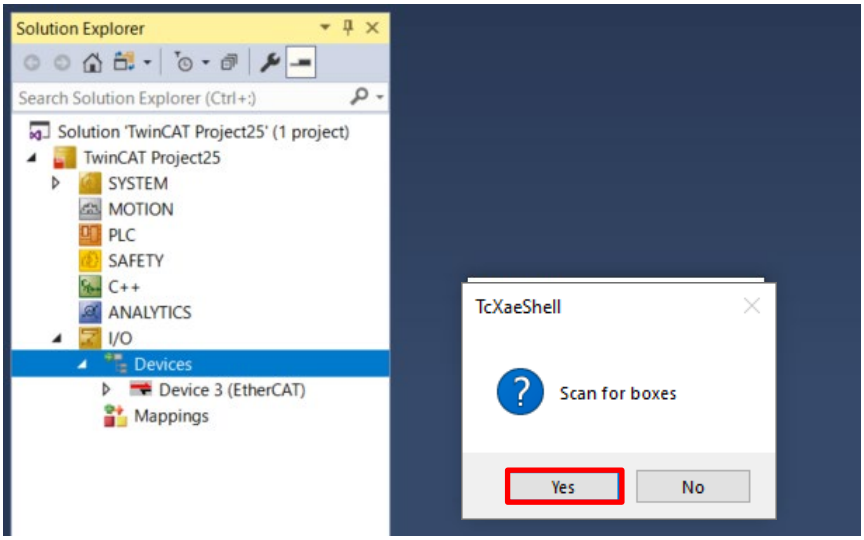


Figure 7.3 Scan SubDevice

4. (Activate SubDevice): The SubDevice is listed in the boxes. Click [Yes] on [Activate free run] dialog box.

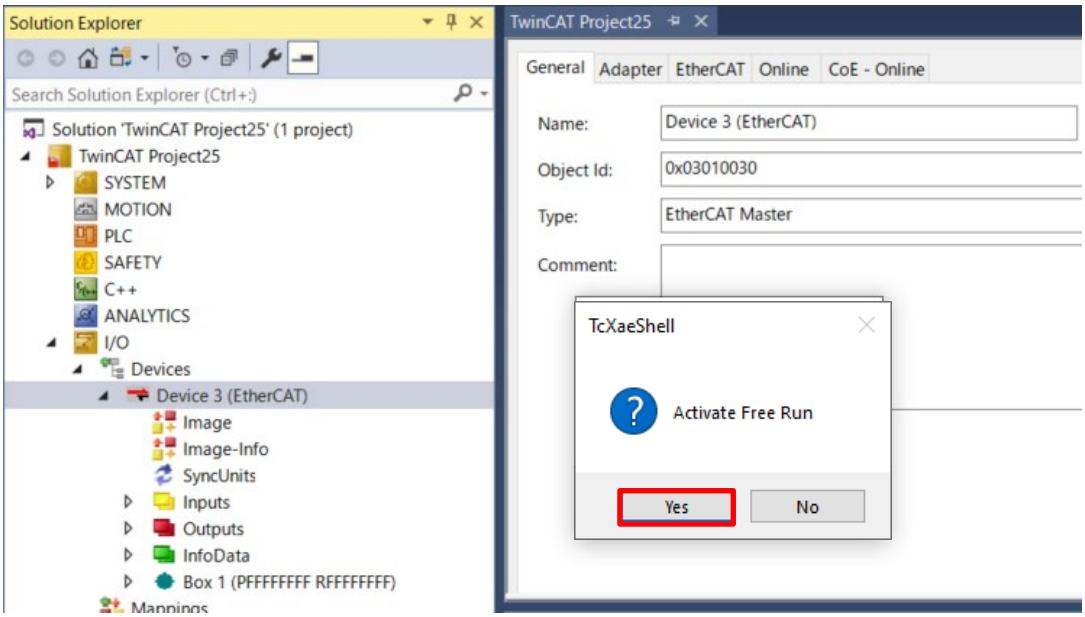


Figure 7.4 Activate Free Run

7.2 Updating EEPROM Data

If the data of another application has already been written to the EEPROM, replace the data.
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.)
   Select the [EtherCAT] tab.

2. Click the [Advanced Setting] button.

3. Select [ESC Access] → [EEPROM] → [Hex Editor].

4. Select [Download from List]

5. Select the ESI according to the following list.

Table 7.1 Select the ESI

Target device	ESI
RZ/T2L-RSK	Renesas EtherCAT RZ/T2 ETG5003 2port
RZ/T2M-RSK, RZ/T2ME-RSK, RZ/T2H EVB	Renesas EtherCAT RZ/T2 ETG5003 3port
RZ/N2L-RSK, RZ/N2H EVB	Renesas EtherCAT RZ/N2 ETG5003 3port

6. OK and Download.

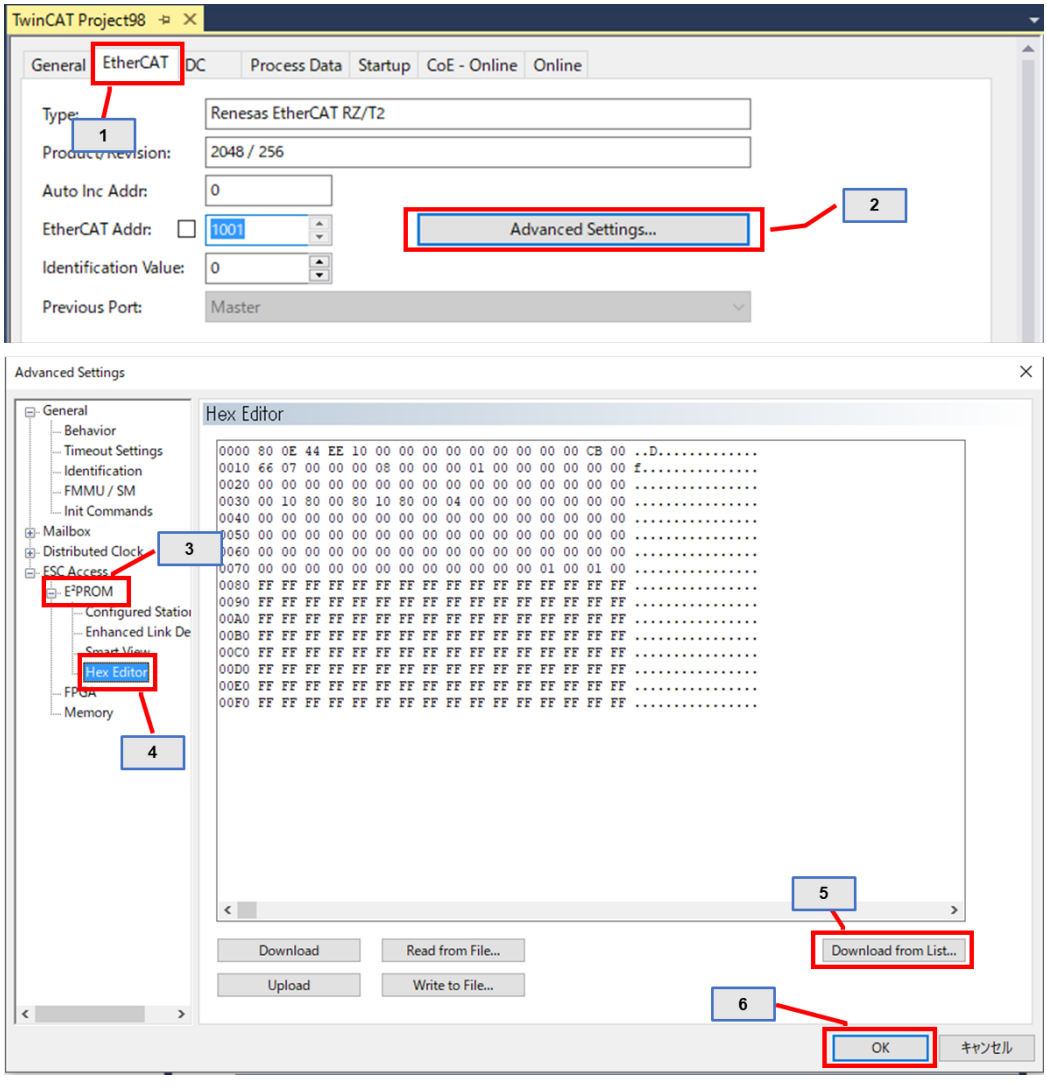


Figure 7.5 Download ESI to EEPROM

7. Apply the ESI file settings.
   Select the device and remove it.

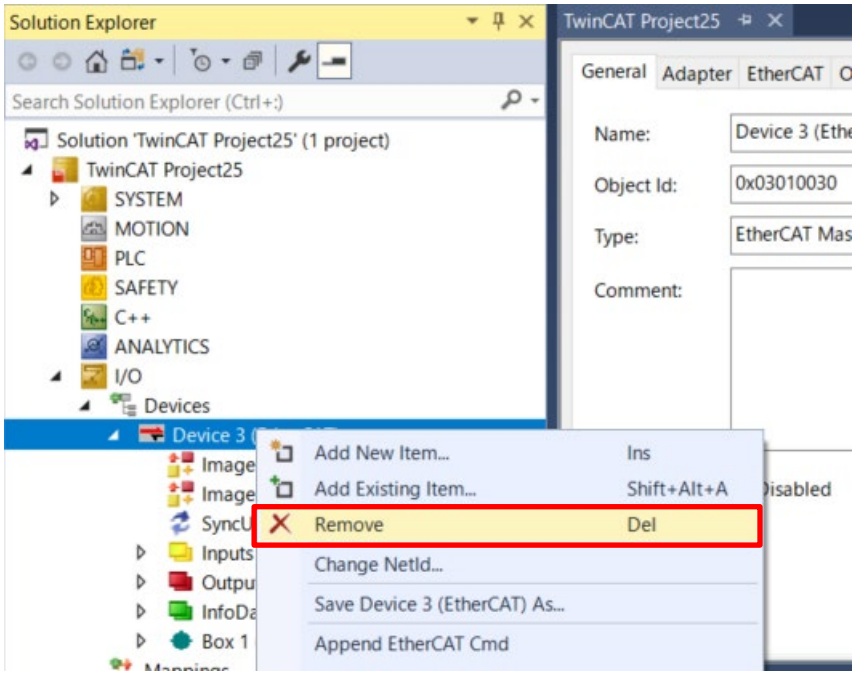


Figure 7.6 Remove the device to reflect

8. Scan the device again.
   If the desired ESI file is displayed, it is correct operation.

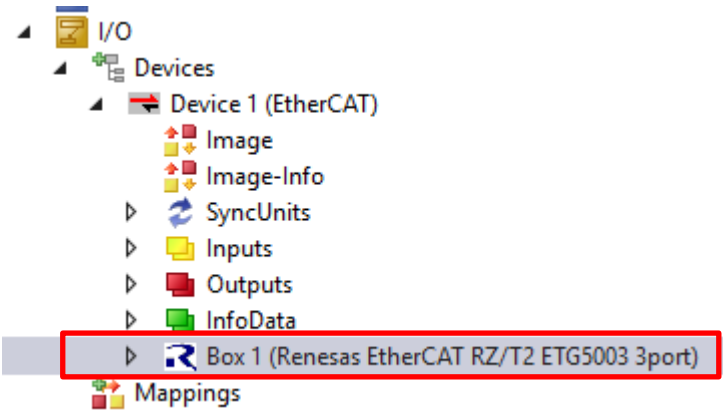


Figure 7.7 The desired ESI file (This is the example of RZ/T2H)

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 [Upload] option.

After the data is replaced, restart the RZ/x2x (by turning it off and on, or resetting it) so that the new data is applied to the microcomputer. Execute [Restart TwinCAT System].

7.3 File over EtherCAT Profile check

In this chapter, check the operation of the File over EtherCAT(FoE) profile.
Follow the steps below to rewrite the firmware.

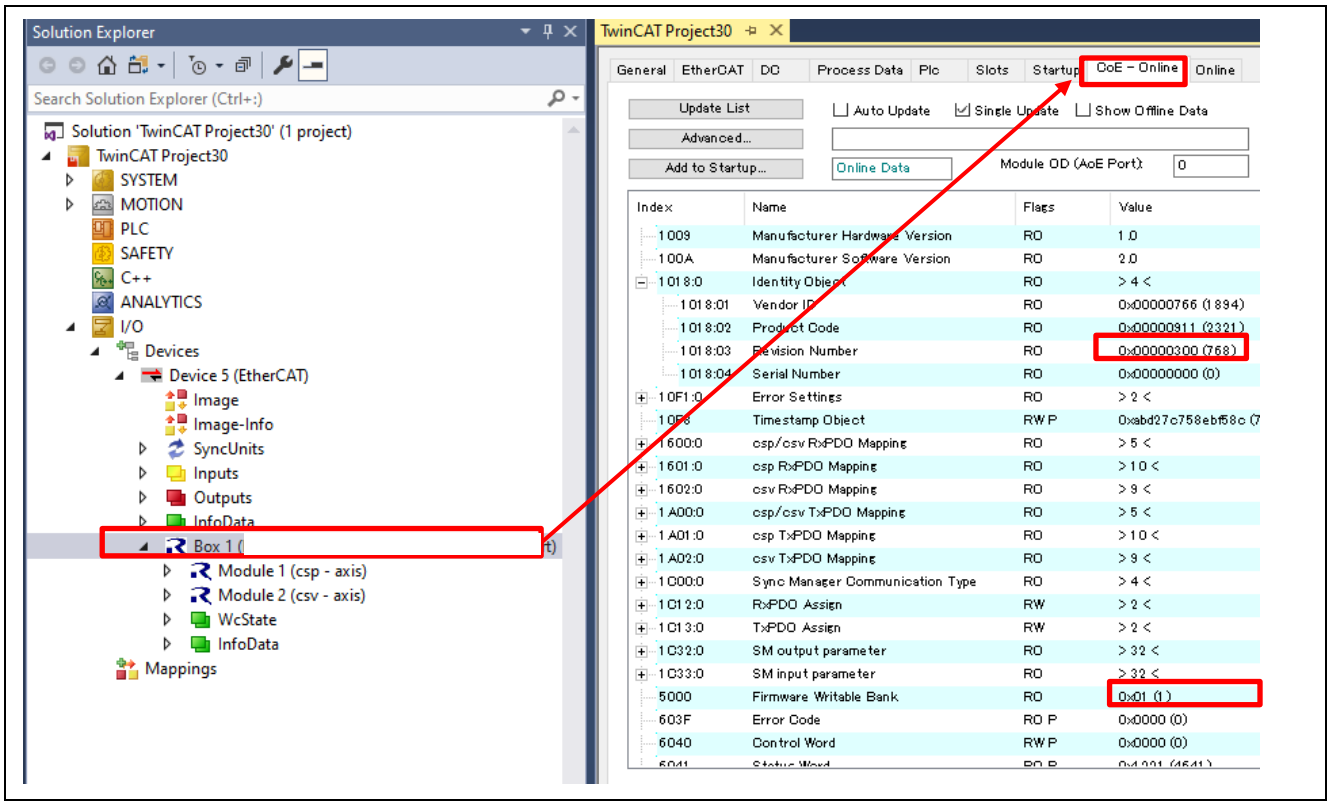
Note: When debugging, the operation of FoE will be failure.
Stop debugging, push the reset button on board to reset RZ/x2x microprocessor.

7.3.1 Confirm initial version

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

2. Confirm [1018:03 Revision Number] is **0x00000300.** At this time, Bank0 is running.
   Confirm [5000:00 Firmware Writable Bank] is **0x01.** At this time, you can do firmware update to BANK1.

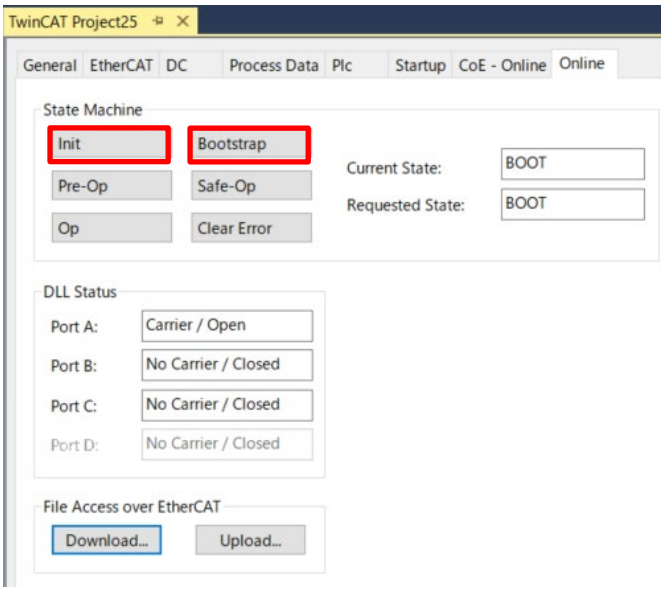
   Note) When using the sample for RZ/T2L-RSK, the revision number is **0x00000200**.



7.3.2 Firmware update

1. Select [Online] tab.

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

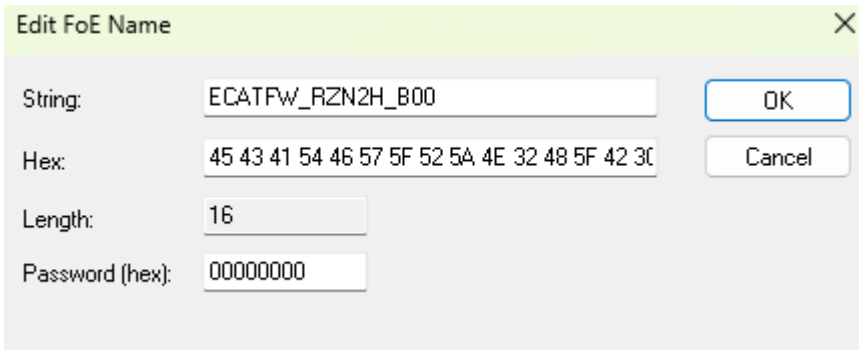


3. Click the [Download] button and select the following download file for BANK1.

Table 7.2 Select the download file for BANK1

Target device	IDE	Download file
RZ/T2L-RSK	e²studio	{Project folder}\BANK1\ECATFW_RZT2L_B01.efw
	EWARM	{Project folder}\BANK1\Exe\ECATFW_RZT2L_B01.efw
RZ/T2M-RSK, RZ/T2ME-RSK	e²studio	{Primary Project folder}\BANK1\ECATFW_RZT2M_B01.efw
	EWARM	{Primary Project folder}\BANK1\Exe\ECATFW_RZT2M_B01.efw
RZ/T2H EVB	e²studio	{Primary Project folder}\BANK1\ECATFW_RZT2H_B01.efw
	EWARM	{Primary Project folder}\BANK1\Exe\ECATFW_RZT2H_B01.efw
RZ/N2L-RSK	e²studio	{Project folder}\BANK1\ECATFW_RZN2L_B01.efw
	EWARM	{Project folder}\BANK1\Exe\ECATFW_RZN2L_B01.efw
RZ/N2H EVB	e²studio	{Primary Project folder}\BANK1\ECATFW_RZN2H_B01.efw
	EWARM	{Primary Project folder}\BANK1\Exe\ECATFW_RZN2H_B01.efw

4. Attributes of the update file are displayed, but if there are no changes, leave them.

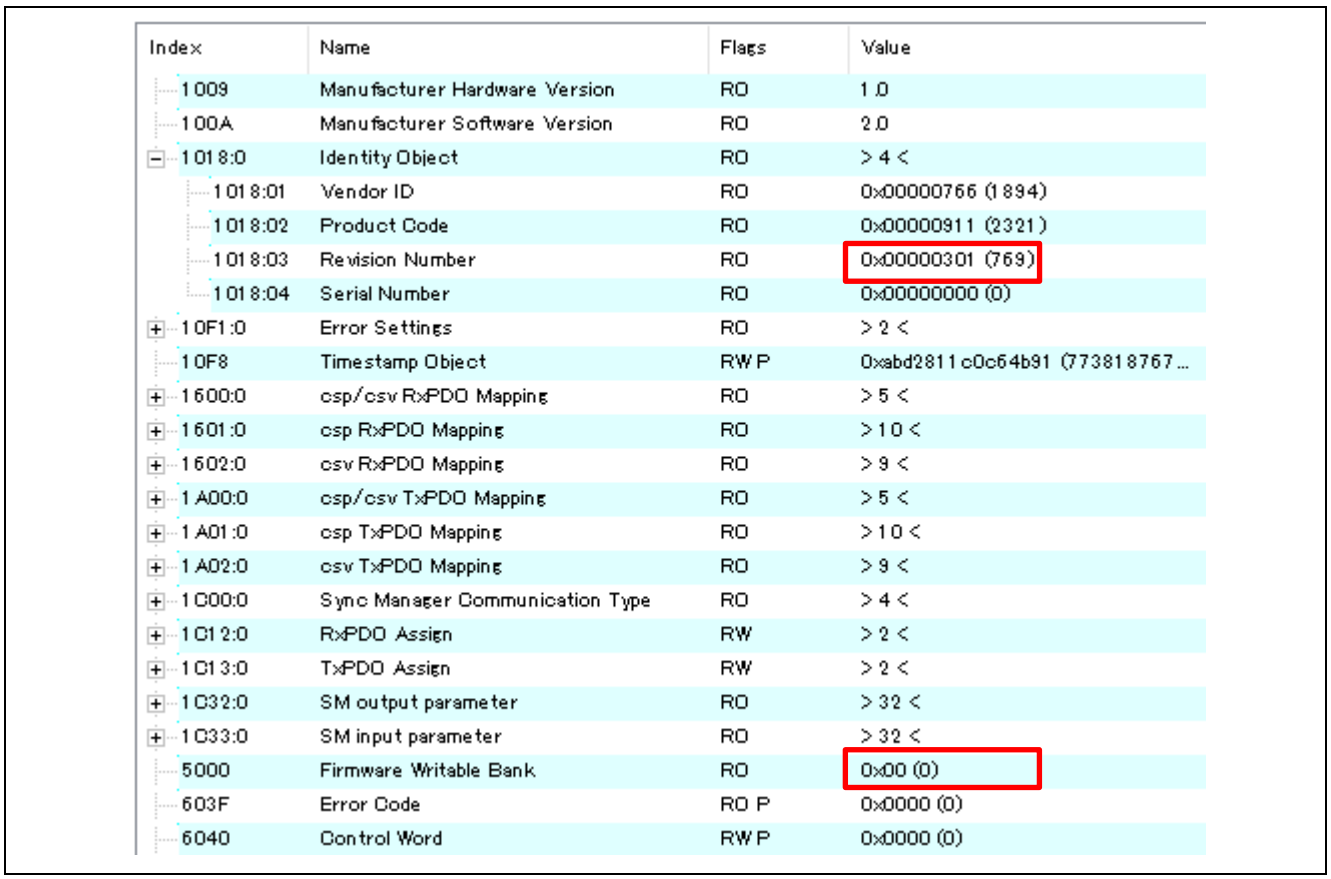


5. After the download is complete, press the [Init] button.
   EVB board will be software reset and the program will restart.

7.3.3 Check for updated version

1. Check the downloaded firmware.
   Confirm that the [1018:03 Revision Number] is **0x0000030**1 on the [CoE-Online] tab. At this time, bank1 is running.
   Confirm [5000:00 Firmware Writable Bank] is **0x00.** At this time, you can do firmware update to bank0.

   Note) When using the sample for RZ/T2L-RSK, the revision number is **0x00000201**.



Index	Name	Flags	Value
1009	Manufacturer Hardware Version	RO	1.0
100A	Manufacturer Software Version	RO	2.0
1018:0	Identity Object	RO	> 4 <
1018:01	Vendor ID	RO	0x00000766 (1894)
1018:02	Product Code	RO	0x00000911 (2321)
1018:03	Revision Number	RO	0x00000301 (769)
1018:04	Serial Number	RO	0x00000000 (0)
10F1:0	Error Settings	RO	> 2 <
10F8	Timestamp Object	RW P	0xabd2811c0c64b91 (773818767...
1600:0	csp/csv RxPDO Mapping	RO	> 5 <
1601:0	csp RxPDO Mapping	RO	> 10 <
1602:0	csv RxPDO Mapping	RO	> 9 <
1A00:0	csp/csv TxPDO Mapping	RO	> 5 <
1A01:0	csp TxPDO Mapping	RO	> 10 <
1A02:0	csv TxPDO Mapping	RO	> 9 <
1C00:0	Sync Manager Communication Type	RO	> 4 <
1C12:0	RxPDO Assign	RW	> 2 <
1C13:0	TxPDO Assign	RW	> 2 <
1C32:0	SM output parameter	RO	> 32 <
1C33:0	SM input parameter	RO	> 32 <
5000	Firmware Writable Bank	RO	0x00 (0)
603F	Error Code	RO P	0x0000 (0)
6040	Control Word	RW P	0x0000 (0)

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

Table 7.3 Select the download file for BANK0

Target device	IDE	Download file
RZ/T2L-RSK	e²studio	{Project folder}\BANK0\ECATFW_RZT2L_B00.efw
	EWARM	{Project folder}\BANK0\Exe\ECATFW_RZT2L_B00.efw
RZ/T2M-RSK, RZ/T2ME-RSK	e²studio	{Primary Project folder}\BANK0\ECATFW_RZT2M_B00.efw
	EWARM	{Primary Project folder}\BANK0\Exe\ECATFW_RZT2M_B00.efw
RZ/T2H EVB	e²studio	{Primary Project folder}\BANK0\ECATFW_RZT2H_B00.efw
	EWARM	{Primary Project folder}\BANK0\Exe\ECATFW_RZT2H_B00.efw
RZ/N2L-RSK	e²studio	{Project folder}\BANK0\ECATFW_RZN2L_B00.efw
	EWARM	{Project folder}\BANK0\Exe\ECATFW_RZN2L_B00.efw
RZ/N2H EVB	e²studio	{Primary Project folder}\BANK0\ECATFW_RZN2H_B00.efw
	EWARM	{Primary Project folder}\BANK0\Exe\ECATFW_RZN2H_B00.efw

8. Software Specifications

8.1 The details of updating the Firmware

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

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

The file for updating the firmware has a checksum, which allows checking the validity of received data.

The updated firmware is written to a different area from that for the factory-default firmware in the serial flash ROM. After the update, the user application program in the form of the updated firmware is loaded to the ATCM to run through the sequence of booting.

If updating fails, restoration of the firmware written at the time of shipment is possible.

8.1.1 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” in the table shows which program is used to implement the corresponding SubDevice operation.

Table 8.1 Procedure for Updating the Firmware

No	Master/User	SubDevice	Function		
			SSC	FW boot loader	FW updater
1	Requset BOOT	Confirm BOOT	✔		
2	Download new SubDevice FW	Download new SubDevice 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.

   Download the new updating firmware from the MainDevice.

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

3. Update the EEPROM.

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

4. 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 ROM then (2) operates with the new firmware.

5. Request the PREOP state.

   Check if the revision number in the EEPROM matches that of the firmware.

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.

8.1.2 Hardware Configuration of the Sample Program

8.1.2.1 Booting from the Serial Flash ROM

To start the boot loader in the serial flash ROM, booting must be set to select xSPIx boot mode (x1 boot serial flash).

8.1.2.2 Memory Map of Serial Flash ROM

Usage of the serial flash ROM is divided into three different areas.

Table 8.2 Classification of the Serial Flash ROM Areas

Address Range			Name (Size)	Description
RZ/N2L RZ/T2M RZ/T2ME	RZ/T2L	RZ/T2H RZ/N2H		
6000_0000H to 6000_FFFFH	6800_0000H to 6800_FFFFH	5000_0000H to 5000_FFFFH	Boot loader parameter area (64 KB)	Parameter area for the boot loader to be referred to by the boot function of the RZ/x2x
6010_0000H to 601F_FFFFH	6810_0000H to 681F_FFFFH	5010_0000H to 501F_FFFFH	BANK 0 area (1MB)	Factory-default firmware area to be written by a serial flash ROM writer, ICE, etc.
6020_0000H to 602F_FFFFH	6820_0000H to 682F_FFFFH	5020_0000H to 502F_FFFFH	BANK 1 area (1MB)	Updating firmware area to be written by the FoE

Figure 8.1 Memory Map shows the memory map of the serial flash.

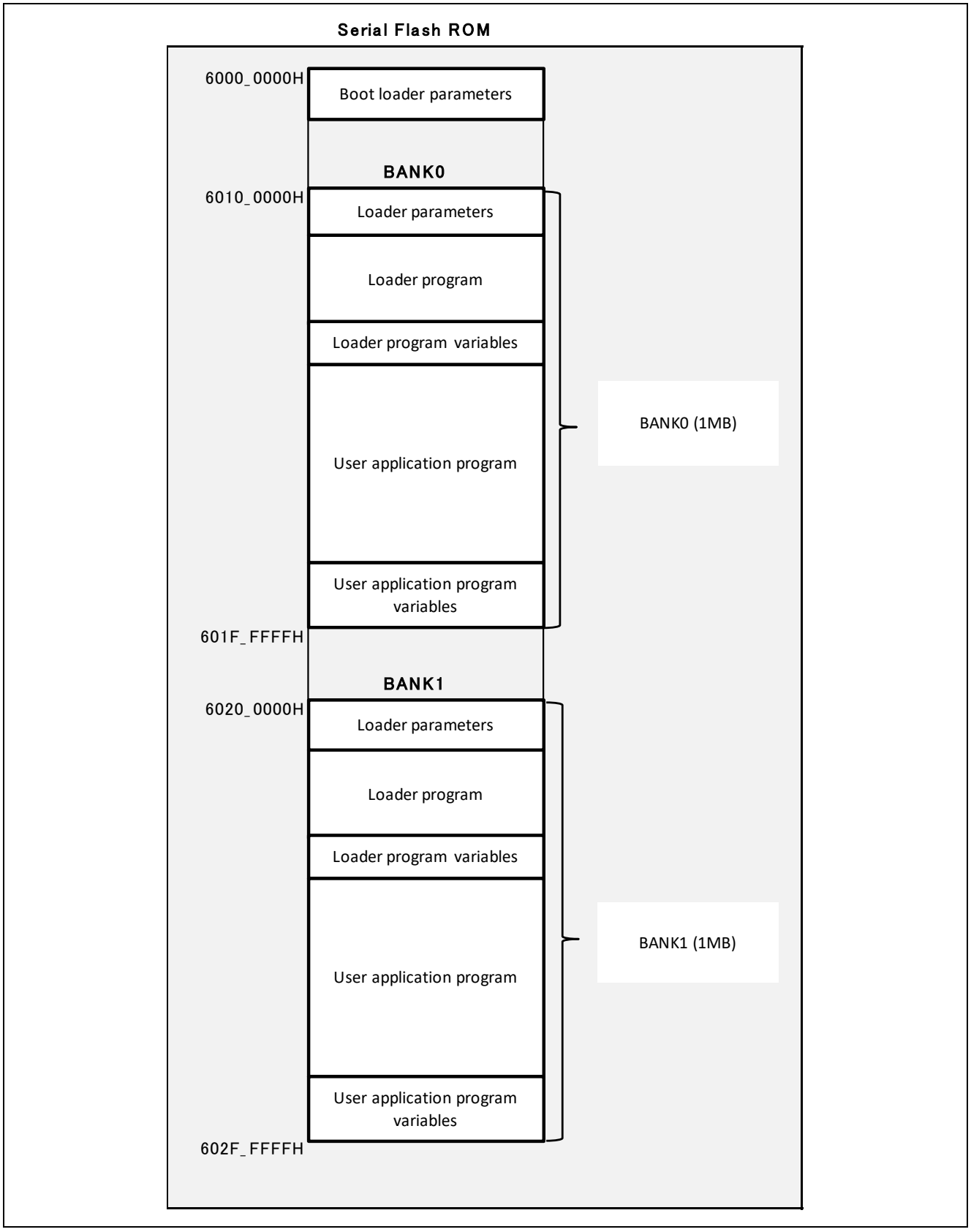


Figure 8.1 Memory Map (This is the example of RZ/N2L)

8.1.2.3 Overview of Bank 0 Boot Operations

This following describes operation for booting the factory-default firmware written to bank 0 through the procedure illustrated in Figure 8.2 BANK 0 Boot Operations.

The boot function of the RZ/N2H

(1) refers to the values in the boot loader parameter area,

(2) transfers the loader program in bank 0 to the BTCM (or SystemRAM), and then.

(3) hands processing to the loader program.

After initializing the various stack pointers, the loader program

(4) transfers the loader program variables to the BTCM (or SystemRAM) and makes settings for peripheral modules, etc.

It also refers to the values in the boot loader parameter area,

(5) transfers the user application program to the System RAM (or ATCM), and then.

(6) hands processing to the user application program.

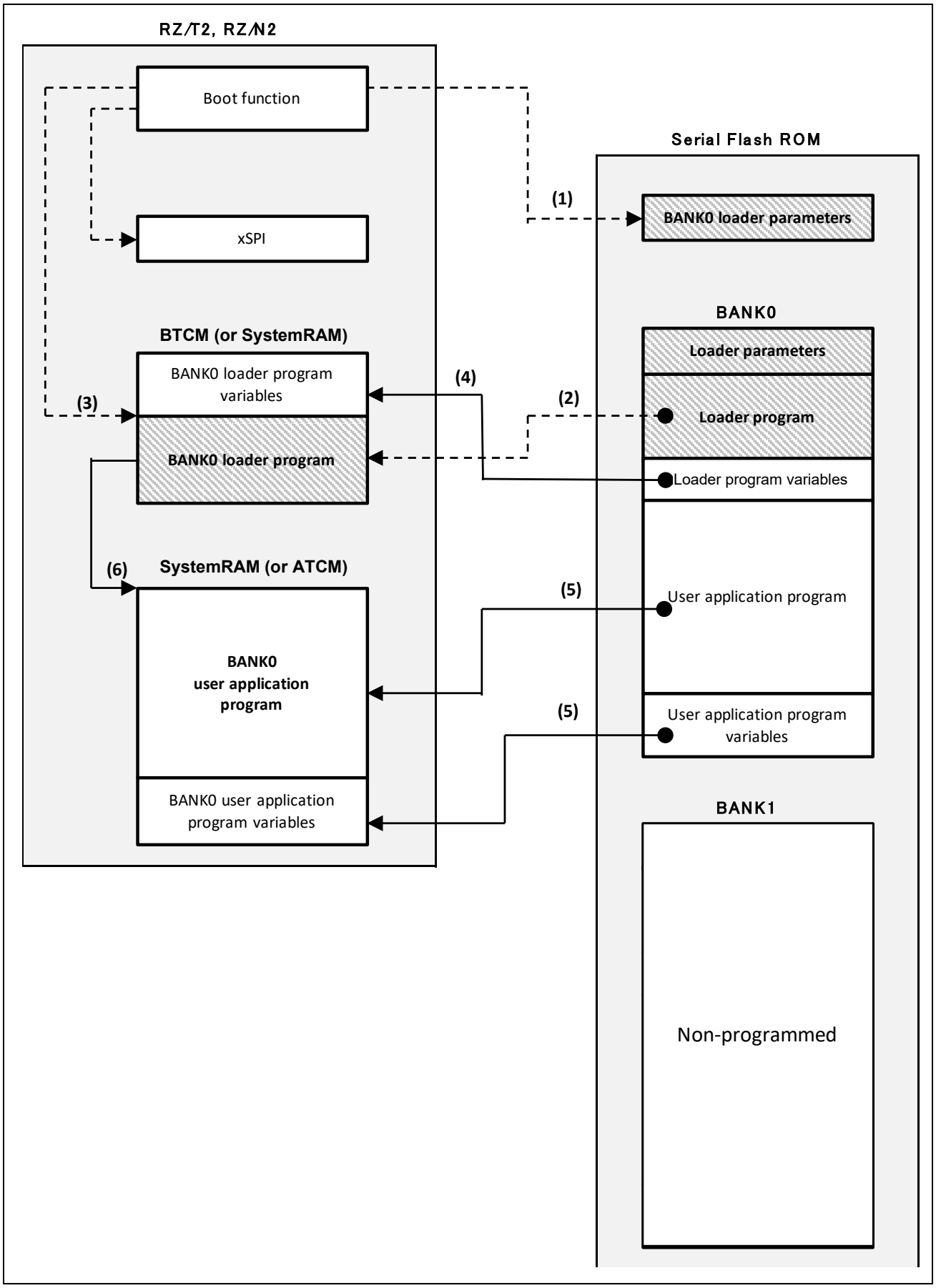


Figure 8.2 BANK 0 Boot Operations

8.1.2.4 Overview of Operation for Updating Firmware in Bank 1

The following describes operation for updating the firmware by using the FoE while a user application program is running from the System RAM, with the procedure illustrated in **Figure 8.3**, Operation for Updating the Firmware in Bank 1.

(1) The MainDevice sends the filename and password of the binary file for the updating firmware at the opening of the FoE, so check if the prefix of the filename and the password are correct. If they are correct, reception of binary data starts.

(2) Erase the boot loader parameter area at the beginning of the serial flash ROM.

(3) If the firmware update is interrupted for any reason, copy the BANK0 loader parameters to the boot loader parameter area so that the factory firmware can be started.

(4) One sector (64 KB) is erased from the address where bank 1 starts. During erasure, the busy status indicator is returned so that the MainDevice does not reach the timeout time.

(5) Data are received on completion of the erasure. The checksum attached to the received data is compared with the calculated. The data are stored in the reception buffer allocated for the storage of user application program variables in the System RAM. The data in the reception buffer are written to the serial flash ROM every time one page (256 bytes) of data is accumulated. An ACK packet is returned to the MainDevice.
On completion of writing one sector of data, erase the next sector (**Figure 8.3**-(5) in the figure).

(6) Repeat step 3 until writing all receiving data ends.

(7) Erase the boot loader parameter area at the start of the serial flash ROM.

(8) Copy the bank 1 loader parameters to the boot loader parameter area (**Figure 8.3**–(8) in the figure).

(9) Update the revision number in the SII memory to the revision number of the updating firmware.

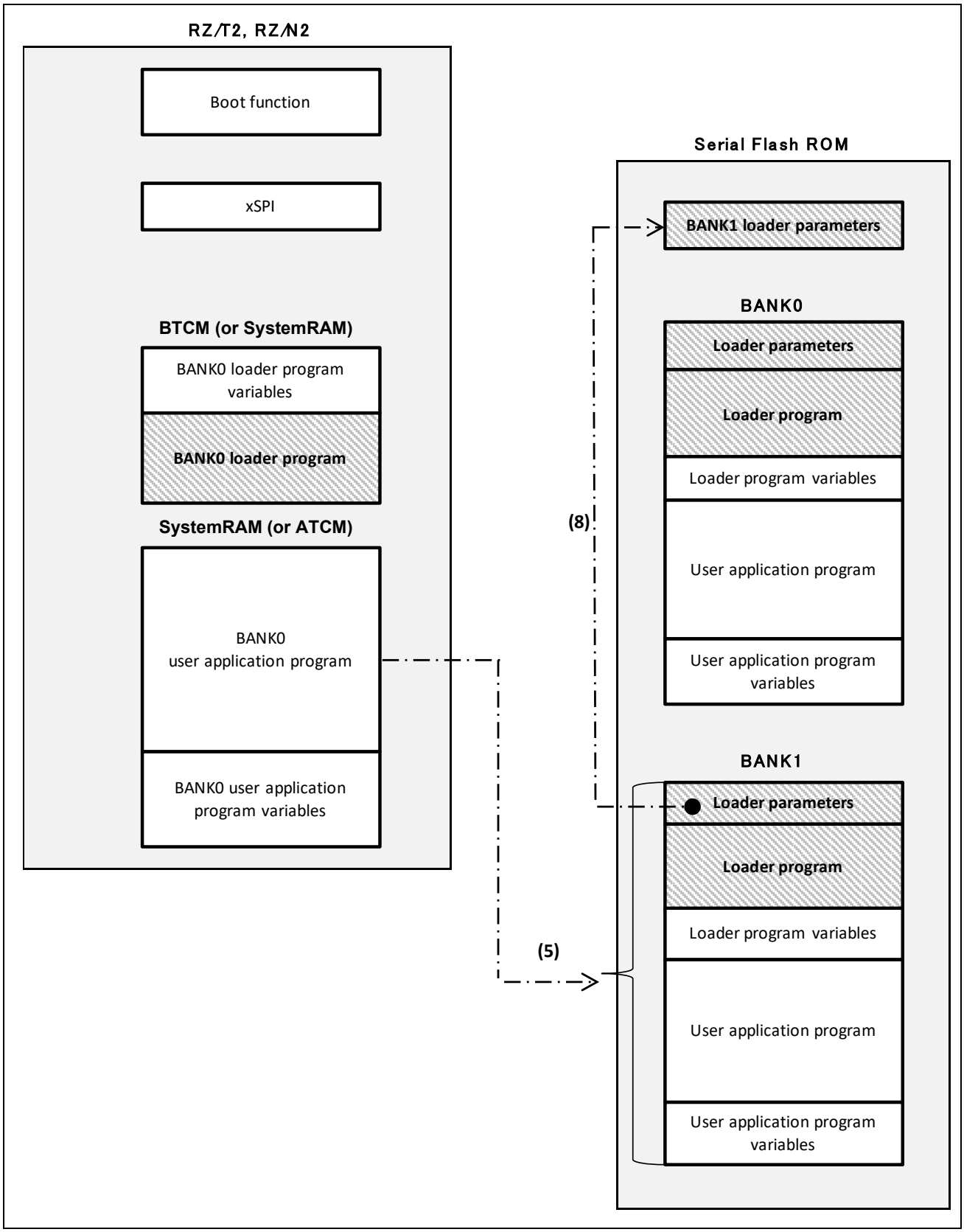


Figure 8.3 Operation for Updating Firmware in Bank 1

8.1.2.5 Overview of Operations to Reboot from Bank 1

The following describes operation from writing of the updating firmware to bank 1 until rebooting, with the procedure illustrated in **Figure 8.4**, Operations to Reboot from Bank 1.

After updating of the firmware in bank 1 has been completed normally,

The user application program operates rebooting by software reset.

The following describes operation for booting from bank after rebooting, with the procedure illustrated in **Figure 8.4**.

(1) The boot function refers to the values in the boot loader parameter area,

(2) transfers the loader program in bank 1 to the BTCM, and then.

(3) hands processing to the loader program.

After initializing the various stack pointers, the loader program

(4) transfers the loader program variables to the BTCM and makes settings for peripheral modules, etc.
It also refers to the values in the boot loader parameter area,

(5) transfers the user application program to the System RAM, and then.

(6) hands processing to the user application program.

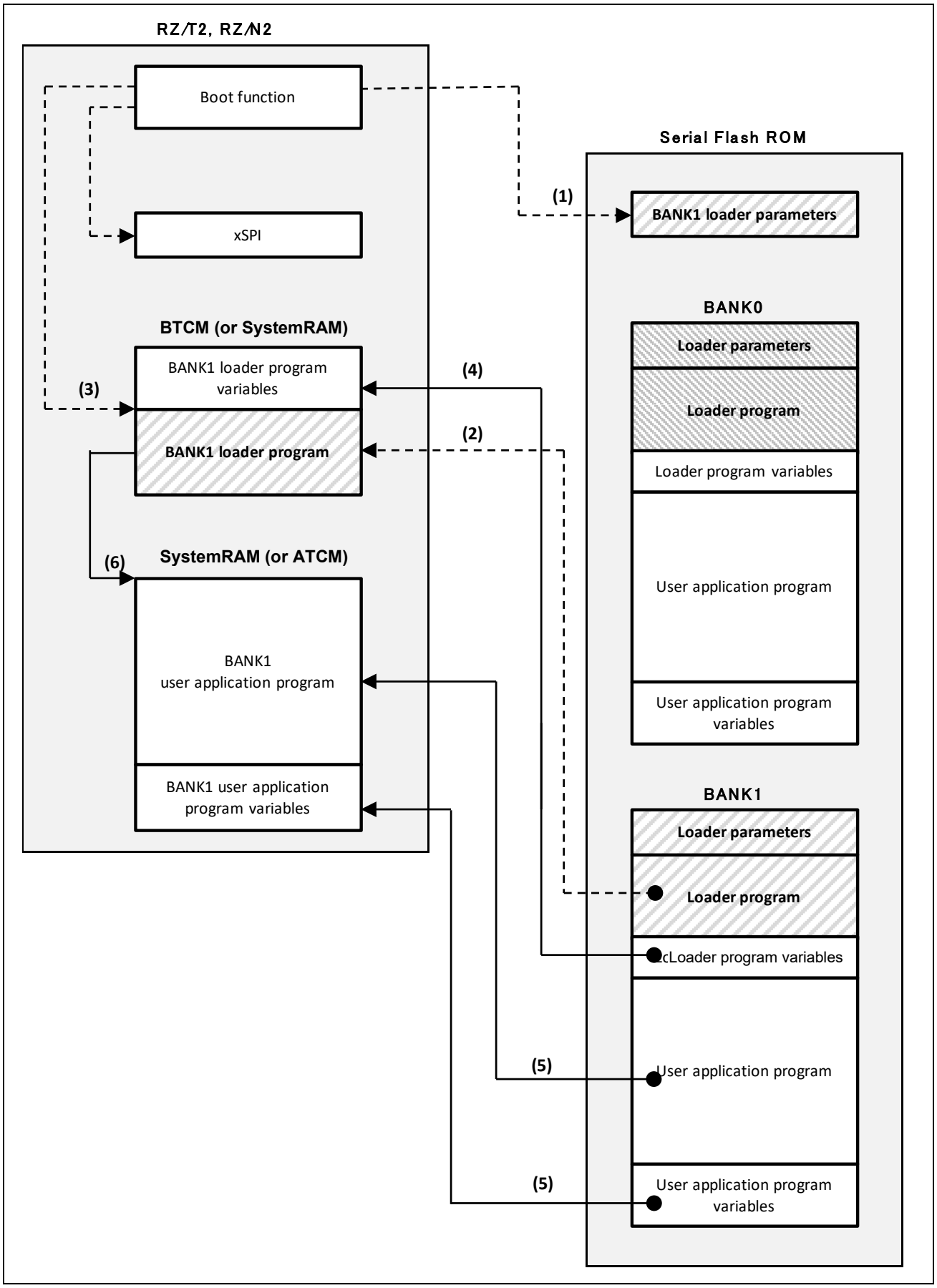


Figure 8.4 Operations to Reboot from Bank 1

8.1.3 Loader Parameters

Table 8.3 shows the loader parameter information for switching the bank area to be executed.

Table 8.3 Loader Parameter Information in xSPI boot mode

Parameter Name	Address			Description
	RZ/N2L RZ/T2M RZ/T2ME	RZ/T2L	RZ/T2H RZ/N2H	
LDR_ADDR_NML	6000_0014H	6800_0014H	5000_0014H	Sets the address where the loader program starts.

8.1.4 Constants

Table 8.4 Constants Used in the Sample Program

Constant Name	Setting			Description
	RZ/N2L RZ/T2M RZ/T2ME	RZ/T2L	RZ/T2H RZ/N2H	
FW_UP_PAGE_SIZE	(256)			Page size of the serial flash ROM
FW_UP_LDRPRM_ADDR	(0x60000000)	(0x68000000)	(0x50000000)	Loader parameters address
FW_UP_LDRPRM_LDR_ADDR	(0x60000014)	(0x68000014)	(0x50000014)	LDR_ADDR_NML address
FW_UP_APPLI_ID_OFFSET	RZ/N2L: (0x00000050) Other: (0x00080900)	(0x00000050)	(0x00080900)	Application identifies information placed address
FW_UP_BANK0_ADDR	(0x60100000)	(0x68100000)	(0x50100000)	Address where bank 0 starts
FW_UP_BANK1_ADDR	(0x60200000)	(0x68200000)	(0x50200000)	Address where bank 1 starts
FW_UP_MIRROR_OFFSET	(0x20000000)	(0x20000000)	(0x00000000)	xSPIx Mirror space minus offset value for Cortex-R52
	(None)	(None)	(- 0x10000000)	xSPIx Mirror space minus offset value for Cortex-A55
FW_UP_APPLI_SIZE	(0x00100000)			Bank area size (1MB)

8.1.5 Functions

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

Table 8.5 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)

9. Appendix

9.1 Appendix A : A point of caution when using SSC Tool

※ When opening SSC Tool

1. Please open 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 not check for updates.

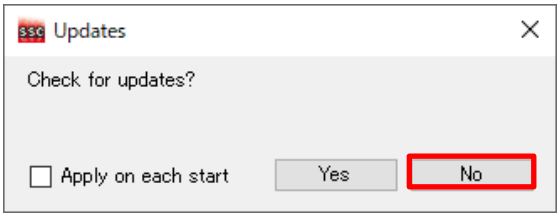


Figure 9.1 The update window

※ Regarding SSC Tool settings

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

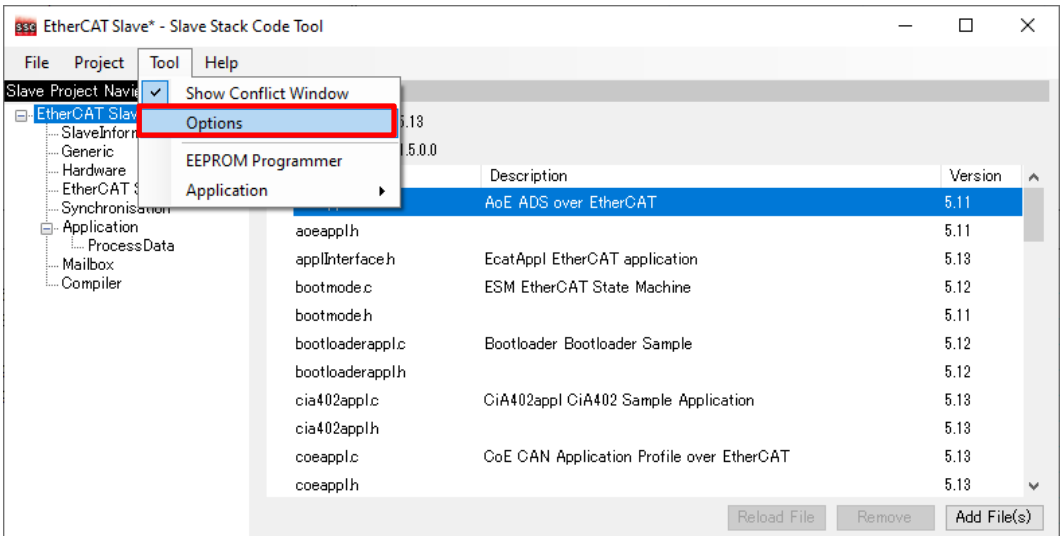


Figure 9.2 SSC tool main window

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

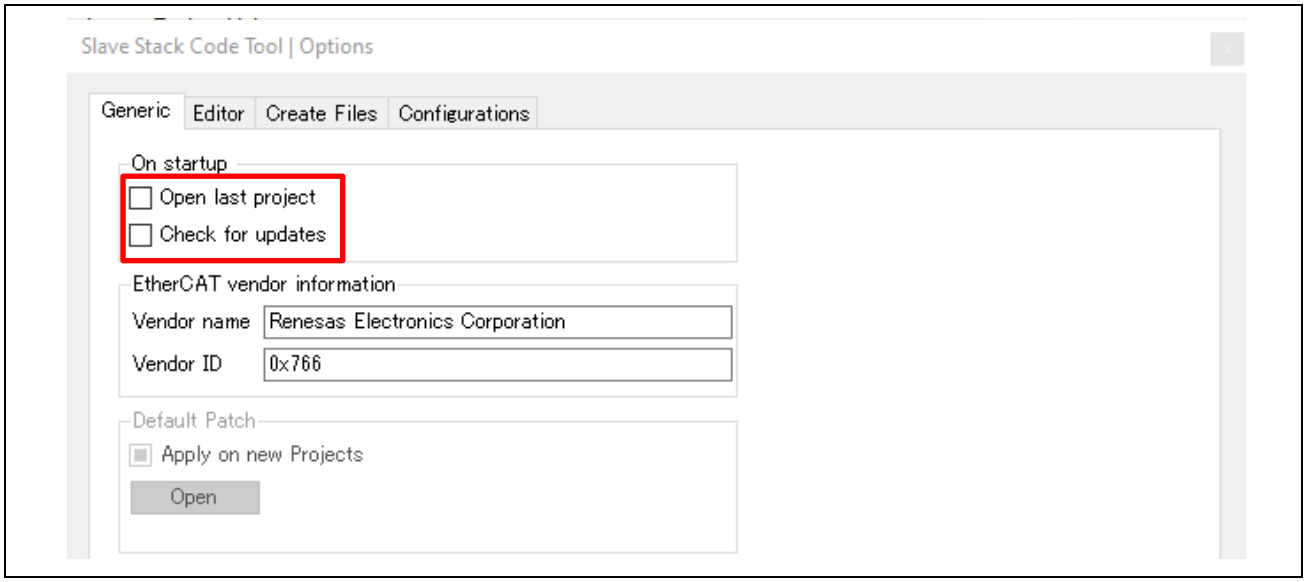


Figure 9.3 Option settings (1)

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

Figure 9.4 Option settings (2)

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

9.2 Appendix B : How to install patch

There are two methods as follows:

1. Via Git for Windows (64bit)
2. Via MinGW Installation Manager

9.2.1 Via Git for Windows (64bit)

This section describes how to install patch via Git for Windows.

1. Download the installer (e.g., Git-x.xx.x-64-bit.exe) from the official Git for Windows website.

   Git for Windows

2. Run the downloaded installer and follow the setup instructions. Use the default settings unless you have specific requirements.

3. After installation, add the path to patch.exe to your system's environment variables.
   For a default installation, the path is typically:

   "C:\Users\<your-username>\AppData\Local\Programs\Git\usr\bin"

   To apply the changes, restart your computer after updating the environment variables.

4. Start Command Prompt, enter "where patch".
   If the path to patch.exe is displayed, the installation was successful.

Figure 9.5 Confirm patch.exe (1)

9.2.2 Via MinGW Installation Manager

This section describes how to install patch.exe of MinGW.

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

   https://osdn.net/projects/mingw/downloads/68260/mingw-get-setup.exe/

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

3. If It’s 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”.

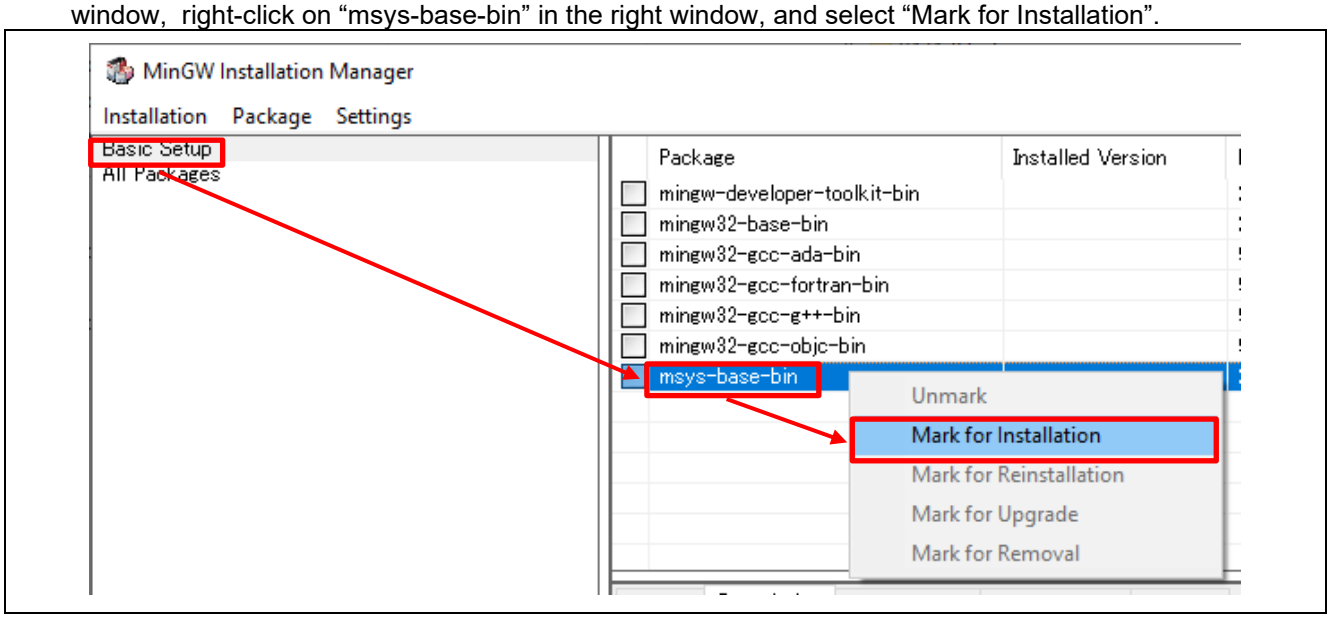


Figure 9.6 MinGW Installation Manager (1)

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

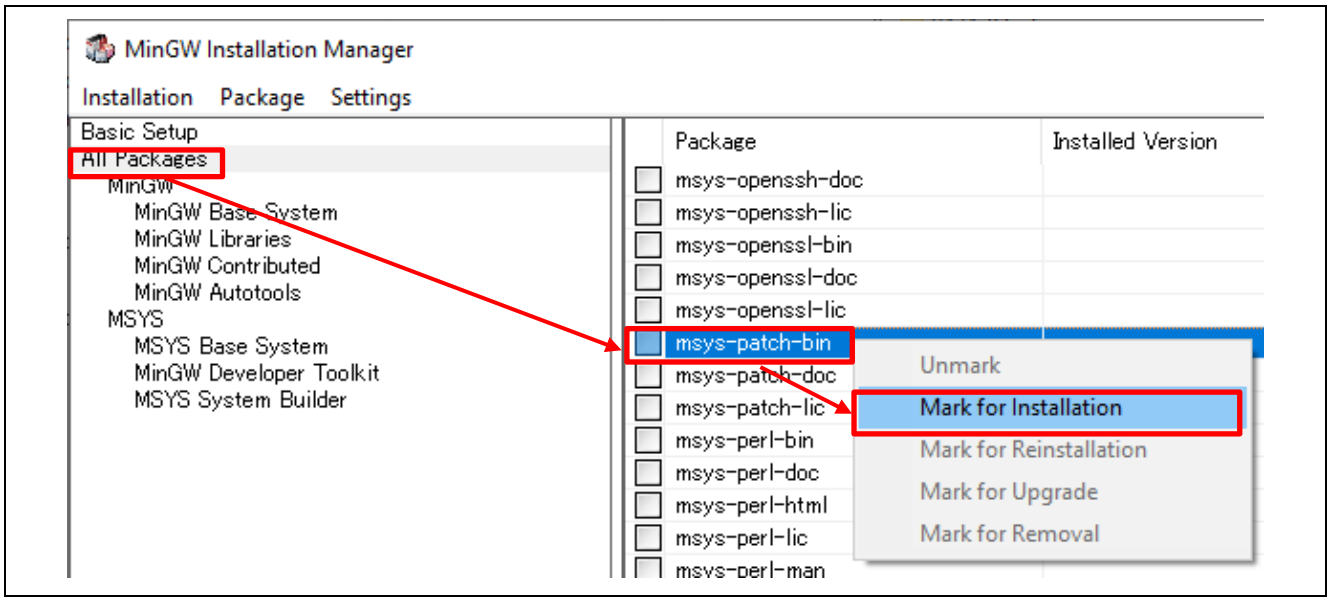


Figure 9.7 MinGW Installation Manager (2)

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

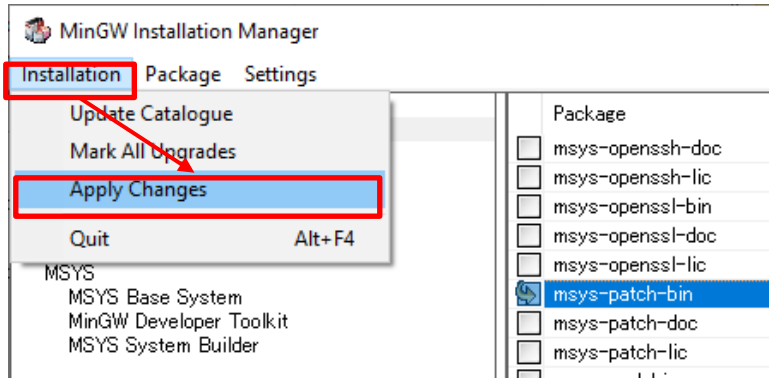


Figure 9.8 MinGW Installation Manager (3)

6. “Schedule of Pending Actions” window is displayed, click “Apply” button.

7. If “All changes were applied successfully; you may now clone this dialogue.” is displayed, Installing patch.exe is succeeded.

8. Add the path to the installed patch.exe to the system environment variables.
   For example, add the following path in the case of default path.

   “C:\MinGW\msys\1.0\bin”

   After updating the system environment variables, restart your computer to apply the changes.

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

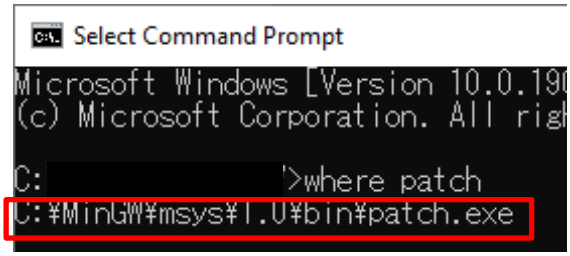


Figure 9.9 Confirm patch.exe (2)

9.3 Appendix C : How to Convert FSP Configuration from RZ/T2M to RZ/T2ME

➢ Open the FSP configuration and the [**BSP**] tab.

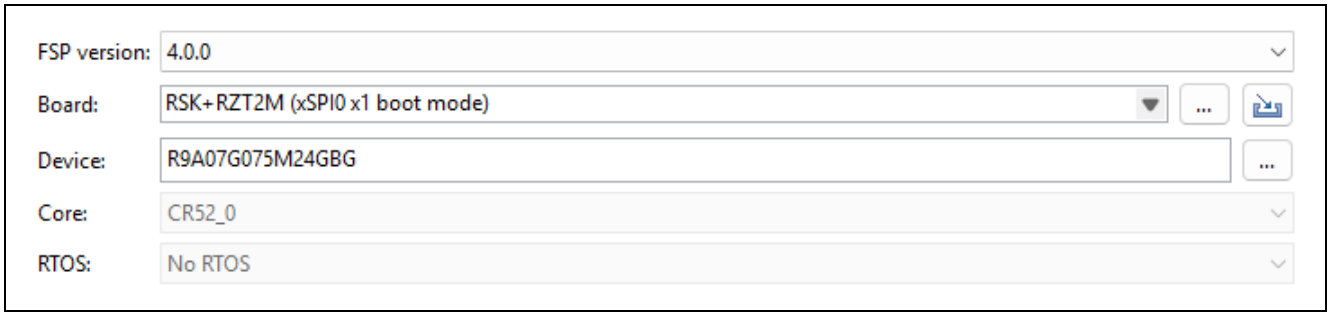


Figure 9.10 Open the FSP configuration

➢ Change [Board] from [RSK+RZT2M (xSPI0 x1 boot mode)] to [RSK+RZT2ME (xSPI0 x1 boot mode)].

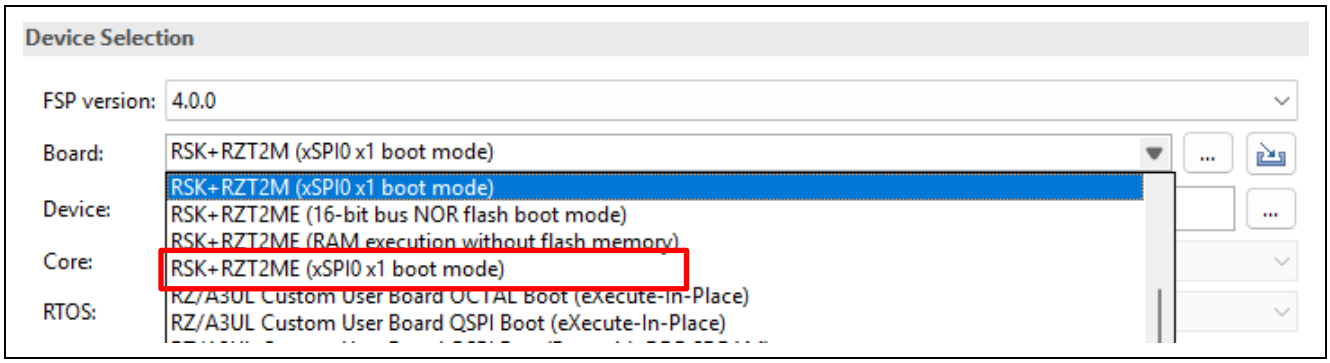


Figure 9.11 Change the board

➢ Open the [**Clocks**] tab. Change [CPU0CLK Mul x1] to [CPU0CLK Mul x4] and [CPU1CLK Mul x1] to [CPU1CLK Mul x4], and [XSPI_CLK0 12.5MHz] to [XSPI_CLK0 100MHz].

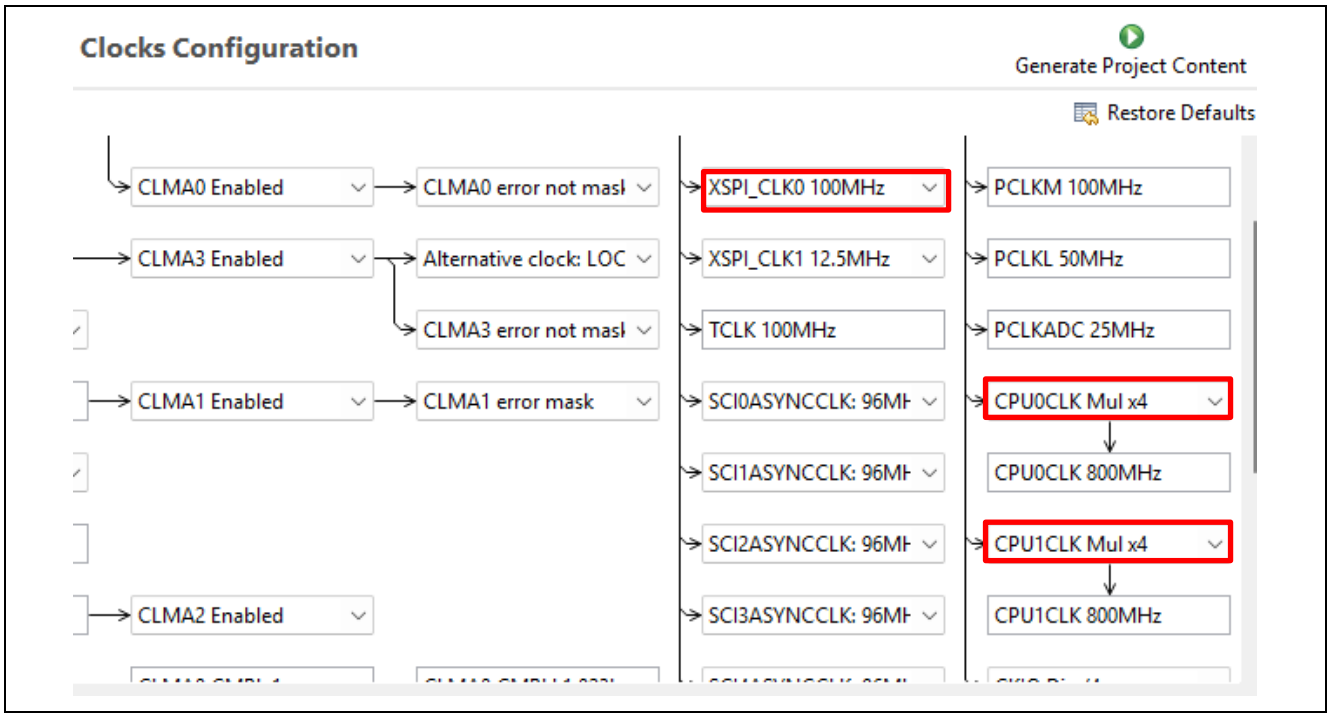


Figure 9.12 Change the clocks

➢ Click the [**Generate Project Content**] button to generate the file for **RZ/T2ME-RSK.**

9.4 Appendix D : ESC MDIO Type setting

It can change the settings from the smart configurator to change the PHY settings and the number of ports.

This appendix shows how to change the PHY management interface from GMAC to ESC.
Initially, it was set to GMAC.

1. Change PHY settings.
   Select each ether_phy stack. [Select MDIO type] → [ESC]

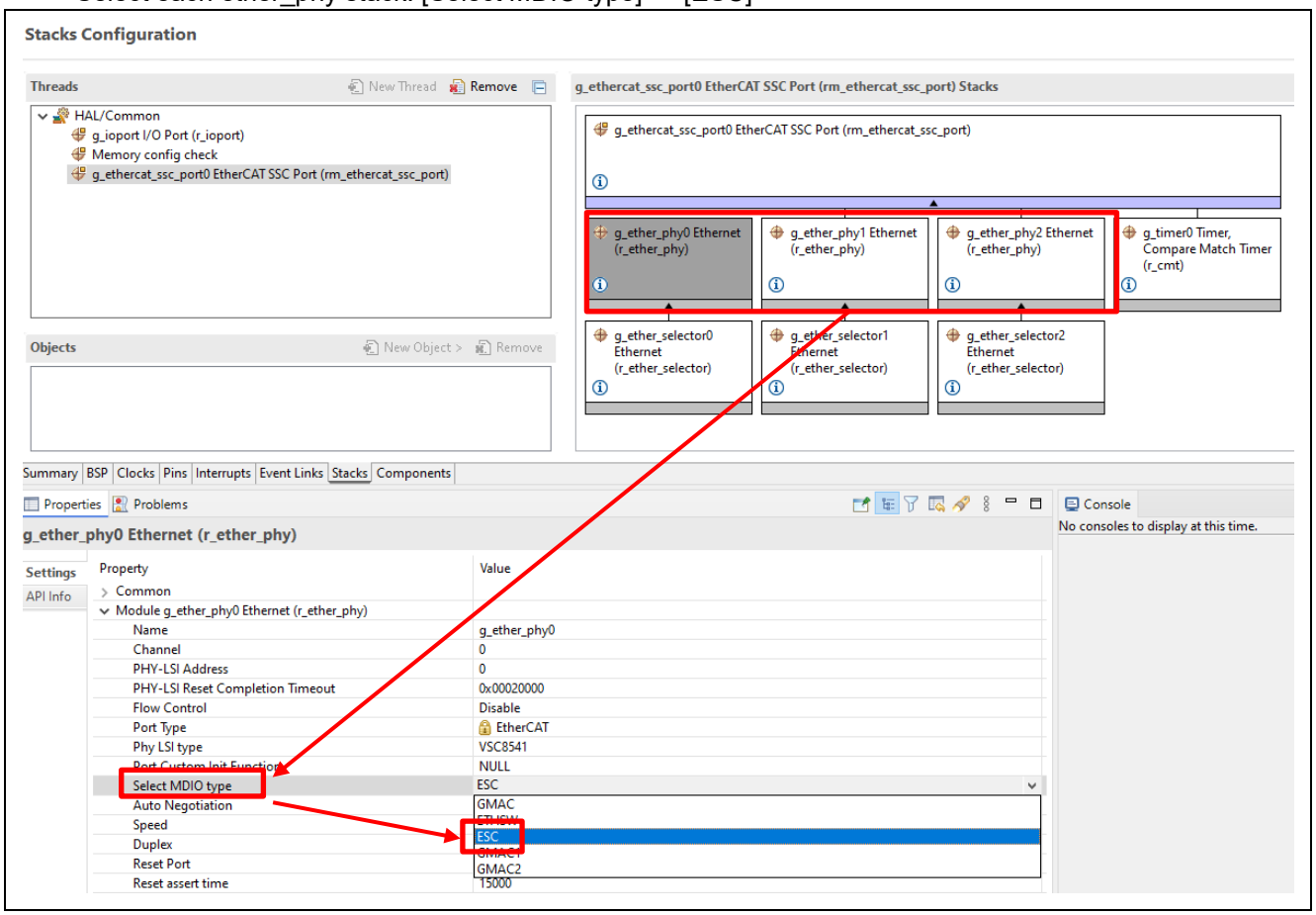


Figure 9.13 Change the MDIO type (1)

2. Change PHY management interface.
   Change ETHER_GMAC0 setting to Disabled.
   - GMAC0_MDC: None
   - GMAC0_MDIO: None

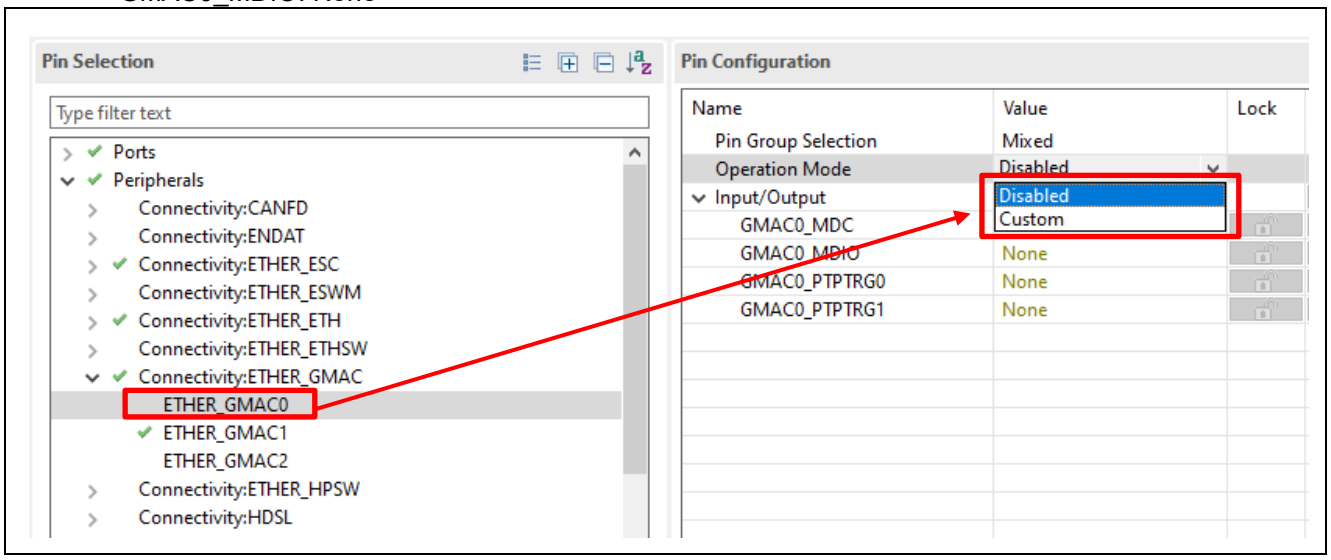


Figure 9.14 Change the MDIO type (2)

Change ETHER_ESC setting.

- In the case of RZ/T2M, RZ/T2ME, RZ/T2L, RZ/N2L RSK
  - ESC_MDC: P08_7
  - ESC_MDIO: P09_0
- In the case of RZ/T2H, RZ/N2H EVB
  - ESC_MDC: P21_4
  - ESC_MDIO: P21_5

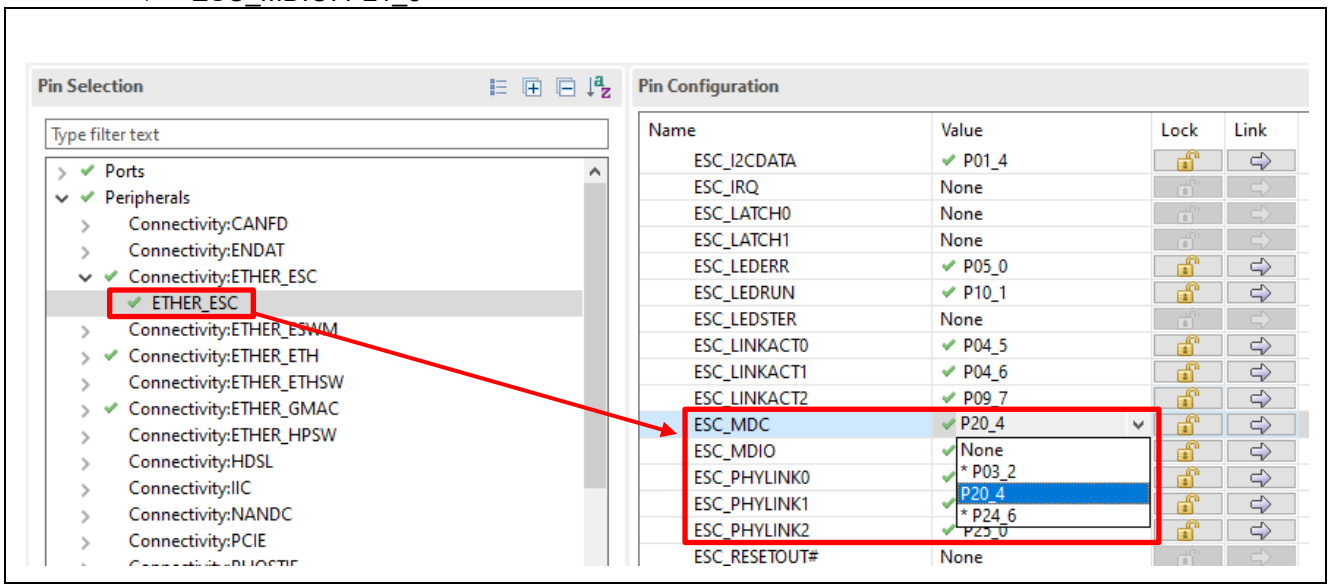


Figure 9.15 Change the MDIO type (3)

3. After completing all the settings, generate the code with "Generate Project Content".

9.5 Appendix E : EEPROM capacity setting change

Use the smart configurator to change the EEPROM capacity size and generate the code.

According to the EtherCAT specification, it is necessary to change the setting with 32 Kbit as the boundary.

1. Select SSC_port stack. [Properties] → [EEPROM Size]

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

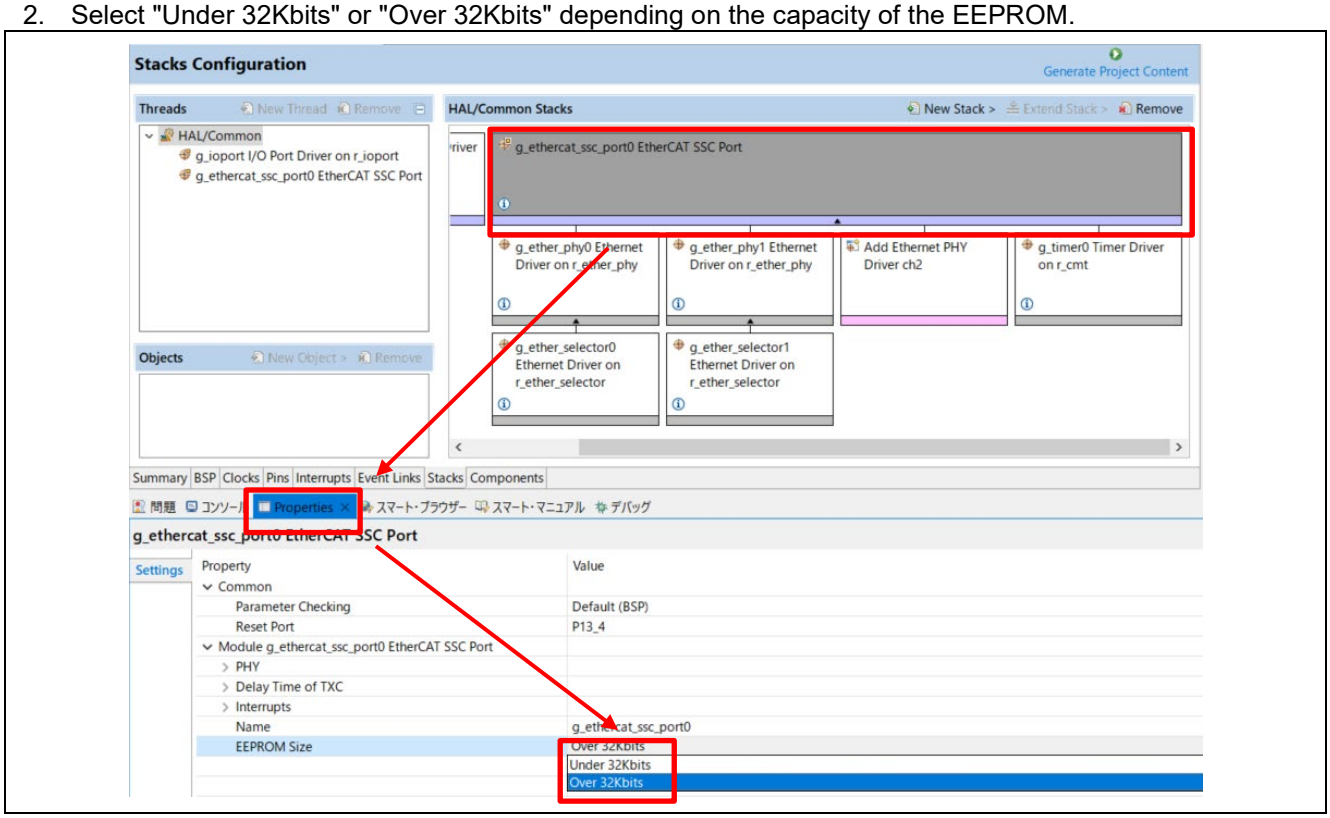


Figure 9.16 Change the EEPROM Size

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

9.6 Appendix F : Semiconductor Device Profile [ETG.5003]

9.6.1 Common Device Profile [ETG.5003.1]

In the case of 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 flamework of object dictionary definition. Separately consider and implement the settings and necessary processing.

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

9.6.2 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 Semi Test Record ETG.7000.2-Annex5003-0001.

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

The procedures in this chapter use **RZ/N2L-RSK** as the representative example.

9.6.2.1 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 be failure.
Stop debugging, push the reset button on RSK board to reset RZ/N2L.

- How to check

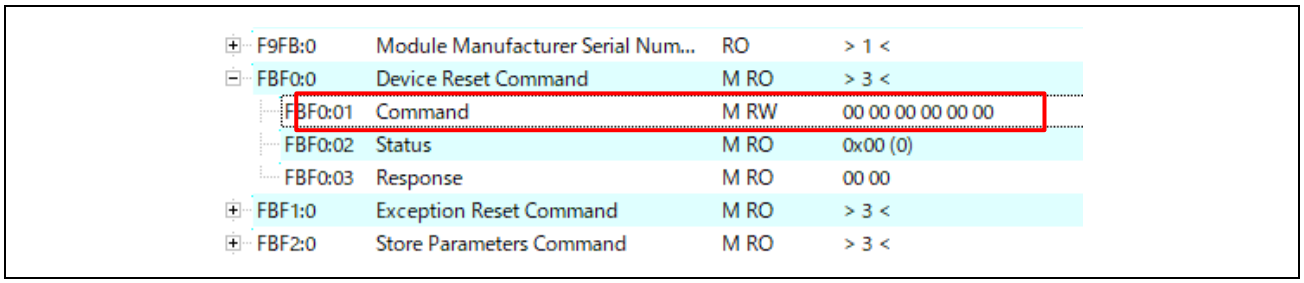
(1) Connect the serial port of the evaluation board to the serial port of the PC and start terminal software such as Tera Term on the PC. In the serial settings of the terminal, set 115200 bps, 8-bit data, no parity, 1 stop bit, no flow control.

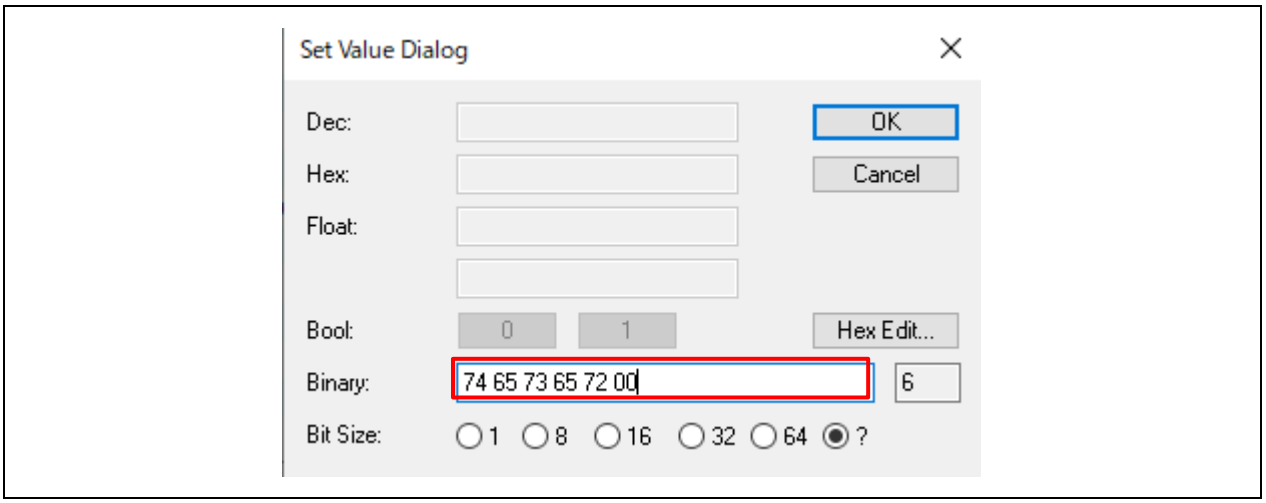
(2) Perform “6. Connecting to TwinCAT3,” Connect the evaluation board to TwinCAT.

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

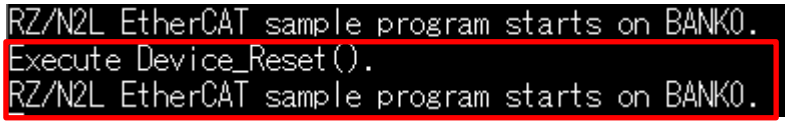
(4) Select the "CoE - Online" tab, double-click on Index FBF0:01 "Command"

and write “74 65 73 65 72 00” in “binary” to reset RZ/N2L.





(5) If the “Execute Device_Reset().” and “RZ/N2L EtherCAT sample program starts on BANKx.” output by serial communication are displayed in terminal software such as Tera Term, the evaluation board has been successfully restarted, Device Reset Command is no problem.



9.6.2.2 Dynamic PDO

In this sample program, the settings related to PDO in the ESI file are shown in Table 9.2.

Table 9.2 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 9-3.

Table 9-3 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) Perform “6. Connecting to TwinCAT3,” Connect the evaluation board to TwinCAT.

(2) Select the “Slots” tab, confirm the current slot.

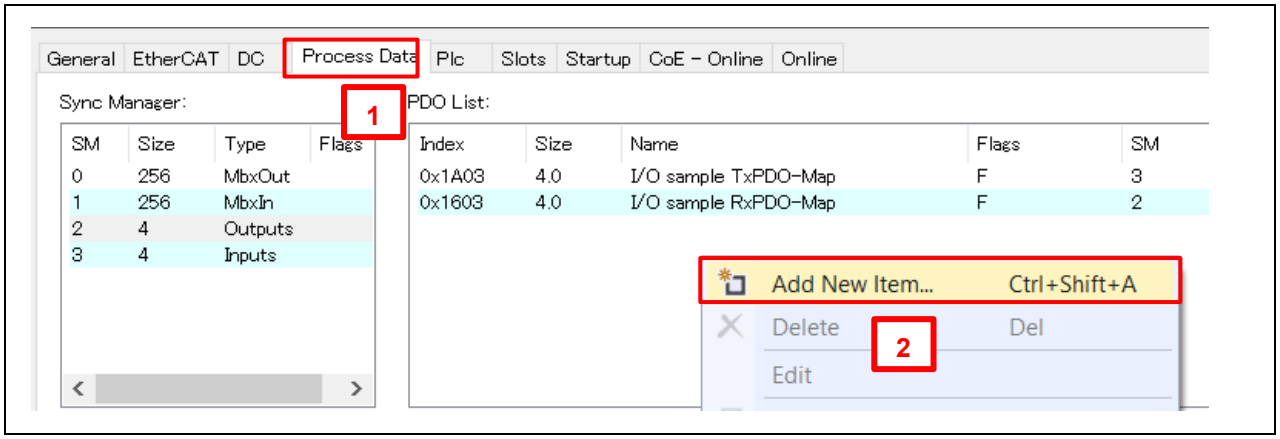
Select something in “Axis 0” of “Slot”of the left window, add it or change to it.

Also, “Axis 1” must be empty.

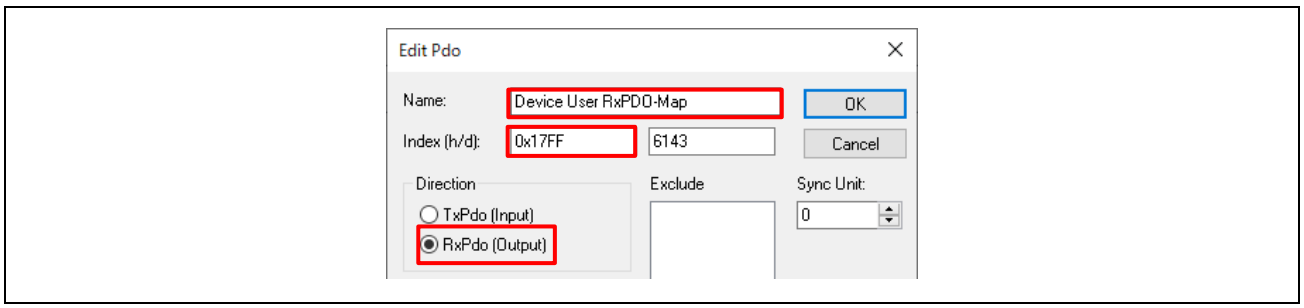
When you have added a module to "Slot" or changed it, restart TwinCAT by pressing [TwinCAT] → [Restart TwinCAT (Config Mode)] in the upper menu bar.

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

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



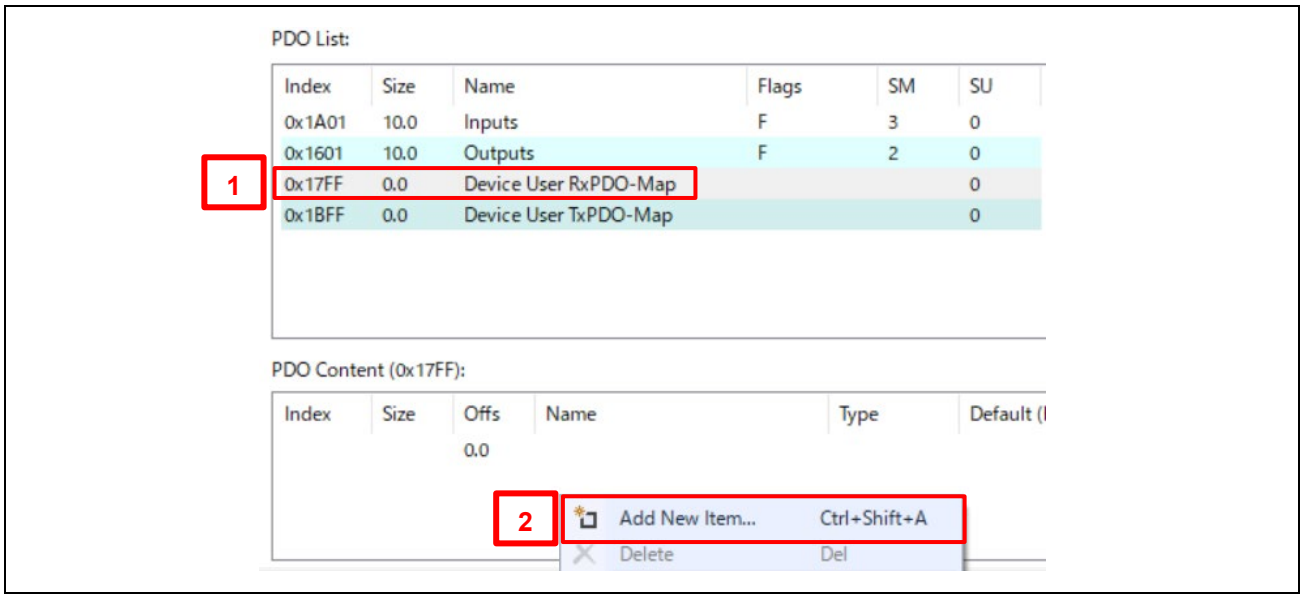
(5) In the "Edit Pdo" window, set the name to "Device User RxPDO-Map", the Index to "0x17FF", the Direction to "RxPdo" and press "OK".



(6) 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".

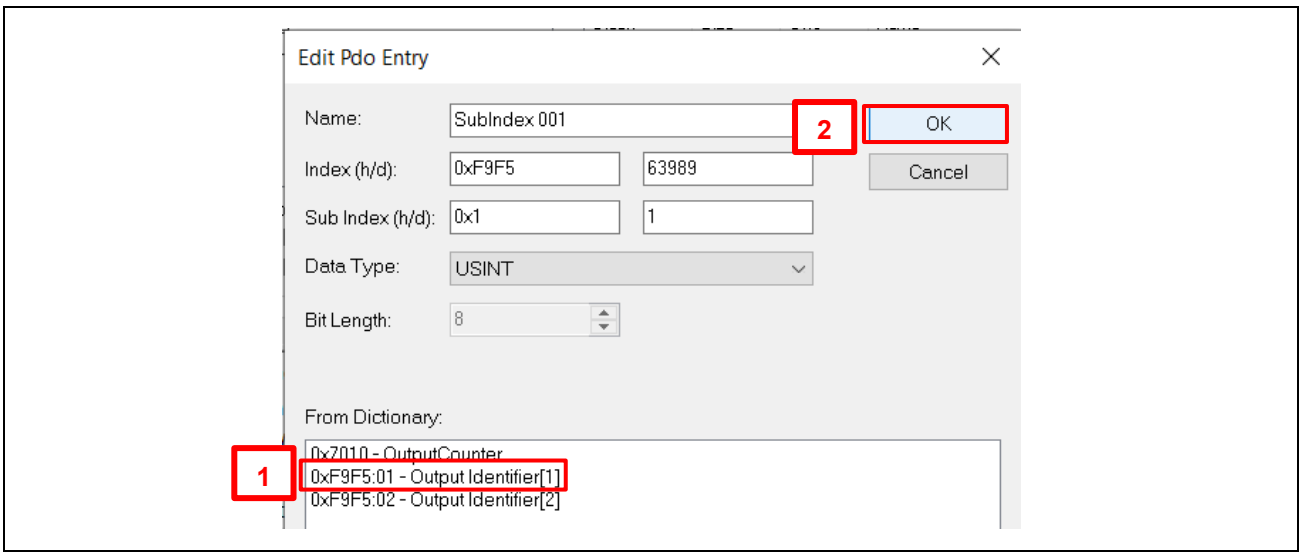
"Device User RxPDO-Map" and "Device User TxPDO-Map" have now been added to the "PDO List".

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



(8) 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.

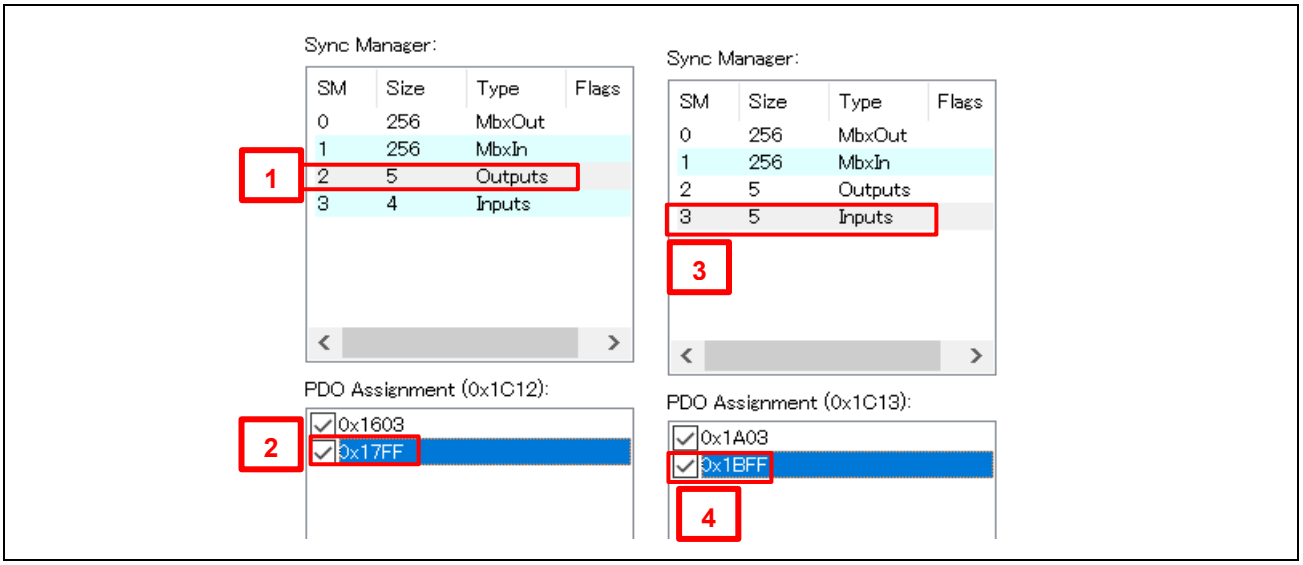


(9) Similarly, click on "Device User TxPDO-Map" in the "PDO List", right-click on the "PDO Content (0x1BFF):" area, select "Add New Item...".

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.

(10) 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.

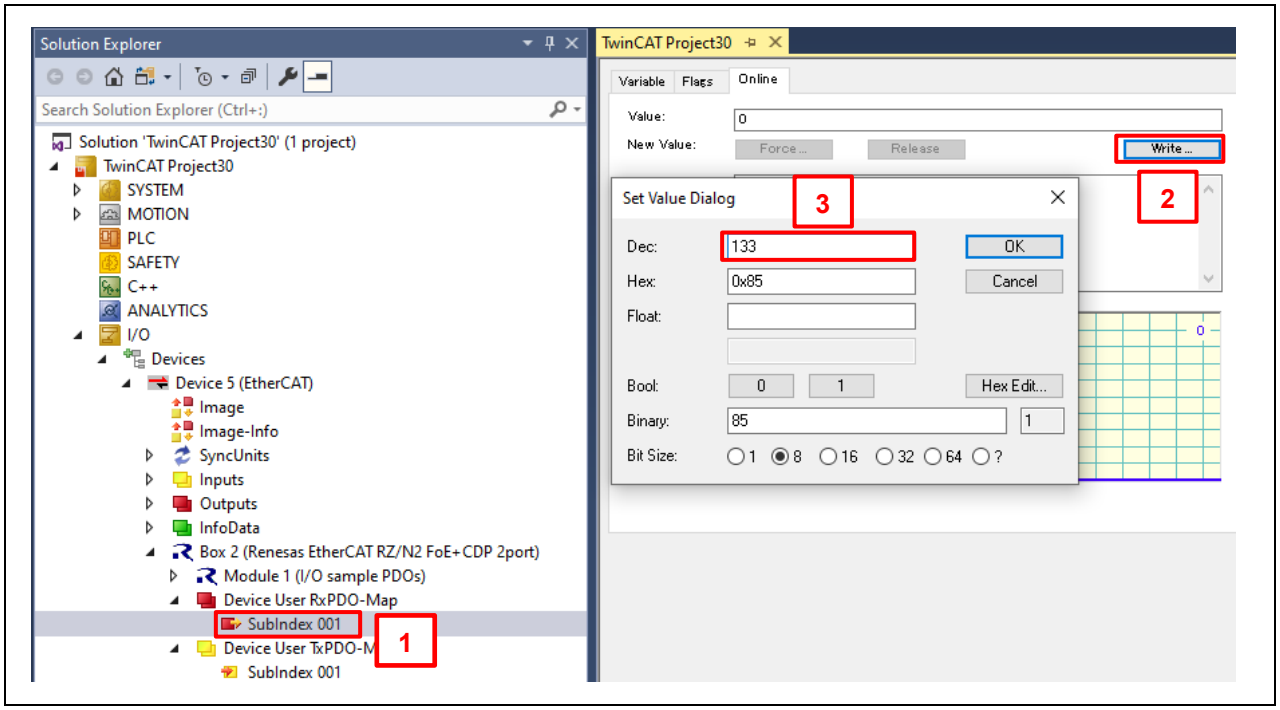
(11) Similarly, in the "Sync Manager" area, click on the row with the "SM" column of 3 and check the 0x1BFF in the "PDO Assignment (0x1C13)" area. You have now assigned an additional 0x17FF to the Index 0x1C12 and an additional 0x1BFF to the 0x1C13.



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

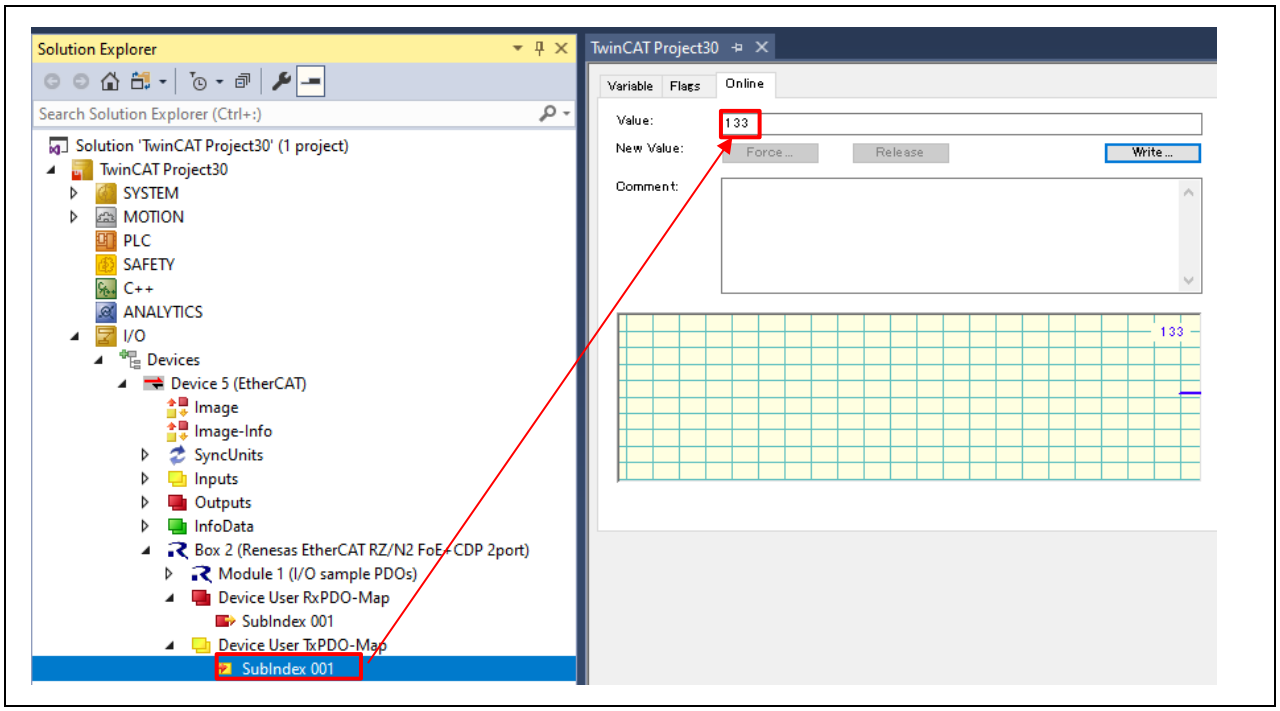
(13) Expand [Device User RxPDO-Map] in [Box 1] and click [SubIndex 001].

(14) Write an arbitrary value of 1~255 in Value.



(15) Expand Device User TxPDO-Map and click SubIndex 001.

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



9.6.2.3 Store Parameters

In this sample program, when the object value with the backup flag is changed by SDO communication, its value will be stored in the non-volatile memory of the evaluation board.

Table 9.4 shows the objects with the backup flag in this sample program.

Table 9.4 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 9.5 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 9.5 Non-volatile Memory Areas for Backup Objects

First address	Last address	Capacity
0x60F0_0000	0x60F0_1000	4KB

Table 9.6 shows the constants used by programs that store in non-volatile memory.

Table 9.6 Constants used in programs storing non-volatile memory (samplesemi.h)

Constant Name	Setting Values	Details
BACKUP_MEMORY_START_ADDRESS	Table 9.5 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 serial port of the evaluation board to the serial port of the PC and start terminal software such as Tera Term on the PC. In the serial settings of the terminal, set 115200 bps, 8-bit data, no parity, 1 stop bit, no flow control.

(2) Connect the evaluation board to TwinCAT 3 by following “5. Connecting to TwinCAT3.”

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

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

(5) If the writing is successful, “Index 0xF3A1 backup is success.” will be displayed in terminal software such as Tera Term.

(6) Reboot the RZ/N2L.

(7) 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.

10. Limitations

1. **When executing the EtherCAT sample program for the first time, be sure to set GMAC_MDC and GMAC_MDIO in Smart Configurator.**
   Write the EEPROM data with this configuration. It is not possible to set ESC_MDC and ESC_MDIO without writing the EEPROM data.

Revision History

Rev.	Date	Description	
		Page	Summary
4.00	Apr. 13, 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/.