

RZ/T2, RZ/N2 Group

Quick Start Guide: EtherCAT AoE SubDevice Software

Introduction

This application note describes how to set up the sample program for EtherCAT® SubDevice functions that use the EtherCAT stack code adapted for the RZ/T2 and RZ/N2 series microprocessors.

This application note also describes how to confirm the SubDevice behavior and stack features using the TwinCAT® MainDevice configuration tool.

Target Device

RZ/T series: RZ/T2M, RZ/T2ME, RZ/T2L, RZ/T2H, RZ/T2N

RZ/N series: RZ/N2L, RZ/N2H

Contents

1. Overview	3
1.1 Abbreviations and Definitions	3
1.2 Reference	3
1.2.1 About RZ/T2, RZ/N2.....	3
2. Features	5
2.1 Folder Structure of the Sample Package	6
3. Requirements (Software and Hardware).....	10
3.1 Requirements for RZ/T2M, RZ/T2ME, RZ/T2L, RZ/T2H, RZ/N2L, RZ/N2H	10
3.2 Requirements for RZ/T2N	11
4. Hardware Setup.....	12
4.1 RZ/T2M and RZ/T2ME RSK Board	12
4.1.1 Jumper and Switch Configuration	13
4.1.2 Board Setup.....	15
4.2 RZ/N2L RSK Board	16
4.2.1 Jumper and Switch Configuration	17
4.2.2 Board Setup.....	19
4.3 RZ/T2L RSK Board.....	20
4.3.1 Jumper and Switch Configuration	21
4.3.2 Board Setup.....	23
4.4 RZ/T2H Evaluation Board	24
4.4.1 Jumper and Switch Configuration	25
4.4.2 Board Setup.....	29
4.5 RZ/N2H Evaluation Board	30
4.5.1 Jumper and Switch Configuration	31

4.5.2	Board Setup.....	36
4.6	RZ/T2N Evaluation Board	37
4.6.1	Jumper and Switch Configuration	38
4.6.2	Board Setup.....	41
5.	Setting up the Host	42
5.1	Setting up TwinCAT 3	42
5.1.1	Copying the ESI Files	42
5.1.2	Adding the Driver	43
6.	Running the Sample Application	44
6.1	Generating the SubDevice Stack Code	45
6.2	Debugging with EWARM	49
6.2.1	Single-Core Project (RZ/T2L, RZ/N2L)	49
6.2.2	Multi-Core Project (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/T2N, RZ/N2H)	52
6.3	Debugging with e ² studio	57
6.3.1	Single-Core Project (RZ/T2L, RZ/N2L)	57
6.3.2	Multi-Core Project (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/T2N, RZ/N2H)	60
7.	Demonstration of the Sample Application	66
7.1	Scanning I/O Devices	66
7.2	Updating EEPROM Data	68
7.3	Evaluating the I/O Controller	71
7.3.1	Output Verification	71
7.3.2	Input Verification	72
7.4	Operation Check of AoE Communication	73
7.4.1	Preparation	73
7.4.2	ADS Read.....	74
7.4.3	ADS Write.....	75
8.	Appendix	77
8.1	Appendix A: Precautions for Using the SSC Tool	77
8.2	Appendix B: How to Install the Patch Command	79
8.2.1	Using Git for Windows (64-bit)	79
8.2.2	Using MinGW Installation Manager	80
8.3	Appendix C: How to Convert the FSP Configuration from RZ/T2M to RZ/T2ME	82
8.4	Appendix D: ESC MDIO Type Setting	83
8.5	Appendix E: Changing the EEPROM Capacity Setting.....	85
9.	Limitations	86
	Revision History	87

1. Overview

This application note describes how to set up the sample program for EtherCAT SubDevice functions that use the EtherCAT stack code adapted for the RZ/T2 and RZ/N2 series microprocessors.

1.1 Abbreviations and Definitions

Table 1.1 Abbreviations and definitions

Index	Abbreviation	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
17	ADS	Automation Device Specification
18	AoE	ADS over EtherCAT

1.2 Reference

1.2.1 About RZ/T2, RZ/N2

Technical information about EtherCAT is available on the ETG member site. Information about RZ/T2 and RZ/N2 is available on the Renesas website.

Table 1.2 Common technical inputs for RZ/T2 and 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****

Table 1.7 Technical inputs for RZ/T2N

Document type	Description	Document title	Document No.
User's Manual	Describes the technical details of the RZ/T2N Evaluation Board hardware.	RZ/T2N Group RZ/T2N Evaluation Board Manual	r20ut5574ej****
User's Manual	Provides technical details of the RZ/T2N microprocessors.	RZ/T2N Group Hardware Manual	r01uh1084eg****

2. Features

This package includes the EtherCAT SubDevice stack firmware that the SSC Tool generates for the 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)
 - AoE (ADS over EtherCAT)
- Synchronization modes:
 - Free Run
 - Sync Manager Synchronization
 - DC Synchronization
- I/O functions:
 - Push switch input
 - LED output



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

2.1 Folder Structure of the Sample Package

When you extract the package, you obtain the following folder structure.

Table 2.1 Sample package overview

Item	Description
r01an8274xx0420-rzt2-rzn2-ethercat-package	RZ/T2, RZ/N2 EtherCAT sample package
├── RZT2M_RZT2ME_EtherCAT_RSK_rev0411.zip	Archive for RZ/T2M, RZ/T2ME (RSK board)
├── RZT2L_EtherCAT_RSK_rev0411.zip	Archive for RZ/T2L (RSK board)
├── RZT2H_EtherCAT_EVB_rev0411.zip	Archive for RZ/T2H (EVB board)
├── RZT2N_EtherCAT_EVB_rev0400.zip	Archive for RZ/T2N (EVB board)
├── RZN2L_EtherCAT_RSK_rev0411.zip	Archive for RZ/N2L (RSK board)
├── RZN2H_EtherCAT_EVB_rev0411.zip	Archive for RZ/N2H (EVB board)
└── r01an8376ej0410-rzt2-rzn2-ecat-aoe.pdf	This application note

When you extract the archive for a multi-core device, you obtain the following folder structure.

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

Item	Description
RZT2M_RZT2ME_EtherCAT_RSK_rev0411	RZ/T2M, RZ/T2ME RSK folder
├── CR52_Dual	Cortex-R52 CPU0 and Cortex-R52 CPU1
│ └── common	Common resources for SSC Tool
│ │ └── AoE	For AoE
│ │ │ └── ESI	EtherCAT SubDevice Information
│ │ └── SSCconfig	SSC Tool setting file
└── project	IDE project folder
└── AoE	For AoE
└── e2studio	The e ² studio projects
├── RZT2M_RSK_ESC_AoE_CR52_Dual_Primary	Primary core project
└── RZT2M_RSK_ESC_AoE_CR52_Dual_Secondary	Secondary core project
└── ewarm	The EWARM projects
├── RZT2M_RSK_ESC_AoE_CR52_Dual_Primary	Primary core project
└── RZT2M_RSK_ESC_AoE_CR52_Dual_Secondary	Secondary core project

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

Item	Description
RZT2H_EtherCAT_EVB_rev0411	RZ/T2H EVB folder
├── CR52_Dual	Cortex-R52 CPU0 and Cortex-R52 CPU1
│ └── common	Common resources for SSC Tool
│ │ └── AoE	For AoE
│ │ │ └── ESI	EtherCAT SubDevice Information
│ │ └── SSCconfig	SSC Tool setting file
└── project	IDE project folder
└── AoE	For AoE
└── e2studio	The e ² studio projects
├── RZT2H_EVB_ESC_AoE_CR52_Dual_Primary	Primary core project
└── RZT2H_EVB_ESC_AoE_CR52_Dual_Secondary	Secondary core project
└── ewarm	The EWARM projects
├── RZT2H_EVB_ESC_AoE_CR52_Dual_Primary	Primary core project
└── RZT2H_EVB_ESC_AoE_CR52_Dual_Secondary	Secondary core project
└── CR52_CA55	Cortex-R52 CPU0 and Cortex-A55 Core2
└── common	Common resources for SSC Tool
└── AoE	For AoE
└── ESI	EtherCAT SubDevice Information

	SSCconfig	SSC Tool setting file
	project	IDE project folder
	AoE	For AoE
	e2studio	The e ² studio projects
	RZT2H_EVB_ESC_AoE_CR52_CA55_Primary	Primary core project
	RZT2H_EVB_ESC_AoE_CR52_CA55_Secondary	Secondary core project
	ewarm	The EWARM project
	RZT2H_EVB_ESC_AoE_CR52_CA55_Primary	Primary core project
	RZT2H_EVB_ESC_AoE_CR52_CA55_Secondary	Secondary core project

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

Item	Description
RZN2H_EtherCAT_EVB_rev0411	RZ/N2H EVB folder
CR52_Dual	Cortex-R52 CPU0 and Cortex-R52 CPU1
common	Common resources for SSC Tool
AoE	For AoE
ESI	EtherCAT SubDevice Information
SSCconfig	SSC Tool setting file
project	IDE project folder
AoE	For AoE
e2studio	The e ² studio projects
RZN2H_EVB_ESC_AoE_CR52_Dual_Primary	Primary core project
RZN2H_EVB_ESC_AoE_CR52_Dual_Secondary	Secondary core project
ewarm	The EWARM projects
RZN2H_EVB_ESC_AoE_CR52_Dual_Primary	Primary core project
RZN2H_EVB_ESC_AoE_CR52_Dual_Secondary	Secondary core project
CR52_CA55	Cortex-R52 CPU0 and Cortex-A55 Core2
common	Common resources for SSC Tool
AoE	For AoE
ESI	EtherCAT SubDevice Information
SSCconfig	SSC Tool setting file
project	IDE project folder
AoE	For AoE
e2studio	The e ² studio projects
RZN2H_EVB_ESC_AoE_CR52_CA55_Primary	Primary core project
RZN2H_EVB_ESC_AoE_CR52_CA55_Secondary	Secondary core project
ewarm	The EWARM projects
RZN2H_EVB_ESC_AoE_CR52_CA55_Primary	Primary core project
RZN2H_EVB_ESC_AoE_CR52_CA55_Secondary	Secondary core project

Table 2.5 Multi-core folder structure (RZ/T2N)

Item	Description
RZT2N_EtherCAT_EVB_rev0400	RZ/T2N EVB folder
├── CR52_Dual	Cortex-R52 CPU0 and Cortex-R52 CPU1
│ └── common	Common resources for SSC Tool
│ │ └── AoE	For AoE
│ │ └── ESI	EtherCAT SubDevice Information
│ │ └── SSCconfig	SSC Tool setting file
│ └── project	IDE project folder
│ │ └── AoE	For AoE
│ │ │ └── e2studio	The e ² studio projects
│ │ │ │ └── RZT2N_EVB_ESC_AoE_CR52_Dual_Primary	Primary core project
│ │ │ │ └── RZT2N_EVB_ESC_AoE_CR52_Dual_Secondary	Secondary core project
│ │ │ └── ewarm	The EWARM projects
│ │ │ │ └── RZT2N_EVB_ESC_AoE_CR52_Dual_Primary	Primary core project
│ │ │ │ └── RZT2N_EVB_ESC_AoE_CR52_Dual_Secondary	Secondary core project
└── CR52_CA55	Cortex-R52 CPU0 and Cortex-A55 Core1
│ └── common	Common resources for SSC Tool
│ │ └── AoE	For AoE
│ │ └── ESI	EtherCAT SubDevice Information
│ │ └── SSCconfig	SSC Tool setting file
│ └── project	IDE project folder
│ │ └── AoE	For AoE
│ │ │ └── e2studio	The e ² studio projects
│ │ │ │ └── RZT2N_EVB_ESC_AoE_CR52_CA55_Primary	Primary core project
│ │ │ │ └── RZT2N_EVB_ESC_AoE_CR52_CA55_Secondary	Secondary core project
│ │ │ └── ewarm	The EWARM project
│ │ │ │ └── RZT2N_EVB_ESC_AoE_CR52_CA55_Primary	Primary core project
│ │ │ │ └── RZT2N_EVB_ESC_AoE_CR52_CA55_Secondary	Secondary core project

When you extract the archive for a single-core device, you obtain the following folder structure.

Table 2.6 Single-core folder structure (RZ/T2L)

Item	Description
RZT2L_EtherCAT_RSK_rev0411	RZ/T2L RSK folder
├── common	Common resources for SSC Tool
│ ├── AoE	For AoE
│ │ ├── ESI	EtherCAT SubDevice Information
│ │ └── SSCconfig	SSC Tool setting file
└── project	IDE project folder
├── AoE	For AoE
├── e2studio	The e ² studio projects
│ └── RZT2L_RSK_ESC_AoE	Single-core project
└── ewarm	The EWARM projects
└── RZT2L_RSK_ESC_AoE	Single-core project

Table 2.7 Single-core folder structure (RZ/N2L)

Item	Description
RZN2L_EtherCAT_RSK_rev0411	RZ/N2L RSK folder
├── common	Common resources for SSC Tool
│ ├── AoE	For AoE
│ │ ├── ESI	EtherCAT SubDevice Information
│ │ └── SSCconfig	SSC Tool setting file
└── project	IDE project folder
├── AoE	For AoE
├── e2studio	The e ² studio projects
│ └── RZN2L_RSK_ESC_AoE	Single-core project
└── ewarm	The EWARM projects
└── RZN2L_RSK_ESC_AoE	Single-core project

3. Requirements (Software and Hardware)

Renesas developed and tested this sample program with the following boards and tools.

3.1 Requirements for RZ/T2M, RZ/T2ME, RZ/T2L, RZ/T2H, RZ/N2L, RZ/N2H

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 The patch for RZ/T2H and RZ/N2H (EWARM_Patch_for_RZT2H_N2H_rev1.0.zip) is required.
	e² 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
	TwinCAT 3	4026.19	Beckhoff Automation TwinCAT 3.1 Build 4026 Beckhoff Worldwide

3.2 Requirements for RZ/T2N

For RZ/T2N, the following table lists only the items that differ from the requirements for the other devices. The other items are the same.

Table 3.2 Requirements for RZ/T2N

Category	Name	Version	Description
Board	RZ/T2N Evaluation Board	—	Renesas
IDE	EWARM	9.60.3	IAR Systems IAR Embedded Workbench for Arm IAR The patch for RZ/T2N (EWARM_Patch_for_RZT2N_rev1.0.zip) is required.
	e² studio	2026-07	Renesas Releases · renesas/rz-fsp
Configurator	FSP Smart Configurator	2026-07	
Flexible Software Package	FSP	4.2.0	
Emulator	J-Link™	9.44	SEGGER SEGGER - The Embedded Experts - Downloads - J-Link / J-Trace

4. Hardware Setup

This chapter describes the main hardware settings. For more information about the boards, see the user manual and schematics for each board.

4.1 RZ/T2M and RZ/T2ME RSK Board

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

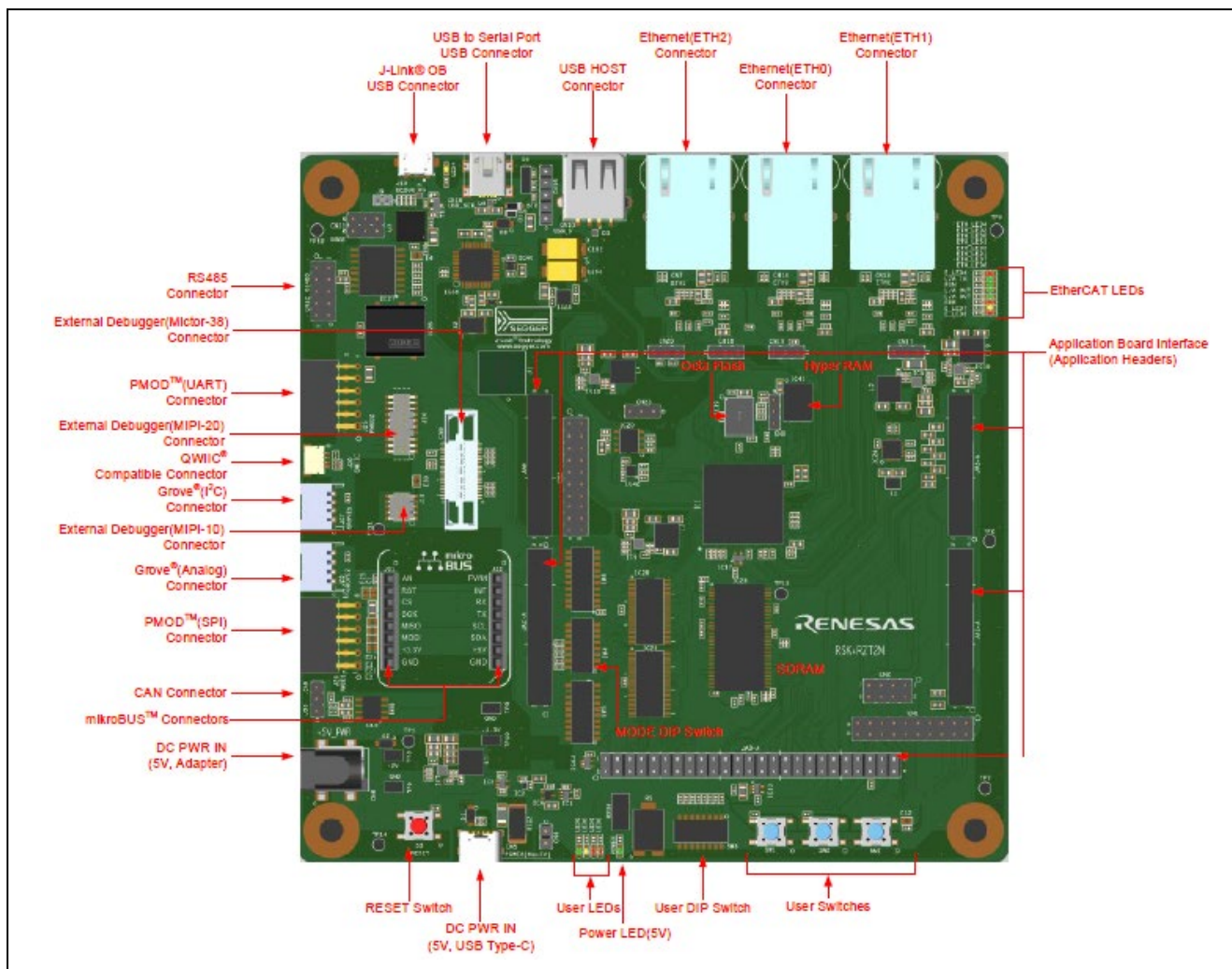


Figure 4.1 RZ/T2M RSK board layout

4.1.1 Jumper and Switch Configuration

The following tables show the jumper pin and switch settings.

Text in bold 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 1.8 V
CN18	Short 1-2	Use three ports in the same PHY mode
CN19	Short 1-2	Use three ports in the same PHY mode
CN20	Short 1-2	Use three ports in the 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 4.2 RZ/T2M-RSK SW4 settings

SW4	Setting	Description
SW4-1	ON	16-bit bus boot mode (NOR Flash)
SW4-2	OFF	See r20ut4939 and set the switches according to the mode to use.
SW4-3	ON	
SW4-4	ON	MDD=0, JTAG Authentication by Hash is disabled.
SW4-5	OFF	MDW=1, ATCM 1 wait. Set this when you use TCM at CPU operating frequencies above 400 MHz.
SW4-6	OFF	—
SW4-7	OFF	—
SW4-8	OFF	—

Table 4.3 RZ/T2M-RSK SW5 settings

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 settings

SW6	Setting	Description
SW6-1	OFF	Enable the ETH2_MDIO and ETH2_MDC signals.
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

This section describes how to set up the board to run the sample program.

To build and run the sample code on the RZ/T2M RSK board, follow these steps:

You can load the program into RAM or flash memory using IAR Embedded Workbench or e² studio.

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

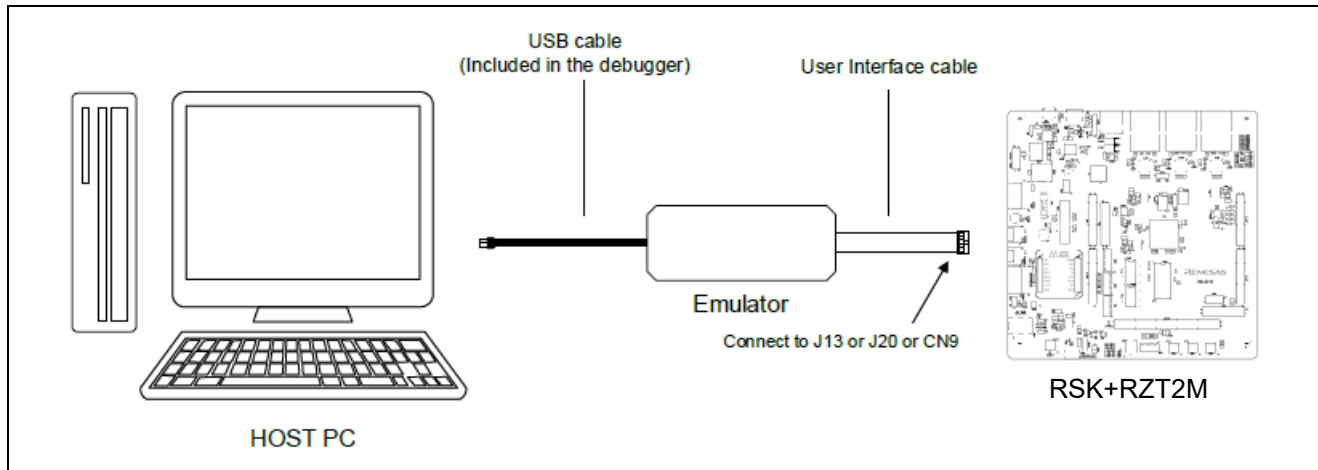


Figure 4.2 RZ/T2M RSK board debug connection diagram

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

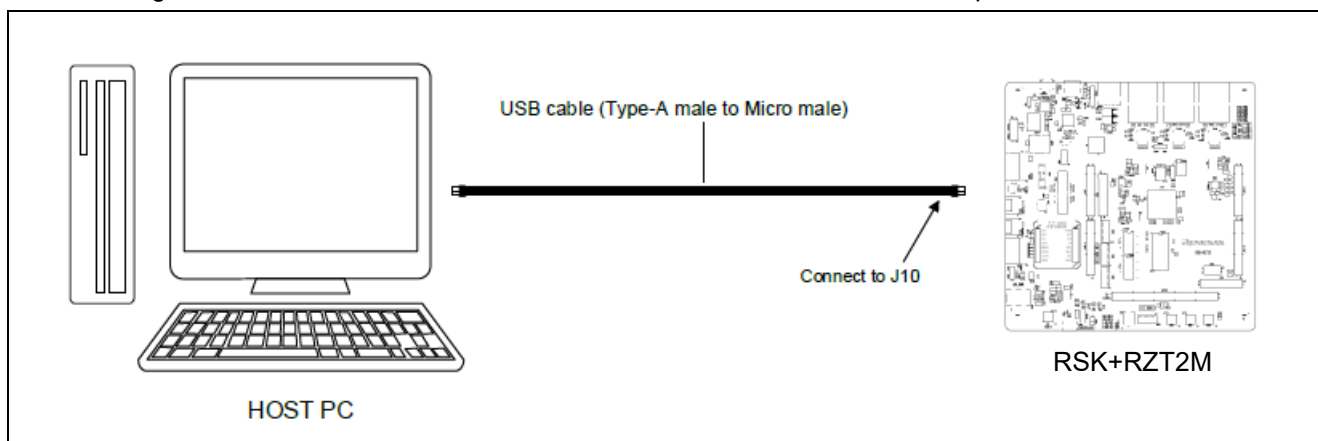


Figure 4.3 RZ/T2M RSK board debug connection diagram (J-Link OB)

2. Power is supplied through 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 using the AC/DC adapter, connect it to the CN6 connector of the RZ/T2M RSK board.
3. Connect the host PC and the RZ/T2M RSK board using an Ethernet cable. Connect the Ethernet cable to the ETH0 connector of the RZ/T2M RSK board. Depending on the protocol specifications, you might be able to connect the cable to ETH1 instead.

4.2 RZ/N2L RSK Board

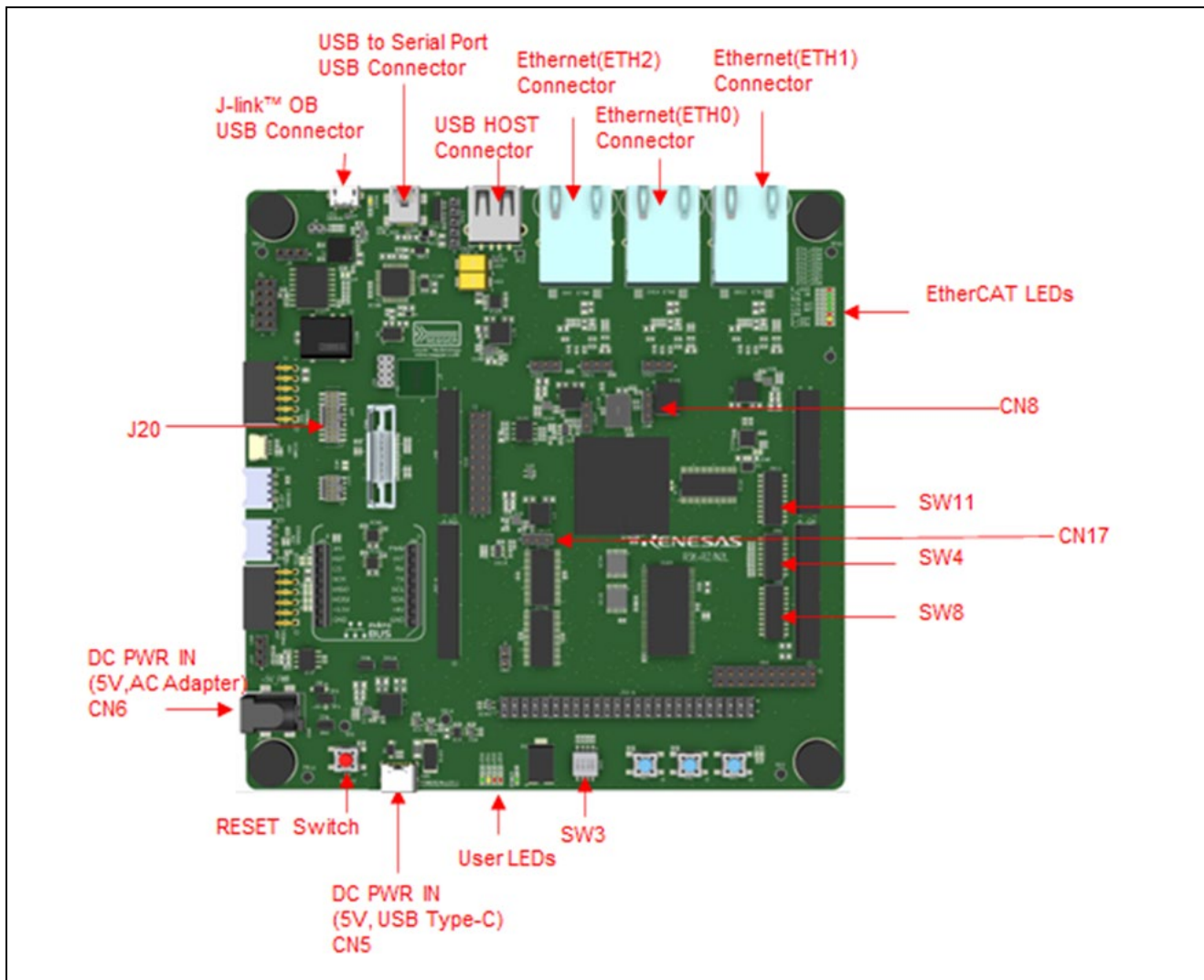


Figure 4.4 RZ/N2L RSK board layout

4.2.1 Jumper and Switch Configuration

The following tables show the jumper pin and switch settings.

Text in bold 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 1.8 V
CN20	Short 1-2	Use three ports in the same PHY mode
CN21	Short 1-2	Use three ports in the same PHY mode
CN22	Short 1-2	Use three ports in the same PHY mode
CN24	Short 2-3	VCC1833_3 Use power supply at 1.8 V
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	16-bit bus boot mode (NOR flash)
SW4-2	OFF	See r20ut4984egxxxx-rskplus-rzn2l-v1-um.pdf and set the switches according to the mode to use.
SW4-3	ON	MDD=0, JTAG Authentication by Hash is disabled.
SW4-4	ON	—
SW4-5	OFF	—
SW4-6	OFF	Enables signals other than the trace signal (for example, motor and RS485 signals).
SW4-7	ON	Enables signals other than the external bus signals (for example, CAN, emulator, and I2C signals).
SW4-8	OFF	Enable SW3.

Table 4.7 RZ/N2L-RSK SW8 settings

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 and M2_VN configuration switch setting
SW8-7	ON	
SW8-8	OFF	CAN_TX, IRQ4, and P02_2 configuration switch setting
SW8-9	OFF	
SW8-10	ON	

Table 4.8 RZ/N2L-RSK SW11 settings

SW11	Setting	Description
SW11-1	ON	Enable the LED_RED2 signal.
SW11-2	OFF	
SW11-3	OFF	
SW11-4	OFF	RS485_RX and M2_UP configuration switch setting
SW11-5	ON	
SW11-6	OFF	P21_5 and M2_VP configuration switch setting
SW11-7	ON	
SW11-8	OFF	CAN_RX, ADTRG, and P01_7 configuration switch setting
SW11-9	OFF	
SW11-10	ON	

4.2.2 Board Setup

This section describes how to set up the board to run the sample program.

To build and run the sample code on the RZ/N2L RSK board, follow these steps:

You can load the program into RAM or flash memory using IAR Embedded Workbench or e² studio.

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

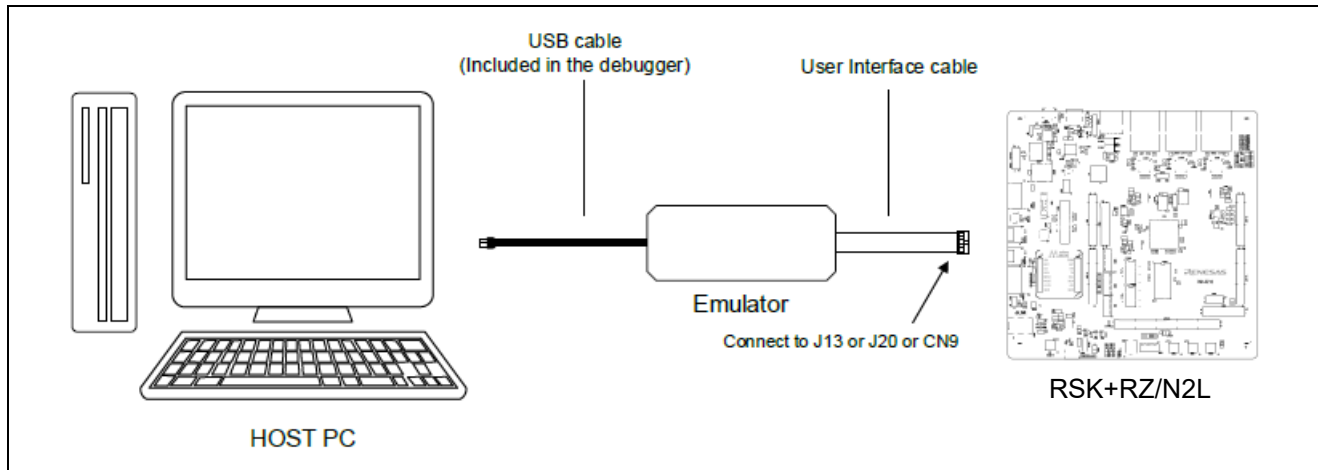


Figure 4.5 RZ/N2L RSK board debug connection diagram

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

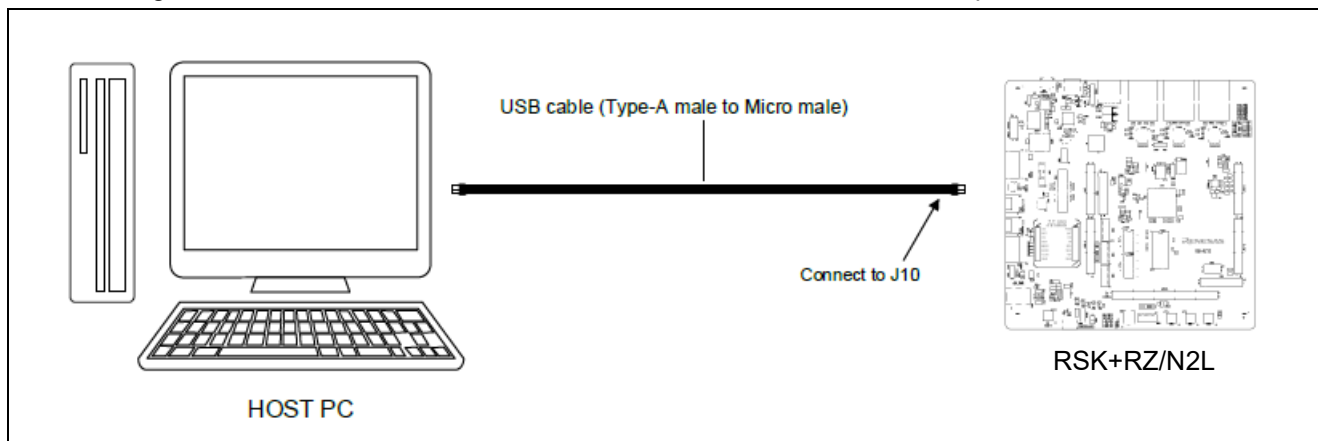


Figure 4.6 RZ/N2L RSK board debug connection diagram (J-Link OB)

2. Power is supplied through 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 using the AC/DC adapter, connect it to the CN6 connector of the RZ/N2L RSK board.
3. Connect the host PC and the RZ/N2L RSK board using an Ethernet cable. Connect the Ethernet cable to the ETH0 connector of the RZ/N2L RSK board. Depending on the protocol specifications, you might be able to connect the cable to ETH1 instead.

4.3 RZ/T2L RSK Board

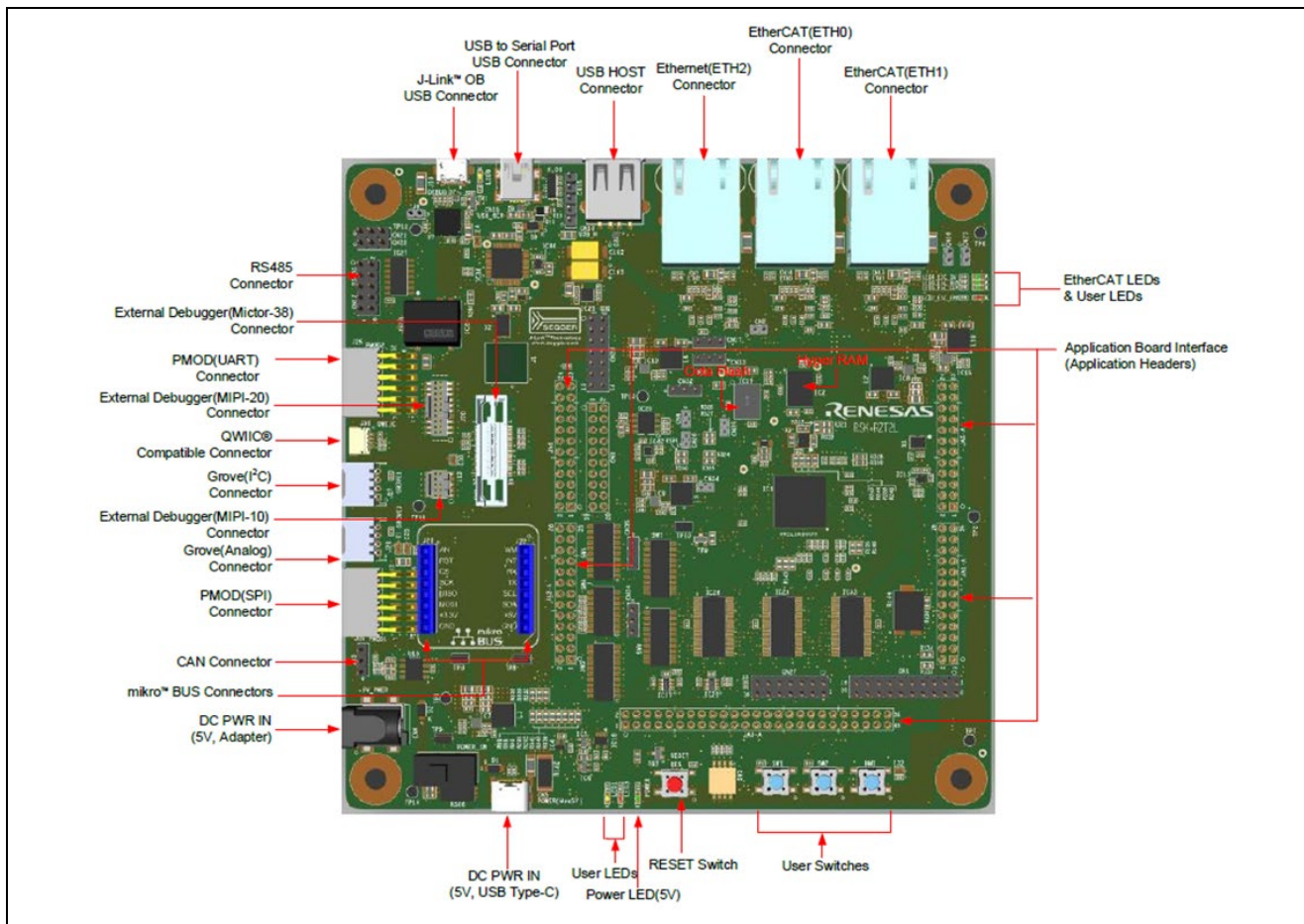


Figure 4.7 RZ/T2L RSK board layout

4.3.1 Jumper and Switch Configuration

The following tables show the jumper pin and switch settings.

Text in bold 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 1.8 V
CN32	Short 1-2	VCC1833_3 Use power supply at 3.3 V
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	xSPI0 x1 boot mode
SW4-2	ON	See r20ut5164ejxxxx-rskrzt2l.pdf and set the switches according to the mode to use.
SW4-3	ON	
SW4-4	OFF	MDD=0, JTAG Authentication by Hash is disabled.
SW4-5	ON	JTAG mode = Normal mode
SW4-6	ON	VCC1833_2 = 1.8 V
SW4-7	OFF	VCC1833_3 = 3.3 V
SW4-8	OFF	—

Table 4.11 RZ/T2L-RSK SW5 settings

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

Table 4.12 RZ/T2L-RSK SW6 settings

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 settings

SW7	Setting	Description
SW7-1	ON	Use P05_5 as ETH1_LINK for EtherCAT Port1
SW7-2	OFF	
SW7-3	OFF	Use P17_6 as LED1 for user LED control
SW7-4	OFF	
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 settings

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
SW8-5	OFF	
		(When SW8-10=OFF:TRACE_OPTION_SEL=H)
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

This section describes how to set up the board to run the sample program.

To build and run the sample code on the RZ/T2L RSK board, follow these steps:

You can load the program into RAM or flash memory using IAR Embedded Workbench or e² studio.

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

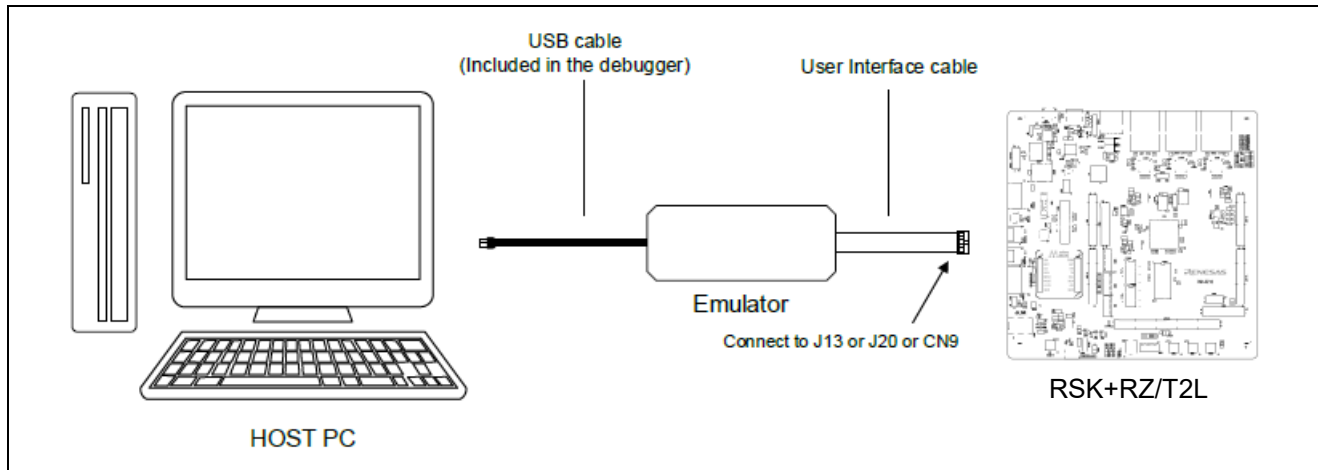


Figure 4.8 RZ/T2L RSK board debug connection diagram

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

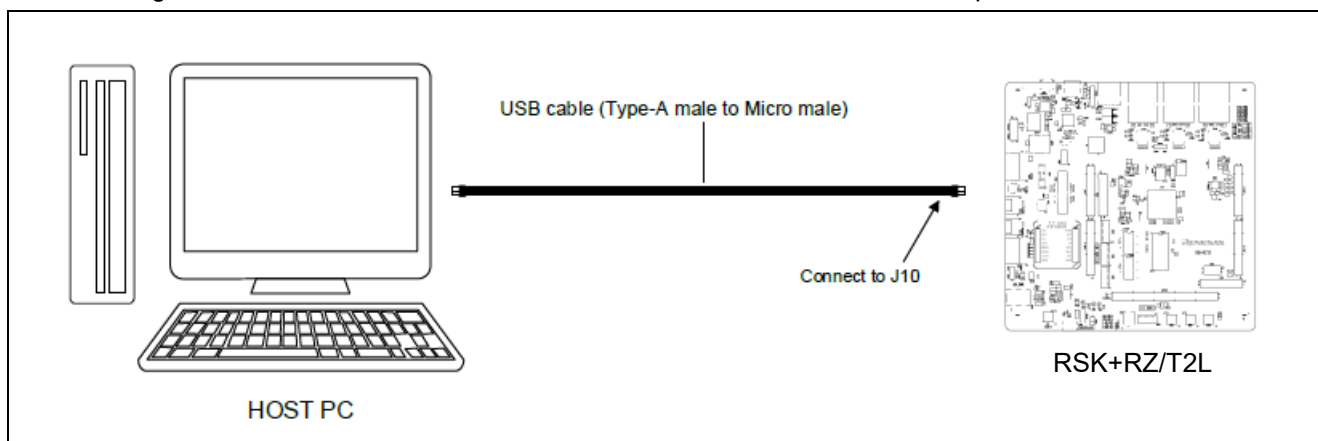


Figure 4.9 RZ/T2L RSK board debug connection diagram (J-Link OB)

2. Power is supplied through 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 using the AC/DC adapter, connect it to the connector CN6 of the RZ/T2L RSK board.
3. Connect the host PC and the RZ/T2L RSK board using an Ethernet cable. Connect the Ethernet cable to the ETH0 connector of the RZ/T2L RSK board.
Depending on the protocol specifications, you might be able to connect the cable to ETH1 instead.

4.4 RZ/T2H Evaluation Board

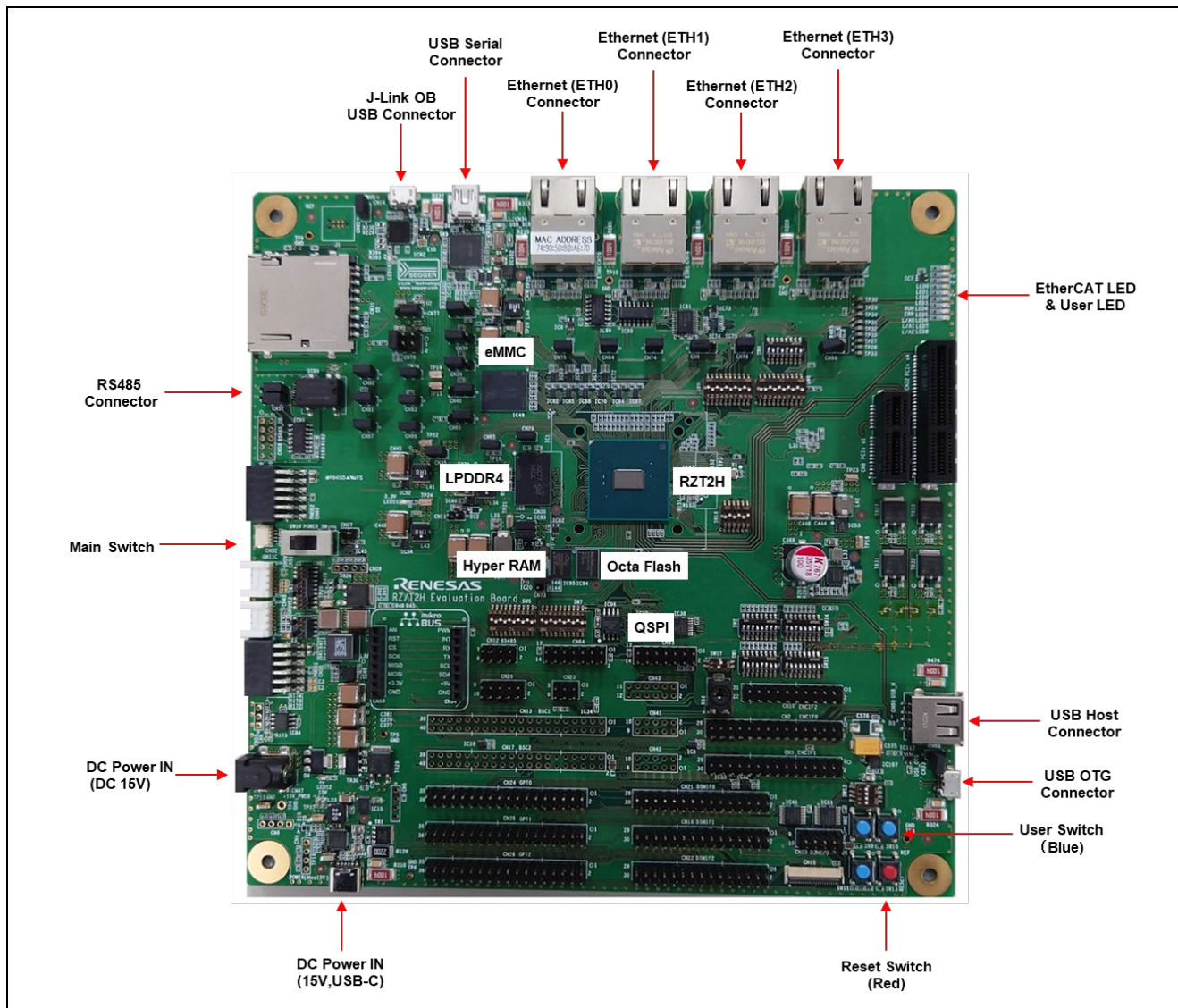


Figure 4.10 RZ/T2H evaluation board layout

4.4.1 Jumper and Switch Configuration

The following tables show the jumper pin and switch settings.

Text in bold 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	Short 2-3	Half-duplex communication
CN57		

Table 4.16 RZ/T2H EVB SW14 settings

SW14	Setting	Description
SW14-1	ON	xSPI1 boot mode (x1 boot serial flash)
SW14-2	OFF	See r20ut5405ejxxxx-rzt2hevb.pdf and set the switches according to the mode to use.
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	P23_4 is used as ESC_I2CDATA.
SW6-9	ON	
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	The PCIe function is used as a 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 the 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

This section describes how to set up the board to run the sample program.

To build and run the sample code on the RZ/T2H Evaluation board, follow these steps:

You can load the program into RAM or flash memory using IAR Embedded Workbench or e² studio.

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

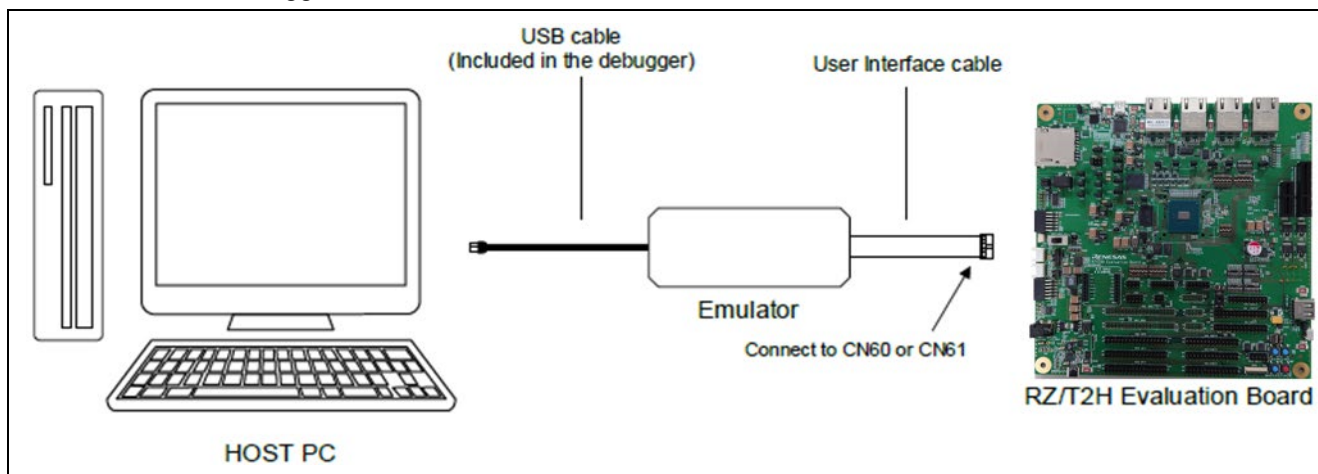


Figure 4.11 RZ/T2H evaluation board debug connection diagram

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

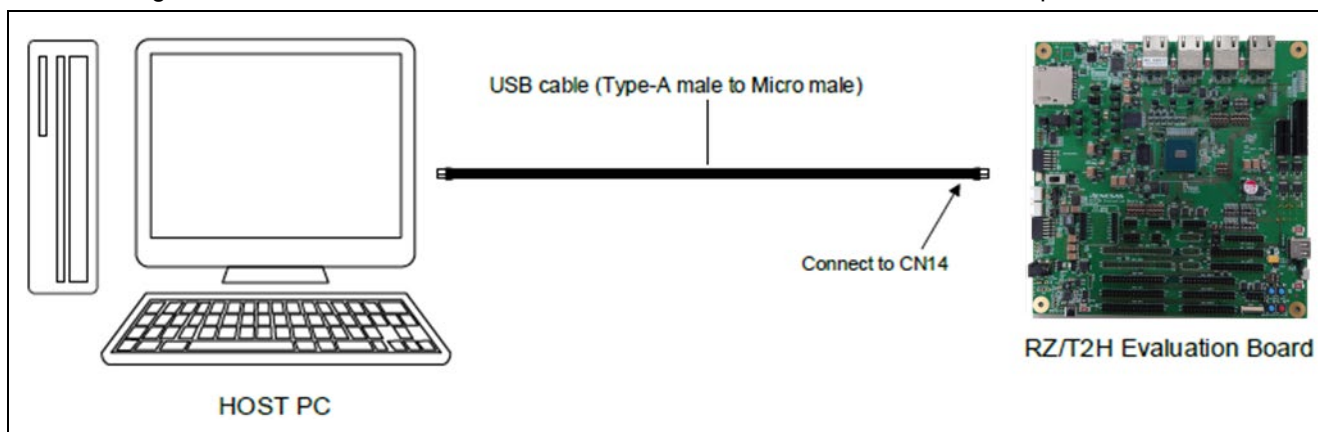


Figure 4.12 RZ/T2H evaluation board debug connection diagram (J-Link OB)

2. Power is supplied through 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 using the AC/DC adapter, connect it to the connector CN47 of the RZ/T2H evaluation board.
3. Connect the host PC and the RZ/T2H evaluation board using an Ethernet cable. Connect the Ethernet cable to the ETH0 connector of the RZ/T2H evaluation board. Depending on the protocol specifications, you might be able to connect the cable to ETH1 instead.

4.5 RZ/N2H Evaluation Board

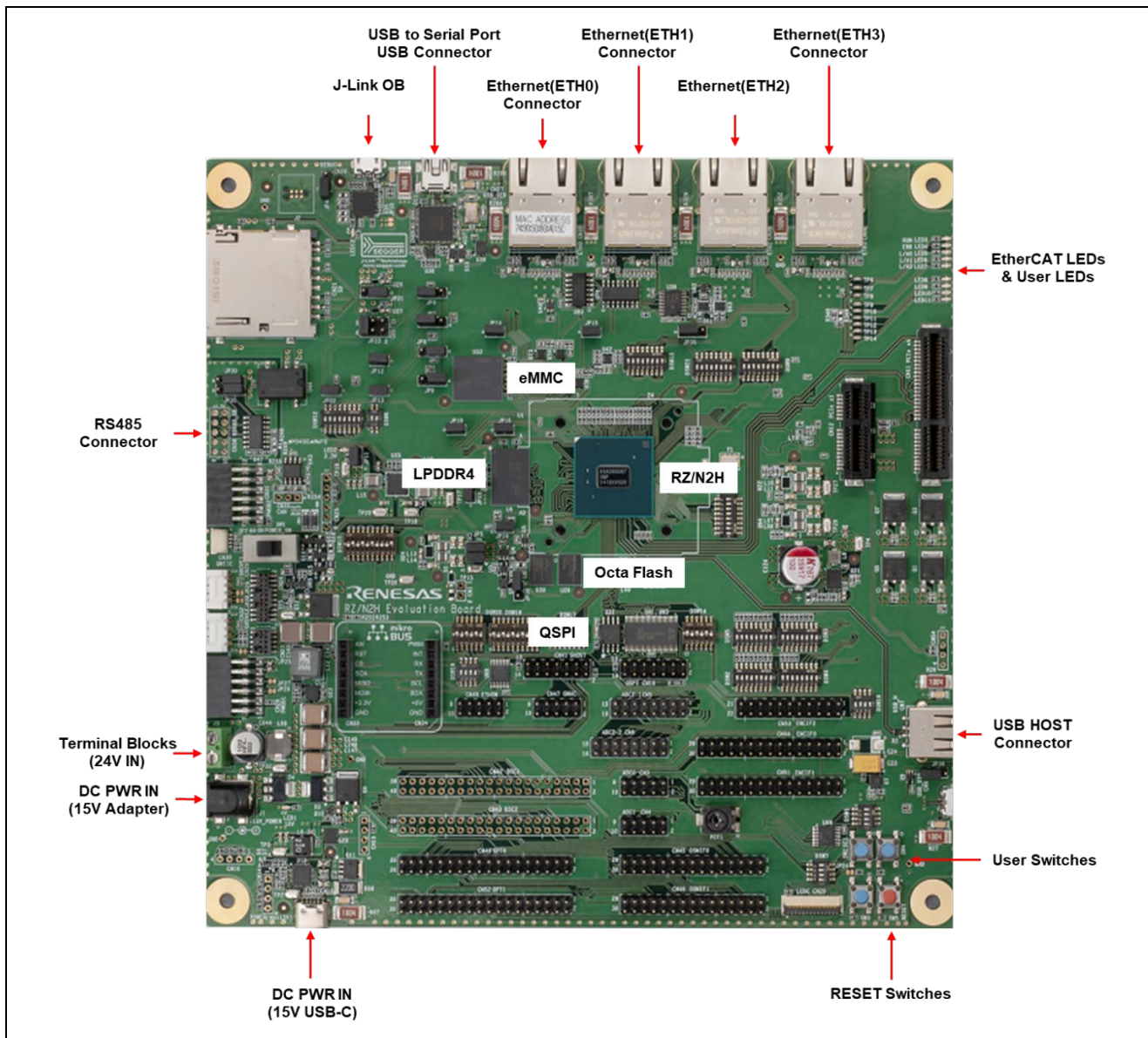


Figure 4.13 RZ/N2H evaluation board layout

4.5.1 Jumper and Switch Configuration

The following tables show the jumper pin and switch settings.

Text in bold 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 (SW5-6 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	See r20ut5522ejxxxx-rzn2hevb.pdf and set the switches according to the mode to use.
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 SW14-1, SW14-3, and SW14-5 to ON and SW14-2, SW14-4, and SW14-6 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 is used as a root complex.
SW4-2	ON	PCIe L1 is used as a 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 in the same way as Ethernet ports 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_CRIS 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	Fixed setting
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.
SW15-9	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_LED RUN(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 LCDG(CN20), INT of PMOD1(CN28), TX of mikroBUS™(CN34), DREQ of BSC1(CN42), ENCIFOE00 of ENCIFO(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).
SW20-2	OFF	In this case, set SW18-5 to OFF, SW18-6 to ON, SW19-3 to OFF, and SW19-4 to ON.
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).
SW21-2	OFF	
SW21-3	ON	In this case, set SW5-8 to OFF.
SW21-4	OFF	P34_5 is used as CS3# of BSC1(CN42).
SW21-5	ON	In this case, set SW5-8 to OFF.
SW21-6	OFF	P34_6 is used as CS5# of BSC1(CN42).
SW21-7	ON	In this case, set SW5-8 to OFF.
SW21-8	OFF	-

4.5.2 Board Setup

This section describes how to set up the board to run the sample program.

To build and run the sample code on the RZ/N2H Evaluation board, follow these steps:

You can load the program into RAM or flash memory using IAR Embedded Workbench or e² studio.

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

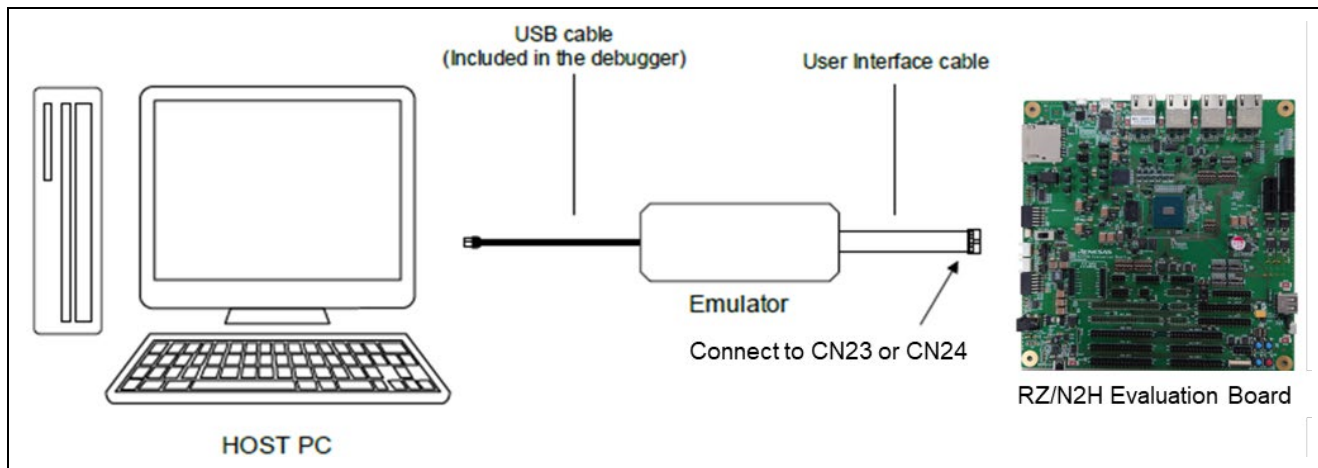


Figure 4.14 RZ/N2H evaluation board debug connection diagram

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

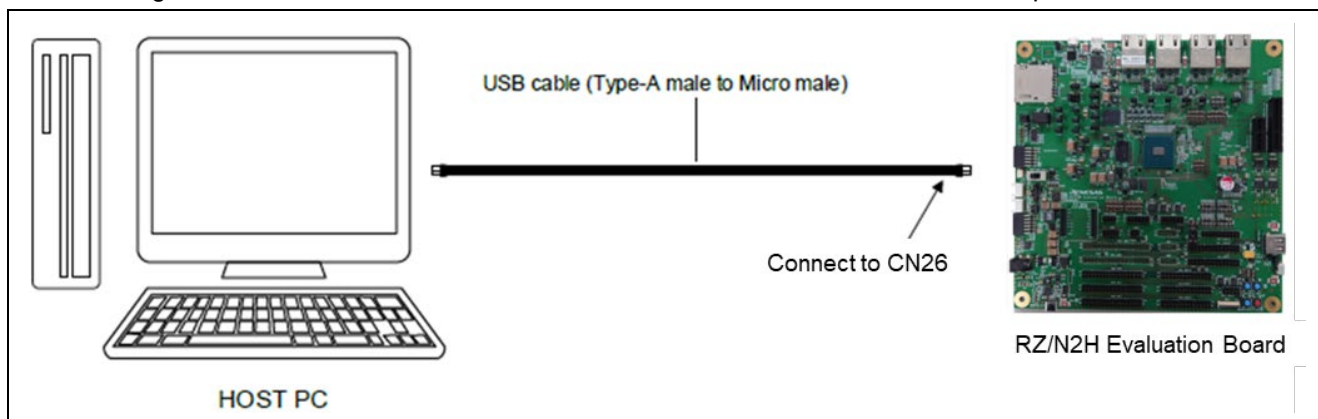


Figure 4.15 RZ/N2H evaluation board debug connection diagram (J-Link OB)

2. Power is supplied through 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 using the AC/DC adapter, connect it to the connector J1 of the RZ/N2H evaluation board.
3. Connect the host PC and the RZ/N2H evaluation board using an Ethernet cable. Connect the Ethernet cable to the ETH0 connector of the RZ/N2H evaluation board. Depending on the protocol specifications, you might be able to connect the cable to ETH1 instead.

4.6 RZ/T2N Evaluation Board

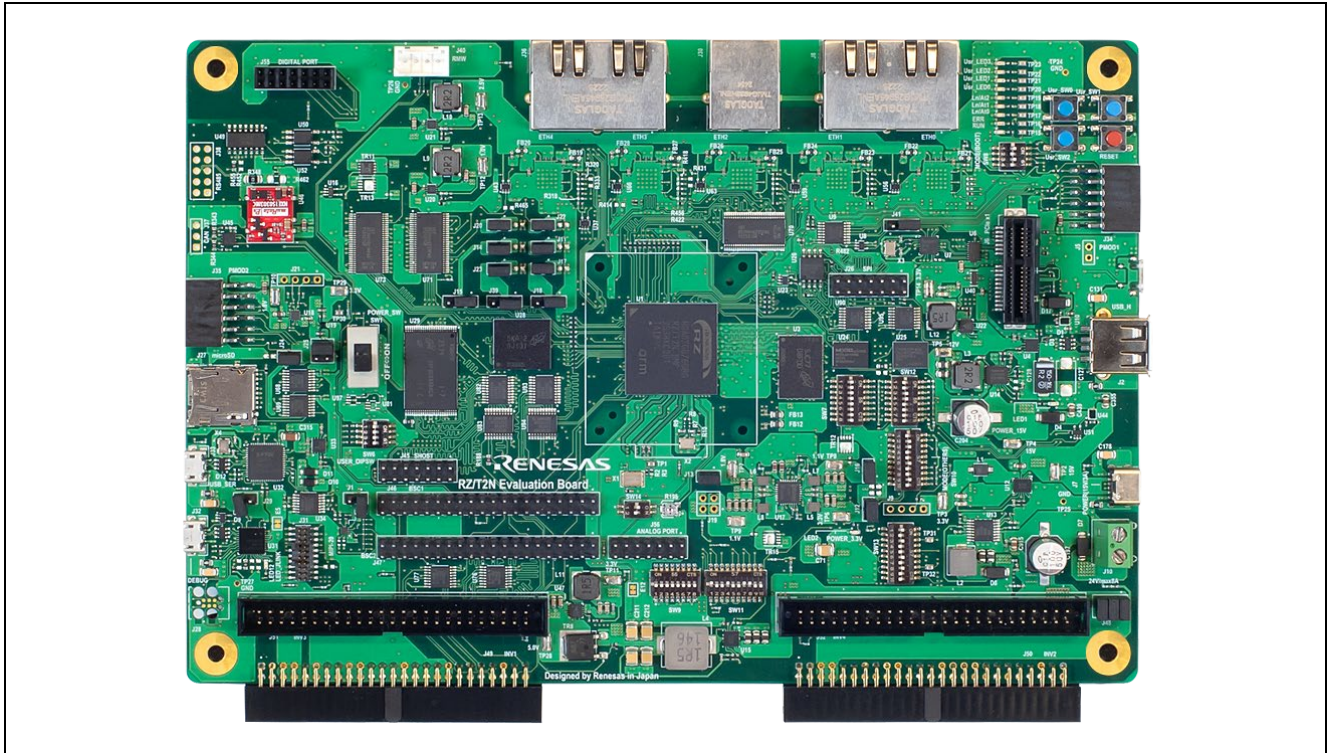


Figure 4.16 RZ/T2N evaluation board layout

4.6.1 Jumper and Switch Configuration

The following tables show the jumper pin and switch settings.

Text in bold type in the tables indicates the settings changed from the time of shipment of the board.

Table 4.46 RZ/T2N EVB jumper configuration

Reference	Jumper position	Description
J29	Open	The on-board debugging function J-Link™ OB is enabled.
J1	Open	PCIe reset is not included in the system reset factors.
J15	Short 1-2	1.8-V power is supplied to VCC1833_1 (for Ethernet port 2, SW7-2 = ON, SW10-2 = ON).
J18	Short 1-2	1.8-V power is supplied to VCC1833_2 (for Ethernet ports 3 and 4 with RGMII, SW10-3 = ON).
J25	Short 3-4	The output from the power-supply control IC for SD1 is supplied to VCC1833_6 (for microSD).
J39	Short 1-2	1.8-V power is supplied to VCC1833_5 (for eMMC, SW10-6 = ON).
J41	Short 2-3	1.8-V power is supplied to VDDMDIO of the Ethernet port 0 PHY (U54) (SW12-3 = ON).
J48	Short	GND_24V for the inverter board is connected to GND.

Table 4.47 RZ/T2N EVB SW8 settings

SW8	Setting	Description
SW8-1	ON	xSPI1 boot mode (x1 boot serial flash)
SW8-2	OFF	See r20ut5574ejxxxx-rzt2nevb.pdf and set the switches according to the mode to use.
SW8-3	ON	—
SW8-4	OFF	—

Table 4.48 RZ/T2N EVB SW10 settings

SW10	Setting	Description
SW10-1	OFF	—
SW10-2	ON	MDV1 = L: VCC1833_1 = 1.8 V
SW10-3	ON	MDV2 = L: VCC1833_2 = 1.8 V
SW10-4	OFF	—
SW10-5	OFF	—
SW10-6	ON	MDV5 = L: VCC1833_5 = 1.8 V
SW10-7	OFF	Cortex-R52 CPU0 ATCM wait cycle = 1 wait
SW10-8	OFF	Cortex-R52 CPU1 ATCM wait cycle = 1 wait
SW10-9	ON	JTAG mode = Normal mode
SW10-10	ON	2nd Boot CPU = Cortex-R52 CPU0

Table 4.49 RZ/T2N EVB SW7 settings

SW7	Setting	Description
SW7-1	ON	P03_6, P03_7, P04_0 to P04_3, and P15_0 are used as microSD control signals.
SW7-2	ON	P23_1, P23_2, P23_4 to P23_7, P24_0 to P24_4, and P25_0 are used as Ethernet port 2 control signals.
SW7-3	ON	P00_0 to P00_7, P01_0 to P01_5, P16_1, and P16_7 are used as Ethernet port 3 control signals.
SW7-4	ON	P01_6, P01_7, P02_0 to P02_7, P03_0, P03_1, P03_4, P03_5, and P15_1 are used as Ethernet port 4 control signals.
SW7-5	ON	P15_7 and P16_2 to P16_5 are used as QSPI flash and SPI expansion connector signals.
SW7-6	ON	P17_5 and P18_3 to P18_6 are used as OctaFlash control signals.
SW7-7	ON	P04_4 to P04_7 and P05_0 to P05_6 are used as eMMC control signals.
SW7-8	ON	

Table 4.50 RZ/T2N EVB SW9 settings

SW9	Setting	Description
SW9-1	OFF	—
SW9-2	OFF	
SW9-3	OFF	P08_5 to P08_7 and P09_3 are used as inverter board INV4, BSC1, and expansion SPI connector control signals.
SW9-4	OFF	—
SW9-5	OFF	ENCIF signals for the inverter board INV1.
SW9-6	OFF	ENCIF signals for the inverter board INV2.
SW9-7	OFF	ENCIF signals for the inverter board INV3.
SW9-8	OFF	ENCIF signals for the inverter board INV4.

Table 4.51 RZ/T2N EVB SW11 settings

SW11	Setting	Description
SW11-1	ON	P11_0 is used as VBUSEN for the USB host.
SW11-2	OFF	
SW11-3	OFF	P11_7 is used for the inverter board INV2 or SHOST.
SW11-4	ON	
SW11-5	OFF	P12_4 is used as the input of user push switch SW4.
SW11-6	OFF	
SW11-7	ON	
SW11-8	ON	P14_3 is used as EXICEN for USB OTG.
SW11-9	OFF	
SW11-10	OFF	—

Table 4.52 RZ/T2N EVB SW12 settings

SW12	Setting	Description
SW12-1	OFF	—
SW12-2	OFF	
SW12-3	ON	P20_4, P20_5, and P07_0 are used as MDC, MDIO, and NRESET of Ethernet port 0.
SW12-4	ON	P20_4, P20_5, and P07_0 are used as MDC, MDIO, and NRESET of Ethernet port 2.
SW12-5	OFF	P16_6, P16_7, and P17_0 are used for the SPI expansion connector and PMOD1.
SW12-6	OFF	PCIe endpoint; the 12-V power of PCIe connector J6 is OFF.
SW12-7	OFF	
SW12-8	OFF	PCIe endpoint; the 3.3-V power of PCIe connector J6 is OFF.

Table 4.53 RZ/T2N EVB SW13 settings

SW13	Setting	Description
SW13-1	ON	P17_0 is used for the SPI expansion connector and INV4.
SW13-2	OFF	
SW13-3	ON	P15_4 is used as SCL for EEPROM.
SW13-4	OFF	
SW13-5	OFF	
SW13-6	ON	P17_1 is used for the SPI expansion connector and INV4.
SW13-7	OFF	
SW13-8	ON	P15_5 is used as SDA for EEPROM.
SW13-9	OFF	
SW13-10	OFF	

Table 4.54 RZ/T2N EVB SW14 settings

SW14	Setting	Description
SW14-1	ON	AN000 is connected to the potentiometer.
SW14-2	OFF	

4.6.2 Board Setup

This section describes how to set up the board to run the sample program.

To build and run the sample code on the RZ/T2N Evaluation board, follow these steps:

You can load the program into RAM or flash memory using IAR Embedded Workbench or e² studio.

1. Connect the debugger to the header J31 on the RZ/T2N evaluation board.

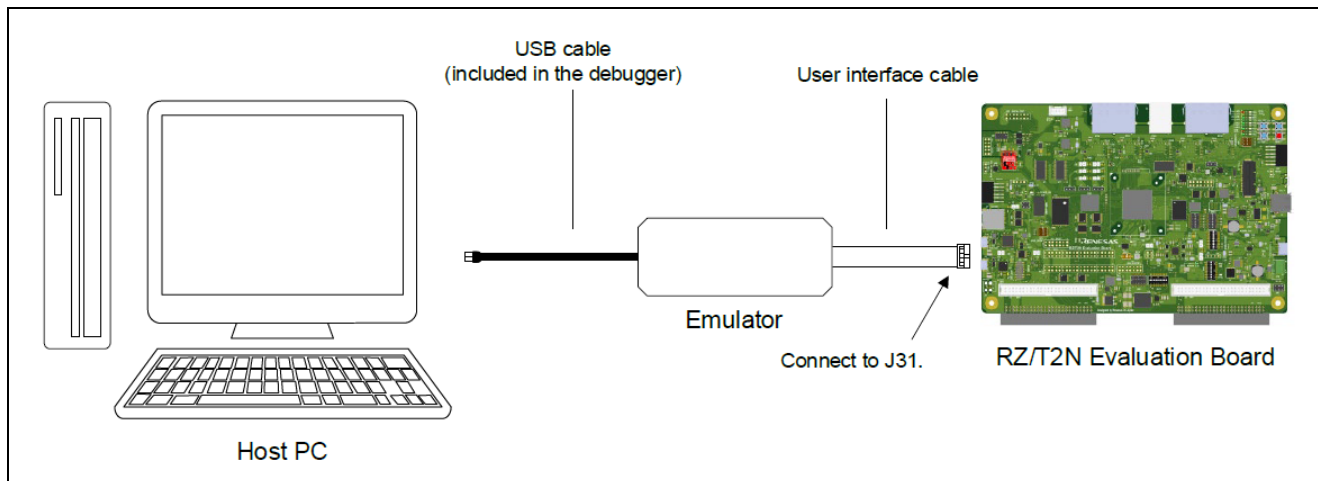


Figure 4.17 RZ/T2N evaluation board debug connection diagram

When using J-Link OB, connect the USB cable to the header J32 and set J29 to open.

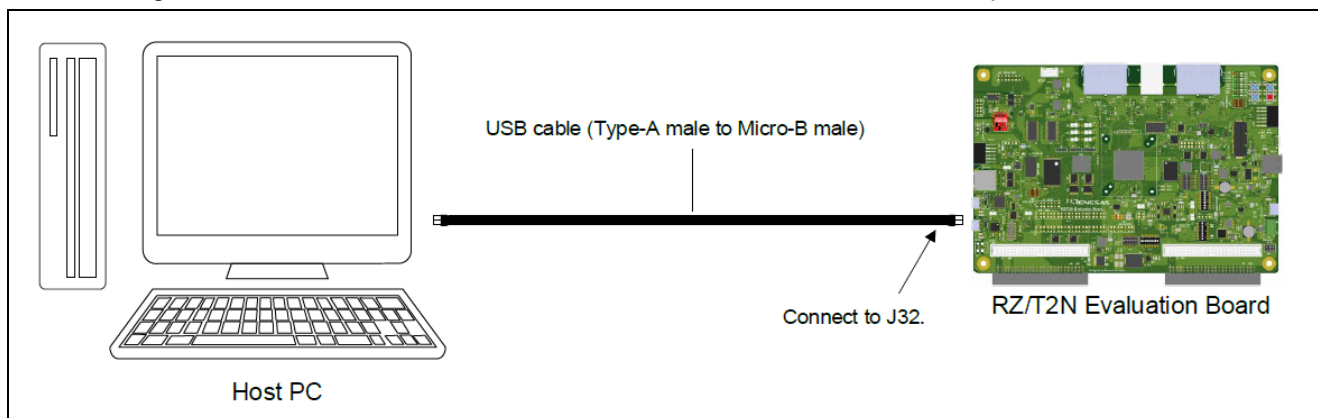


Figure 4.18 RZ/T2N evaluation board debug connection diagram (J-Link OB)

2. Supply power from J7 (USB Type-C, 15 V) or J10 (24 V terminal block), and then turn on power switch SW1.
3. Connect the host PC and the RZ/T2N evaluation board using an Ethernet cable. Connect the Ethernet cable to the ETH0 (J8A) connector of the RZ/T2N evaluation board.

5. Setting up the Host

5.1 Setting up TwinCAT 3

5.1.1 Copying the ESI Files

Before starting TwinCAT, copy the ESI file (.xml) 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_rev0411\common\AoE\ESI

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

- For multi-core projects (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/T2N, RZ/N2H):

RZT2H_EtherCAT_EVB_rev0411\CR52_Dual\common\AoE\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_rev0411\CR52_Dual\common\AoE\ESI
- RZ/N2H (CR52_Dual):
RZN2H_EtherCAT_EVB_rev0411\CR52_Dual\common\AoE\ESI
- RZ/N2H (CR52_CA55):
RZN2H_EtherCAT_EVB_rev0411\CR52_CA55\common\AoE\ESI
- RZ/T2N (CR52_Dual):
RZT2N_EtherCAT_EVB_rev0400\CR52_Dual\common\AoE\ESI
- RZ/T2N (CR52_CA55):
RZT2N_EtherCAT_EVB_rev0400\CR52_CA55\common\AoE\ESI

The ESI file:

- For RZ/T2L, RZ/T2M, RZ/T2ME, RZ/T2H, RZ/T2N
Renesas EtherCAT RZT2 AoE.xml
- For RZ/N2L, RZ/N2H
Renesas EtherCAT RZN2 AoE.xml

The TwinCAT destination folder:

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

5.1.2 Adding the Driver

Add the Ethernet driver for TwinCAT (first time only).

From the Start menu, select **TwinCAT3** → **Show Realtime Ethernet Compatible Devices....**

Select the connected Ethernet port from the communication ports and install the driver.

If the selected Ethernet adapter appears under **Installed and ready to use devices (realtime capable)**, the driver installation is complete.

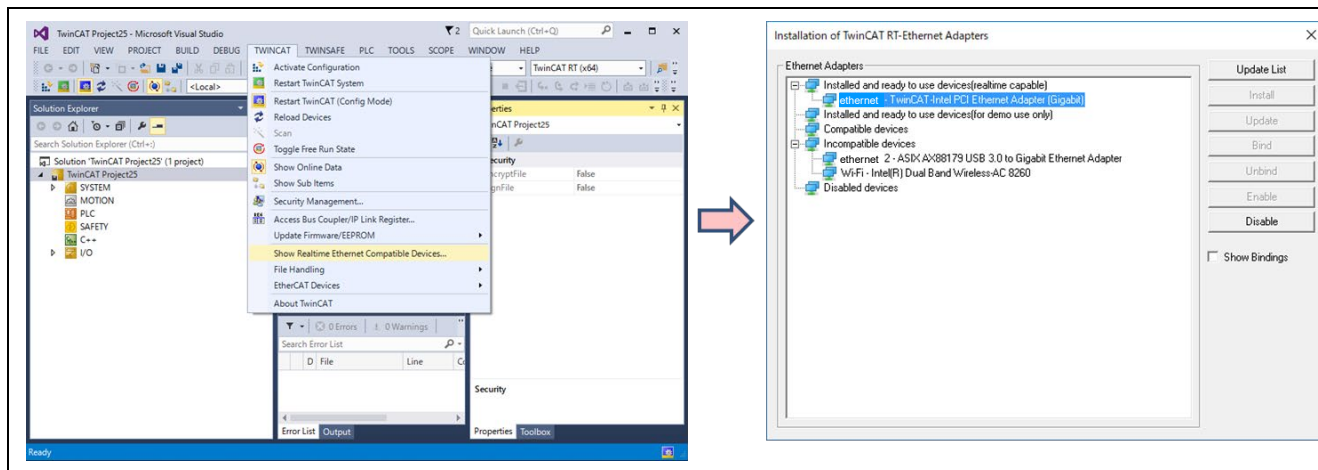


Figure 5.1 Add driver

Note: If you do not use an Intel NIC, the Ethernet adapter appears under **Installed and ready to use devices (for demo use only)**.

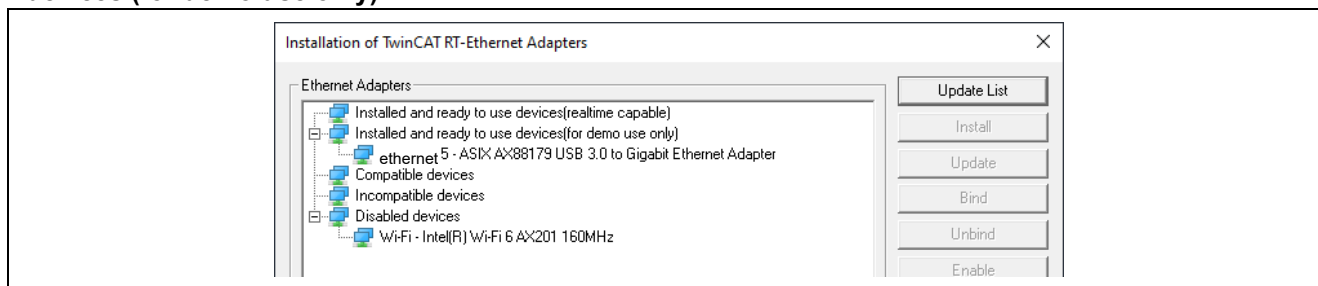


Figure 5.2 In the case of the NIC made by Intel

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, project names, and core configuration in the procedures with the values listed 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_rev0411	Single (Cortex-R52)	RZT2L_RSK_ESC_AoE	–
RZ/N2L	RZN2L_EtherCAT_RSK_rev0411	Single (Cortex-R52)	RZN2L_RSK_ESC_AoE	–
RZ/T2M, RZ/T2ME	RZT2M_RZT2ME_EtherCAT_RSK_rev0411	CR52_Dual (Cortex-R52 CPU0 and Cortex-R52 CPU1)	RZT2M_RSK_ESC_AoE_CR52_Dual_Primary	RZT2M_RSK_ESC_AoE_CR52_Dual_Secondary
RZ/T2H	RZT2H_EtherCAT_EVB_rev0411	CR52_Dual (Cortex-R52 CPU0 and Cortex-R52 CPU1)	RZT2H_EVB_ESC_AoE_CR52_Dual_Primary	RZT2H_EVB_ESC_AoE_CR52_Dual_Secondary
		CR52_CA55 (Cortex-R52 CPU0 and Cortex-A55 Core2)	RZT2H_EVB_ESC_AoE_CR52_CA55_Primary	RZT2H_EVB_ESC_AoE_CR52_CA55_Secondary
RZ/N2H	RZN2H_EtherCAT_EVB_rev0411	CR52_Dual (Cortex-R52 CPU0 and Cortex-R52 CPU1)	RZN2H_EVB_ESC_AoE_CR52_Dual_Primary	RZN2H_EVB_ESC_AoE_CR52_Dual_Secondary
		CR52_CA55 (Cortex-R52 CPU0 and Cortex-A55 Core2)	RZN2H_EVB_ESC_AoE_CR52_CA55_Primary	RZN2H_EVB_ESC_AoE_CR52_CA55_Secondary
RZ/T2N	RZT2N_EtherCAT_EVB_rev0400	CR52_Dual (Cortex-R52 CPU0 and Cortex-R52 CPU1)	RZT2N_EVB_ESC_AoE_CR52_Dual_Primary	RZT2N_EVB_ESC_AoE_CR52_Dual_Secondary
		CR52_CA55 (Cortex-R52 CPU0 and Cortex-A55 Core1)	RZT2N_EVB_ESC_AoE_CR52_CA55_Primary	RZT2N_EVB_ESC_AoE_CR52_CA55_Secondary

6.1 Generating the SubDevice Stack Code

Use the SSC Tool to generate the EtherCAT SubDevice stack code before building the project.

Note: Before using the SSC Tool, check the precautions.

See section "Appendix A: Precautions for Using the SSC Tool".

Note: If the patch command is not installed on your computer, install the 64-bit patch.exe.

For instructions, see section "Appendix B: How to Install the Patch Command".

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_rev0411\common\AoE\SSCconfig\
RZT2_EtherCAT_AoE_CR52.esp**

- For **multi-core projects** (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/T2N, RZ/N2H):

**RZT2H_EtherCAT_EVB_rev0411\CR52_Dual\common\AoE\SSCconfig\
RZT2_EtherCAT_AoE_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**.

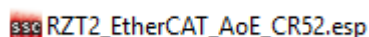


Figure 6.1 The .esp file (example of RZ/T2L)

The following window opens.

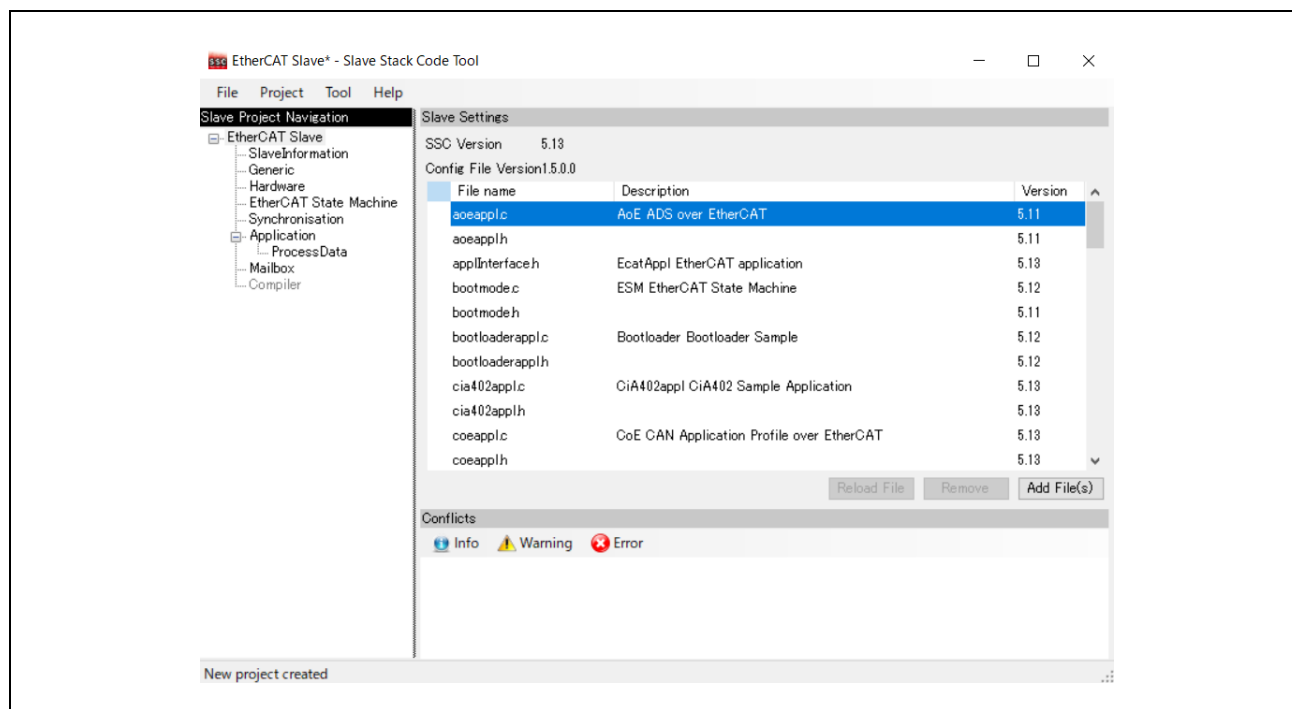


Figure 6.2 SSC Tool main window

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

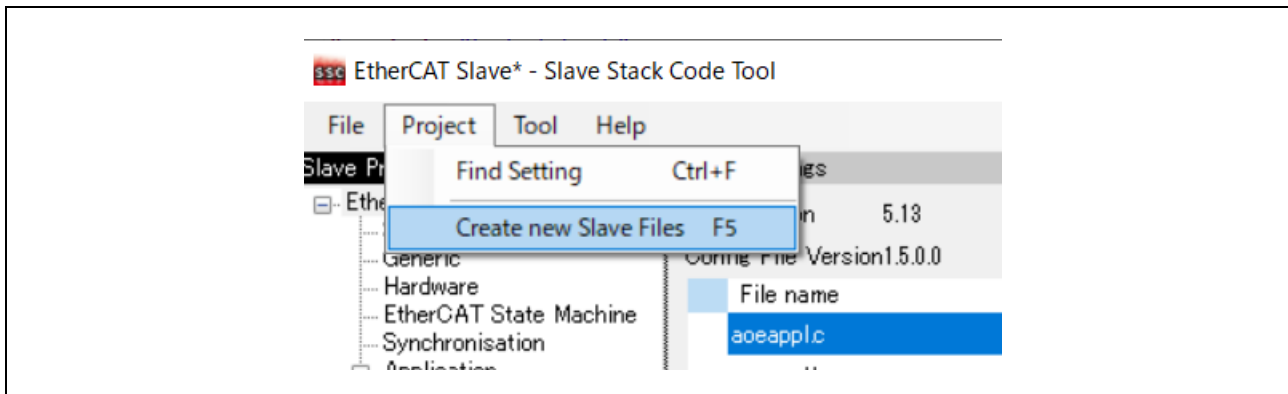


Figure 6.3 Generate EtherCAT SubDevice Stack (1)

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

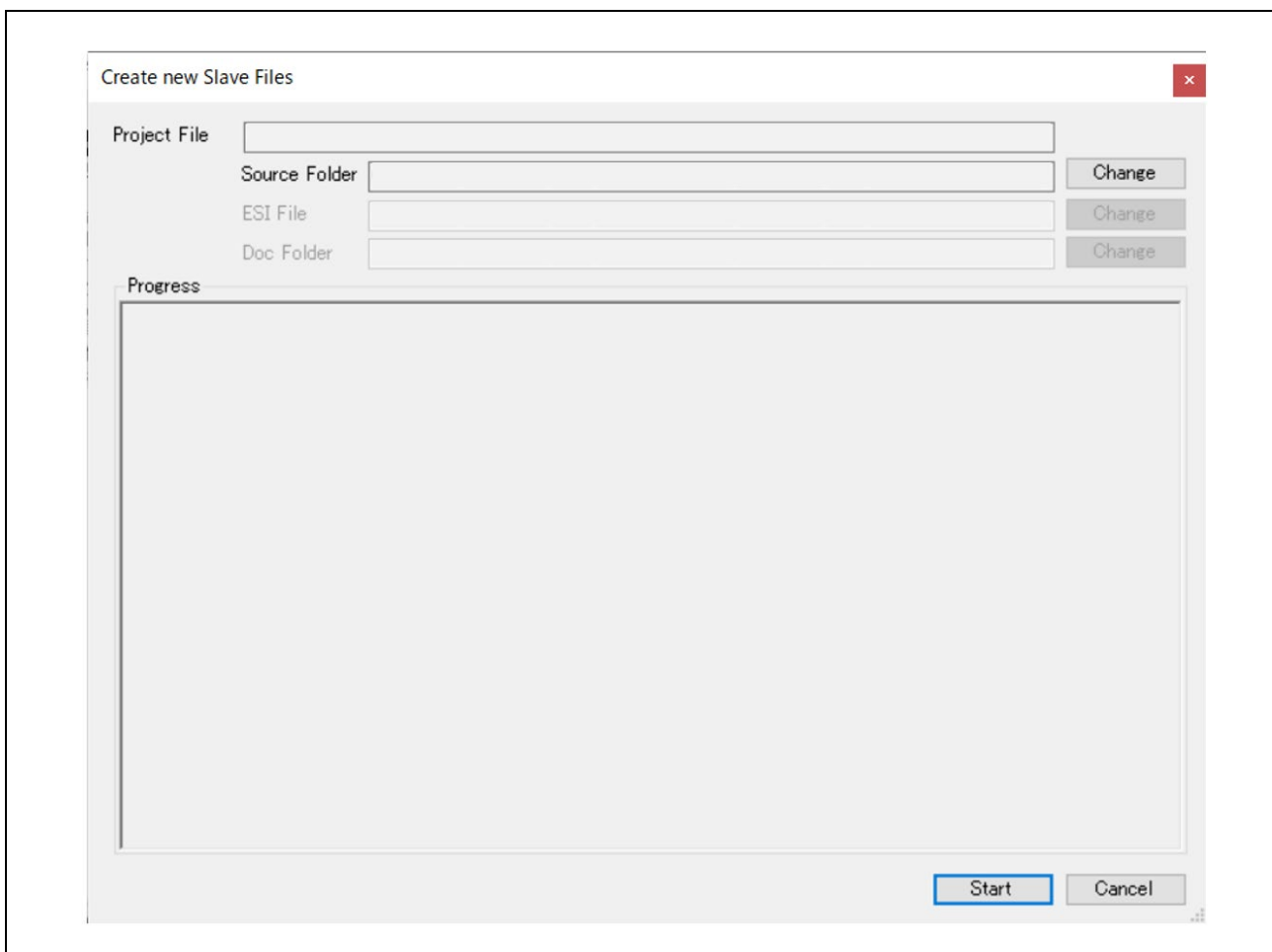


Figure 6.4 Generate EtherCAT SubDevice Stack (2)

4. When the message **"New files created successfully"** appears, the generation is complete.

The source files are generated in the following folder:

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

RZT2L_EtherCAT_RSK_rev0411\common\AoE\SSCconfig

- For **multi-core projects** (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/T2N, RZ/N2H):

RZT2H_EtherCAT_EVB_rev0411\CR52_Dual\common\AoE\SSCconfig

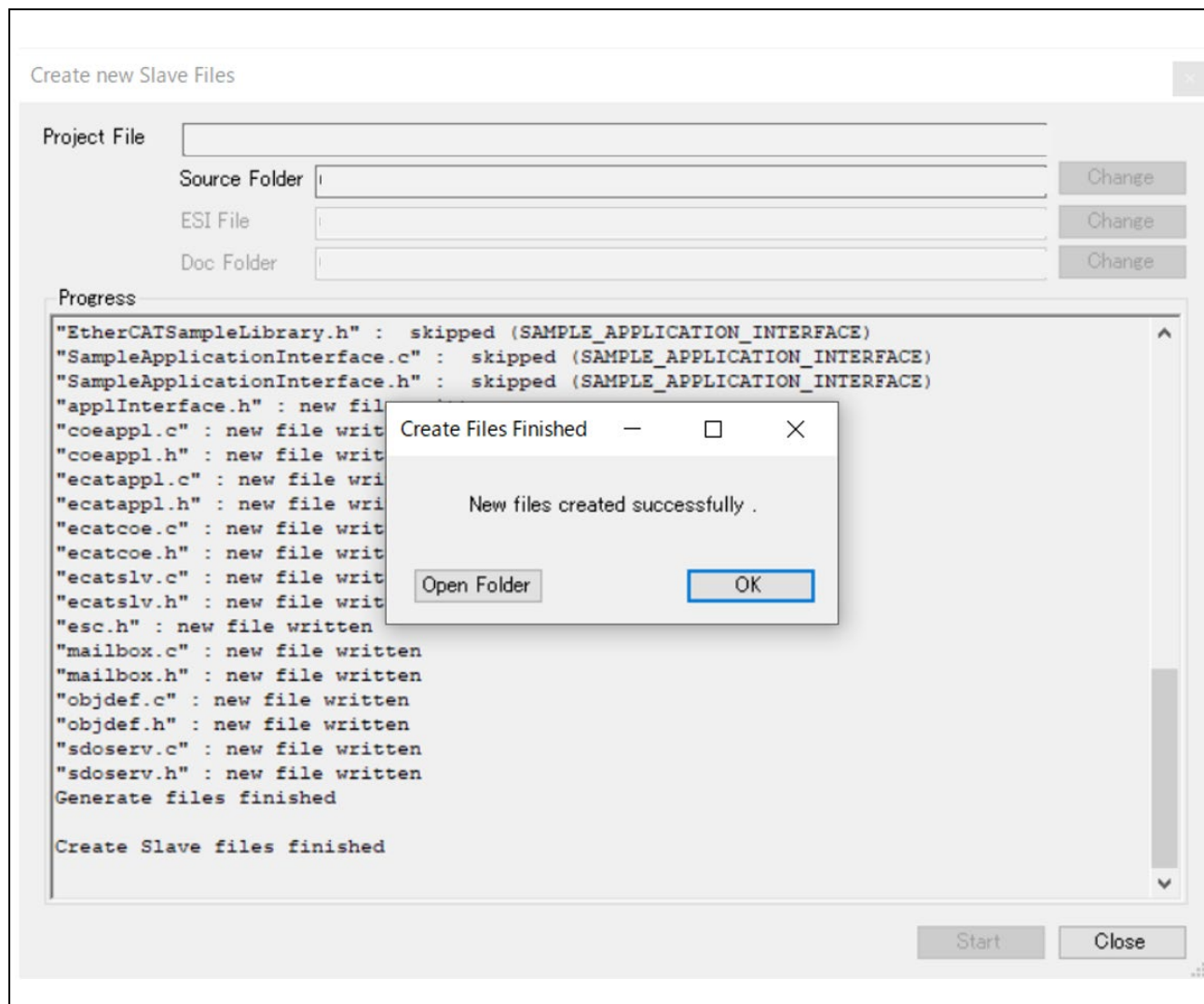


Figure 6.5 Generate EtherCAT SubDevice Stack (3)

5. Copy the EtherCAT SubDevice stack code generated by the SSC Tool to the project folder.

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

Copy the generated Src folder to the following directories:

(1) e² studio project

"project\AoE\e2studio\<project name>\src\ethercat\beckhoff"

(2) IAR Embedded Workbench project

"project\AoE\ewarm\<project name>\src\ethercat\beckhoff"

- For **multi-core projects** (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/T2N, RZ/N2H):

Copy the generated Src folder to the following directories:

(1) e² studio project

"project\AoE\e2studio\<secondary project name>\src\ethercat\beckhoff"

(2) IAR Embedded Workbench project

"project\AoE\ewarm\<secondary project name>\src\ethercat\beckhoff"

Note: For multi-core projects, copy the generated Src folder to the Secondary core project, not to the Primary core project.

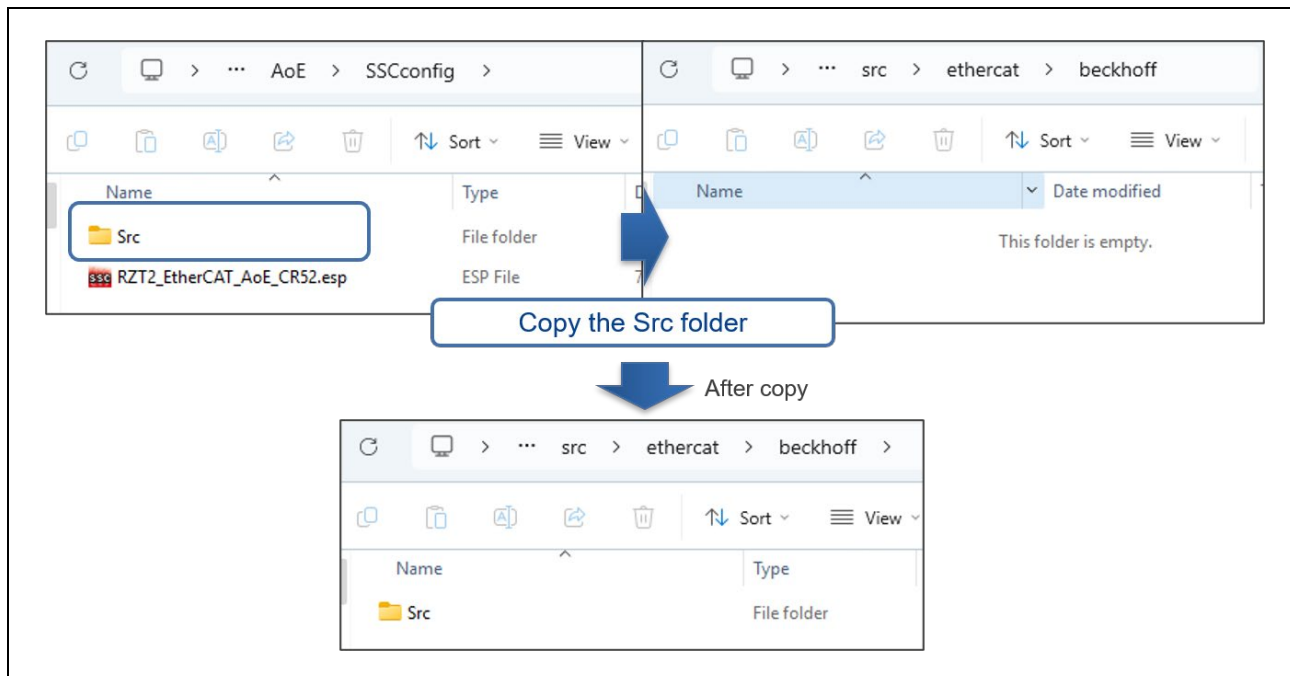


Figure 6.6 Copy the generated stack code to the project folder

6.2 Debugging with EWARM

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

1. Open the sample project (.eww):
RZT2L_EtherCAT_RSK_rev0411\project\AoE\ewarm\RZT2L_RSK_ESC_AoE\
RZT2L_RSK_ESC_AoE.eww

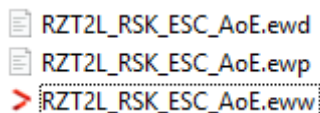


Figure 6.7 Open EWARM workspace file

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

2. Select **Tools > Configure Tools...** and register the following settings.

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\$

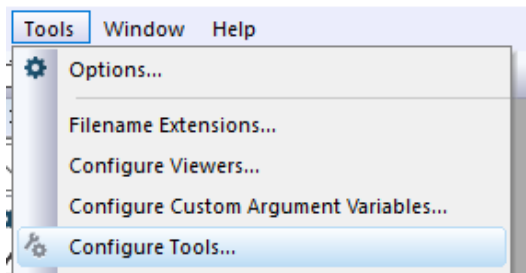


Figure 6.8 Open Configure Tools...

3. Open the buildinfo.ipcf file and enter the path of your FSP Smart Configurator in RASC_EXE_PATH. The FSP Smart Configurator must support RZ FSP v4.0.0 or later.

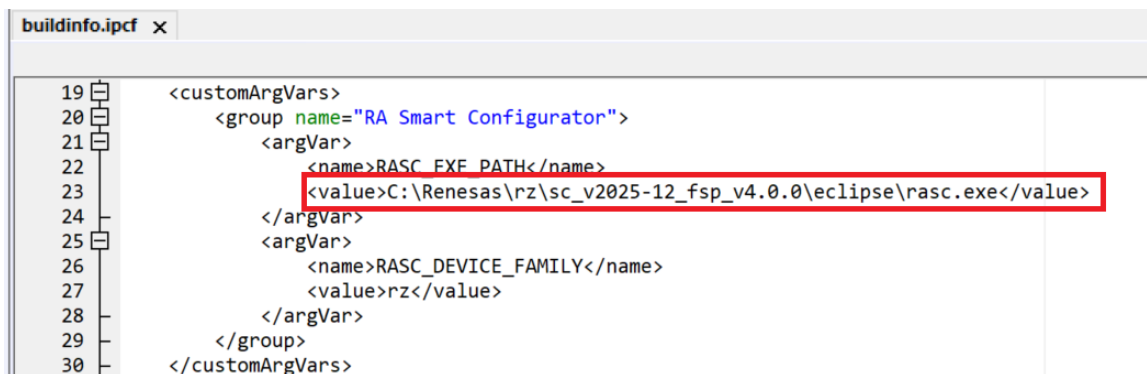


Figure 6.9 RASC_EXE_PATH in buildinfo.ipcf

4. Select the registered menu entry (**RZ Smart Configurator** in this example) to launch the Smart Configurator.

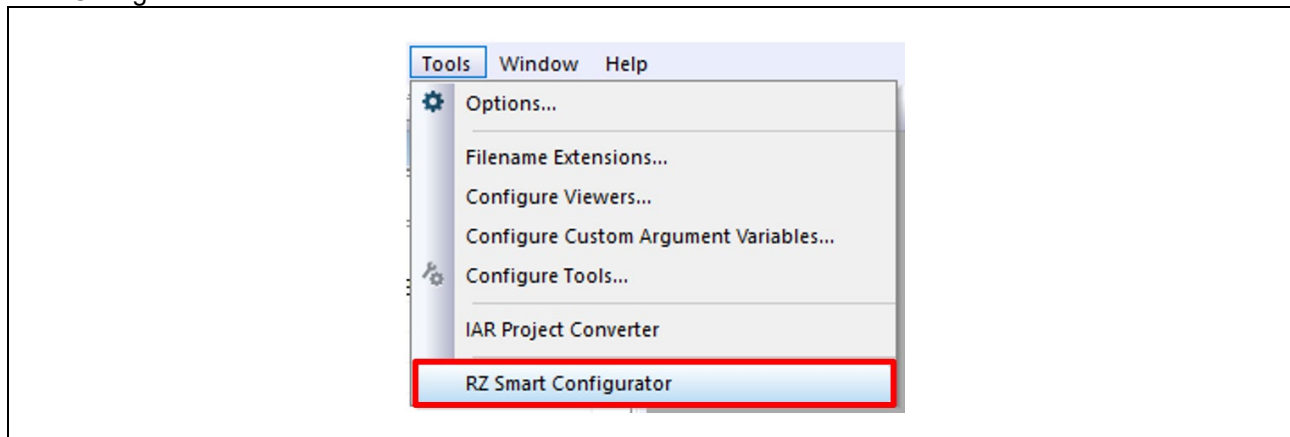


Figure 6.10 Open the Smart Configurator

5. Click **Generate Project Content** to generate the FSP code.

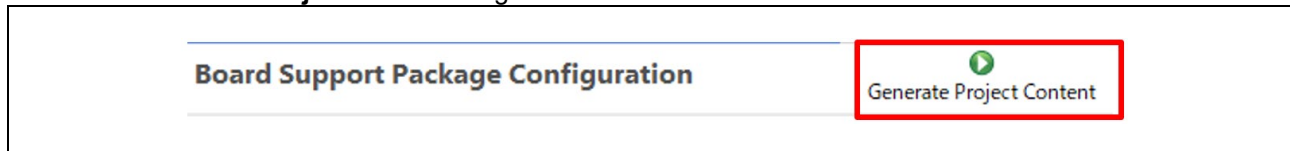


Figure 6.11 Generate Project Content

6. Select **Project > Options...**, and then click the **Config** tab in the **Linker** category. Change Linker configuration file to the following linker script:

\$PROJ_DIR\$/script/fsp_ram_execution_SystemRAM.icf

The linker script places the EtherCAT protocol code in System RAM.

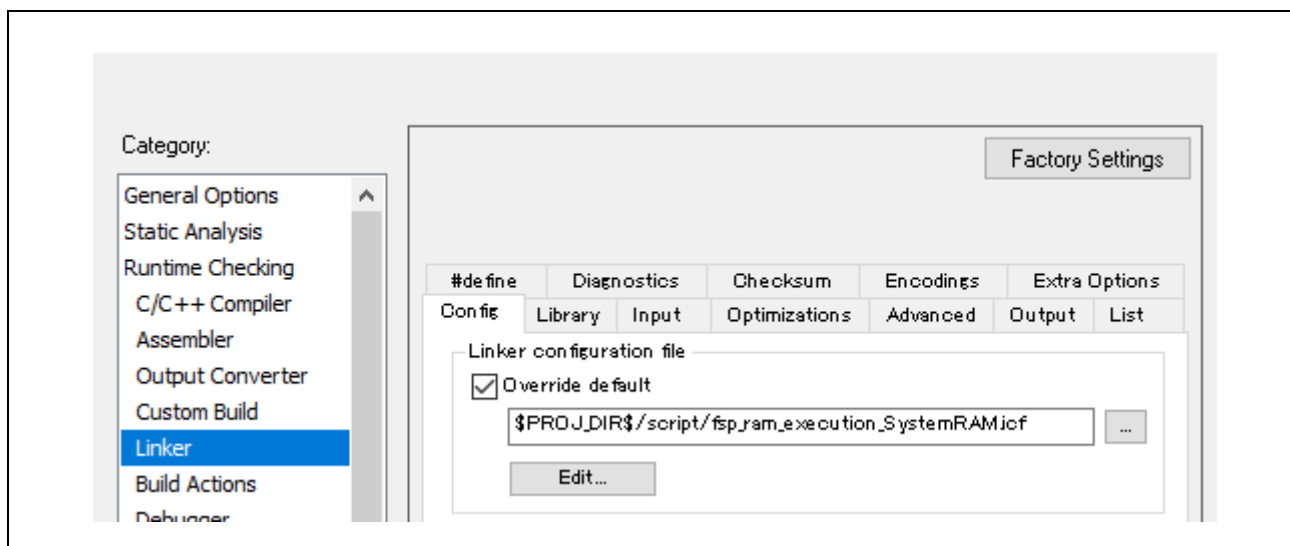


Figure 6.12 Setting the linker script

7. Click **Make** to build the project.

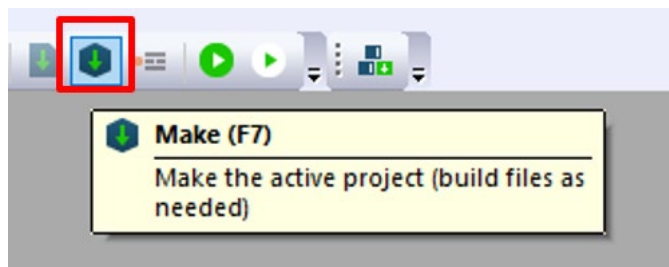


Figure 6.13 Make the project

8. Click **Download and Debug** to download the binary to the board.

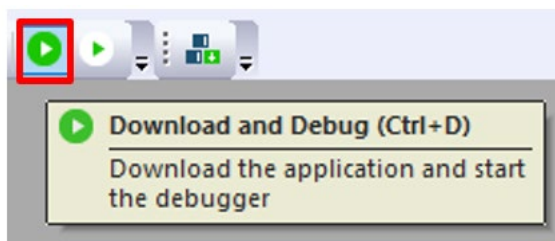


Figure 6.14 Download and Debug

9. Click **Go** to start program execution.

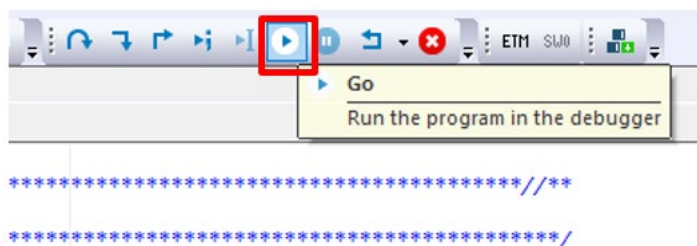


Figure 6.15 Run the program

Note: If a program is already written to flash memory when you start debugging, it might interfere with the debugging process and cause debugging to fail.

If this happens, erase the flash memory first and then retry.

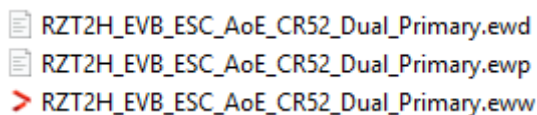
For details, see "Appendix: How to Erase Flash Memory" in the document [r01an6434ej****](#) listed in Table 1.2.

6.2.2 Multi-Core Project (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/T2N, RZ/N2H)

6.2.2.1 Building the Primary Core Project

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

RZT2H_EtherCAT_EVB_rev0411\CR52_Dual\project\AoE\ewarm\
RZT2H_EVB_ESC_AoE_CR52_Dual_Primary\RZT2H_EVB_ESC_AoE_CR52_Dual_Primary.eww



RZT2H_EVB_ESC_AoE_CR52_Dual_Primary.ewd
RZT2H_EVB_ESC_AoE_CR52_Dual_Primary.evp
➤ RZT2H_EVB_ESC_AoE_CR52_Dual_Primary.eww

Figure 6.16 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. Select **Tools > Configure Tools...** and register the following settings.

Table 6.3 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\$

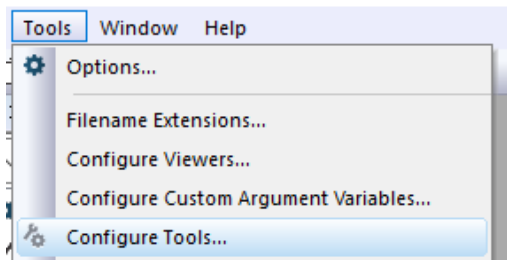
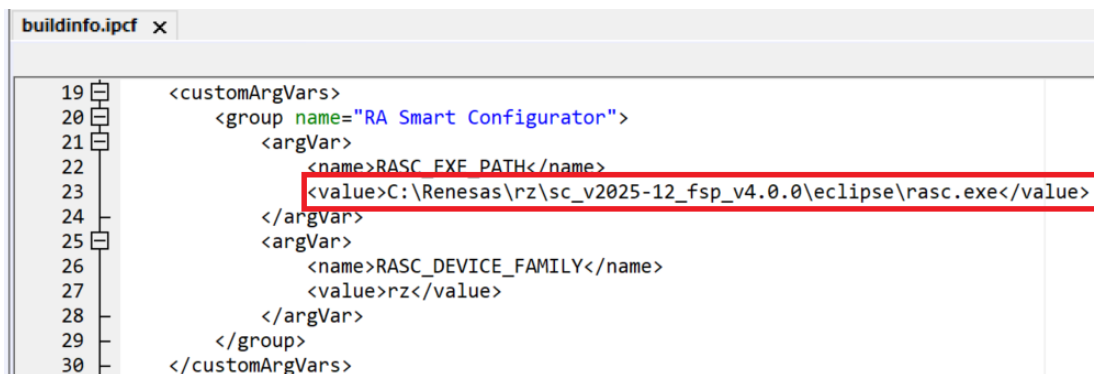


Figure 6.17 Open Configure Tools...

3. Open the buildinfo.ipcf file and enter the path of your FSP Smart Configurator in RASC_EXE_PATH. The FSP Smart Configurator must support RZ FSP v4.0.0 or later.



```

19 <customArgVars>
20   <group name="RA Smart Configurator">
21     <argVar>
22       <name>RASC_EXE_PATH</name>
23       <value>C:\Renesas\rz\sc_v2025-12_fsp_v4.0.0\eclipse\rasc.exe</value>
24     </argVar>
25     <argVar>
26       <name>RASC_DEVICE_FAMILY</name>
27       <value>rz</value>
28     </argVar>
29   </group>
30 </customArgVars>

```

Figure 6.18 RASC_EXE_PATH in buildinfo.ipcf

4. Select the registered menu entry (**RZ Smart Configurator** in this example) to launch the Smart Configurator.

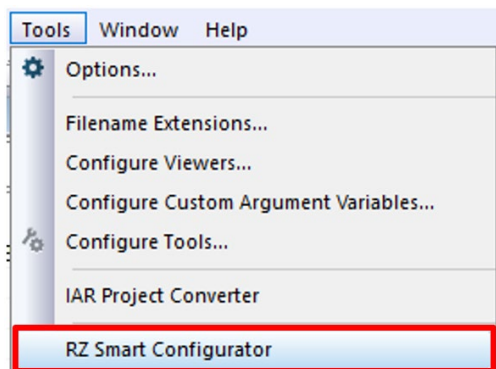


Figure 6.19 Open the Smart Configurator

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

5. Click **Generate Project Content** to generate the FSP code.

Board Support Package Configuration

Generate Project Content

Figure 6.20 Generate Project Content

6. Click **Make** to build the Primary core project.

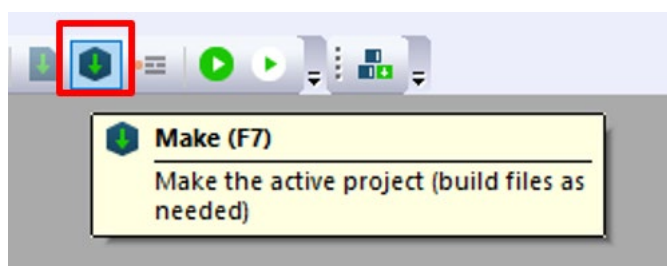


Figure 6.21 Make the Primary project

6.2.2.2 Building the Secondary Core Project

1. Open the Secondary project (.eww):
- 2.

RZT2H_EtherCAT_EVB_rev0411\CR52_Dual\project\AoE\ewarm\

RZT2H_EVB_ESC_AoE_CR52_Dual_Secondary\RZT2H_EVB_ESC_AoE_CR52_Dual_Secondary.eww

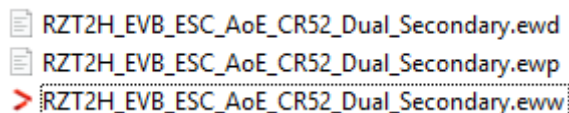


Figure 6.22 Open Secondary EWARM workspace file

3. Open the buildinfo.ipcf file and enter the path of your FSP Smart Configurator in RASC_EXE_PATH. The FSP Smart Configurator must support RZ FSP v4.0.0 or later.

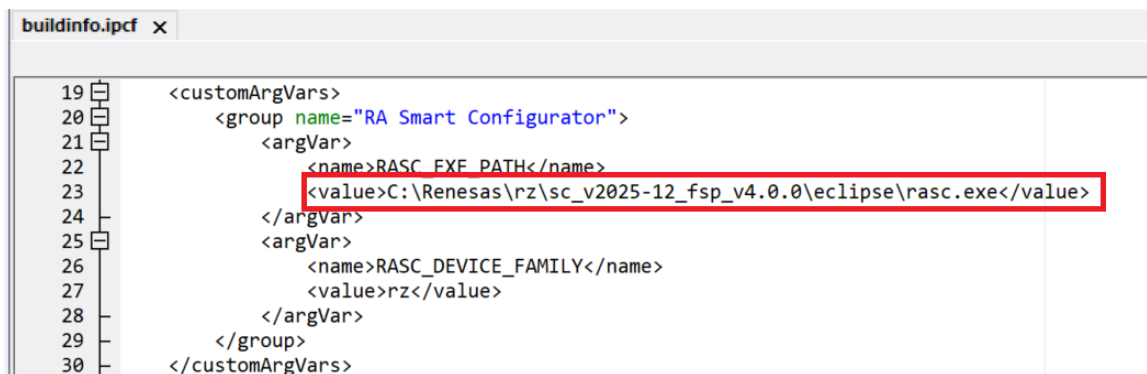


Figure 6.23 RASC_EXE_PATH in buildinfo.ipcf

4. Select the registered menu entry (**RZ Smart Configurator** in this example) to launch the Smart Configurator.

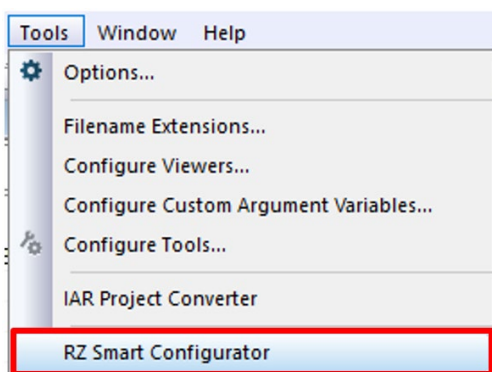


Figure 6.24 Open the Smart Configurator

Note: When using RZ/T2ME-RSK,

if you followed Appendix C: How to Convert the FSP Configuration from RZ/T2M to RZ/T2ME the secondary core FSP configuration is changed automatically.

5. Click **Generate Project Content** to generate the FSP code.

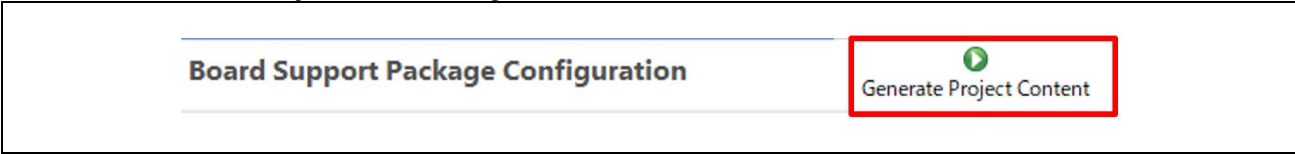


Figure 6.25 Generate Project Content

6. When using the CA55 Secondary core project, modify ecataoe.h as follows:

- **errCode: long** → **UINT32**
- **invokeld: long** → **UINT32**

ecataoe.h (lines 135 to 154)	
Before	After
<pre>typedef struct MBX_STRUCT_PACKED_START { AmsAddr target; AmsAddr sender; UINT16 cmdId; UINT16 stateFlags; UINT32 cbData; long errCode; union { long invokeld; struct { UINT16 fragmentNo; UINT16 packetNo; } frag; } hUser; // long hUser; } MBX_STRUCT_PACKED_END AmsHead;</pre>	<pre>typedef struct MBX_STRUCT_PACKED_START { AmsAddr target; AmsAddr sender; UINT16 cmdId; UINT16 stateFlags; UINT32 cbData; UINT32 errCode; union { UINT32 invokeld; struct { UINT16 fragmentNo; UINT16 packetNo; } frag; } hUser; // long hUser; } MBX_STRUCT_PACKED_END AmsHead;</pre>

7. Click **Make** to build the Secondary core project.

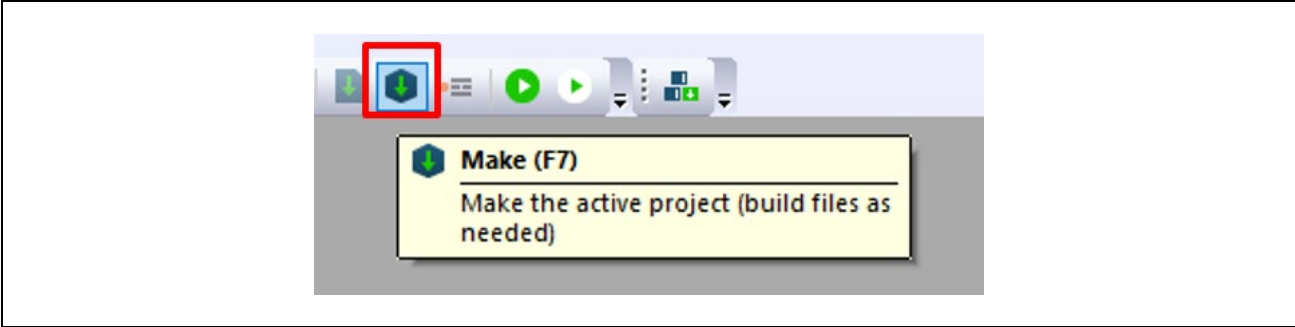


Figure 6.26 Make the Secondary project

6.2.2.3 Downloading and Debugging

1. Close the Secondary core project and reopen the Primary core project.
2. Click **Download and Debug** to start dual-core debugging.
The Secondary core project opens automatically.

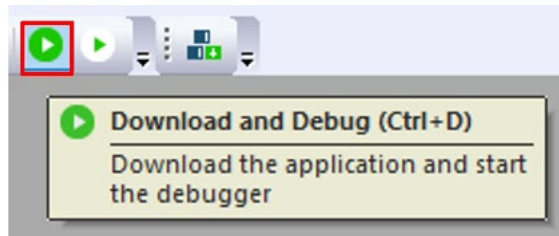


Figure 6.27 Download and Debug

Click **Resume** in the Primary core project.
Then, click the same button in the Secondary core project to start debugging the source code.

Note: If a program is already written to flash memory when you start debugging, it might interfere with the debugging process and cause debugging to fail.

If this happens, erase the flash memory first and then retry.

For details, see "Appendix: How to Erase Flash Memory" in the document [r01an6434ej****](#) listed in Table 1.2.

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**.

Select **Select root directory**, and then click **Browse** to select the following folder:

RZT2L_EtherCAT_RSK_rev0411\project\AoE

Select the following project, and then click **Finish** to import the project:

RZT2L_RSK_ESC_AoE

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

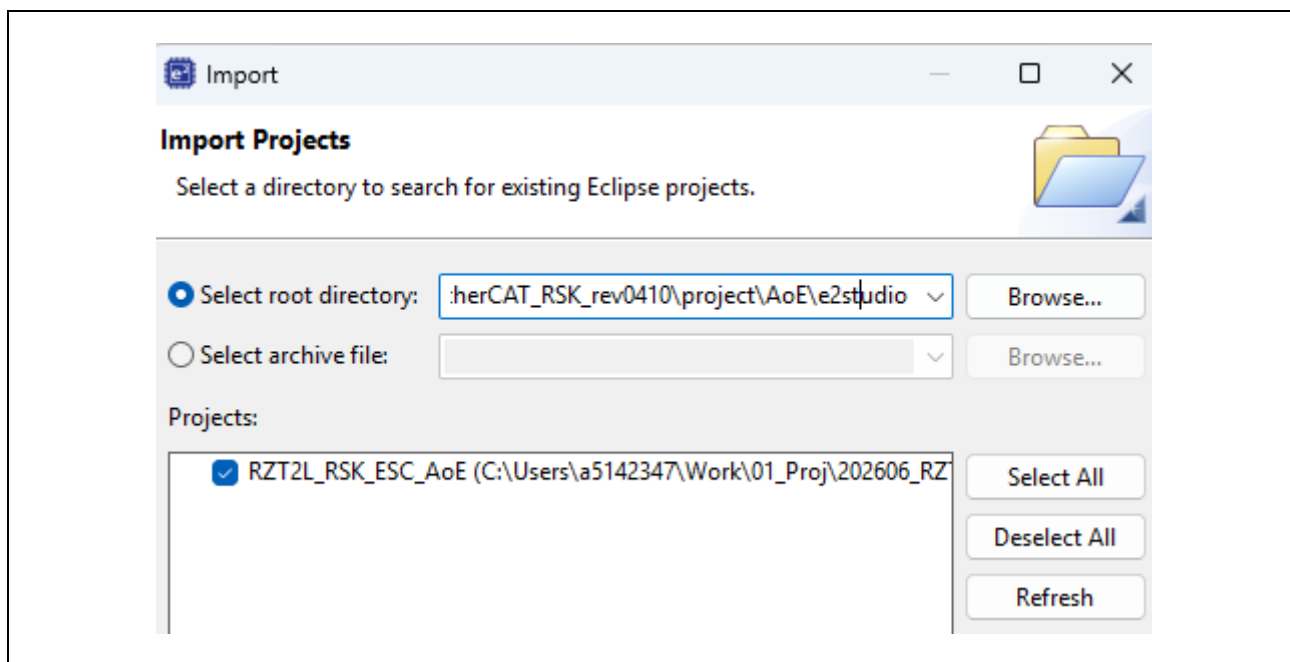


Figure 6.28 Import the project

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

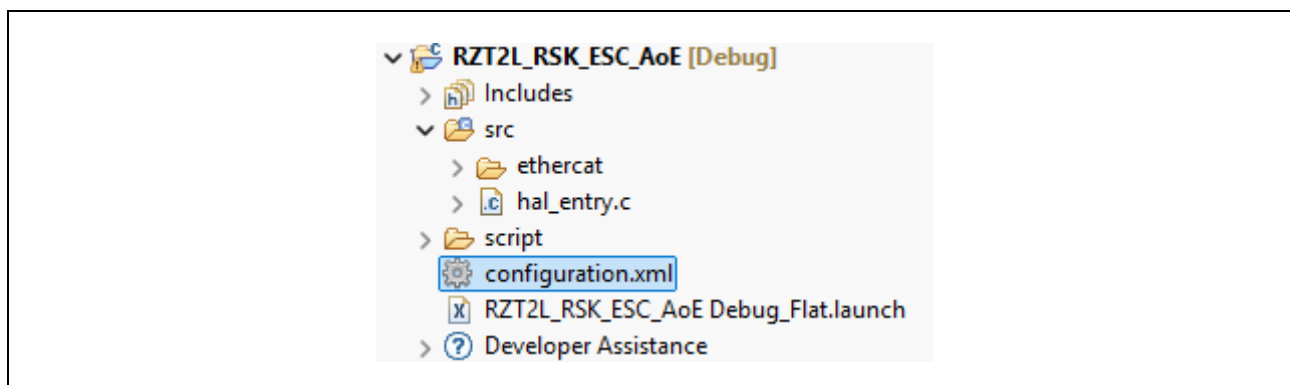


Figure 6.29 Open configuration

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

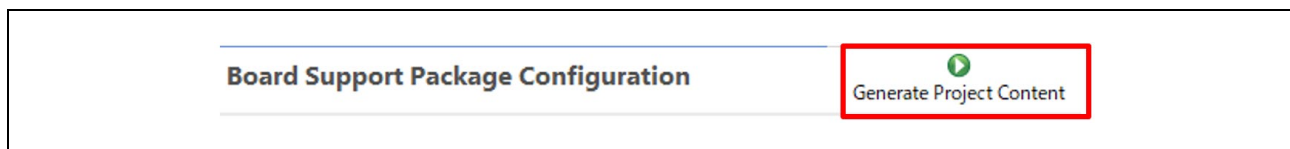


Figure 6.30 Generate Project Content

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

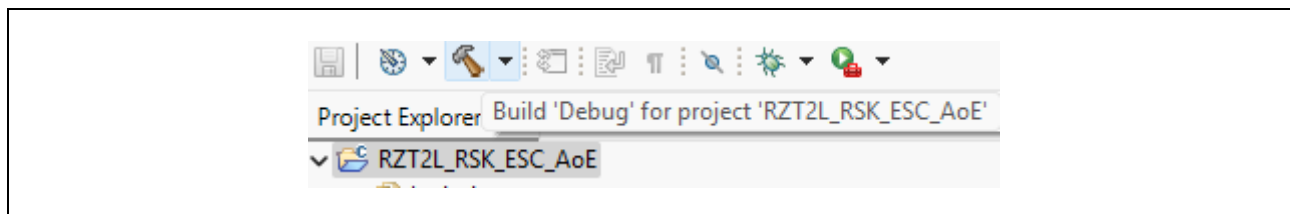


Figure 6.31 Build the project

5. Select the drop-down menu next to the debug icon and select **Debug Configurations...**

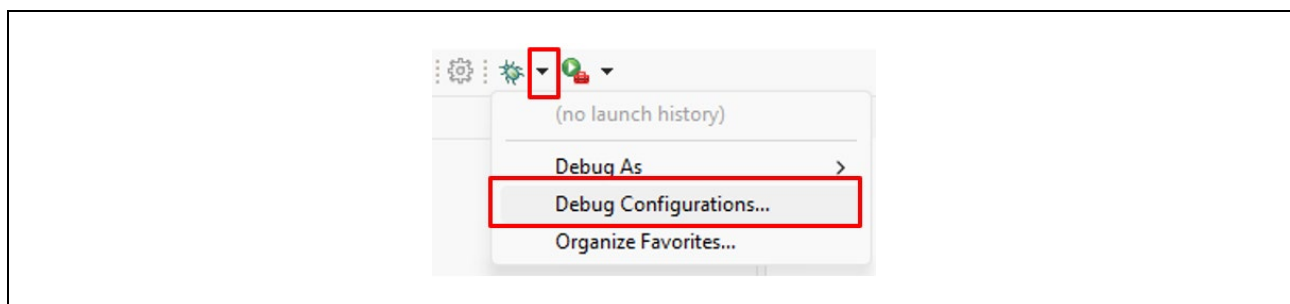


Figure 6.32 Select the drop-down menu

6. Under **Renesas DBG Hardware Debugging**, select RZT2L_RSK_ESC_AoE Debug_Flat and click **Debug**.

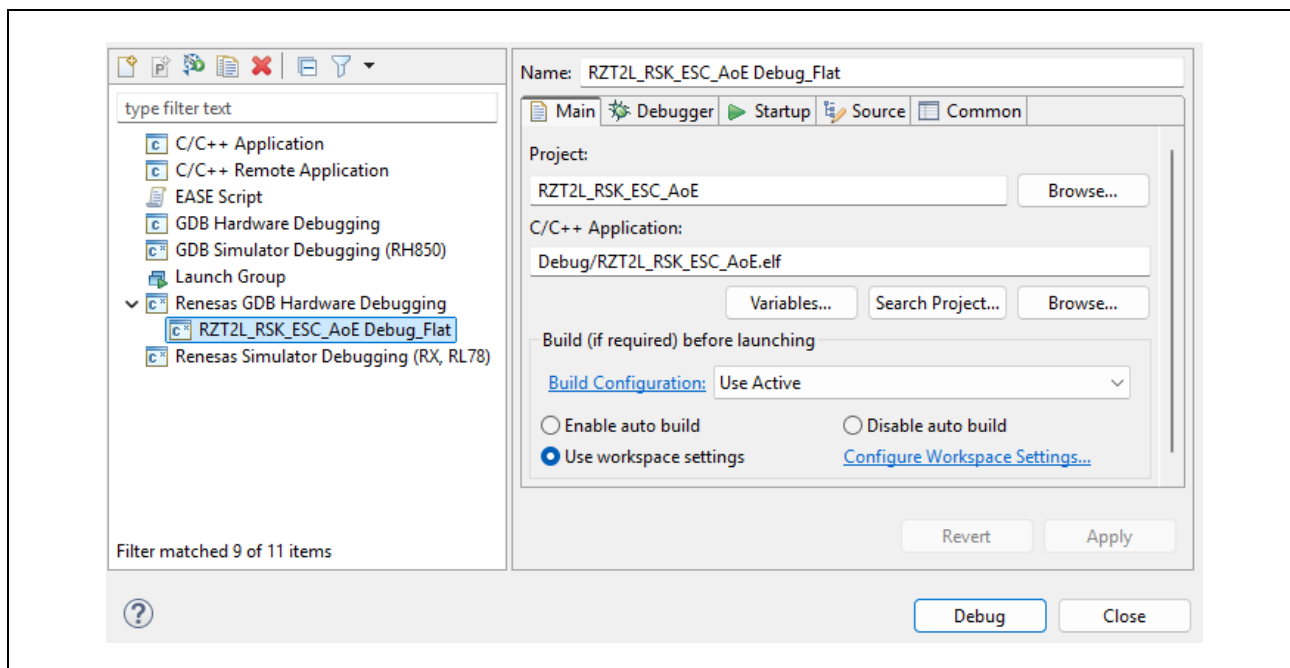


Figure 6.33 Debug Configuration window

Note: A perspective switch dialog box appears. Switch to the debug perspective.

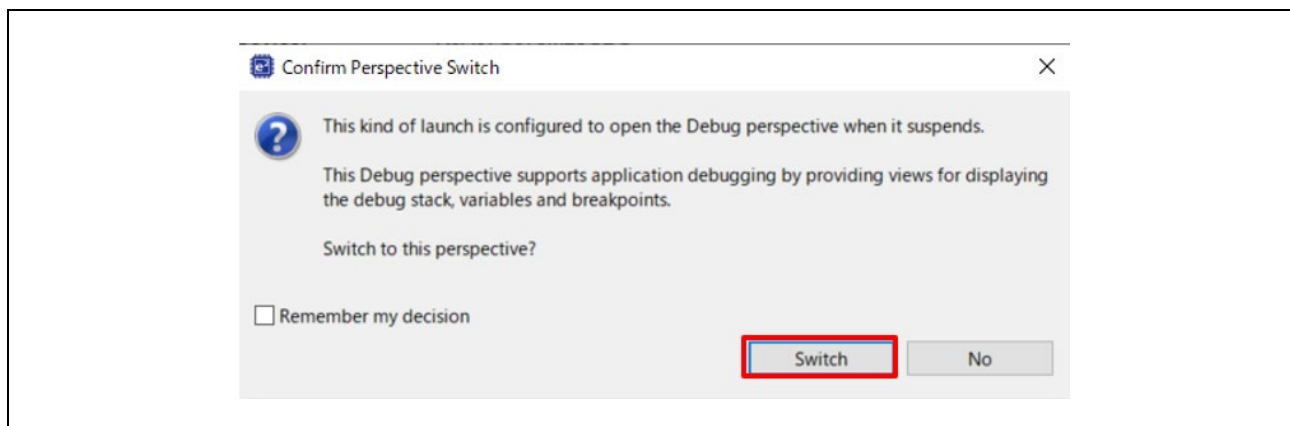


Figure 6.34 Confirm perspective switch dialog box

7. The program is downloaded to RAM and debugging starts. Click **Resume**.

When the program pauses at `hal_entry()` in `main.c`, click **Resume** again to start execution.

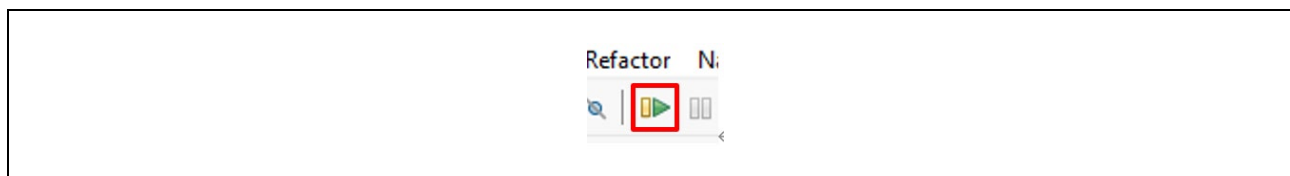


Figure 6.35 Resume button

Note: If a program is already written to flash memory when you start debugging, it might interfere with the debugging process and cause debugging to fail.

If this happens, erase the flash memory first and then retry.

For details, see "Appendix: How to Erase Flash Memory" in the document [r01an6434ej****](#) listed in Table 1.2.

6.3.2 Multi-Core Project (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/T2N, RZ/N2H)

6.3.2.1 Building the Primary Core Project

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

Select **Select root directory**, and then click **Browse** to select the following folder:

RZT2H_EtherCAT_EVB_rev0411\CR52_Dual\project\AoE

Select the following projects, and then click **Finish** to import the projects:

RZT2H_EVB_ESC_AoE_CR52_Dual_Primary

RZT2H_EVB_ESC_AoE_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**.

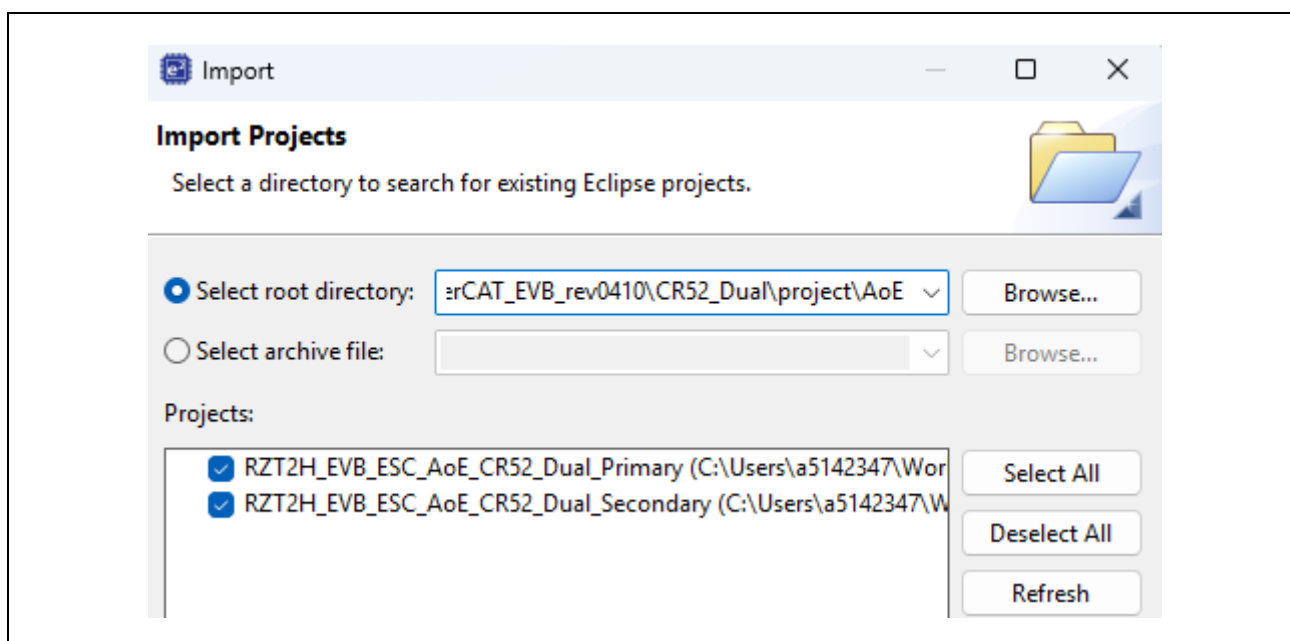


Figure 6.36 Import the projects

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

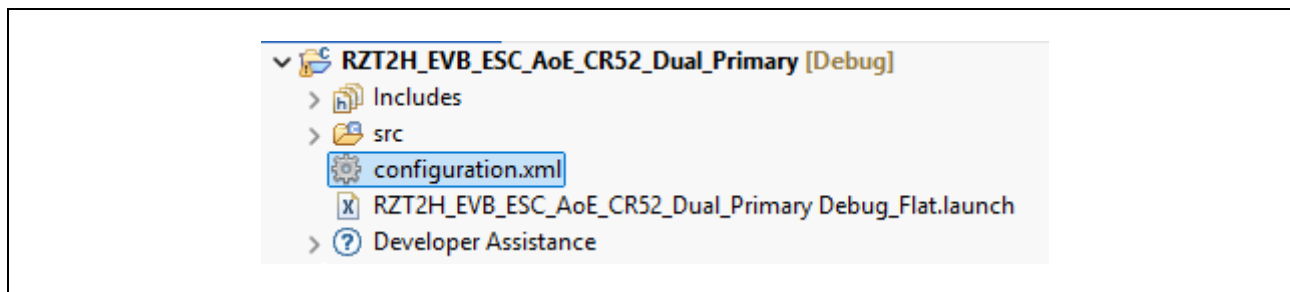


Figure 6.37 Open Primary configuration

Note: When using RZ/T2ME-RSK,

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

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

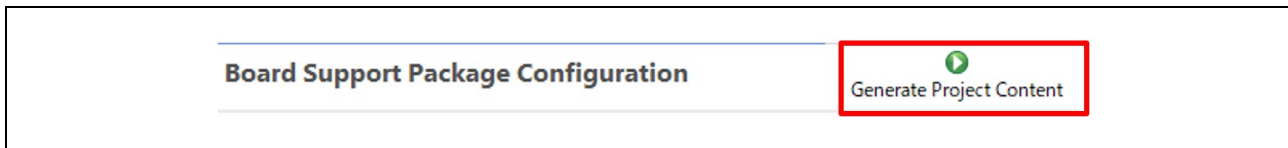


Figure 6.38 Generate Project Content

4. Click the build button (hammer icon) to build the Primary project.

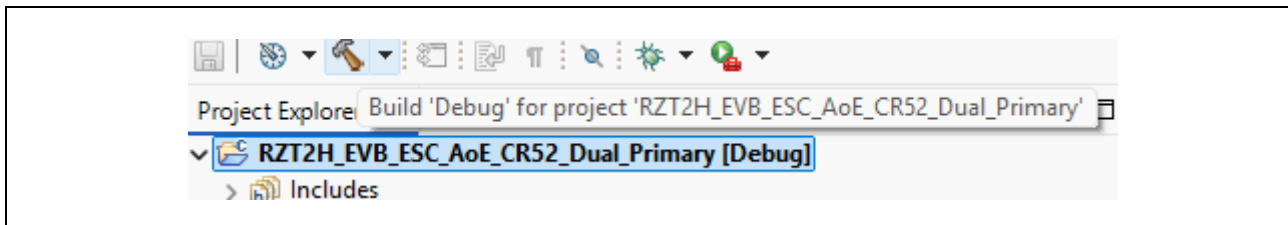


Figure 6.39 Build the Primary project

6.3.2.2 Building the Secondary Core Project

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

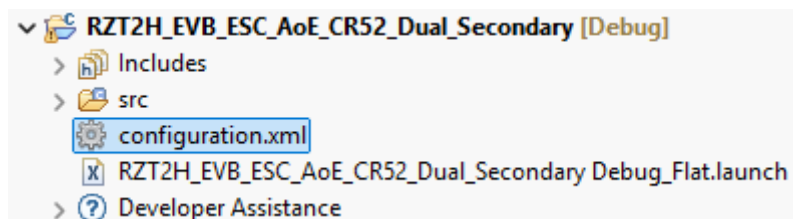


Figure 6.40 Open Secondary configuration

Note: When using RZ/T2ME-RSK,

if you followed Appendix C: How to Convert the FSP Configuration from RZ/T2M to RZ/T2ME the secondary core FSP configuration is changed automatically.

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

Board Support Package Configuration

Generate Project Content

Figure 6.41 Generate Project Content

3. When using the CA55 Secondary core project, modify ecataoe.h as follows:

- **errCode:** long → **UINT32**

- **invokeld:** long → **UINT32**

ecataoe.h (lines 135 to 154)	
Before	After
<pre>typedef struct MBX_STRUCT_PACKED_START { AmsAddr target; AmsAddr sender; UINT16 cmdId; UINT16 stateFlags; UINT32 cbData; long errCode; union { long invokeld; struct { UINT16 fragmentNo; UINT16 packetNo; } frag; } hUser; // long hUser; } MBX_STRUCT_PACKED_END AmsHead;</pre>	<pre>typedef struct MBX_STRUCT_PACKED_START { AmsAddr target; AmsAddr sender; UINT16 cmdId; UINT16 stateFlags; UINT32 cbData; UINT32 errCode; union { UINT32 invokeld; struct { UINT16 fragmentNo; UINT16 packetNo; } frag; } hUser; // long hUser; } MBX_STRUCT_PACKED_END AmsHead;</pre>

- Click the build button (hammer icon) to build the Secondary project.

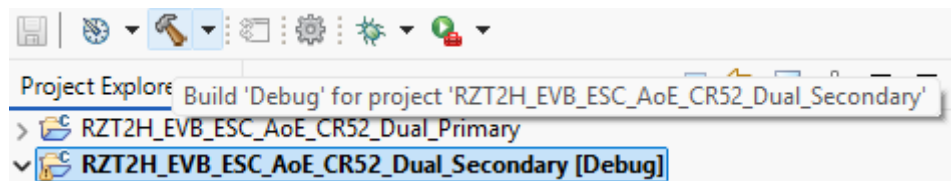


Figure 6.42 Build the Secondary project

6.3.2.3 Downloading and Debugging for Dual Core

- Select the drop-down menu next to the debug icon and select **Debug Configurations...**

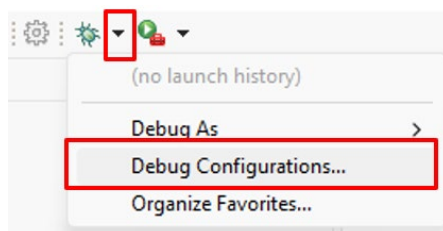


Figure 6.43 Select the drop-down menu

- Under **Renesas DBG Hardware Debugging**, select **RZT2H_EVB_ESC_AoE_CR52_Dual_Primary Debug_Flat** and click **Debug**.

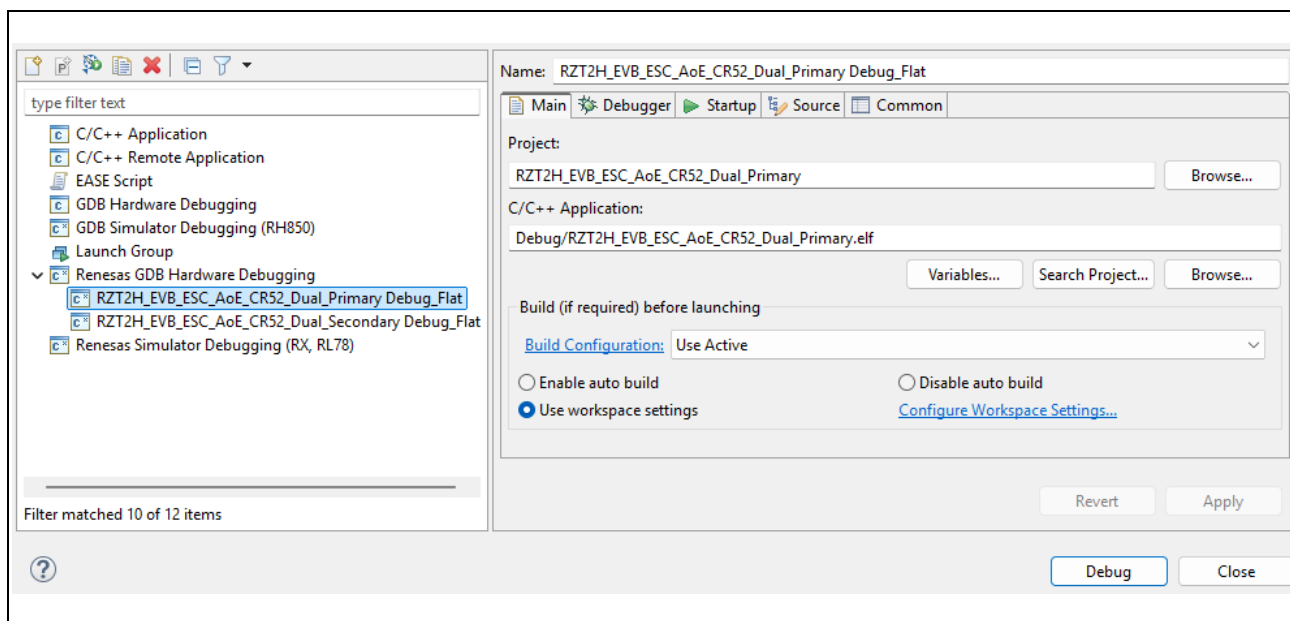


Figure 6.44 Select Primary Debug Configuration

Note: A perspective switch dialog box appears. Switch to the debug perspective.

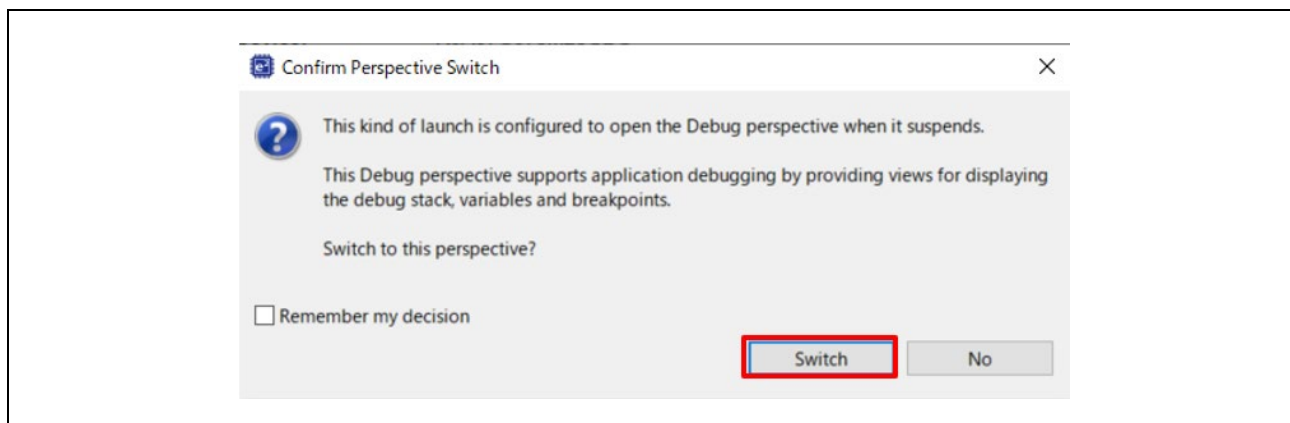


Figure 6.45 Select the drop-down menu

- The Primary project program is downloaded to RAM, and debugging starts. Click the triangle next to the **Debug** button again, and then select **Debug Configurations**.

Select **Renesas DBG Hardware Debugging** →

RZT2H_EVB_ESC_AoE_CR52_Dual_Secondary Debug_Flat item, then click **Debug**.

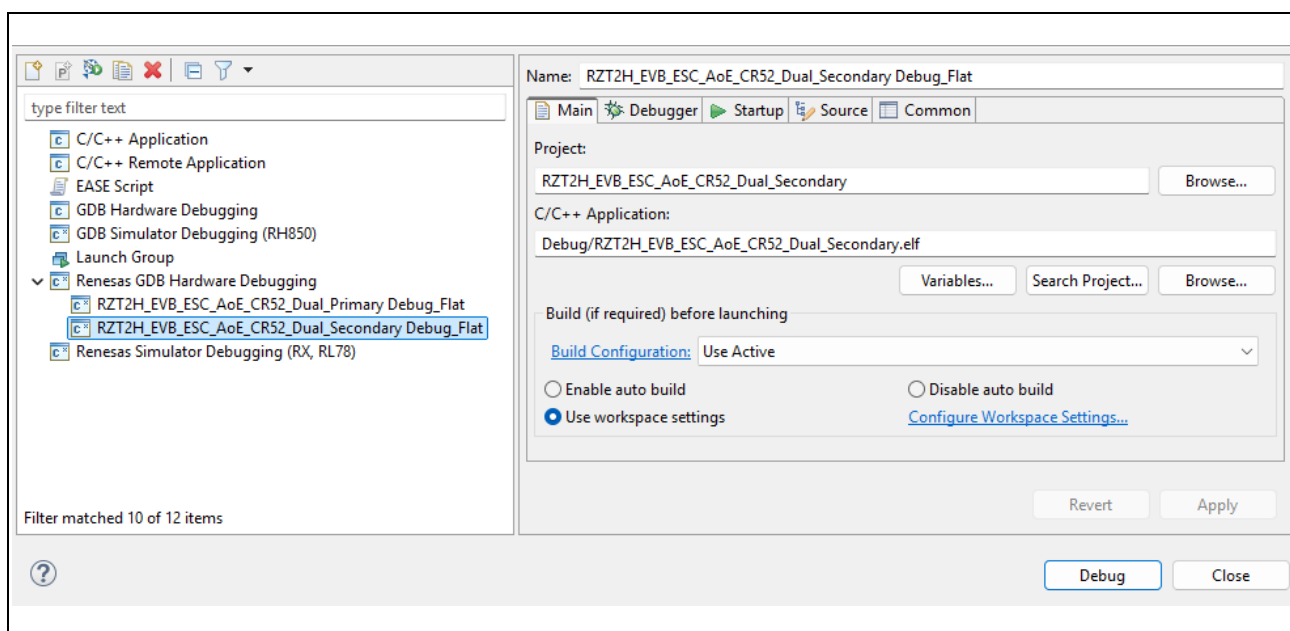


Figure 6.46 Select Secondary Debug Configuration

Note: When the following dialog box appears, click **No**.

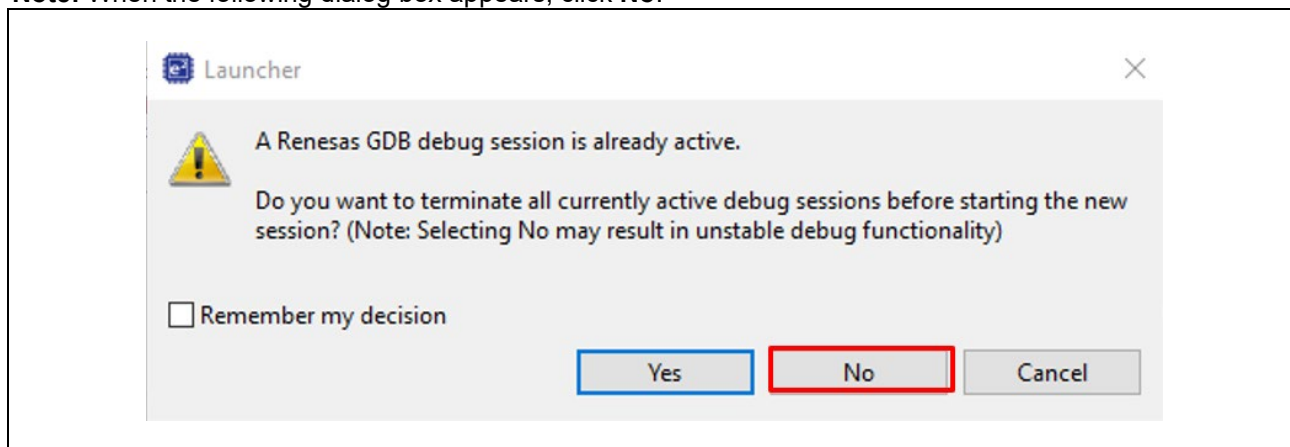


Figure 6.47 Already active dialog box

Note: When the following dialog box appears again, click **Yes**.

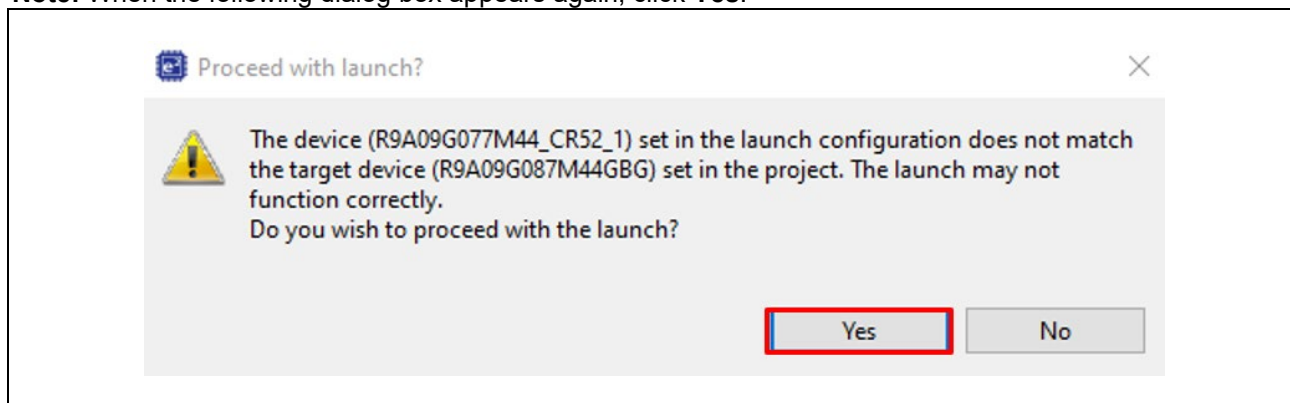


Figure 6.48 Proceed with launch dialog box

4. The Secondary project program is downloaded to RAM, and debugging starts. Click **Resume** in the Primary and Secondary core projects to start debugging the source code.

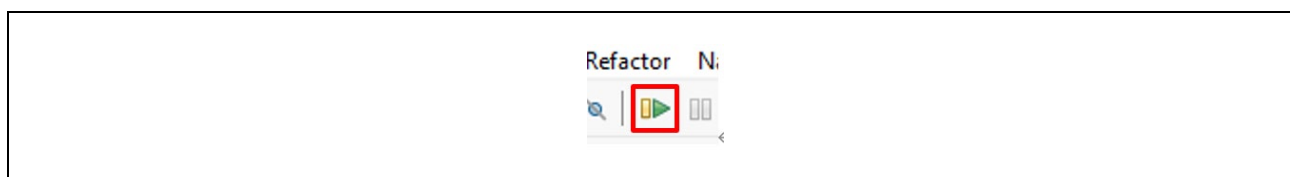


Figure 6.49 Resume button

Note: If a program is already written to flash memory when you start debugging, it might interfere with the debugging process and cause debugging to fail.

If this happens, erase the flash memory first and then retry.

For details, see "Appendix: How to Erase Flash Memory" in the document [r01an6434ej****](#) listed in Table 1.2.

7. Demonstration of the Sample Application

Start TwinCAT 3 as follows:

From the Start menu, select **Beckhoff** → **TwinCAT3** → **TwinCAT XAE (VS2013)**.

After TwinCAT starts, select **File** → **New** → **Project** to create a new project of the TwinCAT XAE Project type. The following sections describe the subsequent procedures.

7.1 Scanning I/O Devices

- In Solution Explorer, under **I/O** → **Devices**, select **Scan**.

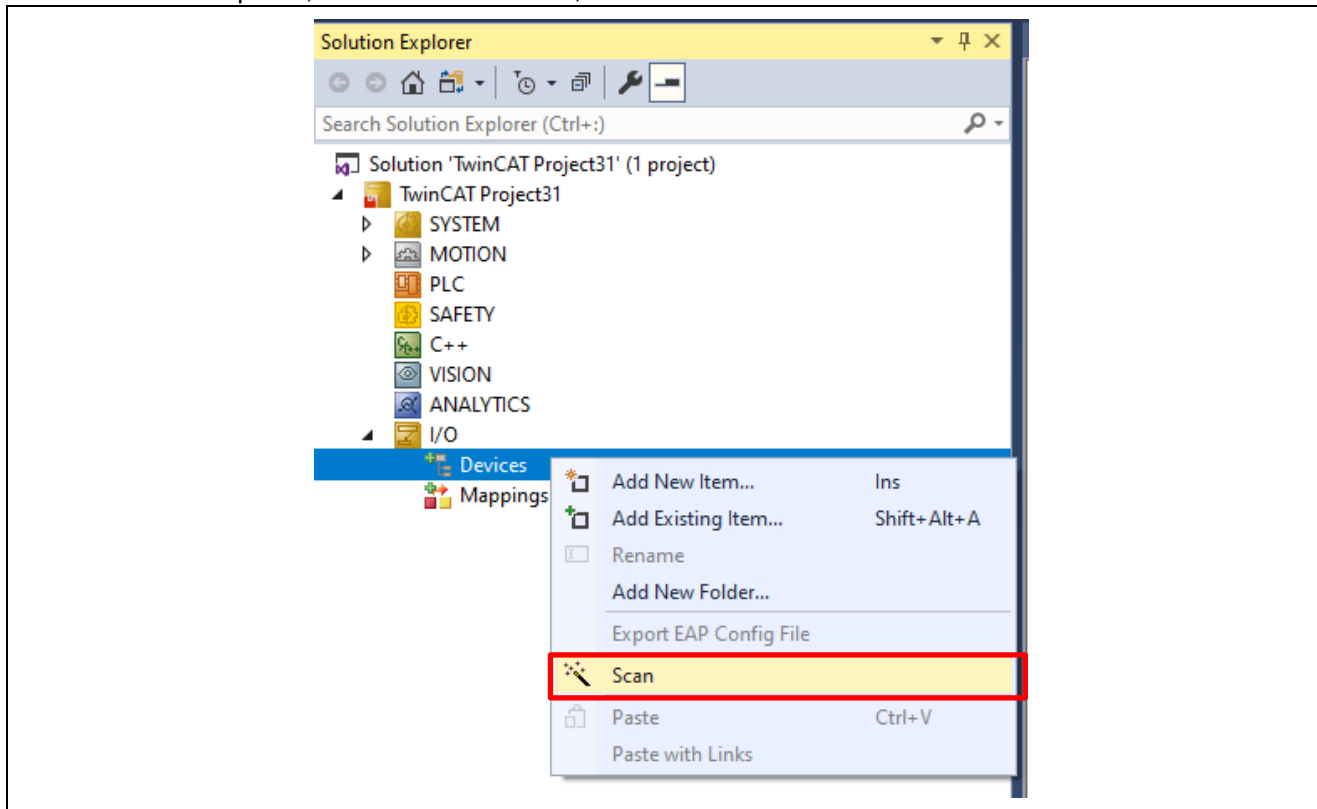


Figure 7.1 Scan devices

2. The EtherCAT port is displayed. Select the port and click **OK**.

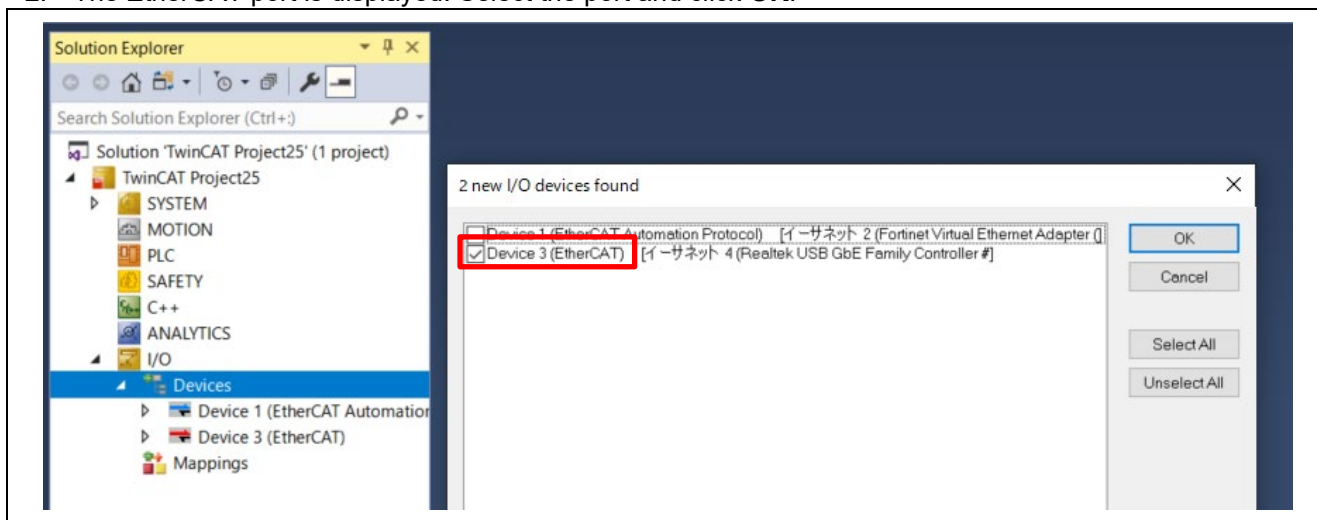


Figure 7.2 EtherCAT port is displayed

Note: If a valid EtherCAT SubDevice exists in the network, TwinCAT displays the candidate with a check box.

3. Start scanning the SubDevice.

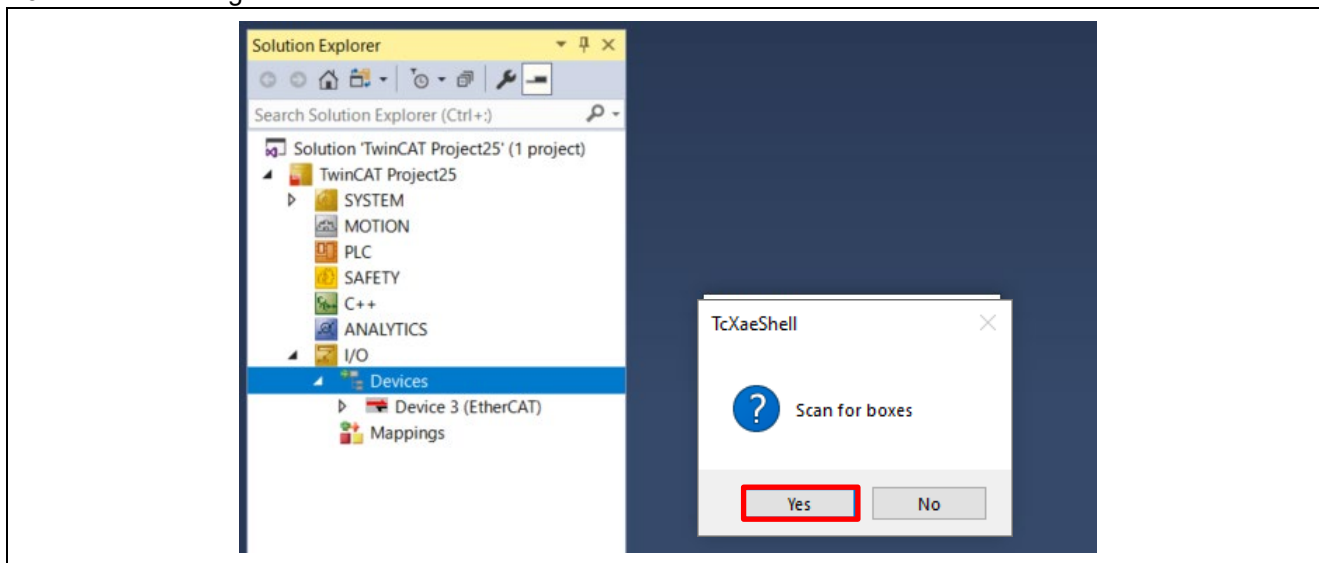


Figure 7.3 Scan SubDevice

4. The SubDevice is listed in the boxes. Click **Yes** in the Activate free run dialog box.

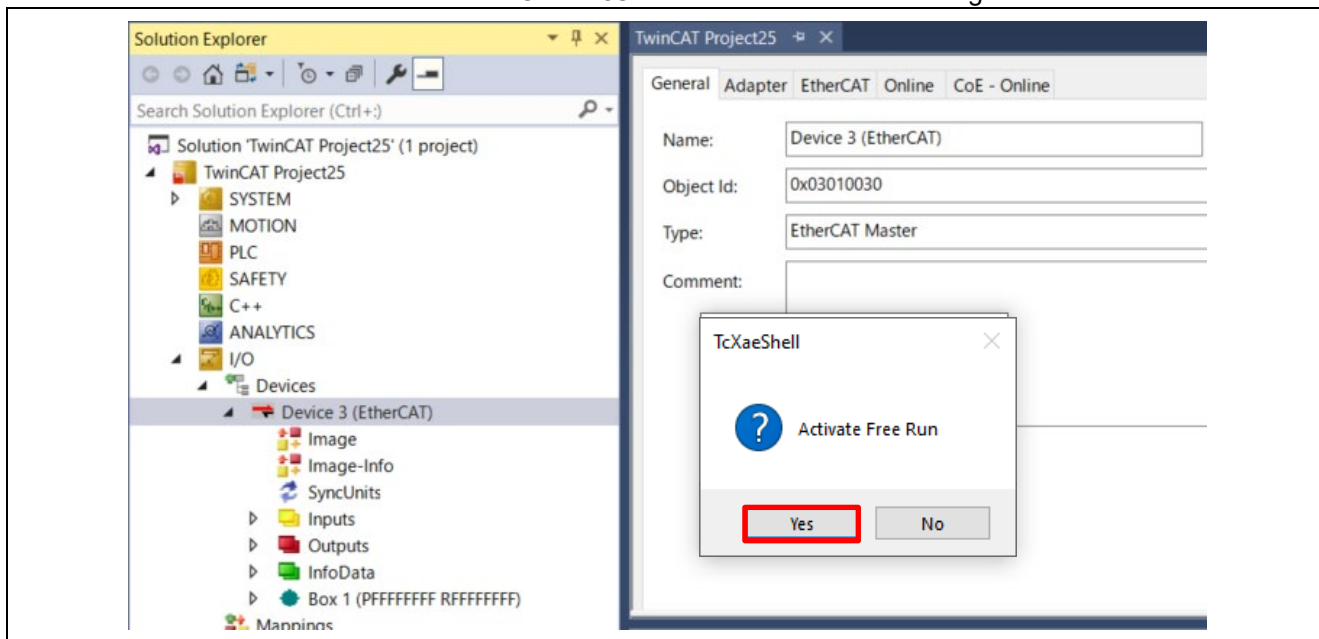


Figure 7.4 Activate free run

7.2 Updating EEPROM Data

If the data of another application has already been written to the EEPROM, replace the data.
Use the following procedure to replace the EEPROM data:

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 ESI selection by target device

Target device	ESI
RZ/T2L-RSK	Renesas EtherCAT RZ/T2 AoE 2port
RZ/T2M-RSK, RZ/T2ME-RSK, RZ/T2H EVB, RZ/T2N EVB	Renesas EtherCAT RZ/T2 AoE 3port
RZ/N2L-RSK, RZ/N2H EVB	Renesas EtherCAT RZ/N2 AoE 3port

6. Click **OK**, and then click **Download**.

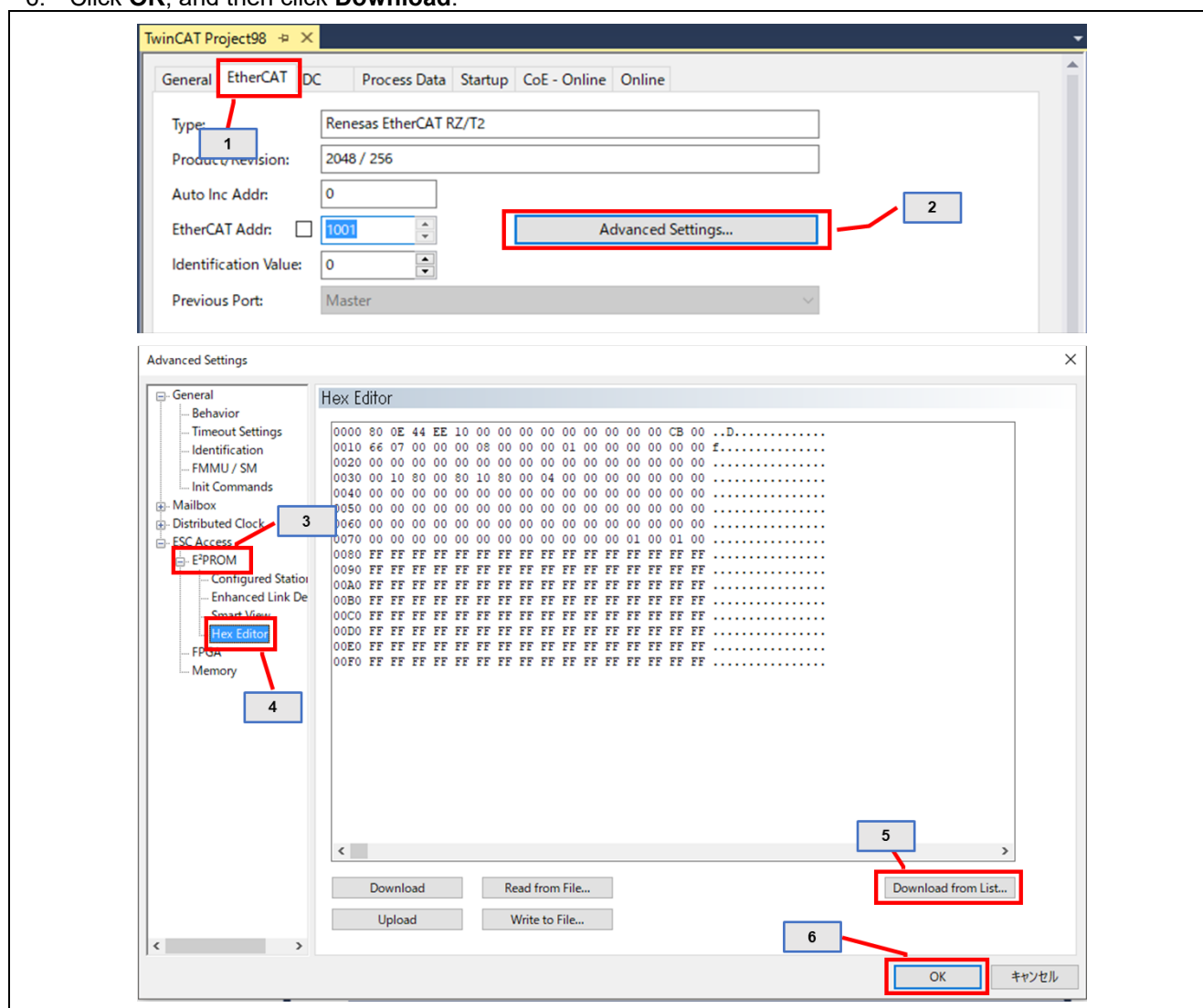


Figure 7.5 Download ESI to EEPROM

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

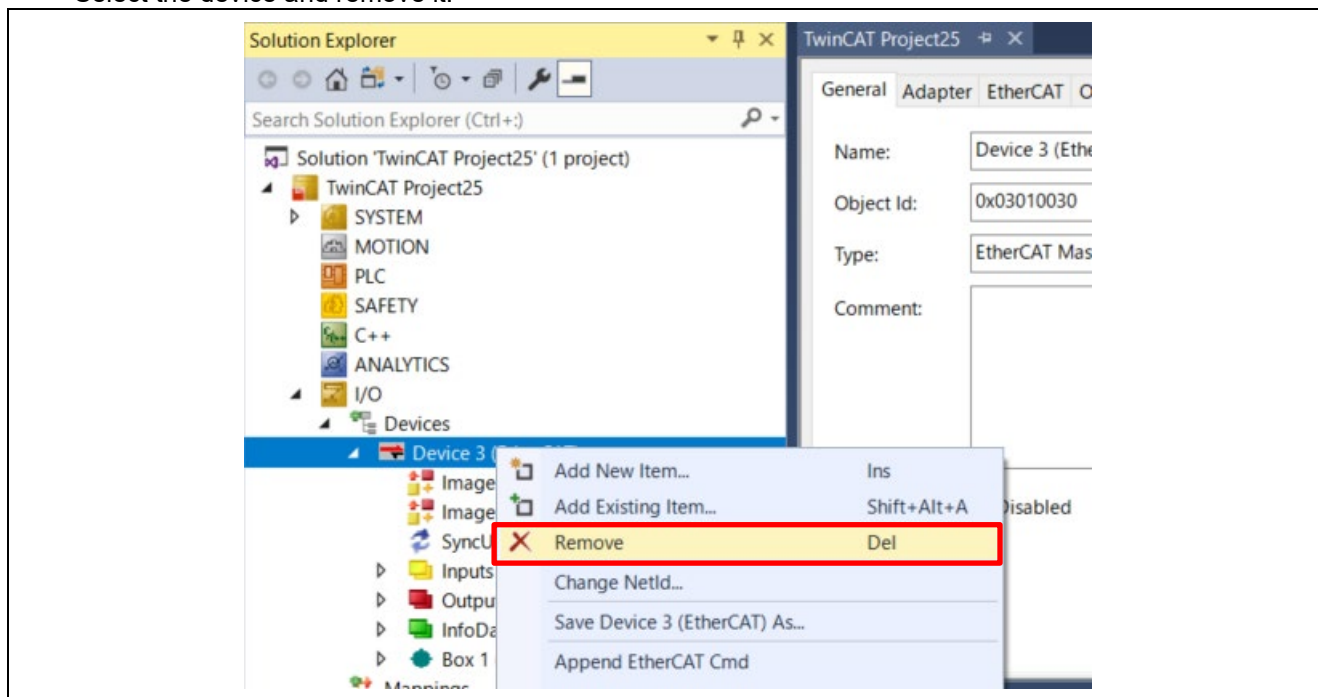


Figure 7.6 Remove the device to apply the settings

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

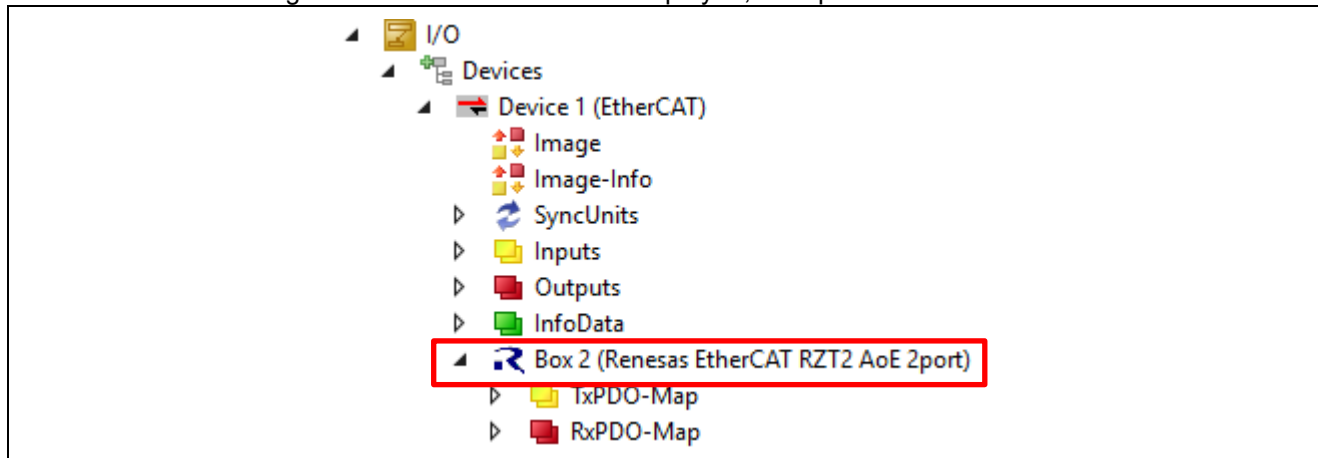


Figure 7.7 The desired ESI file (example of RZ/T2L)

Option A: Create an ESI binary file from the ESI XML file and download the binary file.

1. In the SSC Tool, select **Tool** → **EEPROM Programmer**.
2. Select **FILE** → **OPEN**, and then browse to and select the ESI file.
3. Select **FILE** → **Save AS**, and then select binary as the file type.
4. A binary file is generated in the specified folder.
5. Click **Read from File**, select the ESI binary file, and then click **Download**.
6. Use the **Upload** option to confirm the write status.

After the data is replaced, restart the board (power it down and power it up again, or reset it) so that the new data is applied to the device. Then, select **Restart TwinCAT System**.

7.3 Evaluating the I/O Controller

7.3.1 Output Verification

1. In the TwinCAT 3 System Manager tree, select Output Counter → Online.
2. Click **Write** and enter a desired value.
3. The on-board LEDs light according to the written value.

Note: The number of controllable LEDs and the available bit width depend on the target board.

Table 7.2 LED assignment by target device

Target device	Output LEDs	Bit width
RZ/T2L-RSK	LED1, LED3	2-bit
RZ/T2M-RSK, RZ/T2ME-RSK	LED0, LED1, LED2, LED3	4-bit
RZ/N2L-RSK	LED0, LED1, LED2, LED3	4-bit
RZ/T2H EVB	LED0, LED1, LED2, LED3	4-bit
RZ/N2H EVB	LED10, LED11	2-bit
RZ/T2N EVB	LED8, LED9, LED10, LED11	4-bit

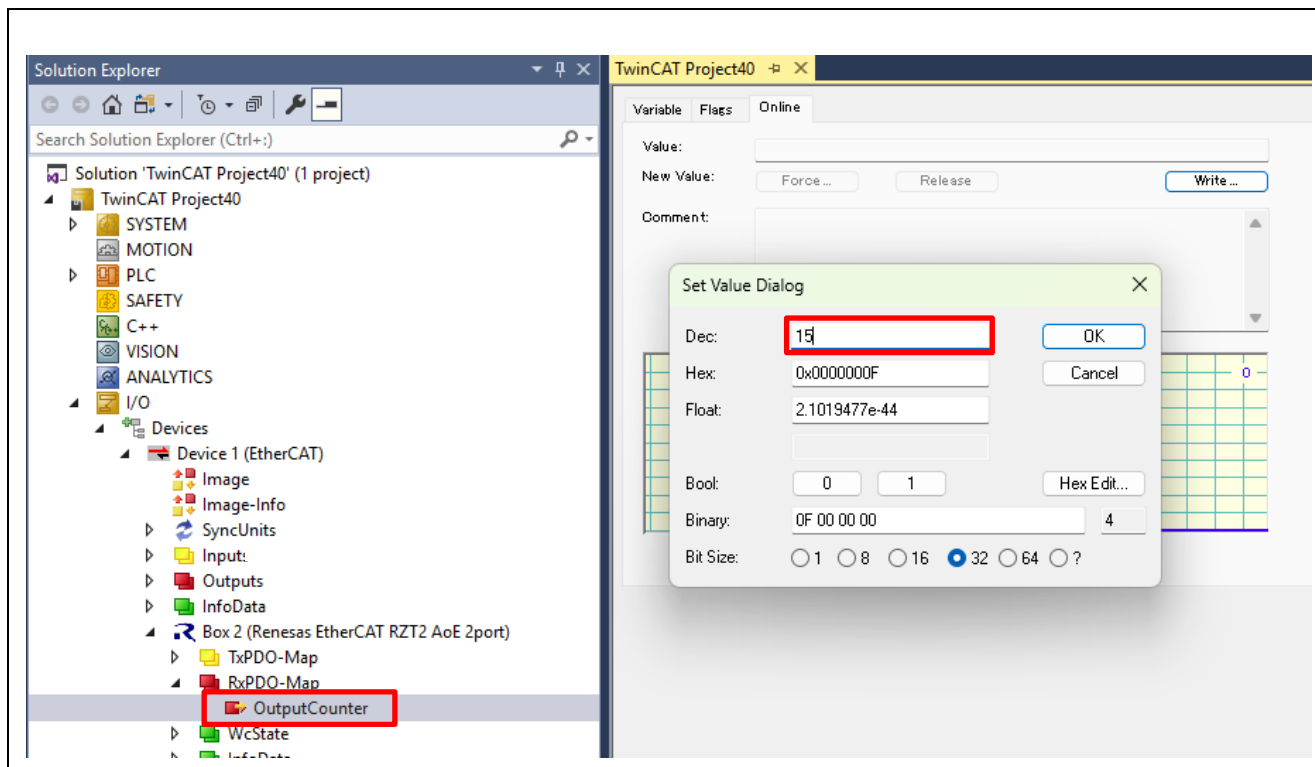


Figure 7.8 Output LED

7.3.2 Input Verification

1. In the TwinCAT 3 System Manager tree, select InputCounter → Online.
2. The current state of the on-board **push switch** is displayed in the Value field as a multi-bit input value.

Note: The assignments for the push switches depend on the target board.

Table 7.3 Push switch assignment by target device

Target device	Push switch name	Bit width
RZ/T2L-RSK	SW1, SW2, NMI	3-bit
RZ/T2M-RSK, RZ/T2ME-RSK	SW1, SW2, NMI	3-bit
RZ/N2L-RSK	SW1, SW2, NMI	3-bit
RZ/T2H EVB	SW9, SW10, SW11	3-bit
RZ/N2H EVB	SW2, SW3, SW4	3-bit
RZ/T2N EVB	SW2, SW3, SW4	3-bit

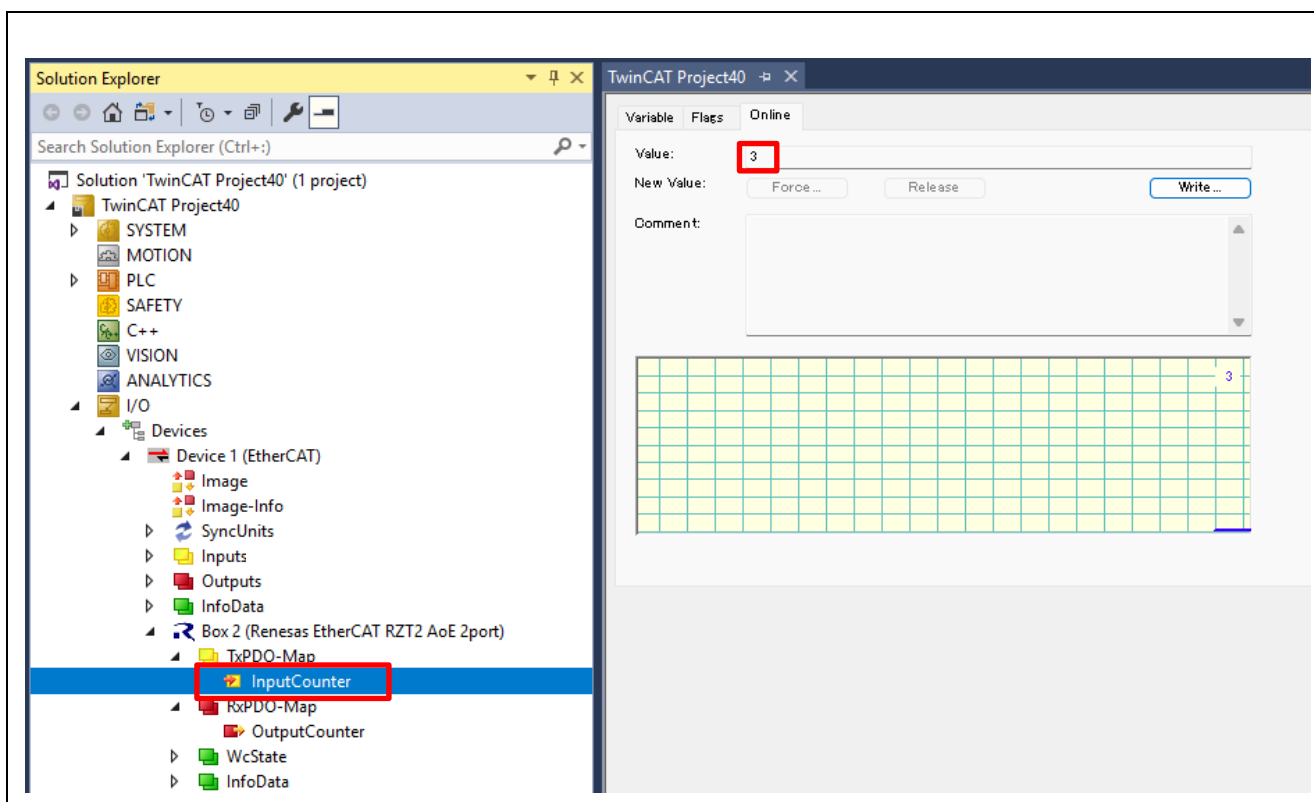


Figure 7.9 Push switch input

7.4 Operation Check of AoE Communication

With this sample program, you can access the Object Dictionary of the EtherCAT SubDevice by executing ADS Read and ADS Write operations from the AoE-Online tab in TwinCAT 3. For more information about the Automation Device Specification (ADS), see [ADS in the Beckhoff Information System](#).

7.4.1 Preparation

1. Double-click **Box 1** in the System Manager tree to display a panel on the right side of the screen.
2. Click the **Online** tab and check that Current Status changes to OP.
3. Click the **AoE-Online** tab.

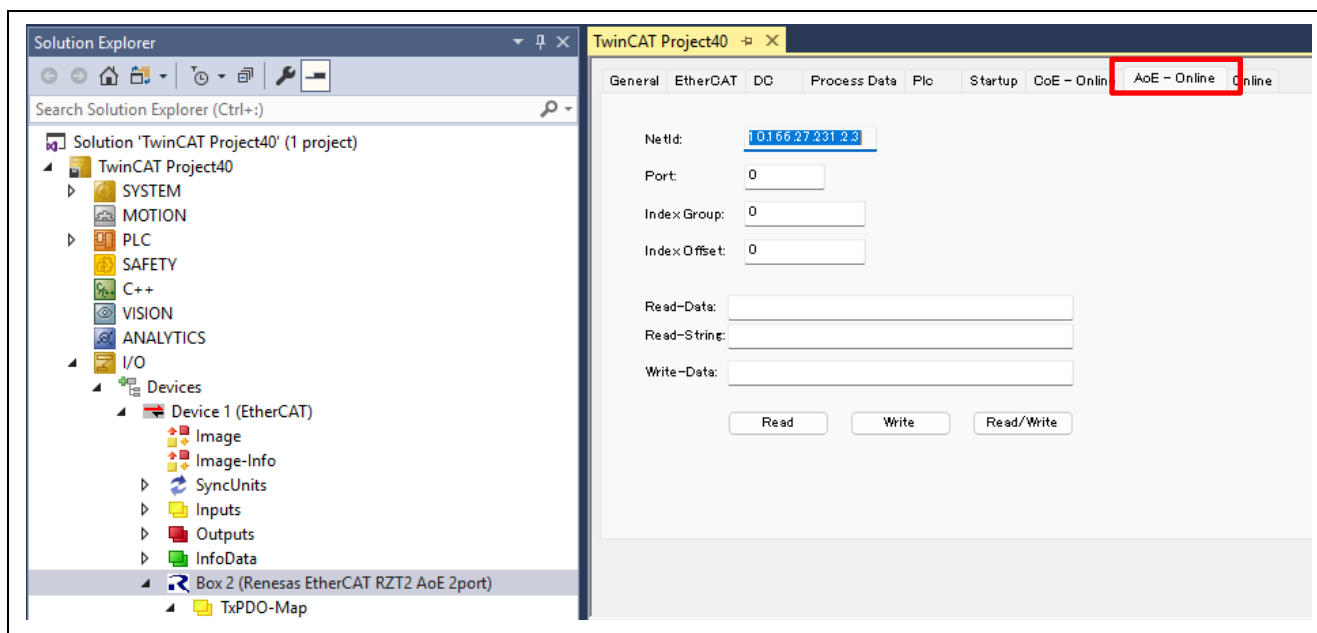


Figure 7.10 Open AoE-Online

7.4.2 ADS Read

Perform an ADS Read operation to read the device name from Object Dictionary entry 0x1008:0.

1. Enter the following values in the **AoE-Online** tab:

Settings	Value	Description
Port	65535	AoE port number
Index Group	0xF302	Object Dictionary access
Index Offset	0x10080000	Object Dictionary entry 0x1008:0

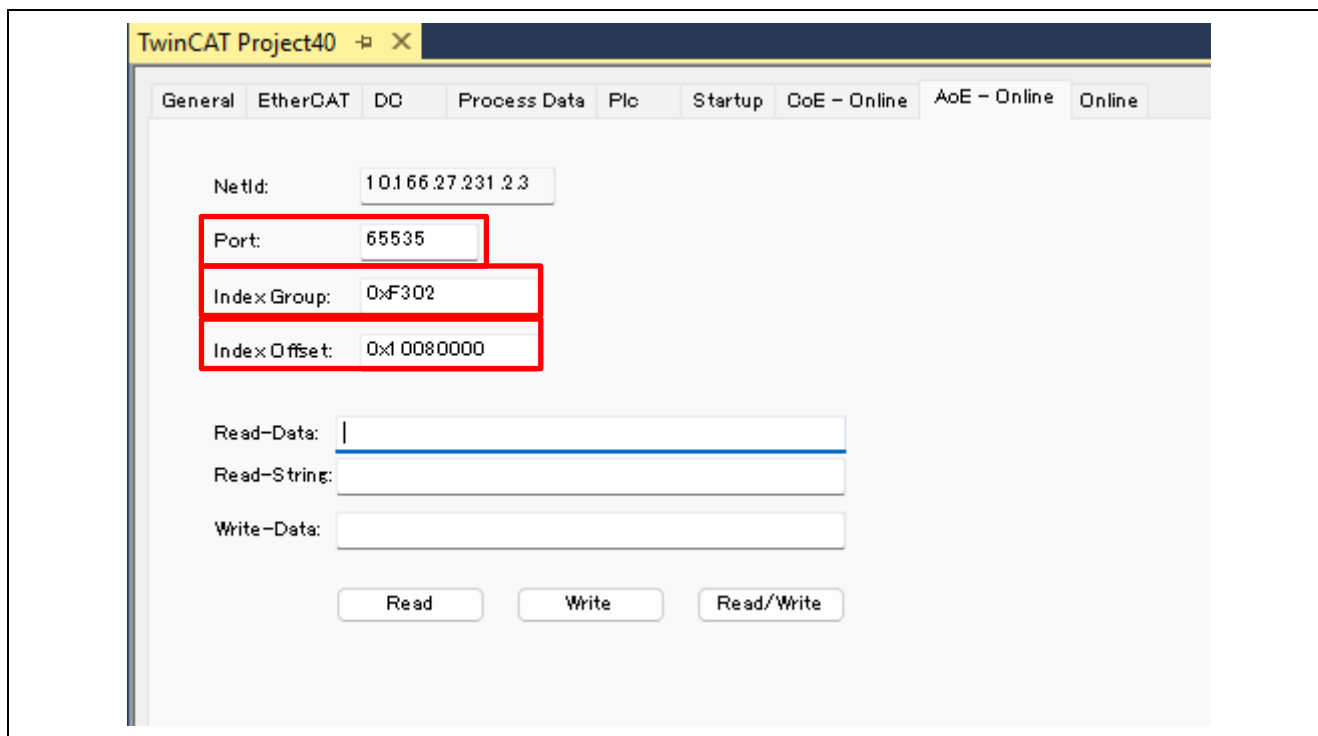


Figure 7.11 Setting for ADS Read

2. Click **Read** to execute the ADS Read operation.
3. Confirm that the value of Object Dictionary entry 0x1008:0 (Device name) is displayed in the read result.

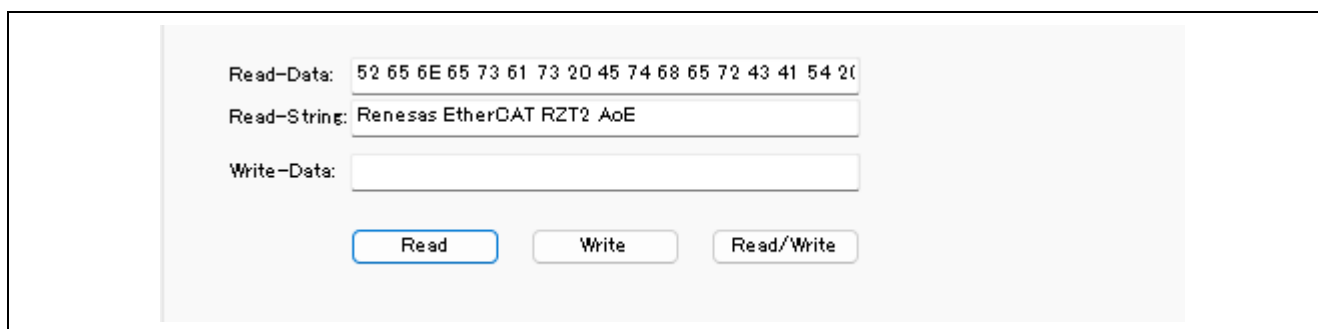


Figure 7.12 ADS Read from Object Dictionary entry 0x1008:0

7.4.3 ADS Write

Perform an ADS Write operation to modify Object Dictionary entry 0x10F1:2 (Sync Error Counter Limit).

- 1. Enter the following values in the **AoE-Online** tab:

Settings	Value	Description
Port	65535	AoE port number
Index Group	0xF302	Object Dictionary access
Index Offset	0x10F10002	Object Dictionary entry 0x10F1:2

GeneralEtherCATDCProcess DataPlcStartupCoE - OnlineAoE - OnlineOnline

NetId:10.166.27.231.2.3

Port:65535

Index Group:0xF302

Index Offset:0x10F10002

Read-Data:

Read-String:

Write-Data:

ReadWriteRead/Write

Figure 7.13 Setting for ADS Write

- 2. Click **Read** to execute the ADS Read operation.
- 3. Confirm the value of Object Dictionary entry 0x10F1:2 (Sync Error Counter Limit).

NetId:10.166.27.231.2.3

Port:65535

Index Group:0xF302

Index Offset:0x10F10002

Read-Data:04 00

Read-String:

Write-Data:

ReadWriteRead/Write

Figure 7.14 ADS Read from Object Dictionary entry 0x10F1:2

- Enter 07 00 in the Write-data area.
- Click **Write** to execute the ADS Write operation.

The screenshot shows the 'EtherCAT' tab in the software interface. The 'DC' sub-tab is active. The 'Write-Data' field is highlighted with a red rectangle and contains the value '07 00'. The 'Write' button is also highlighted with a red rectangle. Other fields include 'NetId: 10.166.27.231.23', 'Port: 65535', 'Index Group: 0xF302', and 'Index Offset: 0x10F10002'. The 'Read-Data' and 'Read-String' fields are empty. The 'Read' and 'Read/Write' buttons are also visible.

Figure 7.15 ADS Write to Object Dictionary entry 0x10F1:2

- Click **Read** again to confirm the value of Object Dictionary entry 0x10F1:2.

The screenshot shows the same 'EtherCAT' tab and 'DC' sub-tab. The 'Read-Data' field is highlighted with a red rectangle and contains the value '07 00'. The 'Read' button is also highlighted with a red rectangle. The 'Write-Data' field now contains '07 00'. The 'NetId', 'Port', 'Index Group', and 'Index Offset' fields remain the same as in the previous screenshot.

Figure 7.16 Confirm the value of Object Dictionary entry 0x10F1:2

8. Appendix

8.1 Appendix A: Precautions for Using the SSC Tool

※ When opening the SSC Tool

1. Open the SSC Tool as an administrator.
Otherwise, the SSC code generation might fail.
2. When you open the SSC Tool for the first time, the following window might appear.
Select **No** to skip checking for updates.

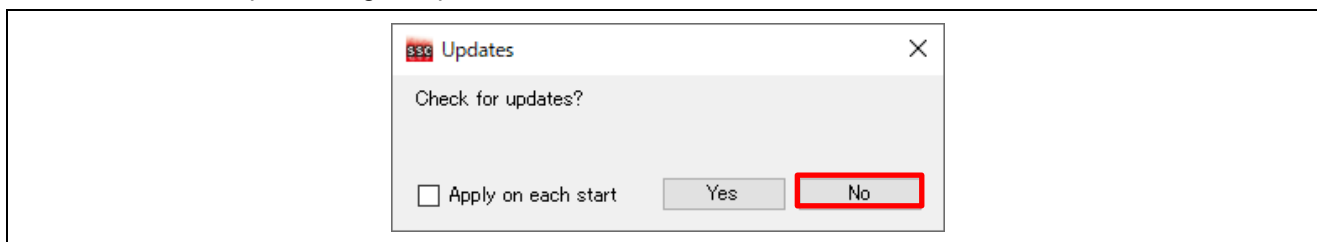


Figure 8.1 The update window

※ SSC Tool settings

3. Select **Tool > Options** to confirm the SSC Tool settings.

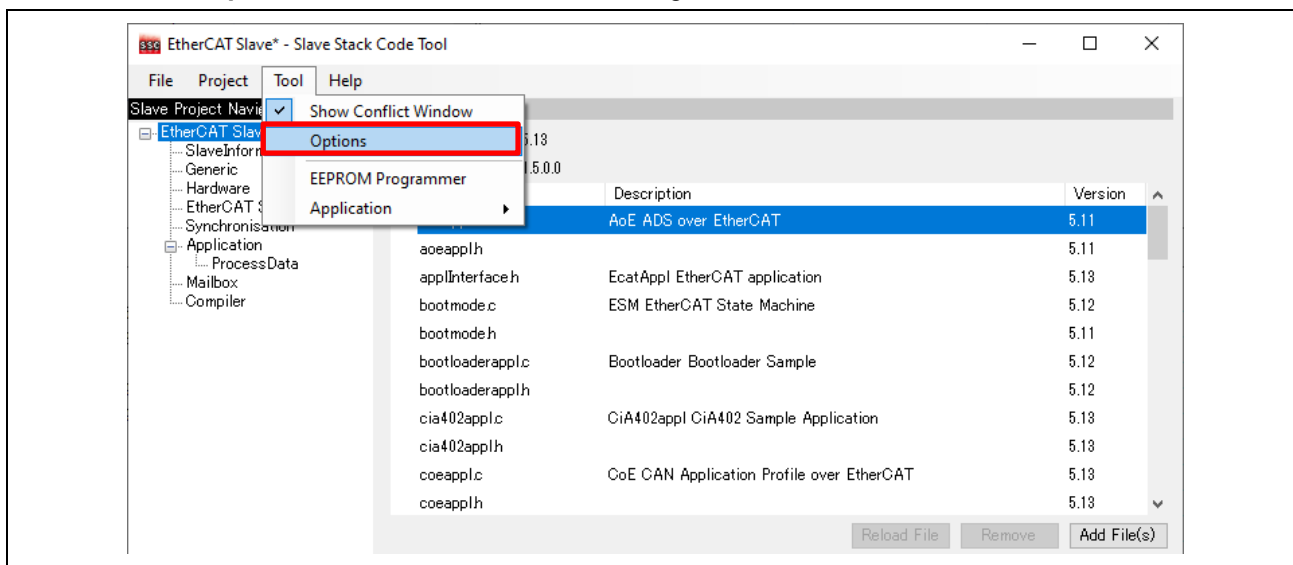


Figure 8.2 SSC Tool main window

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

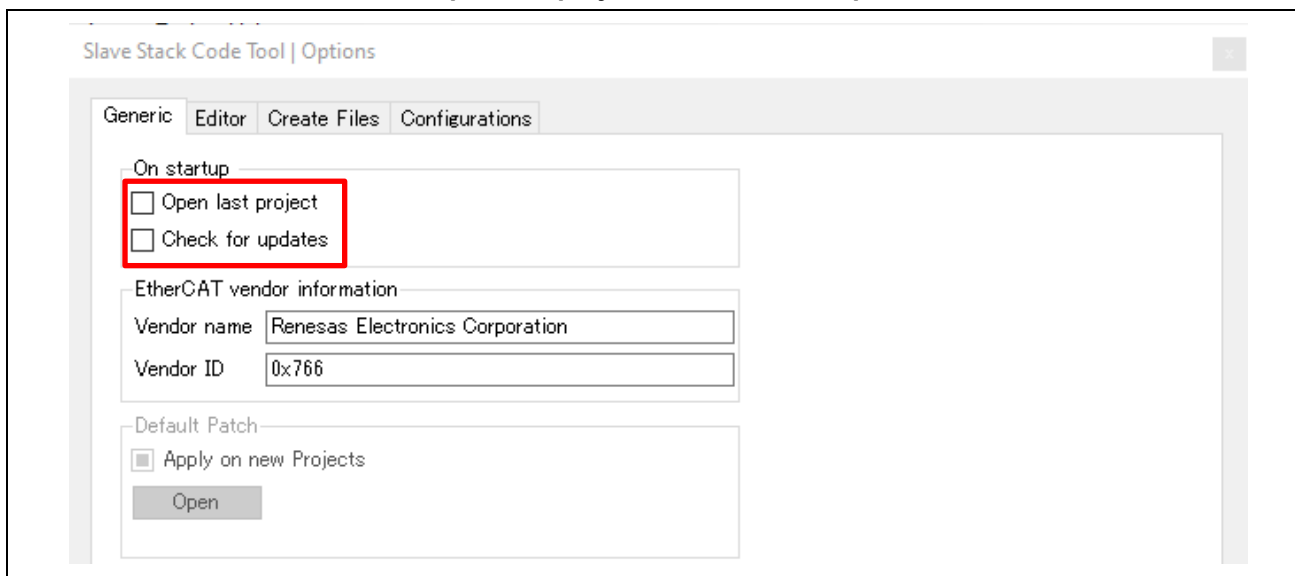


Figure 8.3 Option settings (1)

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

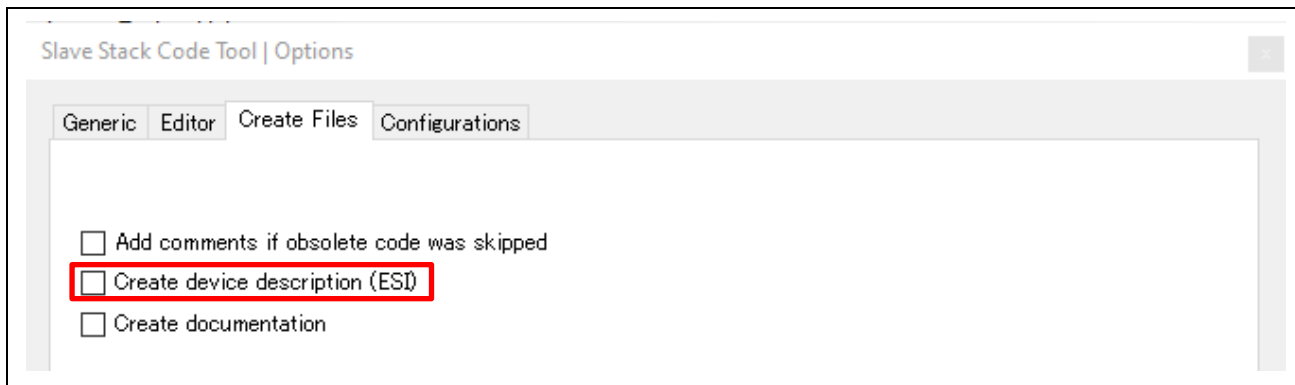


Figure 8.4 Option settings (2)

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

8.2 Appendix B: How to Install the Patch Command

You can install patch using one of the following methods:

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

8.2.1 Using Git for Windows (64-bit)

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

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

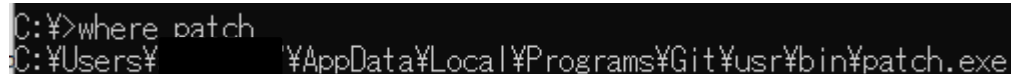
[Git for Windows](#)

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 of patch.exe to the system 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 and enter "where patch" .
If the path to patch.exe is displayed, the installation was successful.



```
C:\>where patch
C:\Users\%username%\AppData\Local\Programs\Git\usr\bin\patch.exe
```

Figure 8.5 Confirm patch.exe (1)

8.2.2 Using MinGW Installation Manager

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

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

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

2. Run mingw-get-setup.exe and install the MinGW Installation Manager by following the instructions in the dialog box.
3. When the installation is complete and the MinGW Installation Manager window appears, select **Basic Setup** in the left window, right-click **msys-base-bin** in the right window, and select **Mark for Installation**.

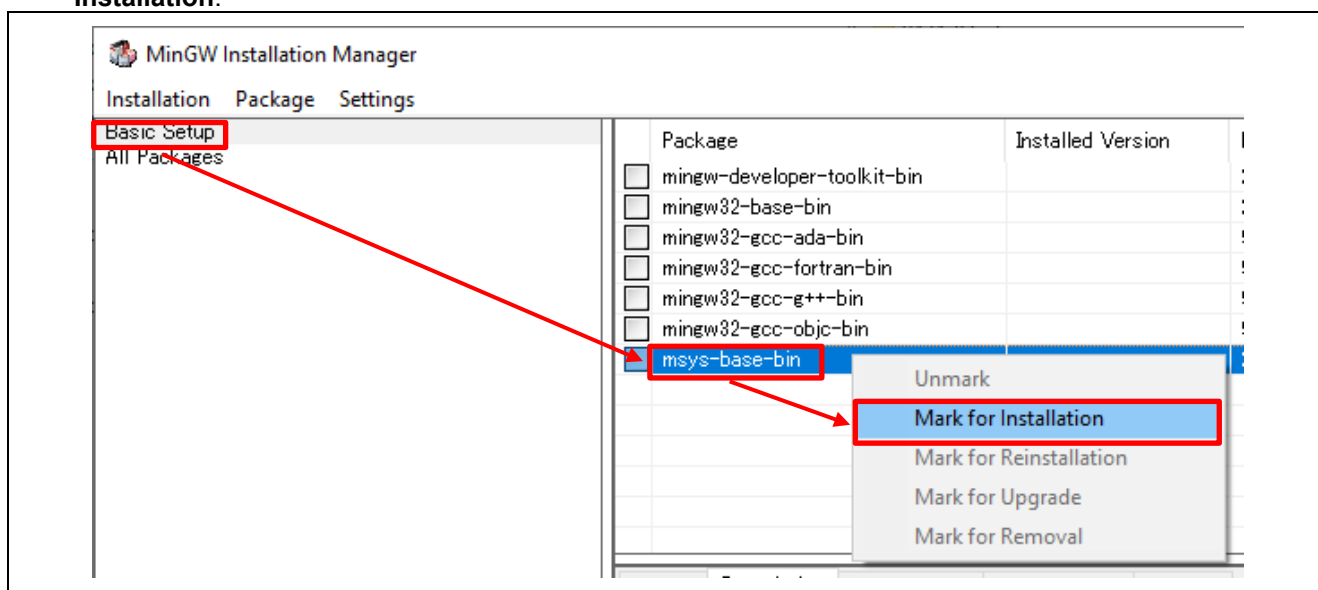


Figure 8.6 MinGW Installation Manager (1)

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

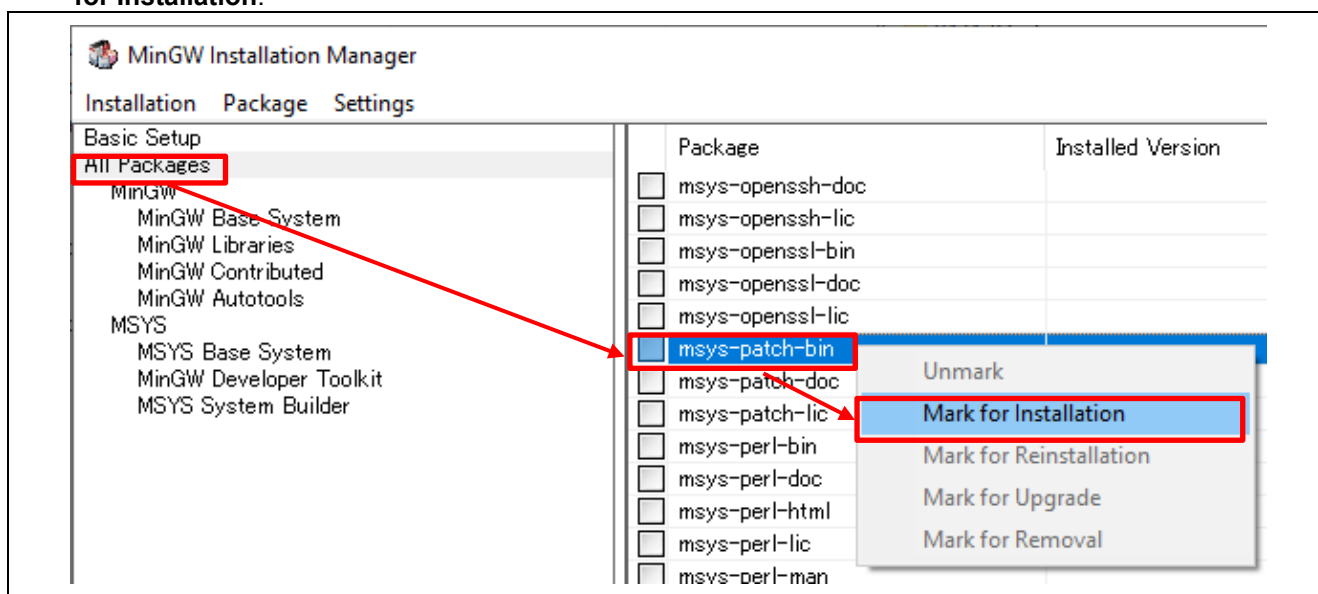


Figure 8.7 MinGW Installation Manager (2)

5. Select **Installation > Apply Changes** in the menu bar.

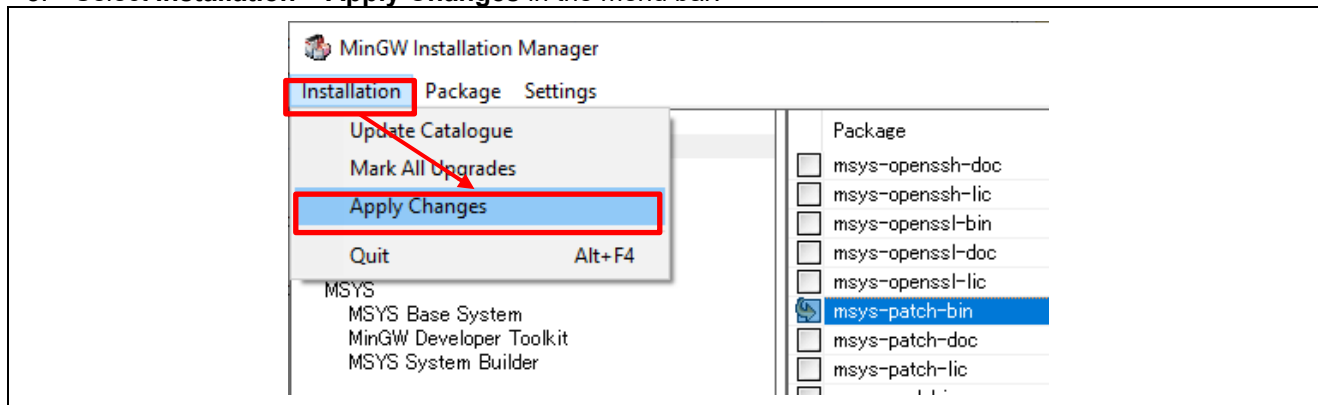


Figure 8.8 MinGW Installation Manager (3)

6. When the Schedule of Pending Actions window appears, click **Apply**.
7. If the message “All changes were applied successfully; you may now clone this dialogue.” appears, patch.exe is installed successfully.
8. Add the path to the installed patch.exe to the system environment variables.
For example, for the default installation, add the following path:

“C:\MinGW\msys\1.0\bin”

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

9. Start Command Prompt and enter “where patch” .
If the path to the installed patch.exe is displayed, the installation was successful.

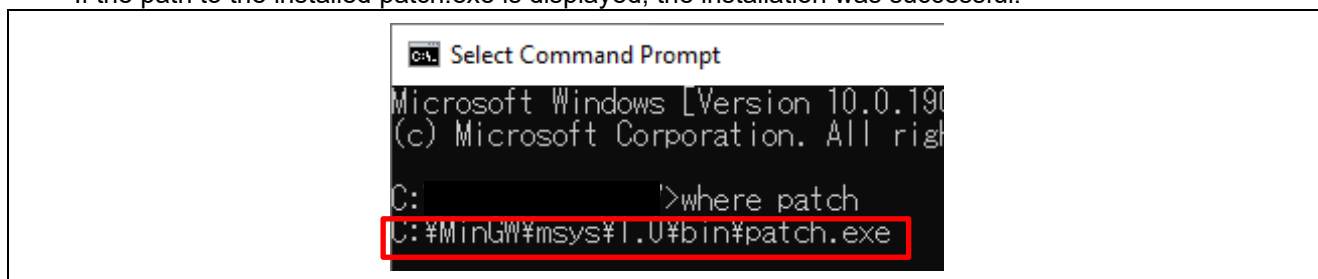


Figure 8.9 Confirm patch.exe (2)

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

- Open the FSP configuration and click the **BSP** tab.

The screenshot shows the 'Device Selection' window. The 'FSP version' is set to 4.0.0. The 'Board' dropdown is set to 'RSK+RZT2M (RAM execution without flash memory)'. The 'Device' dropdown is set to 'R9A07G075M24GBG'. The 'Core' dropdown is set to 'CR52_0'. The 'RTOS' dropdown is set to 'No RTOS'.

Figure 8.10 Open the FSP configuration

- Change **Board** from **RSK+RZT2M (RAM execution without flash memory)** to **RSK+RZT2ME (RAM execution without flash memory)**.

The screenshot shows the 'Device Selection' window with the 'Board' dropdown menu open. The dropdown list includes several options, and 'RSK+RZT2ME (RAM execution without flash memory)' is highlighted with a red box. Other options in the list include 'RSK+RZT2M (xSPI0 x1 boot mode)', 'RSK+RZT2ME (16-bit bus NOR flash boot mode)', 'RSK+RZT2ME (xSPI0 x1 boot mode)', 'RZ/A3UL Custom User Board OCTAL Boot (eXecute-In-Place)', 'RZ/A3UL Custom User Board QSPI Boot (eXecute-In-Place)', and 'RZ/A3UL Custom User Board QSPI Boot(Exec with DDR SDRAM)'.

Figure 8.11 Change the board

- Click the **Clocks** tab. Change **CPU0CLK Mul x1** to **CPU0CLK Mul x4** and **CPU1CLK Mul x1** to **CPU1CLK Mul x4**.

The screenshot shows the 'Clocks Configuration' window. The 'Generate Project Content' button is visible at the top right. The 'Restore Defaults' button is also visible. The 'CPU0CLK Mul' dropdown is set to 'x4' and the 'CPU1CLK Mul' dropdown is set to 'x4'. Both dropdowns are highlighted with red boxes. The 'CLKIO Div' is set to '/4'. The 'Summary' tab is selected at the bottom.

Figure 8.12 Change the clocks

- Click **Generate Project Content** to generate the files for RZ/T2ME-RSK.

8.4 Appendix D: ESC MDIO Type Setting

You can change the PHY settings and the number of ports in the Smart Configurator.

This appendix shows how to change the PHY management interface from GMAC to ESC. The initial setting is GMAC.

1. Change the PHY settings.

Select each ether_phy stack, and then set **Select MDIO type** to **ESC**.

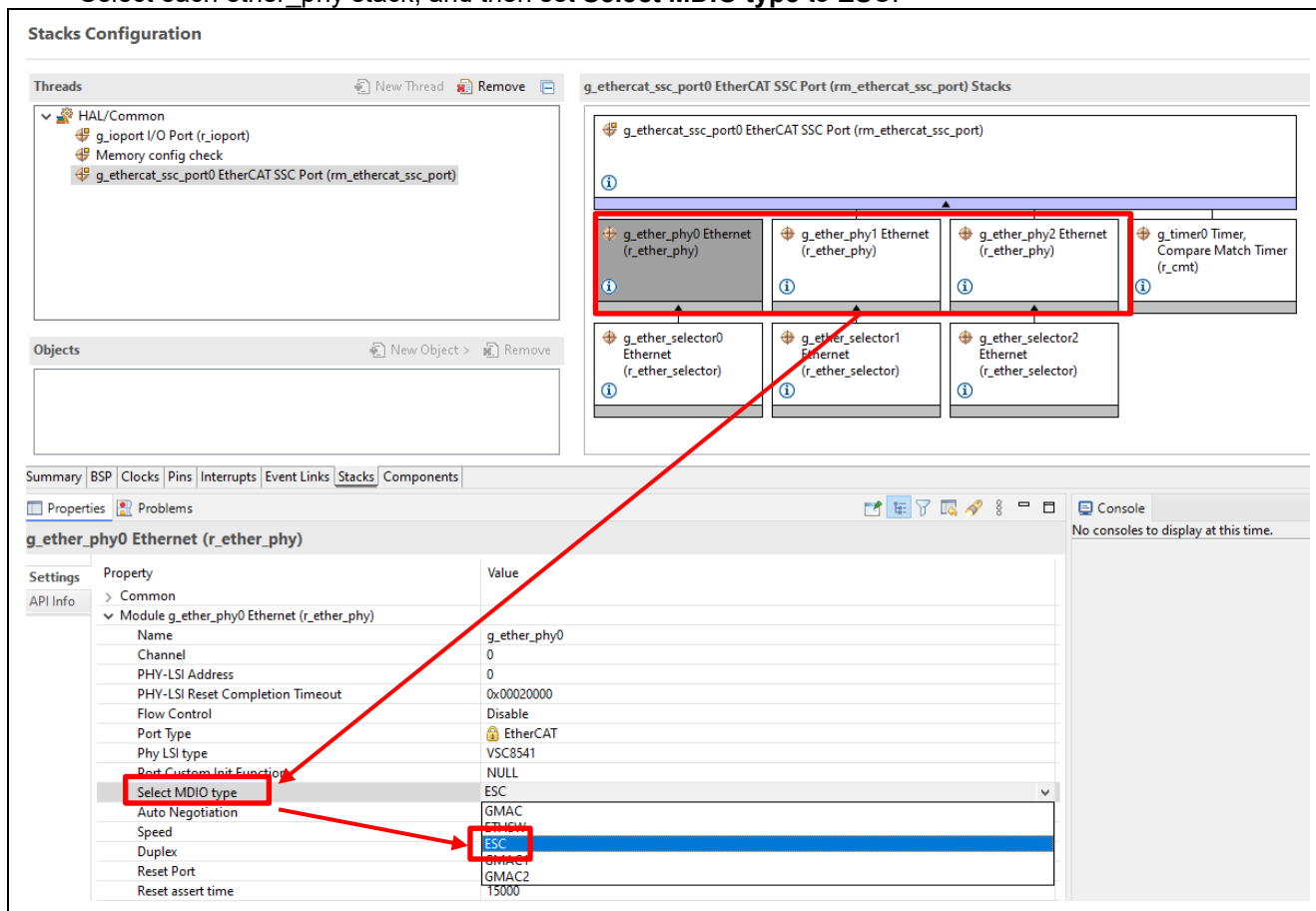


Figure 8.13 Change the MDIO type (1)

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

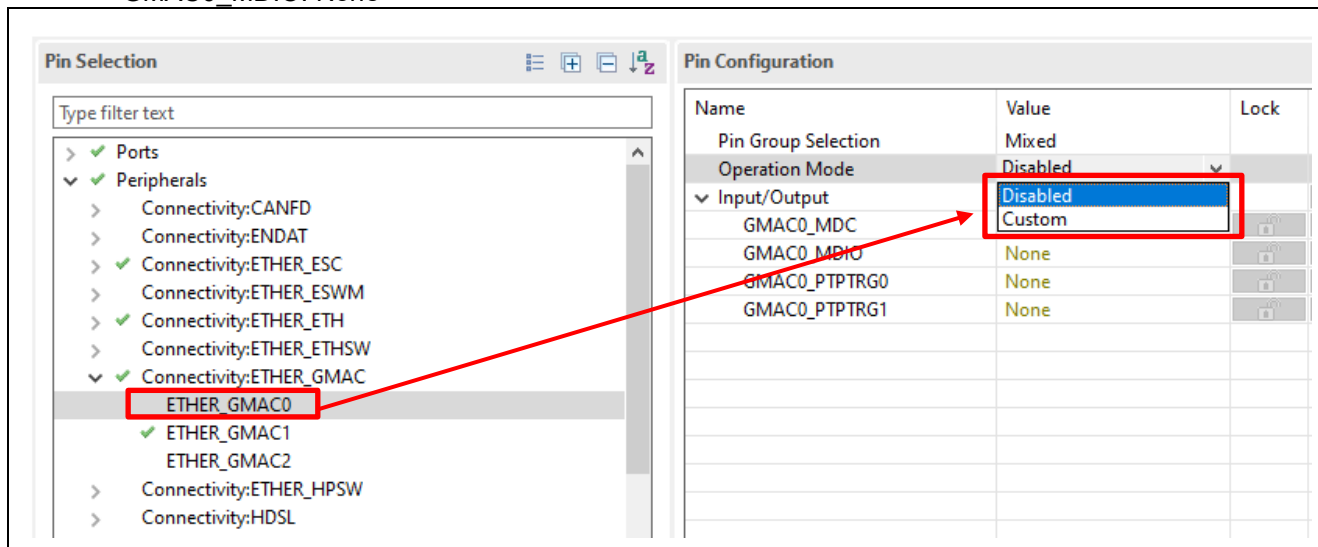


Figure 8.14 Change the MDIO type (2)

Change the ETHER_ESC settings.

- For RZ/T2M, RZ/T2ME, RZ/T2L, and RZ/N2L RSK:
 - ✧ ESC_MDC: P08_7
 - ✧ ESC_MDIO: P09_0
- For RZ/T2H and RZ/N2H EVB:
 - ✧ ESC_MDC: P21_4
 - ✧ ESC_MDIO: P21_5
- For RZ/T2N EVB:
 - ✧ ESC_MDC: P20_4
 - ✧ ESC_MDIO: P20_5

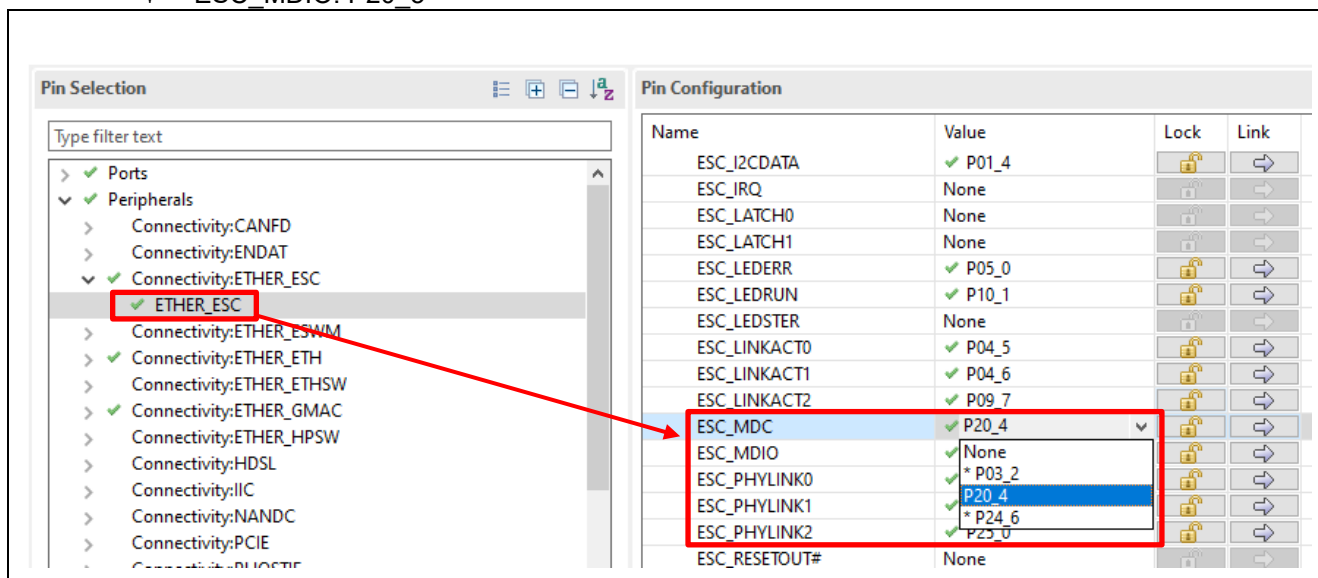


Figure 8.15 Change the MDIO type (3)

3. After completing all the settings, click **Generate Project Content** to generate the code.

8.5 Appendix E: Changing the EEPROM Capacity Setting

Use the Smart Configurator to change the EEPROM capacity setting and generate the code.

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

1. Select the SSC_port stack, and then select **Properties** → **EEPROM Size**.
2. Select **Under 32 Kbits** or **Over 32 Kbits** according to the EEPROM capacity.

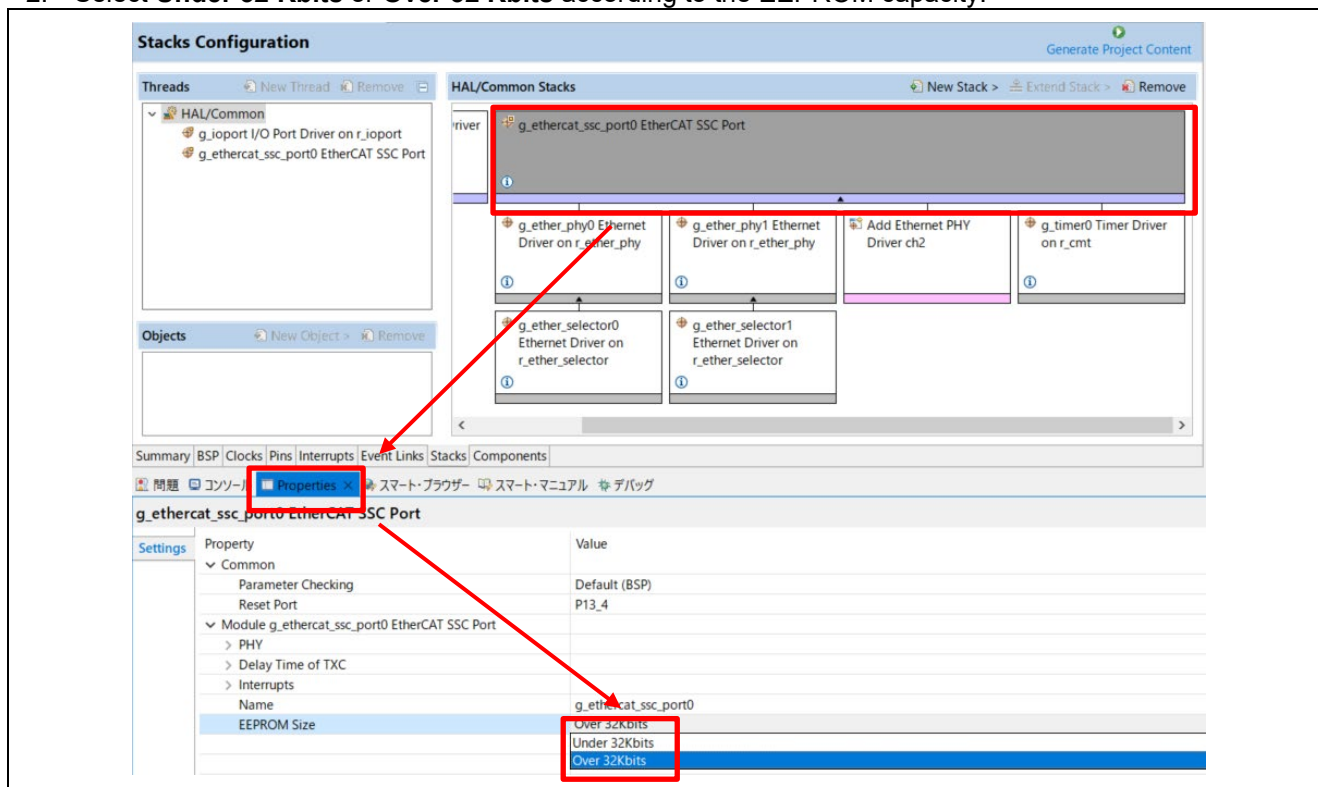


Figure 8.16 Change the EEPROM Size

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

9. Limitations

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

Revision History

Rev.	Date	Description	
		Page	Summary
4.00	July 17, 2026	-	First release
4.10	Nov. 4, 2026	-	Supported RZ/T2N. Updated package revision number.

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.

- Arm® and Cortex® are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.
- Ethernet is a registered trademark of Fuji Xerox Co. Ltd.
- IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers Inc
- EtherCAT® and TwinCAT® are registered trademarks and patented technologies, licensed by Beckhoff Automation GmbH, Germany.
- J-Link is a trademark of SEGGER Microcontroller GmbH.
- Additionally, all product names and service names in this document are a trademark or a registered trademark which belongs to the respective owners.

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/.