RZ/T2, RZ/N2

Quick Start Guide: EtherNet/IP OpENer Sample Program

Introduction

This document describes the setup procedure of the sample program for RZ/T2 and RZ/N2 port of EtherNet/IP™ “OpENer”.

Note: In this document, the Cortex®-R52 is referred to as CR52 and the Cortex®-A55 is referred to as CA55.

Target Device

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

Contents

1. Overview 4
1.1 Abbreviations / Definitions 4
1.2 Reference 5
1.3 Limitations and Restrictions / Known Issues 6
2. Features and folder structure 7
2.1 Features 7
2.2 Package folder structure 8
3. Requirements 11
4. Hardware Setup 12
4.1 RZ/T2M RSK board 12
4.1.1 Jumper and Switch configuration 12
4.1.2 Setup Board 13
4.2 RZ/T2ME RSK board 15
4.2.1 Jumper and Switch configuration 15
4.2.2 Setup Board 15
4.3 RZ/T2H Evaluation board 16
4.3.1 Jumper and Switch configuration 16
4.3.2 Setup Board 18
4.4 RZ/N2L RSK board 19
4.4.1 Jumper and Switch configuration 19
4.4.2 Setup Board 20
4.5 RZ/N2H Evaluation board 22
4.5.1 Jumper and Switch configuration 22
4.5.2 Setup Board 23
5. Setup the Host Device 25
5.1 Configure the Host IP Address 25
5.2 Setup the CODESYS Software 25
5.2.1 How to get CODESYS 25
5.2.2 Startup CODESYS Tools 26
5.2.3 Install EDS File into CODESYS 28
6. Running Sample Application 31
6.1 Board IP Address Setting 32
6.2 Debugging with EWARM 33
6.2.1 Single-Core Project (RZ/N2L) 33
6.2.2 Multi-Core Project (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/N2H) 36
6.3 Debugging with e^2 studio 43

6.3.1 Single-Core Project (RZ/N2L) 43
6.3.2 Multi-Core Project (RZ/T2M, RZ/T2ME, RZ/T2H, RZ/N2H) 49

7. Demonstration of the application with the CODESYS 58
7.1 Application Behavior 59
7.2 IP and MAC Address Configuration 60
7.2.1 IP Configuration 60
7.2.2 MAC Address Configuration 61
7.3 Startup Software PLC 62
7.3.1 Open CODESYS project 62
7.3.2 Network Configuration 64
7.3.3 Interface and IP address configuration 66
7.4 Operation Check 68
7.4.1 Build Project and Start Application 68
7.4.2 Check Network Connection 68
7.4.3 Check Application Behavior 69

8. Software Specification 70
8.1 Stack Diagram 70
8.2 System Configuration Diagram 71
8.3 Support Object Classes, Services and Attributes 72
8.4 Device Data Configuration 79

9. Appendix 80

1. Overview

This document describes the setup procedure of the sample program for RZ/T2 and RZ/N2 port of EtherNet/IP “OpENer” and explains the procedure for connecting the CODESYS software programmable logic controller (PLC).

In this sample program, these open-source software are used.

- EtherNet/IP OpENer
- FreeRTOS
- lwIP

For demonstration, the application of this sample program has an Exclusive Connection and an Input Only Connection. This document also explains how to connect to the CODESYS software prepared in this package.

1.1 Abbreviations / Definitions

Table 1.1 Abbreviations/Definitions

Index	Abbreviations/Definitions	Description
1	ACD	Address Conflict Detection
2	CIP	Common Industrial Protocol
3	DLR	Device Level Ring
4	EIP	EtherNet/IP
5	EVB	Evaluation Board
6	EWARM	Embedded Workbench® for ARM
7	FSP	Flexible Software Package
8	FuSa	Functional Safety
9	IP	Internet Protocol
10	lwIP	lightweight IP
11	LLDP	Link Layer Discovery Protocol
12	OSS	Open-Source Software
13	PC	Personal Computer
14	QoS	Quality of Service
15	RSK	Renesas Starter Kit
16	SCI	Serial Communication Interface
17	SW	Switch
18	TCP	Transmission Control Protocol
19	UDP	User Datagram Protocol
20	USB	Universal Serial Bus

1.2 Reference

Technical information is available via Renesas.

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

Document Type	Description	Document Title	Document No.
Application Note	Describes how to use the Renesas Flexible Software Package (FSP) for writing applications for the RZ/T2, RZ/N2 microprocessor series.	RZ/T2, RZ/N2 Getting Started with Flexible Software Package	r01an6434

Table 1.3 Technical Inputs for RZ/T2M, 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	r20ut4939
User's Manual	Provides technical details of the RZ/T2M microprocessor.	RZ/T2M Group User's Manual Hardware	r01uh0916
User's Manual	Provides technical details of the RZ/T2ME microprocessor.	RZ/T2ME Group User's Manual Hardware	r01uh1062

Table 1.4 Technical Inputs for RZ/T2H, 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	r20ut5405
User's Manual	Describes the technical details of the RZ/N2H EVB hardware.	RZ/N2H Evaluation Board User's Manual	r20ut5522
User's Manual	Provides technical details of the RZ/T2H and RZ/N2H microprocessor.	RZ/T2H and RZ/N2H Groups User's Manual: Hardware	r01uh1039

Table 1.5 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	r20ut4984
User's Manual	Provides technical details of the RZ/N2L microprocessor.	RZ/N2L Group User's Manual Hardware	r01uh0955

1.3 Limitations and Restrictions / Known Issues

Table 1.6 Limitations and Restrictions / Known Issues

Device	Category	Detail
-	-	-

2. Features and folder structure

2.1 Features

The "OpENer" is an open-source software for I/O communication adapters of EtherNet/IP. It supports multiple I/O and explicit connections and includes objects and services for making EtherNet/IP-compliant products as defined in the ODVA specification.

This package is the RZ/T2 and RZ/N2 port of OpENer and includes OpENer source codes. Regarding the open-source license of OpENer, please see the following file.

```
common\oss\OpENer\license.txt
```

And this sample software has the functionality required by the ODVA Conformance Test Suite CT21 (hereinafter referred to as CT21).

The Class Objects implemented in this sample software are as follows. For details, please see Chapter 8.3.

Table 2.1 CIP Object Classes supported this sample software.

Object Class #	Object Class Name
0x01	Identity
0x02	Message Router
0x04	Assembly
0x06	Connection Manager
0x47	Device Level Ring
0x48	QoS
0xF5	TCP/IP Interface
0xF6	Ethernet Link
0x109	LLDP Management

Option: CIP Safety feature

This sample code also includes the implementation of the CIP Safety feature as an option. By default, this feature is disabled, and it can be enabled by changing the project settings.

Note: The enabled CIP Safety feature has only been tested for basic operation in the connection configurations described later. A self-test based on the official ODVA guidelines has not been conducted.

The objects that become active upon enabling are as follows:

Table 2.2 List of CIP Safety Objects Enabled

Object Class #	Object Class Name
0x39	Safety Supervisor
0x3A	Safety Validator
0x49	Safety Analog Input Point
0x64	Safety Analog Output Point

For details on how to enable the CIP Safety feature, services, and operation, please refer to Chapter 9 Appendix C CIP Safety feature.

2.2 Package folder structure

The folder structure of the sample application is shown in the following table.

Table 2.3 Sample package overview

Item	Description
r01an8286xx0410-rzt2-rzn2-ethernetip-opener-package.zip	RZ/T2, RZ/N2 EtherNet/IP Sample Package
├─── RZT2M_EtherNetIP_OpENer_RSK_rev0410.zip	Archive for RZ/T2M (RSK board)
├─── RZT2H_EtherNetIP_OpENer_EVB_rev0410.zip	Archive for RZ/T2H (EVB board)
├─── RZN2L_EtherNetIP_OpENer_RSK_rev0410.zip	Archive for RZ/N2L (RSK board)
├─── RZN2H_EtherNetIP_OpENer_EVB_rev0410.zip	Archive for RZ/N2H (EVB board)
├─── r01an6601ej0410_rzt2_rzn2_ethernetip_rsk.pdf	Application note
├─── r01an8286ej0410-rzt2_rzn2_ethernetip_package.pdf	Release note
└─── SBOM	SBOM files

Table 2.4 Common folder structure

Item	Description
common	Common files for all projects
├──oss	Open-Source Software
│ ├──amazon-freertos	FreeRTOS
│ ├──lwip	TCP/IP stack light-weight IP
│ └──OpENer	EtherNet/IP stack OpENer
└──renesas	Renesas Software
├──application	User EtherNet/IP Application
├──module	Modules
│ ├──cip_safety	CIP Safety module
│ ├──dlr	DLR module
│ ├──ether_netif	NETIF module
│ ├──lwip_lldp	LLDP module
│ ├──lwip_port	lwIP port module
│ ├──opener_port	OpENer port module
│ └──serial_io	Serial interface module
└──oss_deps	OSS dependencies
├──lwip	lwIP dependency
└──opener	OpENer dependency

Table 2.5 Files for scanners

Item	Description
scanner	-
│ renesas_opener_sample_app.eds	Electronic Data Sheet (EDS) file
│ renesas_opener_sample_app.soc	SoC file
│	-
├──codesys	-
│ opener_sample_app.project	CODESYS project file for Demo
│	-
└──EDS_SOC_for_CipSafety	Files for CIP Safety feature

Table 2.6 RZ/T2M folder structure

Item	Description
RZT2M_EtherNetIP_OpENer_RSK_rev0410	-
├──common	Common files for all projects
├──project	-
│ └──rzt2m_rsk	Projects for RZ/T2M RSK
│ └──CR52_Dual	Dual core project
│ ├──e2studio	e² studio project
│ │ └──RAM	RAM-execution project
│ │ ├──RZT2M_RSK_OpENer_CR52_Dual_Prim	For CR52_0 (primary core)
│ │ └──RZT2M_RSK_OpENer_CR52_Dual_Second	For CR52_1 (secondary core)
│ └──ewarm	EWARM project
│ └──RAM	RAM-execution project
│ ├──RZT2M_RSK_OpENer_CR52_Dual_Prim	For CR52_0 (primary core)
│ └──RZT2M_RSK_OpENer_CR52_Dual_Second	For CR52_1 (secondary core)
└──scanner	Common files for scanners

Table 2.7 RZ/T2H folder structure

Item	Description
RZT2H_EtherNetIP_OpENer_EVB_rev0410	-
├──common	Common files for all projects
├──project	-
│ └──rzt2h_evb	Projects for RZ/T2H EVB
│ ├──CR52_CA55	Dual core project
│ │ ├──e2studio	e² studio project
│ │ │ └──RAM	RAM-execution project
│ │ │ ├──RZT2H_EVB_OpENer_CR52_CA55_Prim	For CR52_0 (primary core)
│ │ │ └──RZT2H_EVB_OpENer_CR52_CA55_Second	For CA55_2 (secondary core)
│ │ └──ewarm	EWARM project
│ │ └──RAM	RAM-execution project
│ │ ├──RZT2H_EVB_OpENer_CR52_CA55_Prim	For CR52_0 (primary core)
│ │ └──RZT2H_EVB_OpENer_CR52_CA55_Second	For CA55_2 (secondary core)
│ └──CR52_Dual	Dual core project
│ ├──e2studio	e² studio project
│ │ └──RAM	RAM-execution project
│ │ ├──RZT2H_EVB_OpENer_CR52_Dual_Prim	For CR52_0 (primary core)
│ │ └──RZT2H_EVB_OpENer_CR52_Dual_Second	For CR52_1 (secondary core)
│ └──ewarm	EWARM project
│ └──RAM	RAM-execution project
│ ├──RZT2H_EVB_OpENer_CR52_Dual_Prim	For CR52_0 (primary core)
│ └──RZT2H_EVB_OpENer_CR52_Dual_Second	For CR52_1 (secondary core)
└──scanner	Common files for scanners

Table 2.8 RZ/N2L folder structure

Item	Description
RZN2L_EtherNetIP_OpENer_RSK_rev0410	-
├──common	Common files for all projects
├──project	-
│ └──rzn2l_rsk	Projects for RZ/N2L RSK
│ └──CR52	Single core project
│ ├──e2studio	e² studio project
│ │ └──RAM	RAM-execution project
│ │ ├──RZN2L_RSK_OpENer_CR52	For CR52_0 (primary core)
│ └──ewarm	EWARM project
│ └──RAM	RAM-execution project
│ └──RZN2L_RSK_OpENer_CR52	For CR52_0 (primary core)
└──scanner	Common files for scanners

Table 2.9 RZ/N2H folder structure

Item	Description
RZN2H_EtherNetIP_OpENer_EVB_rev0410	-
├──common	Common files for all projects
├──project	-
│ └──rzn2h_evb	Projects for RZ/N2H EVB
│ ├──CR52_CA55	Dual core project
│ │ ├──e2studio	e² studio project
│ │ │ └──RAM	RAM-execution project
│ │ │ ├──RZN2H_EVB_OpENer_CR52_CA55_Prim	For CR52_0 (primary core)
│ │ │ └──RZN2H_EVB_OpENer_CR52_CA55_Second	For CA55_2 (secondary core)
│ │ └──ewarm	EWARM project
│ │ └──RAM	RAM-execution project
│ │ ├──RZN2H_EVB_OpENer_CR52_CA55_Prim	For CR52_0 (primary core)
│ │ └──RZN2H_EVB_OpENer_CR52_CA55_Second	For CA55_2 (secondary core)
│ └──CR52_Dual	Dual core project
│ ├──e2studio	e² studio project
│ │ └──RAM	RAM-execution project
│ │ ├──RZN2H_EVB_OpENer_CR52_Dual_Prim	For CR52_0 (primary core)
│ │ └──RZN2H_EVB_OpENer_CR52_Dual_Second	For CR52_1 (secondary core)
│ └──ewarm	EWARM project
│ └──RAM	RAM-execution project
│ ├──RZN2H_EVB_OpENer_CR52_Dual_Prim	For CR52_0 (primary core)
│ └──RZN2H_EVB_OpENer_CR52_Dual_Second	For CR52_1 (secondary core)
└──scanner	Common files for scanners

3. Requirements

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

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
	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
	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	CODESYS	v3.5.21.10 64-bit	CODESYS GmbH CODESYS Group

4. Hardware Setup

4.1 RZ/T2M RSK board

4.1.1 Jumper and Switch configuration

This document describes the major hardware properties. Refer to Renesas Starter Kit+ for RZ/T2M user's manual and schematic for more board details.

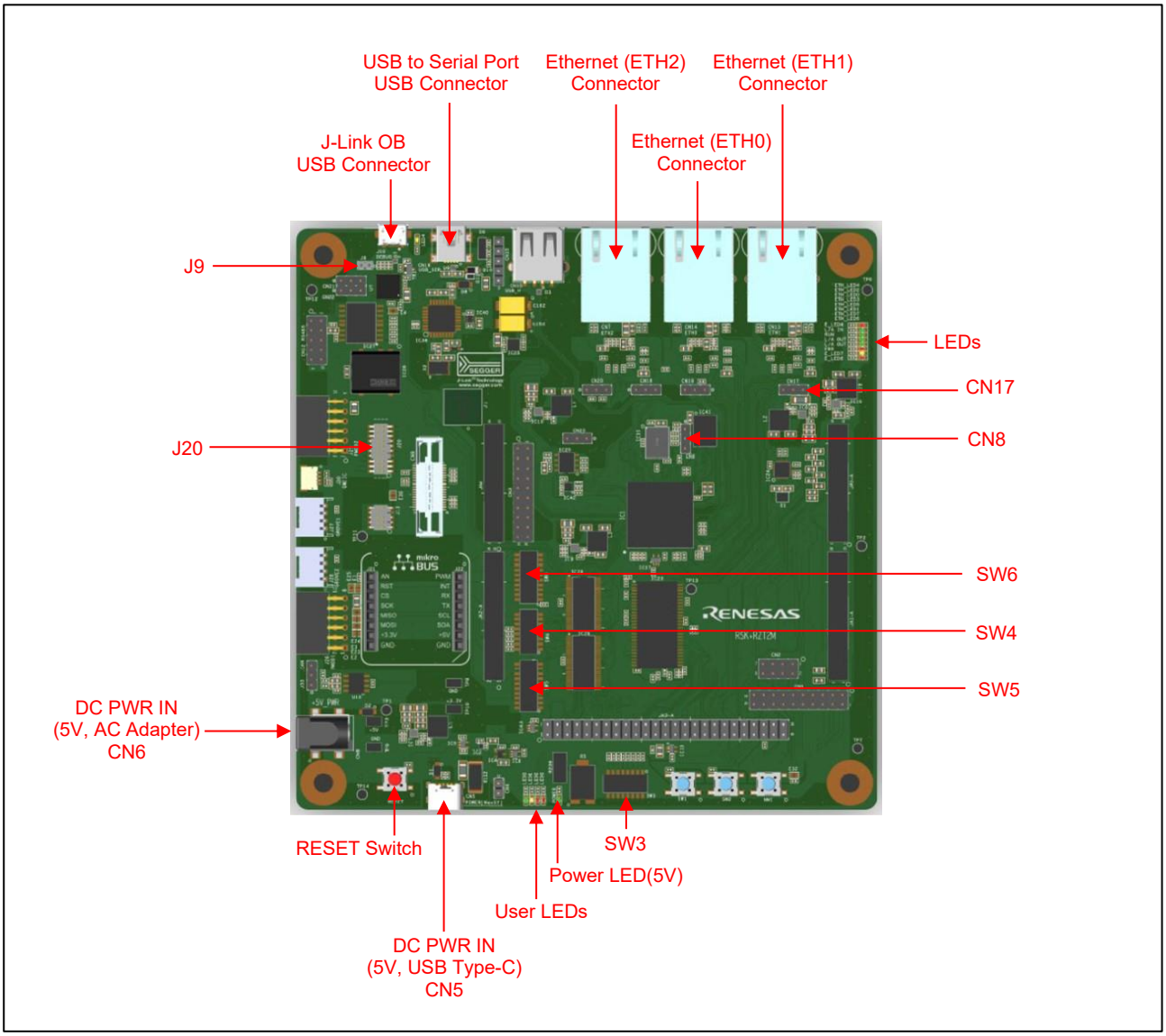


Figure 4.1 RZ/T2M RSK board layout

Table 4.1 Jumper pin settings

Reference	Jumper Position	Description
CN8	Shorted Pin 2-3	Enable QSPI (IC21).
CN17	Shorted Pin 2-3	Connect 1.8V Power rail to VCC1833_2. (When using Ethernet)
J9	Open	Enable the J-Link OB.

Table 4.2 SW4 Settings

SW4	Setting	Description
SW4-1	ON	16-bit bus boot mode (NOR flash) Sets this SW settings when the program is executed by RAM.
SW4-2	OFF	
SW4-3	ON	
SW4-1	ON	xSPI0 boot mode (x1 boot serial flash) Sets these SW settings when the program is executed by flash writing.
SW4-2	ON	
SW4-3	ON	
SW4-4	ON	JTAG Authentication by Hash is disabled.
SW4-5	OFF	ATCM 1 wait

Table 4.3 SW5 Settings

SW5	Setting	Description
SW5-3	ON	Enable SCI_RTS
SW5-4	OFF	
SW5-5	ON	Enable SCI_RXD
SW5-6	OFF	
SW5-7	OFF	
SW5-8	OFF	Enable SCK3
SW5-9	ON	
SW5-10	OFF	

Table 4.4 SW6 Settings

SW6	Setting	Description
SW6-1	OFF	Enables signals other than the external bus. (CAN, Emulator, I2C, etc.)
SW6-3	ON	Enable TRACE_CTL
SW6-4	OFF	
SW6-5	OFF	Enable SCI_TXD
SW6-6	ON	
SW6-7	OFF	Enable MB_RST
SW6-8	ON	
SW6-9	OFF	Enable CAN_RX_OB
SW6-10	ON	

Other SW settings refer to the User's Manual r20ut4939.

4.1.2 Setup Board

Setting the board for running sample program is shown below.

1. Connect an emulator to the connector .
   - When you use I-jet or J-Link emulator, connect it to J20 on RZ/T2M RSK board.
   - When you use J-Link On-Board emulator, connect USB micro-B to J10 on RZ/T2M RSK board. (Please disconnect J9 for powering up J-Link OB.)

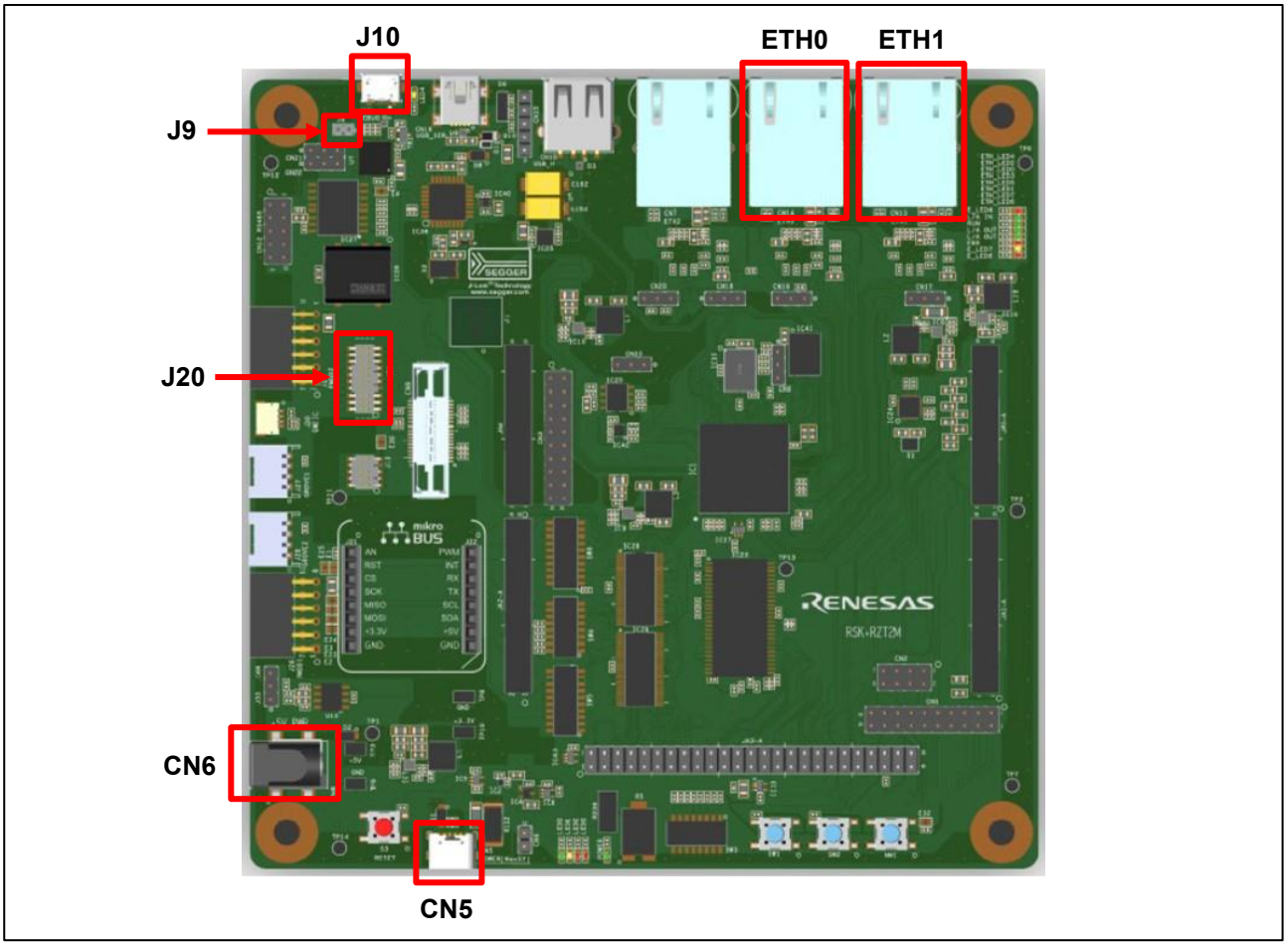


Figure 4.2 Setup RZ/T2M RSK board

2. Power is supplied using USB cable (Type-C) or AC / DC adapter.
   - When using USB cable (Type-C), connect it to the USB connector “CN5” on the RZ/T2M RSK board.
   - When using AC/DC adapter, connect it to the connector “CN6” on the RZ/T2M RSK board.
3. Connect Ethernet cable to the Ethernet Connector “ETH0” or “ETH1”.

4.2 RZ/T2ME RSK board

4.2.1 Jumper and Switch configuration

For each configuration, see the section 4.1.1 RSK+RZT2M.

4.2.2 Setup Board

For each setting, see the section 4.1.2 RSK+RZ/T2M.

4.3 RZ/T2H Evaluation board

4.3.1 Jumper and Switch configuration

This document describes the major hardware properties. Refer to the Evaluation Board for RZ/T2H user's manual and schematic for more board details.

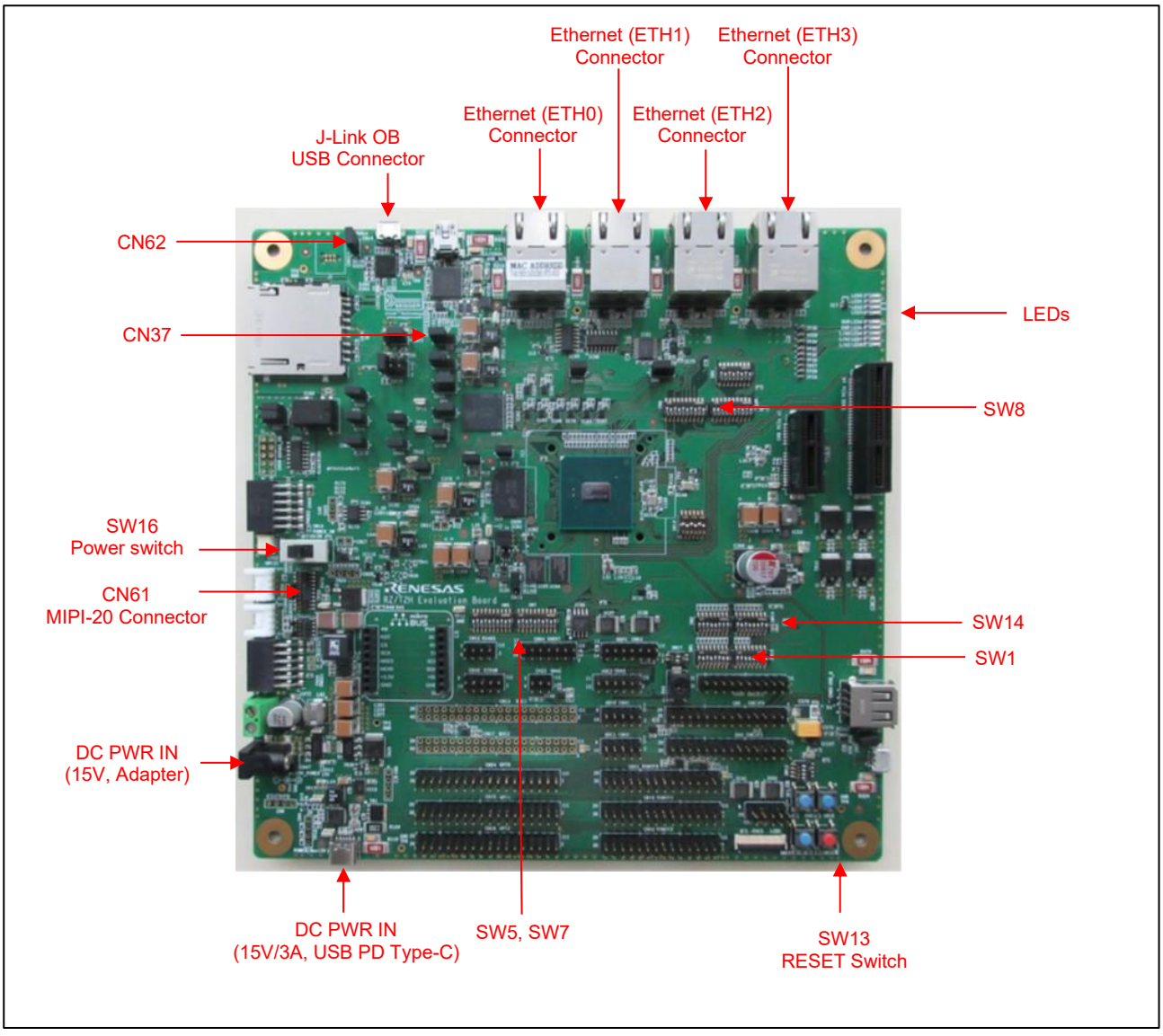


Figure 4.3 RZ/T2H Evaluation Board Layout

Table 4.5 Jumper pin settings

Reference	Jumper Position	Description
CN37	2-3 are short-circuit	3.3-V power is supplied to VCC1833_0. (For Ethernet Port 0)
CN62	Open circuit	The on-board debugging function J-Link OB is enabled.

Table 4.6 SW1 settings

SW1	Setting	Description
SW1-1	ON	XTALSEL = 'L' Select oscillator for RZ/T2H clock input.
SW1-3	ON	P35_3,4,5,6 are connected to SW12 and used as user DIPSW inputs.
SW1-6	ON	P01_0,1,2,4,5,6,7 and P02_0,1,2,3 are used as XSPI1 signals.

Table 4.7 SW2 settings

SW2	Setting	Description
SW2-3	OFF	LED3 Enable

Table 4.8 SW5 settings

SW5	Setting	Description
SW5-1	ON	P22_7 is used as ESC_LINKACT0.
SW5-2	OFF	
SW5-3	ON	P23_0 is used as ESC_LINKACT1.
SW5-4	OFF	
SW5-9	ON	P32_2 is used as USER_LED1 control.
SW5-10	OFF	

Table 4.9 SW7 settings

SW7	Setting	Description
SW7-5	ON	P23_5 is used as ESC_LINKACT2.
SW7-6	OFF	

Table 4.10 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	
SW8-9	ON	P23_1 is used as USER_LED0.
SW8-10	OFF	

Table 4.11 SW14 settings

SW14	Setting	Description
SW14-1	ON	MD = 2 (L, H, L) xSPI1 boot mode (x1 boot serial flash)
SW14-2	OFF	
SW14-3	ON	
SW14-4	OFF	CPU0 ATCM wait cycle = 1 wait cycle
SW14-5	OFF	CPU1 ATCM wait cycle = 1 wait cycle
SW14-6	OFF	Supply voltage of boot peripheral is 3.3 V
SW14-7	ON	JTAG mode = Normal mode

Other SW settings refer to the User's Manual r20ut5405.

4.3.2 Setup Board

Setting the board for running sample program is shown below.

1. Connect an emulator to the connector .
   - When you use I-jet or J-Link emulator, connect it to CN61.
   - When you use J-Link On-Board emulator, connect USB micro-B to CN14. (Please disconnect CN62 for powering up J-Link OB.)

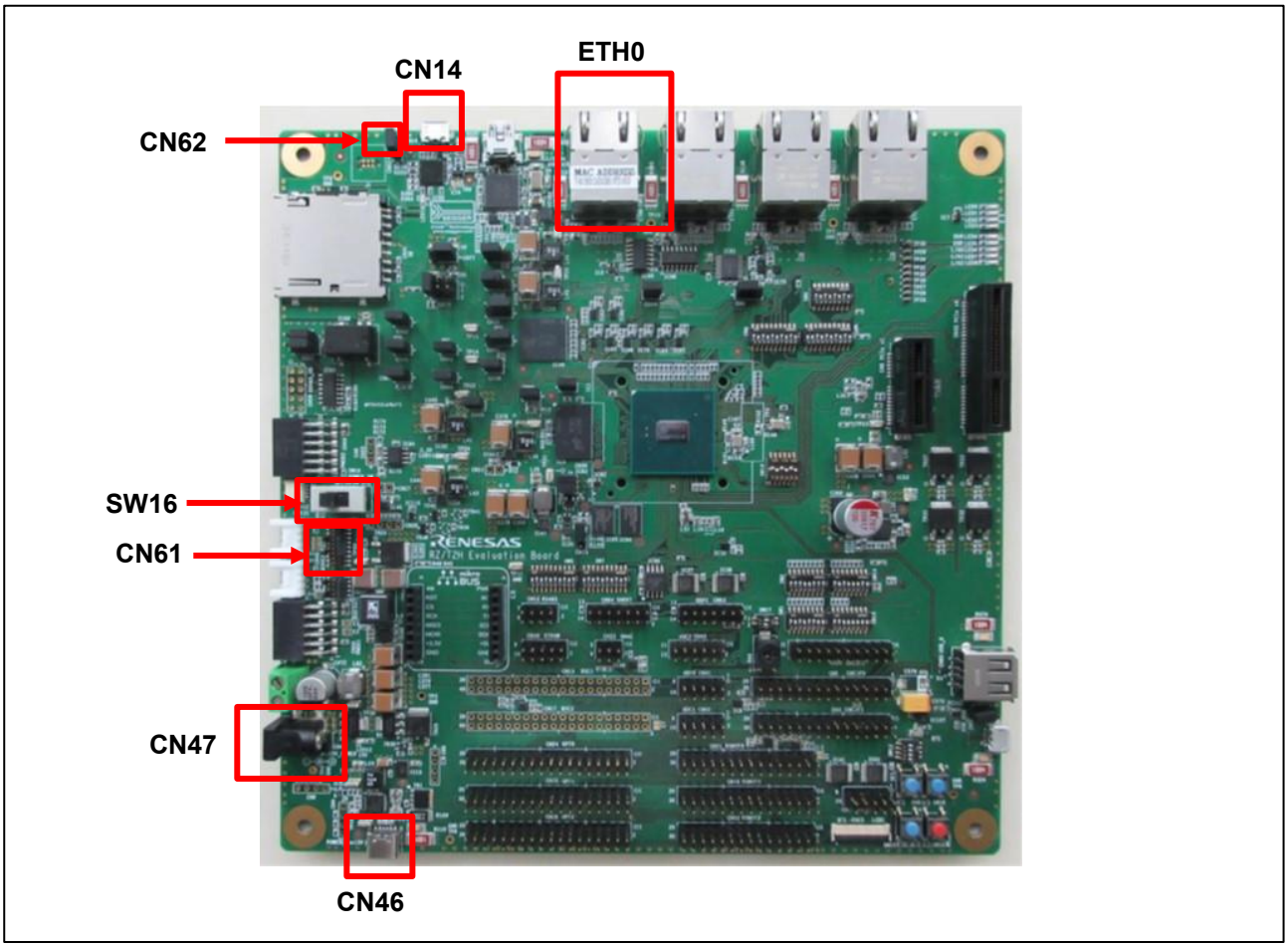


Figure 4.4 Setting RZ/T2H Evaluation Board

2. Power is supplied using a 15V AC/DC adapter or a USB cable (15V/3A USB PD Type-C).
   - When using a 15V AC/DC adapter, connect it to the connector CN47.
   - When using a USB cable (15V/3A USB PD Type-C), connect it to the USB connector CN46.
3. Connect Ethernet cable to the Ethernet Connector ETH0.
4. Slide the SW16 POWER_SW to ON. (Provide power to the board)

4.4 RZ/N2L RSK board

4.4.1 Jumper and Switch configuration

This document describes the major hardware properties. Refer to Renesas Starter Kit+ for RZ/N2L user's manual and schematic for more board details.

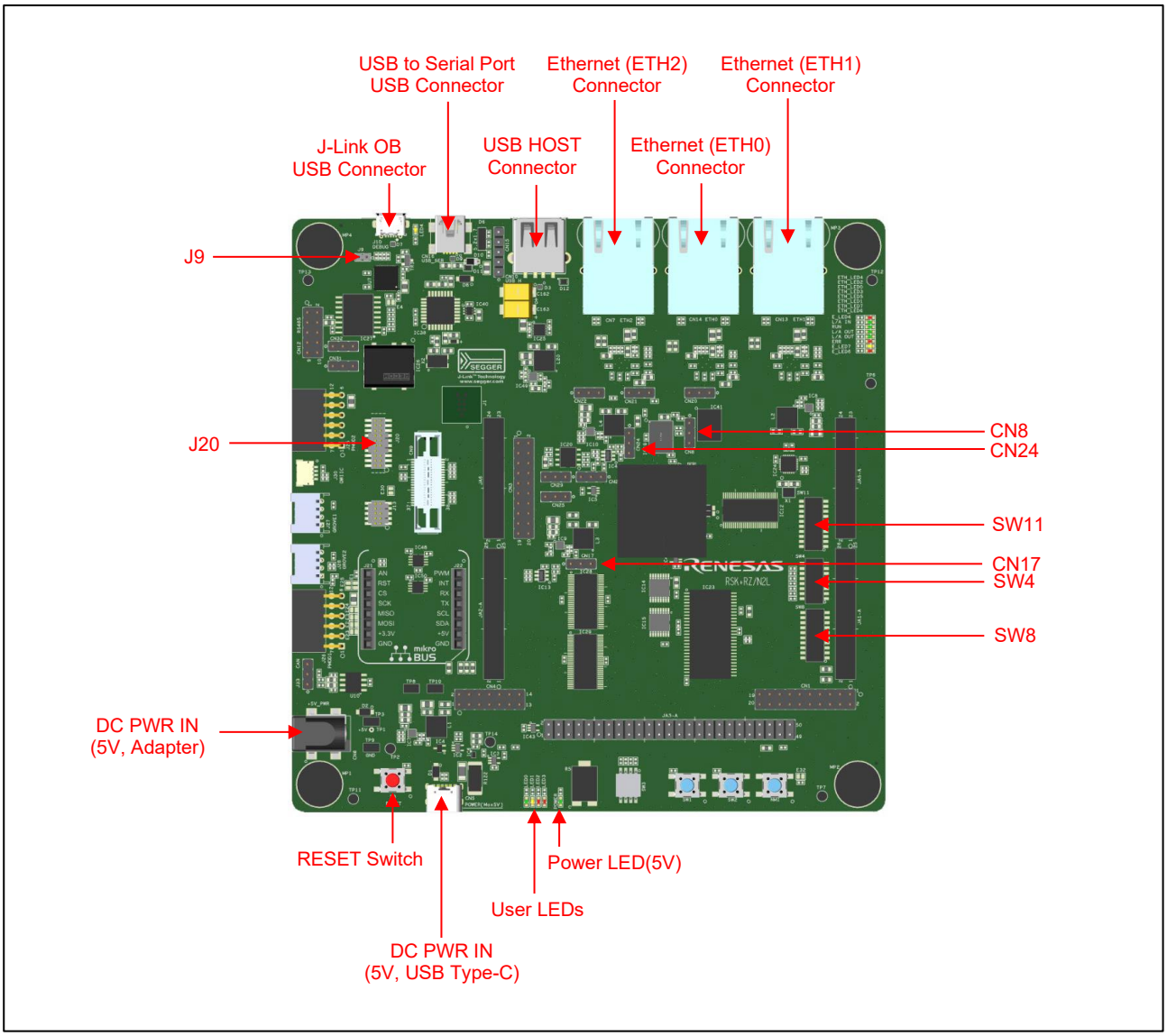


Figure 4.5 RZ/N2L RSK board layout

Table 4.12 Jumper pin settings

Reference	Jumper Position	Description
CN8	Shorted Pin 2-3	Enable QSPI (IC21).
CN17	Shorted Pin 2-3	Connect 1.8V Power rail to VCC1833_2. (When using Ethernet)
CN24	Shorted Pin 2-3	Connect 1.8V Power rail to VCC1833_3. (When using QSPI Flash IC21)
J9	Open	Enable the J-Link OB.

Table 4.13 SW4 Settings

SW4	Setting	Description
SW4-1	ON	16-bit bus boot mode (NOR flash) Sets this SW settings when the program is executed by RAM.
SW4-2	OFF	
SW4-3	ON	
SW4-1	ON	xSPI0 boot mode (x1 boot serial flash) Sets this SW settings when the program is executed by flash writing.
SW4-2	ON	
SW4-3	ON	
SW4-4	ON	JTAG Authentication by Hash is disabled.
SW4-6	OFF	Enables signals other than the trace signal. (Motor, RS485, etc.)
SW4-7	ON	Enables signals other than the external bus. (CAN, Emulator, I2C, etc.)
SW4-8	OFF	Enable SW3.

Table 4.14 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	

Table 4.15 SW11 Settings

SW11	Setting	Description
SW11-1	ON	Enable the " LED_RED2" signal.
SW11-2	OFF	
SW11-3	OFF	

Other SW settings refer to the User's Manual r20ut4984.

4.4.2 Setup Board

Setting the board for running sample program is shown below.

1. Connect an emulator to the connector .
   - When you use I-jet or J-Link emulator, connect it to J20 on RZ/N2L RSK board.
   - When you use J-Link On-Board emulator, connect USB micro-B to J10 on RZ/N2L RSK board. (Please disconnect J9 for powering up J-Link OB.)

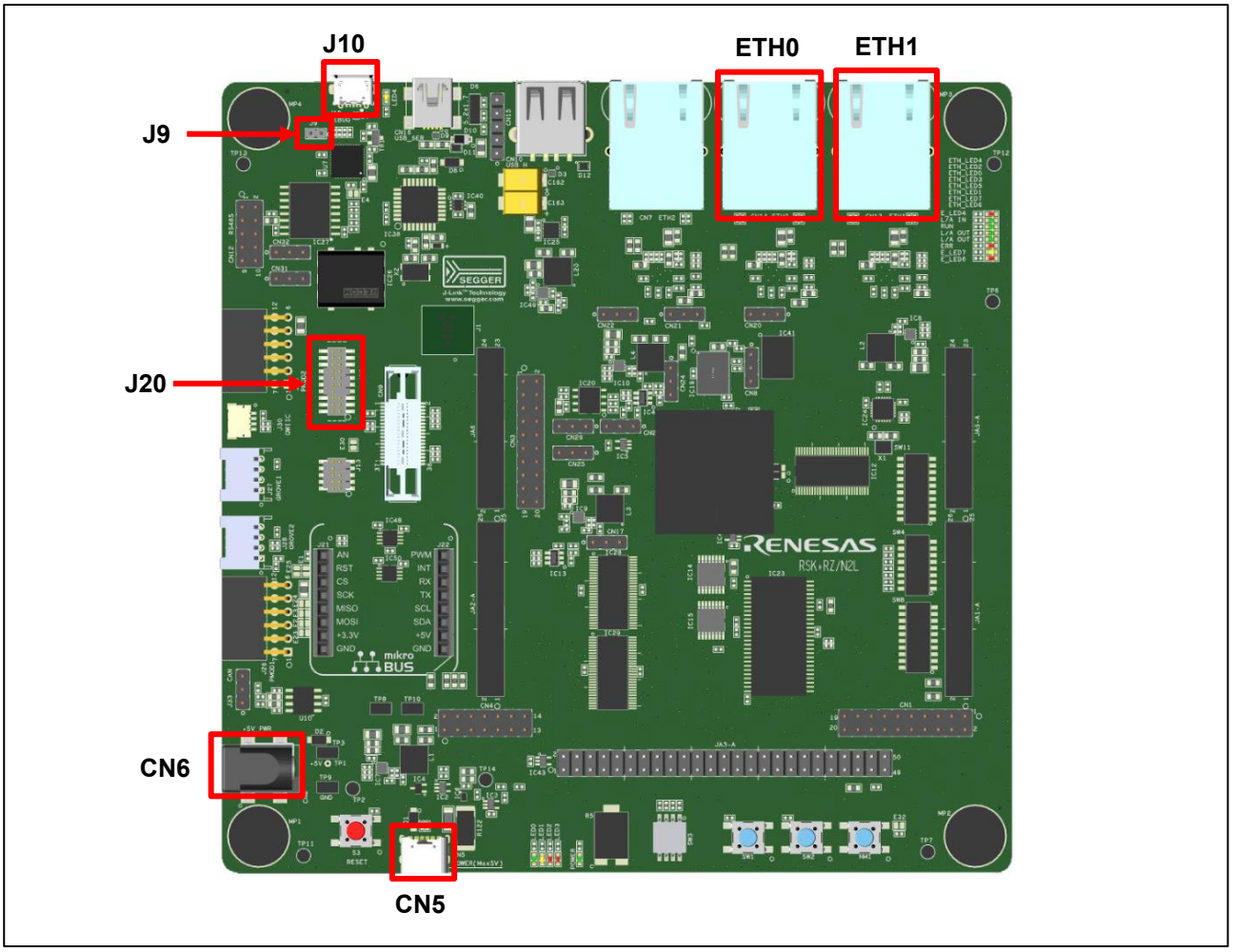


Figure 4.6 Setup RZ/N2L RSK board

2. Power is supplied using USB cable (Type-C) or AC / DC adapter.
   - When using USB cable (Type-C), connect it to the USB connector “CN5” on the RZ/N2L RSK board.
   - When using AC/DC adapter, connect it to the connector “CN6” on the RZ/N2L RSK board.
3. Connect Ethernet cable to the Ethernet Connector “ETH0” or “ETH1”.

4.5 RZ/N2H Evaluation board

4.5.1 Jumper and Switch configuration

This document describes the major hardware properties. Refer to the Evaluation Board for RZ/N2H user's manual and schematic for more board details.

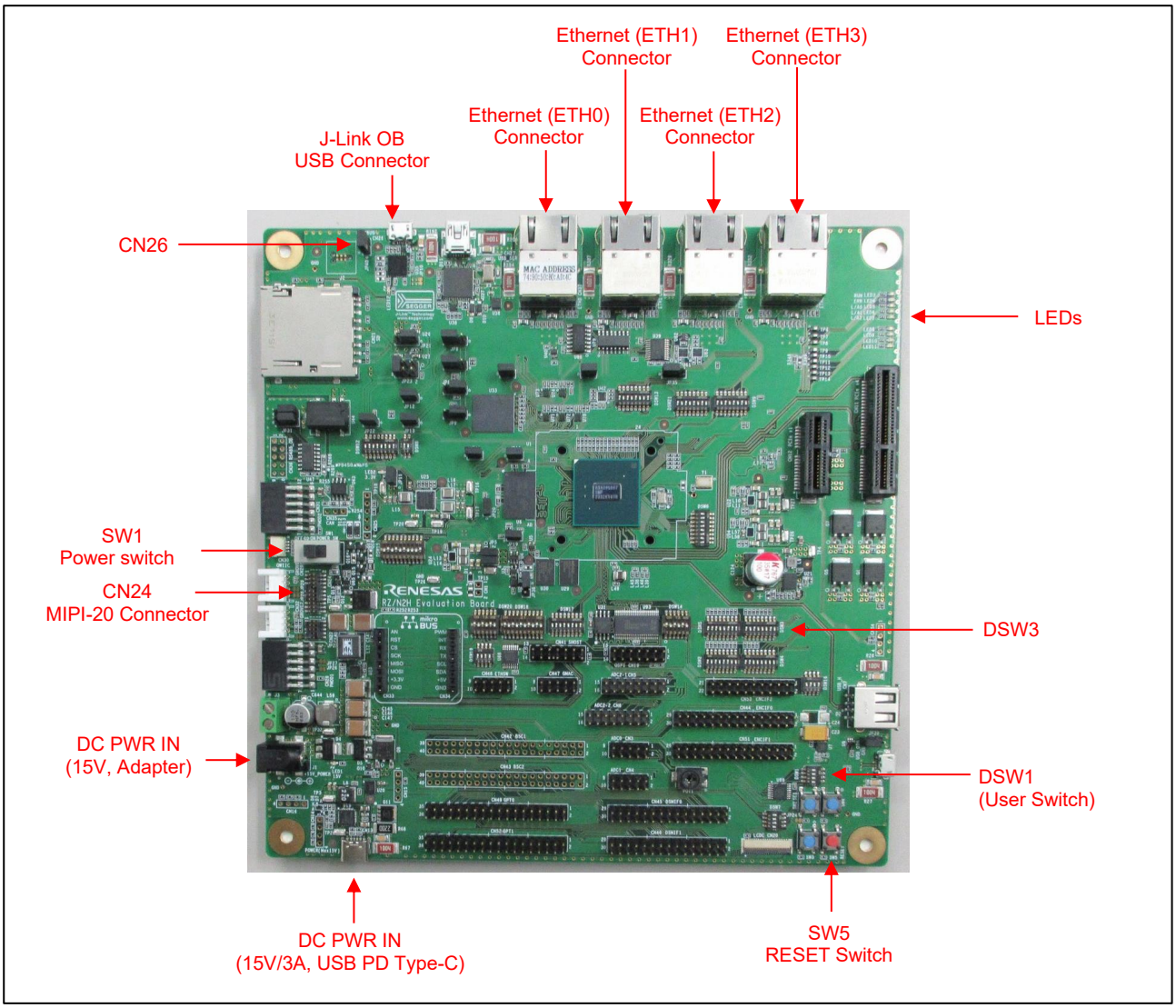


Figure 4.7 RZ/N2H Evaluation Board Layout

Table 4.16 Jumper pin settings

Reference	Jumper Position	Description
CN26	Open-circuit	The on-board debugging function J-Link OB is enabled.

Table 4.17 DSW3 settings

DSW3	Setting	Description
DSW3-1	ON	xSPI1 boot mode (x1 boot serial flash)
DSW3-2	OFF	
DSW3-3	ON	
DSW3-4	ON	CPU0 ATCM 0 wait
DSW3-5	OFF	CPU1 ATCM 1 wait
DSW3-6	OFF	The power-supply voltage of the boot peripheral is 3.3 V.
DSW3-7	ON	JTAG mode = Normal mode
DSW3-8	Unused	-

Other SW settings refer to the User's Manual r20ut5522.

4.5.2 Setup Board

Setting the board for running sample program is shown below.

1. Connect an emulator to the connector.
   - When you use I-jet or J-Link emulator, connect it to CN24.
   - When you use J-Link On-Board emulator, connect USB micro-B to CN26. (Please disconnect JP40 for powering up J-Link OB.)

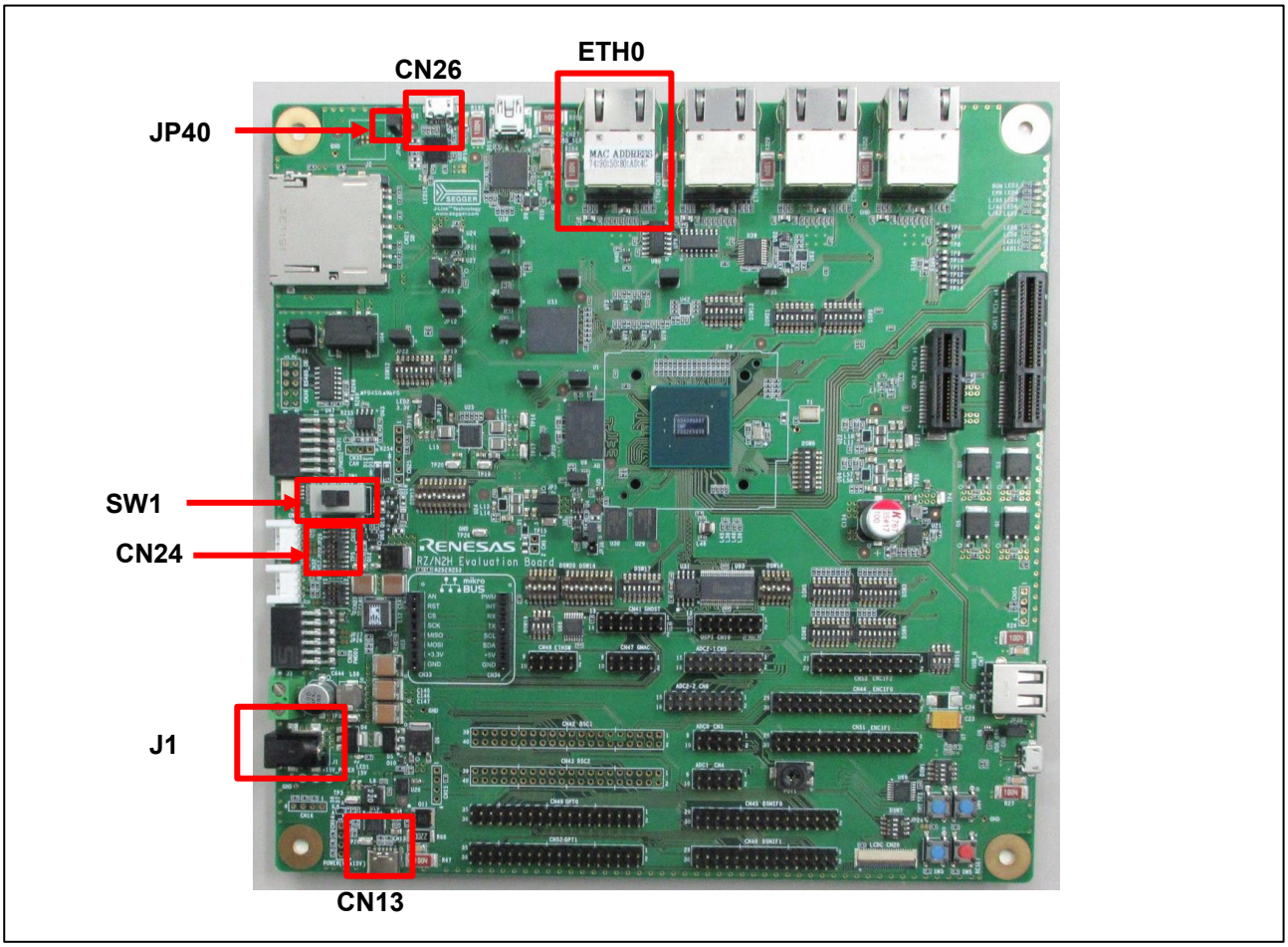


Figure 4.8 Setting RZ/N2H Evaluation Board

2. Power is supplied using a 15V AC/DC adapter or a USB cable (15V/3A USB PD Type-C).
   - When using a 15V AC/DC adapter, connect it to the connector J1.
   - When using a USB cable (15V/3A USB PD Type-C), connect it to the USB connector CN13.
3. Connect Ethernet cable to the Ethernet Connector ETH0.
4. Slide the SW1 POWER_SW to ON. (Provide power to the board)

5. Setup the Host Device

5.1 Configure the Host IP Address

Set an IP address that can communicate with the device in the Ethernet adapter settings on the PC side.

This sample software sets the IP address 192.168.1.10 and subnet mask 255.255.255.0 for the device by default. Therefore, for example, set as follows on the PC side.

- Example setting on the PC side.
  - IP address: 192.168.1.100
  - Subnet mask: 255.255.255.0

5.2 Setup the CODESYS Software

This chapter describes the setup of the CODESYS software.

5.2.1 How to get CODESYS

CODESYS Development system is available from the following web sites.

- CODESYS Store
  - To create your account and login is required to download the CODESYS tools.
    - When you create an account as a business customer, you need:
      - VAT Number if you are European VAT registered Customers.
      - Certificate of Registration as Taxpayer (entrepreneur) if you are non-EU customers.
  - Click the “All versions" tab to get the specified version.

- CODESYS Store North America
  - To create your account and login is required to download the CODESYS tools.
    - United States, Canada and Mexico only can be registered in the "Country" form of the "Address Information".
  - Click the “Versions" tab to get the specified version.

- LINX (Distributor in Japan)
  - To create your account and login is required to download the CODESYS tools.
    - Only Japanese companies can create the account.

5.2.2 Startup CODESYS Tools

After installing the CODESYS, please launch the CODESYS tools shown below.

Table 5.1 CODESYS tools

Name	Description	Note
CODESYS V3	IDE	-
CODESYS Gateway V3	Software Gateway	This may be already started by Windows Start Up Process.
CODESYS Control Win V3	Software PLC	This may be already started by Windows Start Up Process.

If the CODESYS is launched properly, the following window is shown.

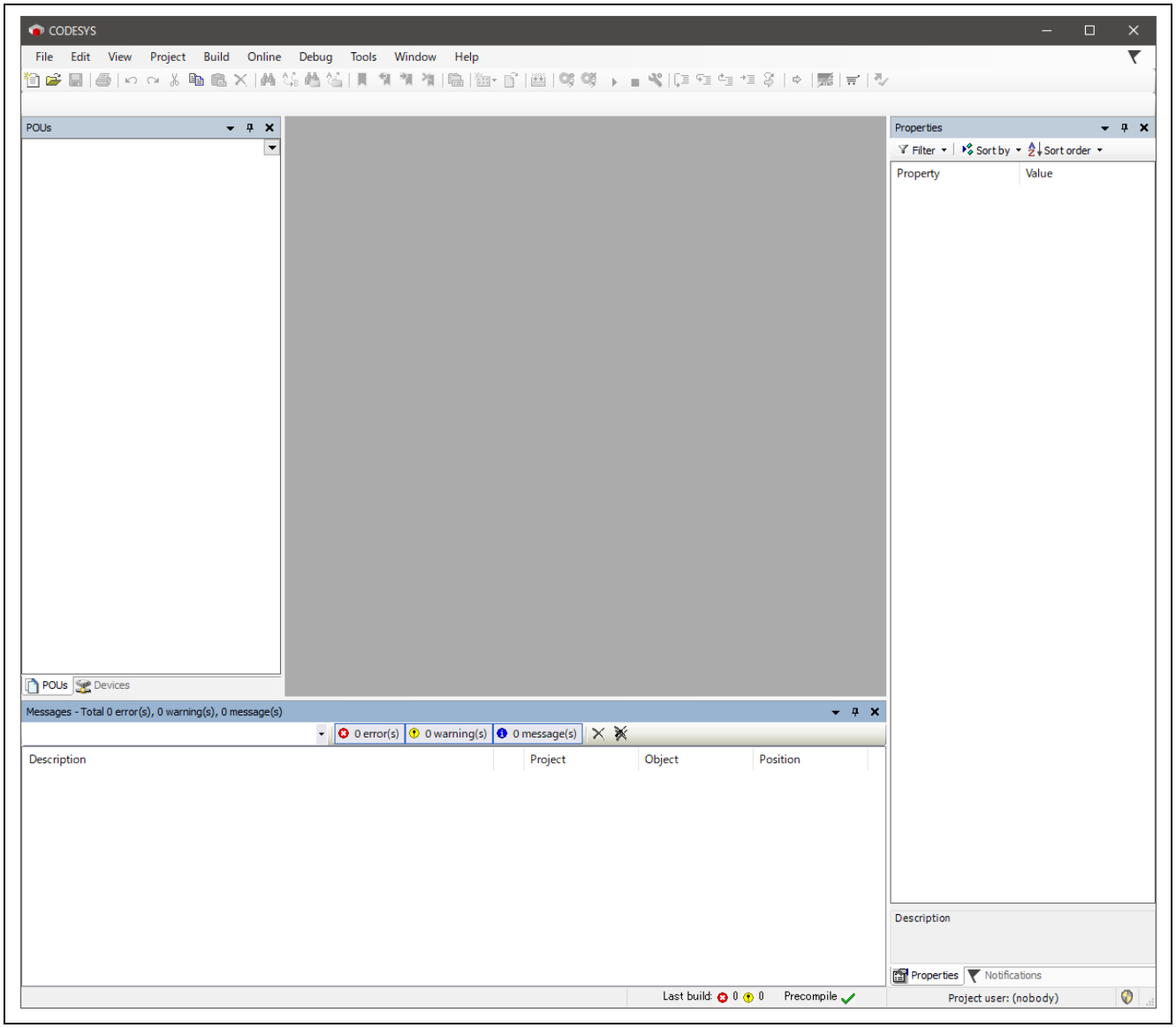


Figure 5.1 CODESYS Initial Window

If the CODESYS Gateway and Control Win SysTray is launched properly, the following icons are shown in notification area of Windows Tool Bar. (The left icon is of the CODESYS Gateway, and the right one is of the CODESYS Control Win SysTray)

Figure 5.2 CODESYS Icons

Please click each icon and click “Start Gateway” and “Start PLC”.

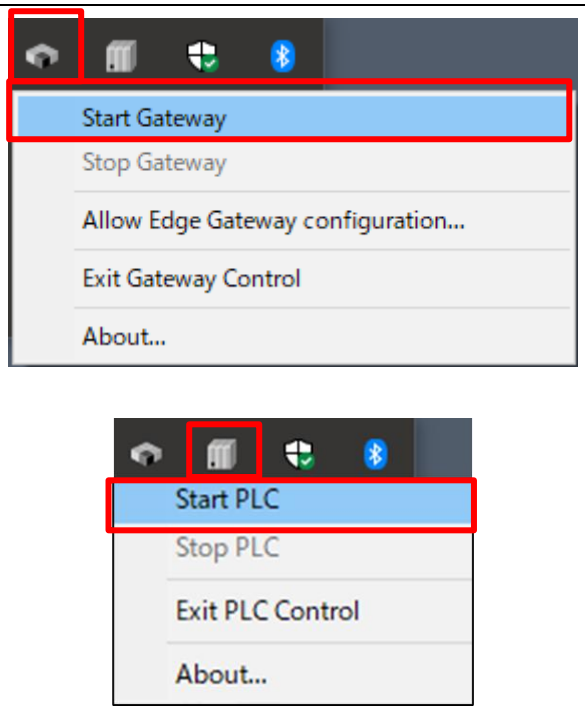


Figure 5.3 Start Gateway and PLC

If the Gateway and PLC are started properly, the icons change like the following image.

Figure 5.4 Icons successfully started

5.2.3 Install EDS File into CODESYS

In the CODESYS, please open “tools” > “Device Repository” in tool bar.

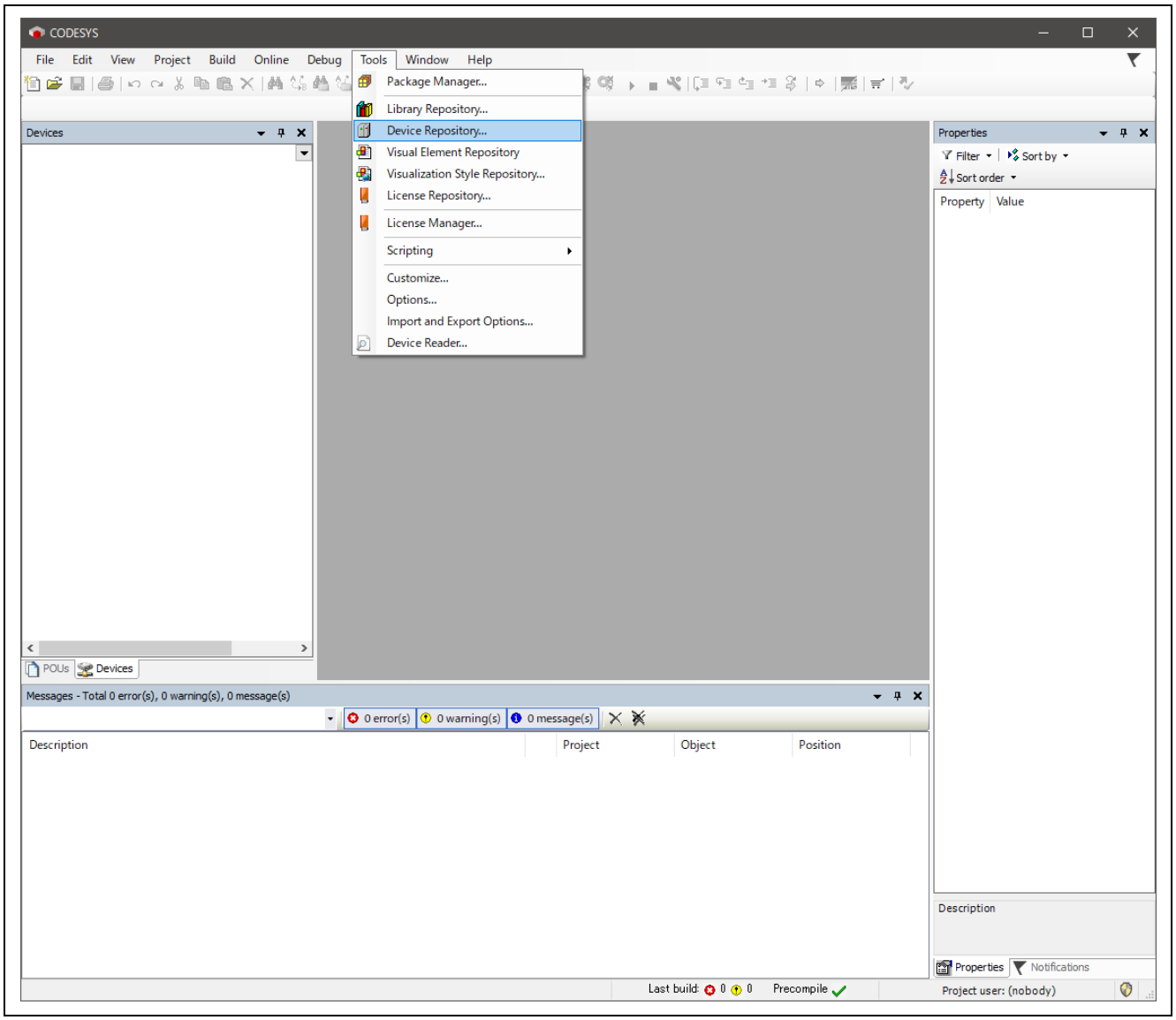


Figure 5.5 Device Repository in CODESYS

Please click the “Install” to open dialog to select EDS file, and select the EDS file “renesas_opener_sample_app.eds” in “scanner” directory.

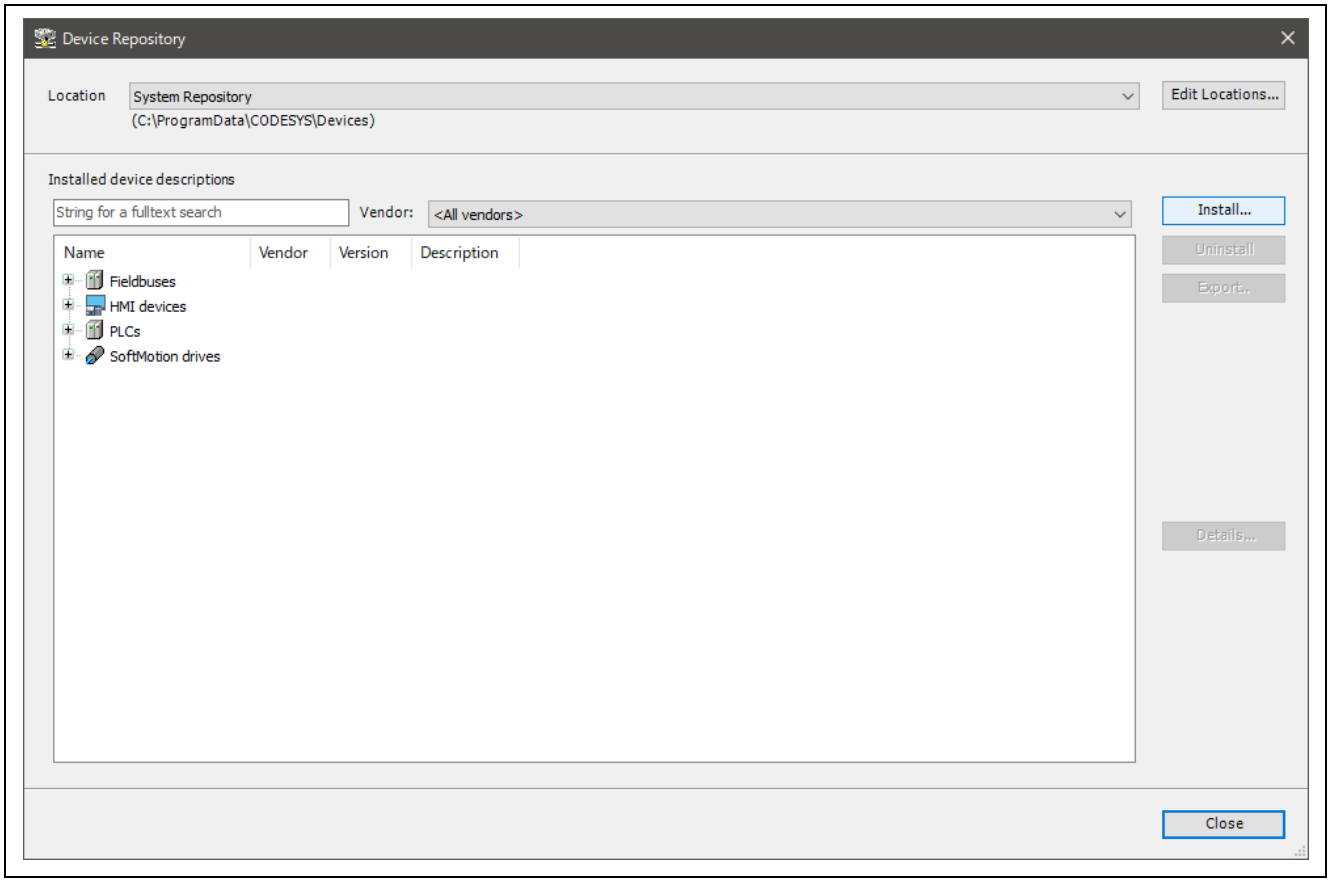


Figure 5.6 Click "Install" in Device Repository Window

If the Renesas OpENer Device is shown in the blue line as EtherNet/IP Remote Adapter, please click “close” to close this window.

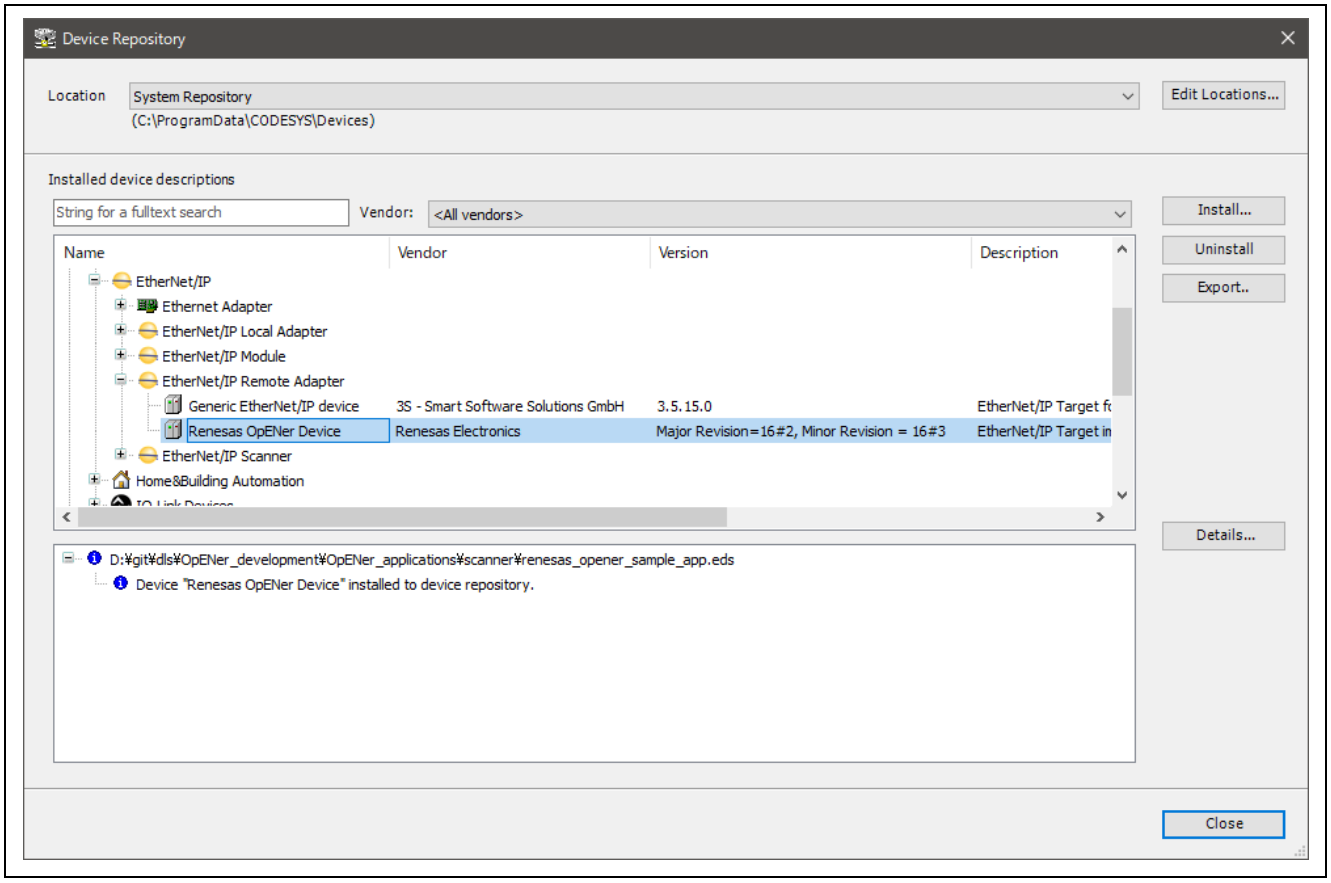


Figure 5.7 EtherNet/IP Remote Adapter

Preparation for using CODESYS is now complete. Continue with the program operation in Chapter 6 and then operate CODESYS again in Chapter 7.

6. Running Sample Application

This chapter describes how to download and run the program.

Since the procedure varies by device, please refer to the appropriate section according to the device you are using.

The procedures in this chapter use RZ/N2L as the representative example for single-core projects, and RZ/T2M (CR52_Dual) as the representative example for multi-core projects.

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

Table 6.1 Folder Names and Project Names by Target Device

Target Device	Folder Name	Core	Project Name (Primary)	Project Name (Secondary)
RZ/N2L	RZN2L_EtherNetIP_OpENer_RSK_rev0410	Single (Cortex-R52)	RZN2L_RSK_OpENer_CR52	(None)
RZ/T2M, RZ/T2ME	RZT2M_EtherNetIP_OpENer_RSK_rev0410	CR52_Dual (Cortex-R52 CPU0 and Cortex-R52 CPU1)	RZT2M_RSK_OpENer_CR52_Dual_Prim	RZT2M_RSK_OpENer_CR52_Dual_Second
RZ/T2H	RZT2H_EtherNetIP_OpENer_EVB_rev0410	CR52_Dual (Cortex-R52 CPU0 and Cortex-R52 CPU1)	RZT2H_EVB_OpENer_CR52_Dual_Prim	RZT2H_EVB_OpENer_CR52_Dual_Second
		CR52_CA55 (Cortex-R52 CPU0 and Cortex-A55 Core2)	RZT2H_EVB_OpENer_CR52_CA55_Prim	RZT2H_EVB_OpENer_CR52_CA55_Second
RZ/N2H	RZN2H_EtherNetIP_OpENer_EVB_rev0410	CR52_Dual (Cortex-R52 CPU0 and Cortex-R52 CPU1)	RZN2H_EVB_OpENer_CR52_Dual_Prim	RZN2H_EVB_OpENer_CR52_Dual_Second
		CR52_CA55 (Cortex-R52 CPU0 and Cortex-A55 Core2)	RZN2H_EVB_OpENer_CR52_CA55_Prim	RZN2H_EVB_OpENer_CR52_CA55_Second

If you are using FSP v4.1.0 or later v4.x.x, refer Appendix F to resolve issues with building or running the sample program.

6.1 Board IP Address Setting

The IP address is set in src/renesas/application/opener_port_instance.c.

The following addresses are used in the example:

IP address : 192.168.1.10

Subnet mask : 255.255.255.0

Default gateway : 192.168.1.2

Host Name : OPENER_NETIF0

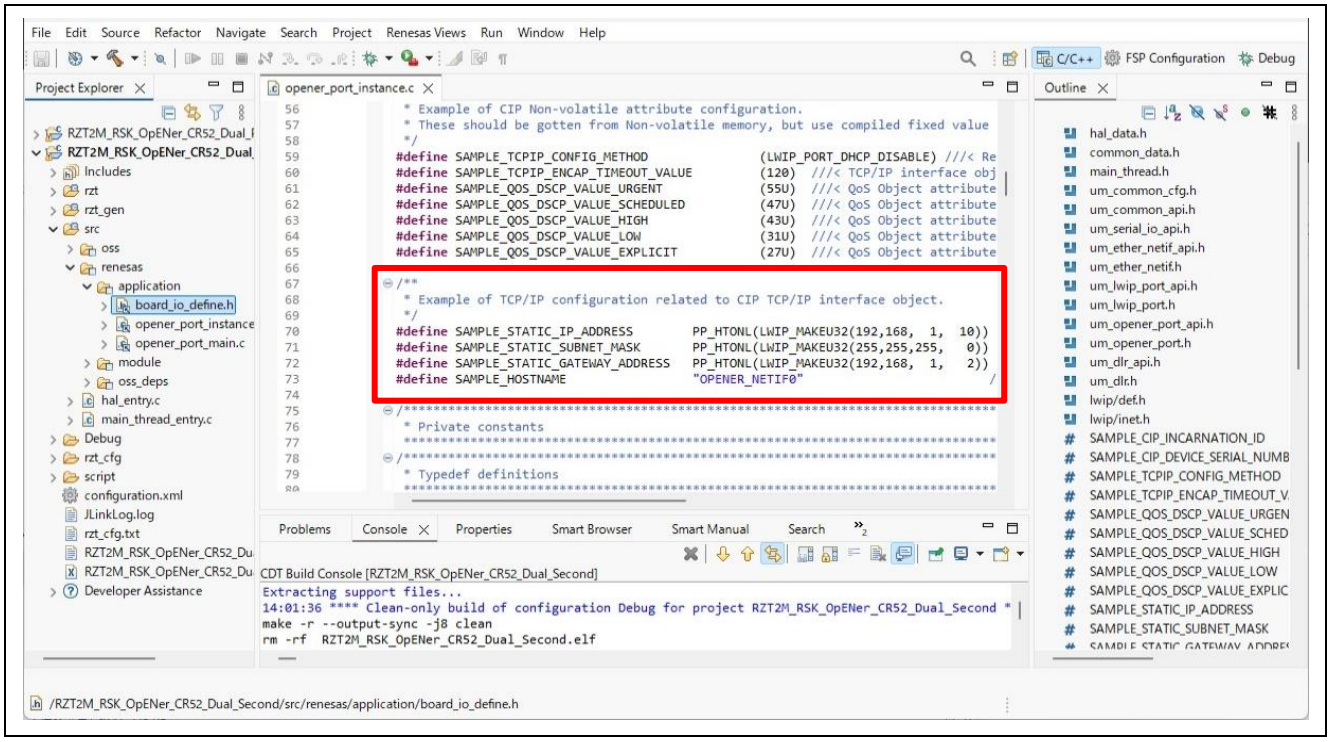


Figure 6.1 Static IP address

6.2 Debugging with EWARM

6.2.1 Single-Core Project (RZ/N2L)

6.2.1.1 Startup EWARM

1. Open the EWARM.
2. Click “Open Workspace…” in the File tab.

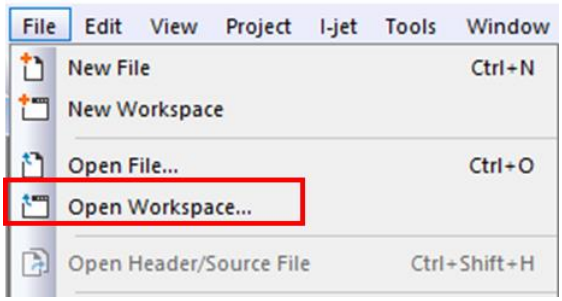


Figure 6.2 EWARM file tab

3. Select the Workspace File(.eww) and click the Open button.

   “(Extracted folder path)\project\rzn2l_rsk\CR52\ewarm\RAM\RZN2L_RSK_OpENer_CR52”

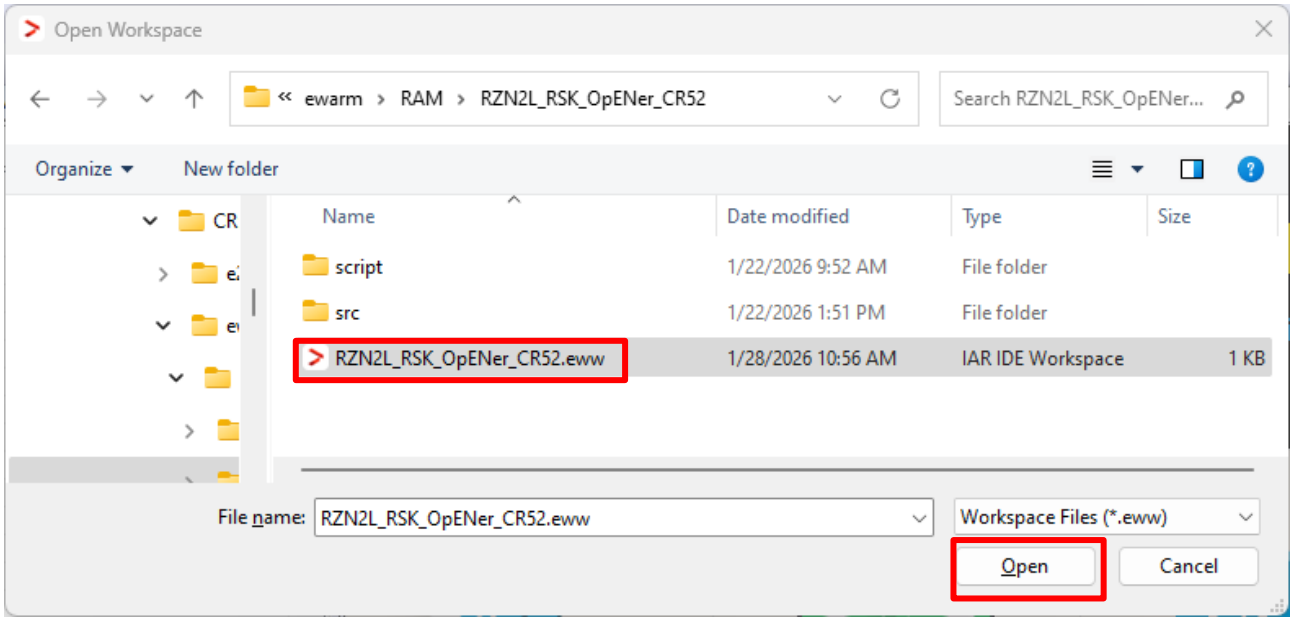


Figure 6.3 Open project file

6.2.1.2 Program execution

(1) How to generate source code and how to build

A) Click the “Tools” -> “FSP Smart Configurator” on tool bar. If you have not set up FSP Smart Configurator yet on EWARM, refer to r01an6434ejxxxx-rzt2-rzn2-fsp-getting-started.pdf in which section 5.4 describes how to set it up.

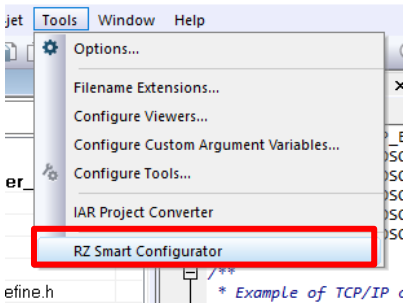


Figure 6.4 Tools tab

B) Click “Generate Project Content” button then generate rz, rz_gen, rz_cfg folder.

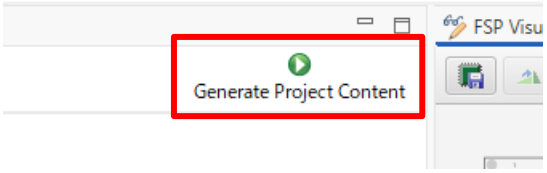


Figure 6.5 Generate Project Content

C) Click the Make button on the tool bar to build. Once the build is completed, the build message is displayed in the Build Console window that displays compilation target files and the number of error/warnings.

Figure 6.6 Make button

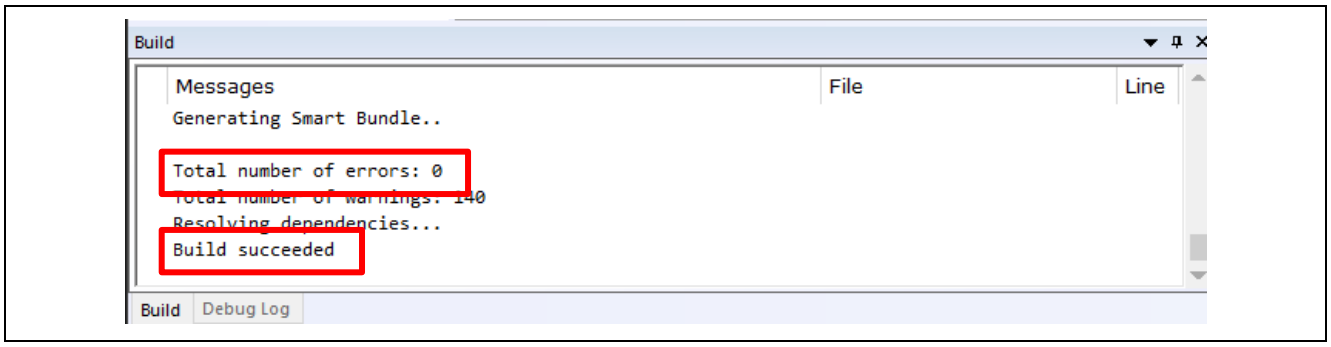


Figure 6.7 Build console

(2) Download application and run debugger

A) Click the Debug button in tool bar to download the built application program and launch the debugger. The program will break at the first code in main function.

Figure 6.8 Debug button

B) Click the Go button.

Figure 6.9 Go button

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

6.2.2.1 Startup EWARM

1. Open the EWARM.
2. Click “Open Workspace…” in the File tab.

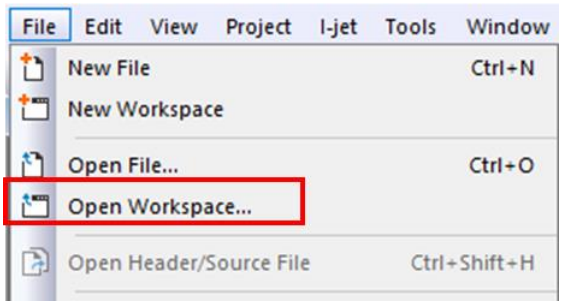


Figure 6.10 EWARM file tab

3. Select the Workspace File(.eww) in the Primary project and click the Open button.
   “\project\rzt2m_rsk\CR52_Dual\ewarm\RAM\RZT2M_RSK_OpENer_CR52_Dual_Prim”

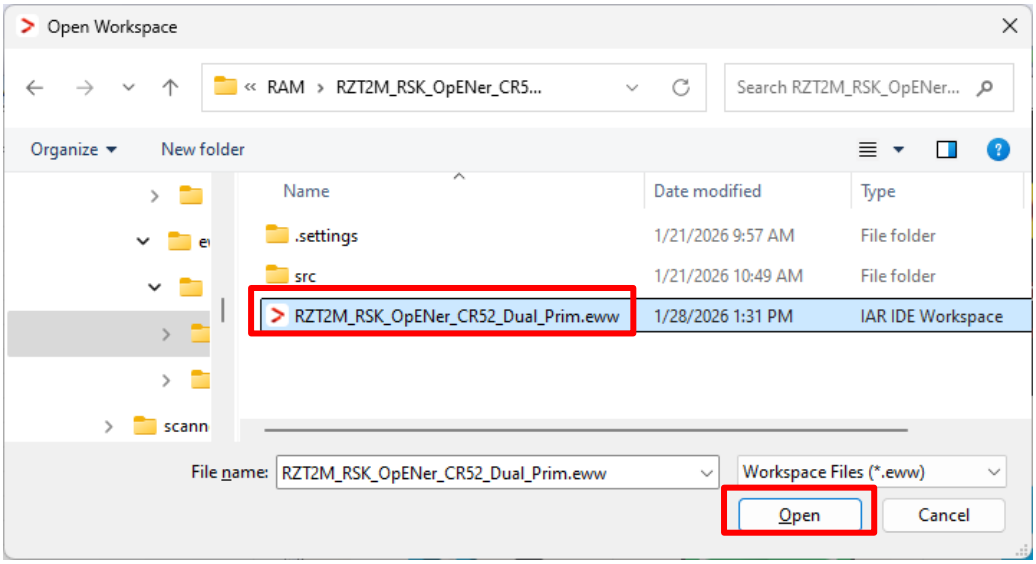


Figure 6.11 Open the Workspace of the Primary project

4. Open another EWARM window.
5. Click “Open Workspace…” in the File tab, and open the Workspace File(.eww) in the Secondary project.

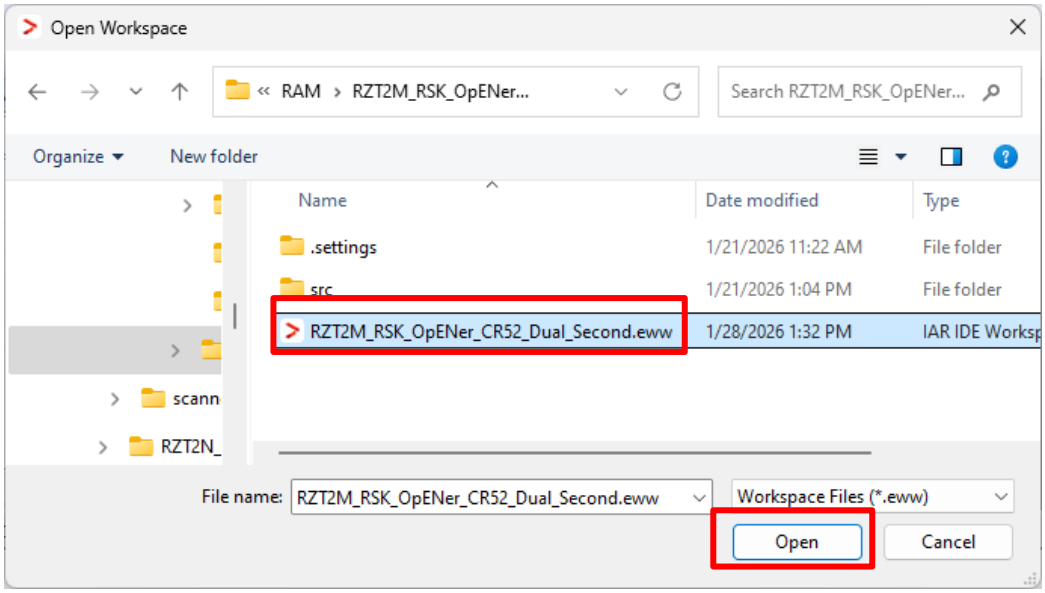


Figure 6.12 Open the Workspace of the Secondary project

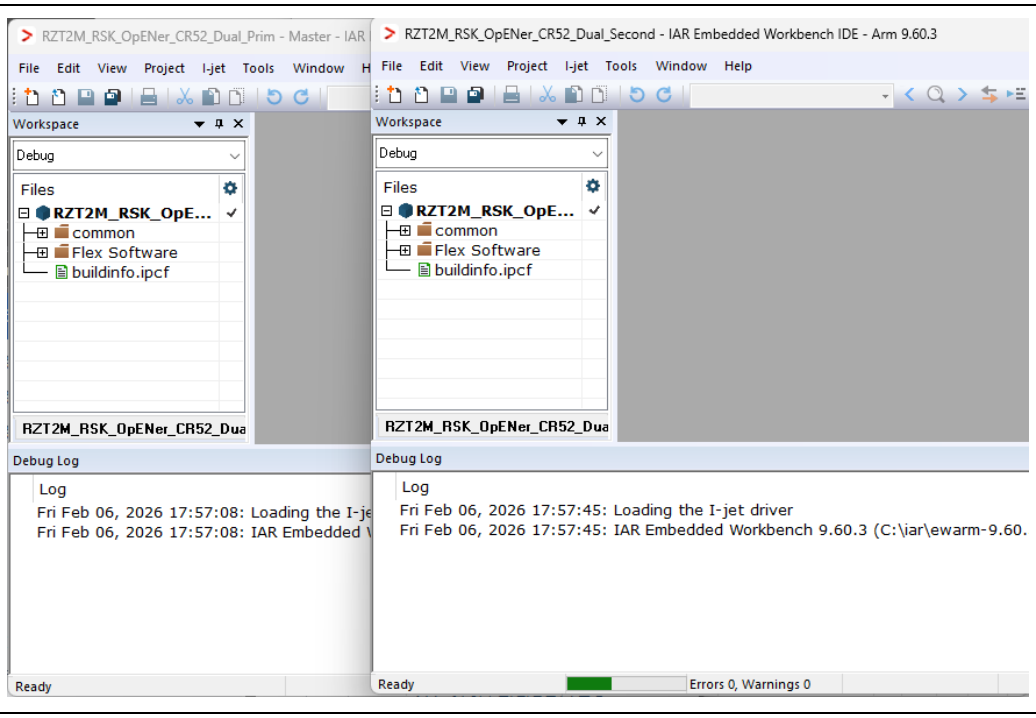


Figure 6.13 EWARM windows

6.2.2.2 Program execution

(1) How to generate source code and how to build

A) Click the “Tools” -> “FSP Smart Configurator” on the Primary project.

Note: If you have not set up FSP Smart Configurator yet on EWARM, refer to r01an6434ejxxxx-rzt2-rzn2-fsp-getting-started.pdf in which section 5.4 describes how to set it up.

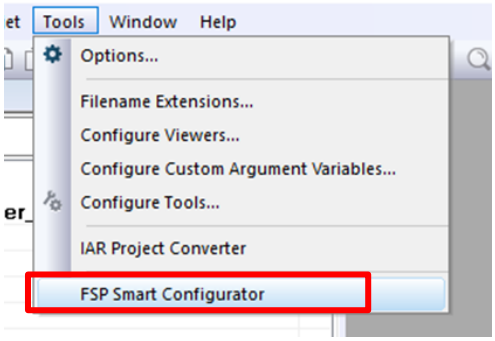


Figure 6.14 Tools tab on the Primary project

B) Click “Generate Project Content” button then generate rz, rz_gen, rz_cfg folder.

Figure 6.15 Generate Project Content

C) Click the Make button to build the Primary project. Once the build is completed, the build message is displayed in the Build Console window that displays compilation target files and the number of error/warnings.

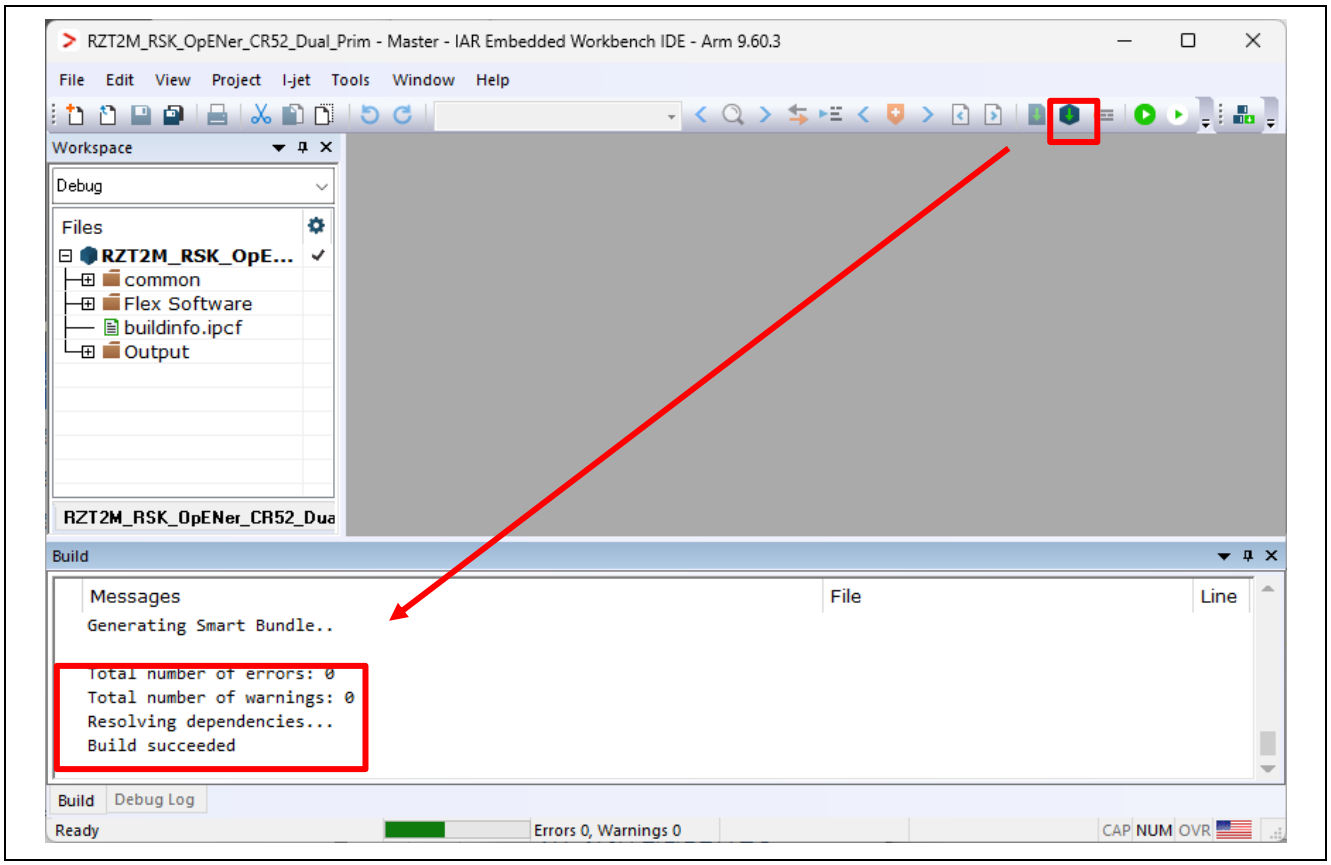


Figure 6.16 Make button on the Primary project

D) Click the “Tools” -> “FSP Smart Configurator” on the Secondary project.

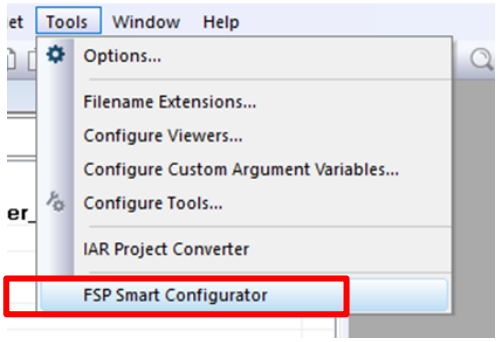


Figure 6.17 Tools tab on the Secondary project

E) Click “Generate Project Content” button then generate rz, rz_gen, rz_cfg folder.

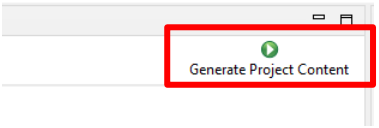


Figure 6.18 Generate Project Content

F) Click the Make button to build the Secondary project. Once the build is completed, the build message is displayed in the Build Console window that displays compilation target files and the number of error/warnings.

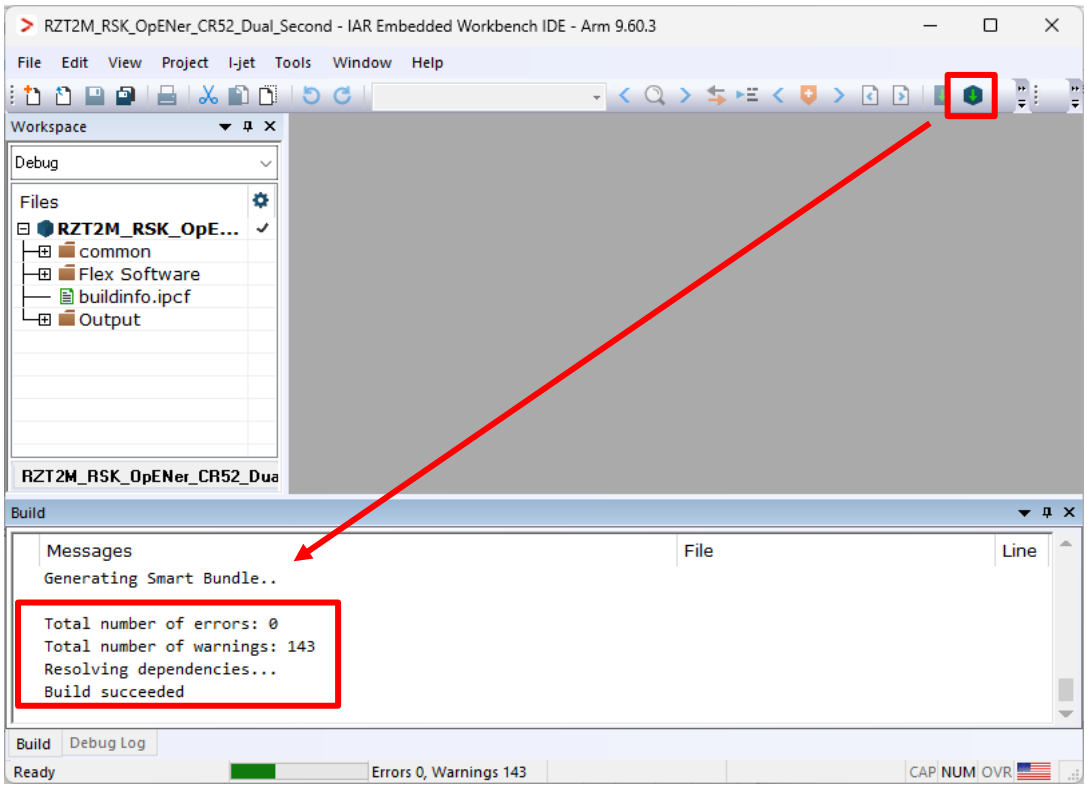


Figure 6.19 Make button on the Secondary project

G) Close the Workspace of the Secondary project.

(2) Download application and run debugger

A) Click the Debug button to download the application program of the Primary project and launch the debugger. Then automatically the Workspace of the Secondary project will open.

Figure 6.20 Debug button

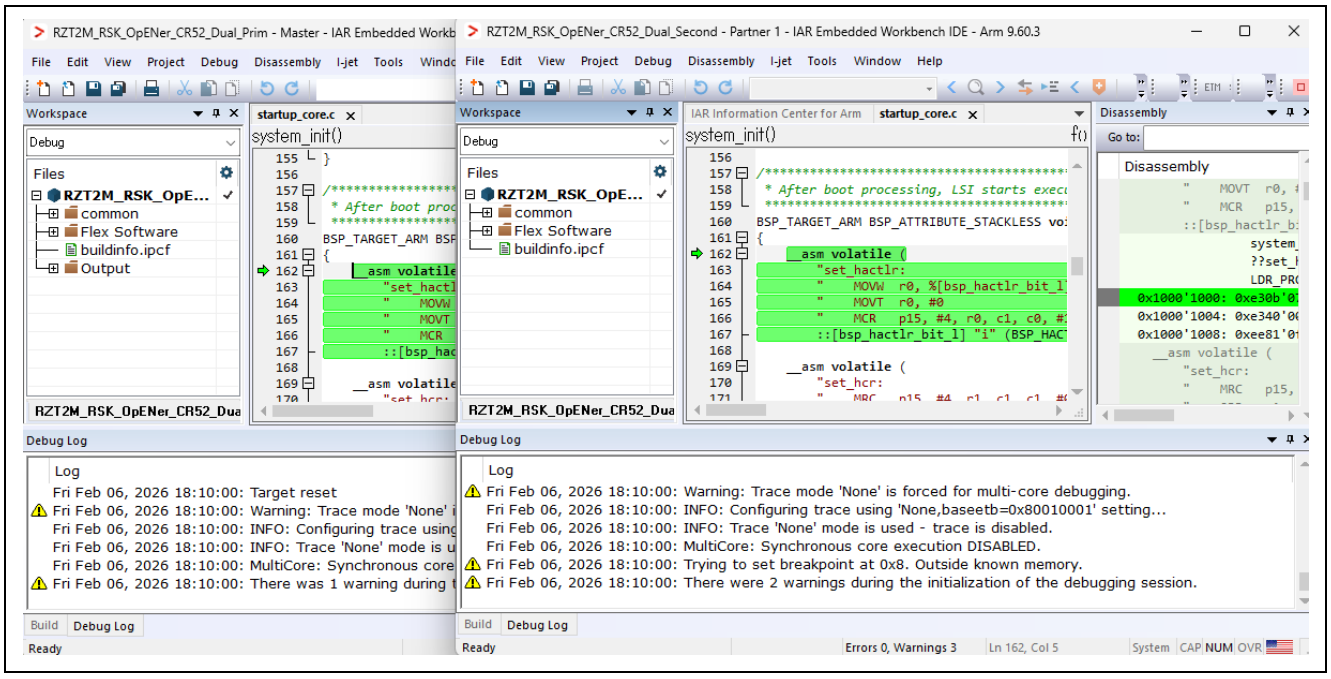


Figure 6.21 The Workspace of the Secondary project automatically opens

B) Click the Go button on the Primary project.

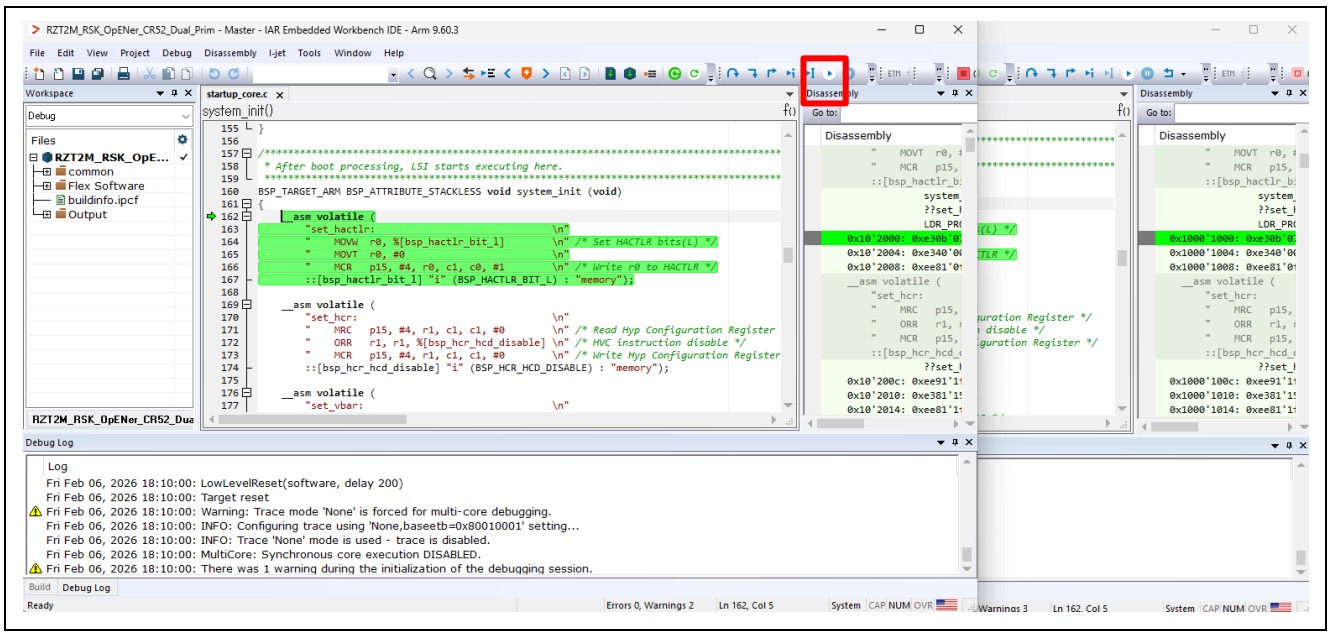


Figure 6.22 Go button on the Primary project

C) Click the Go button on the Secondary project.

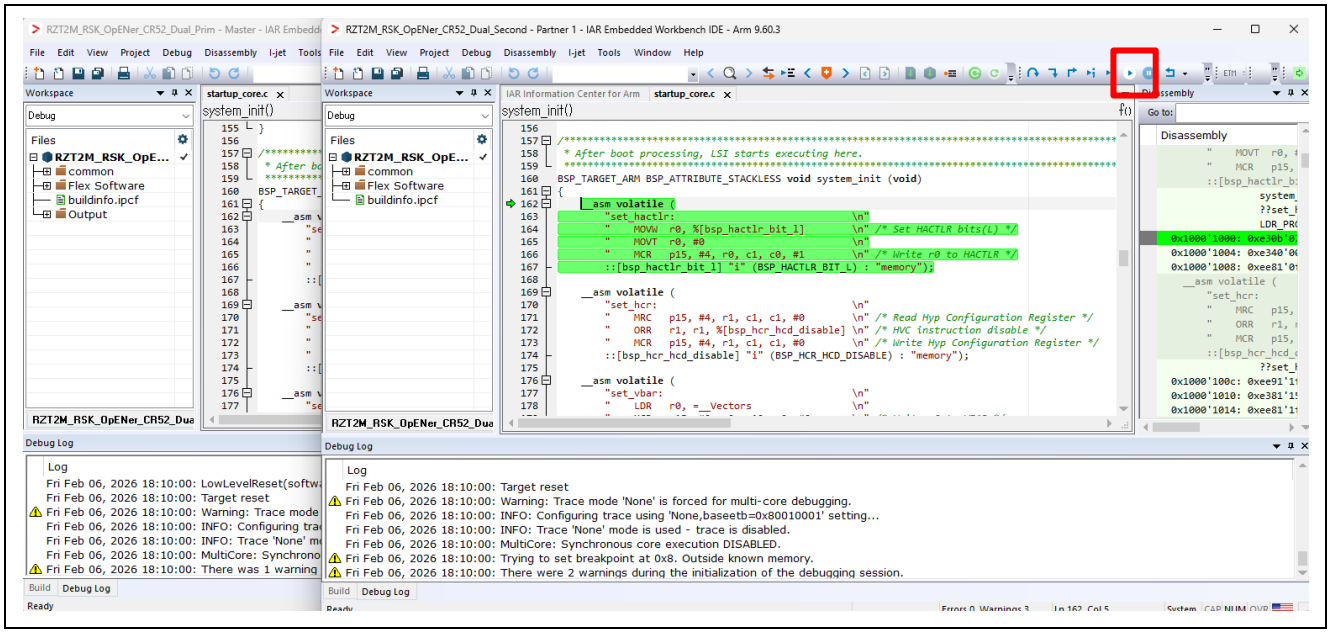


Figure 6.23 Go button on the Secondary project

6.3 Debugging with e² studio

6.3.1 Single-Core Project (RZ/N2L)

6.3.1.1 Startup e² studio

1. Open the e² studio and select a directory as workspace.
2. Click “Import” in File tab.

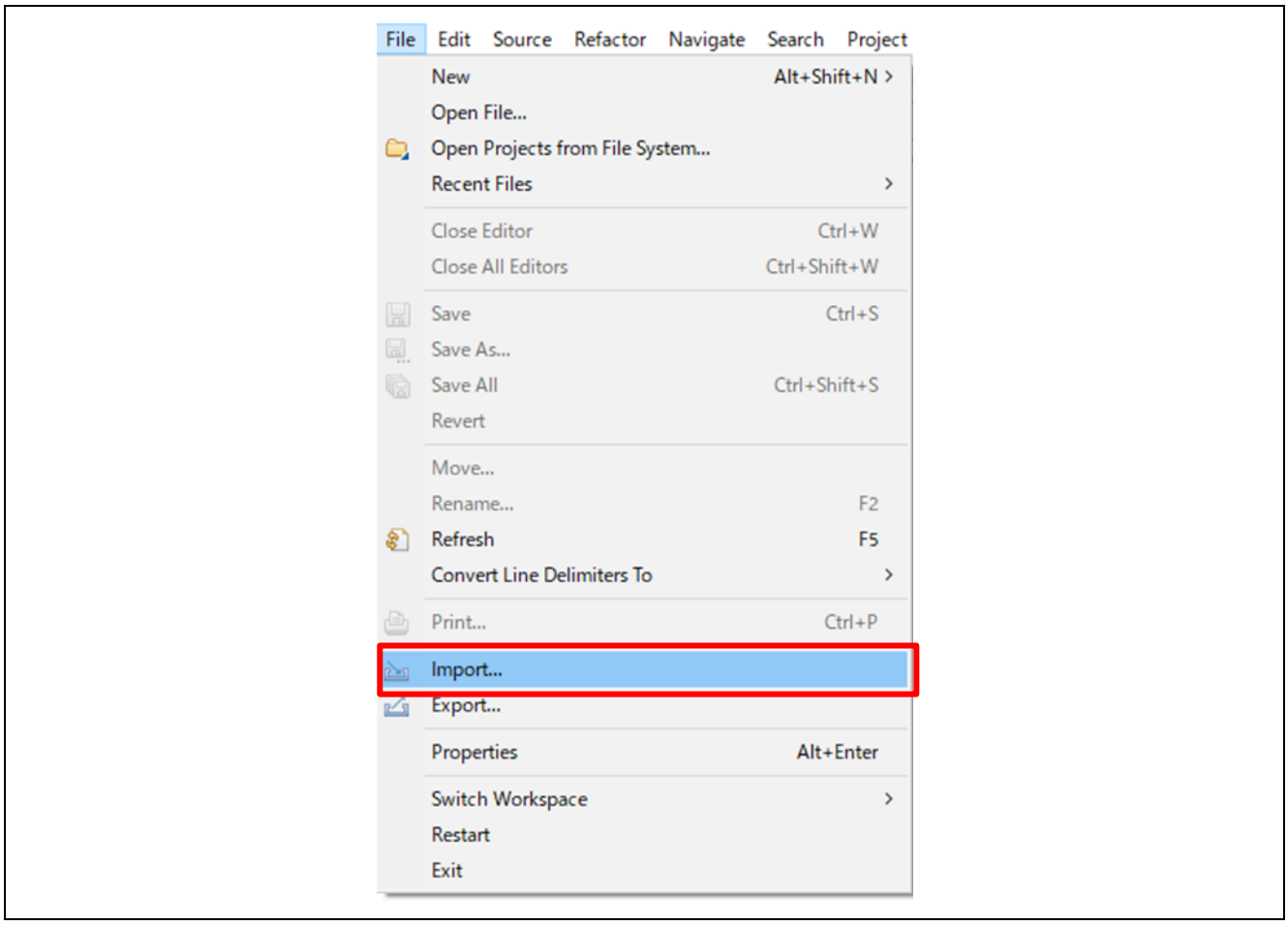


Figure 6.24 e² studio File tab

3. Click “Existing Projects into Workspace” in “General” and Click “Next”.

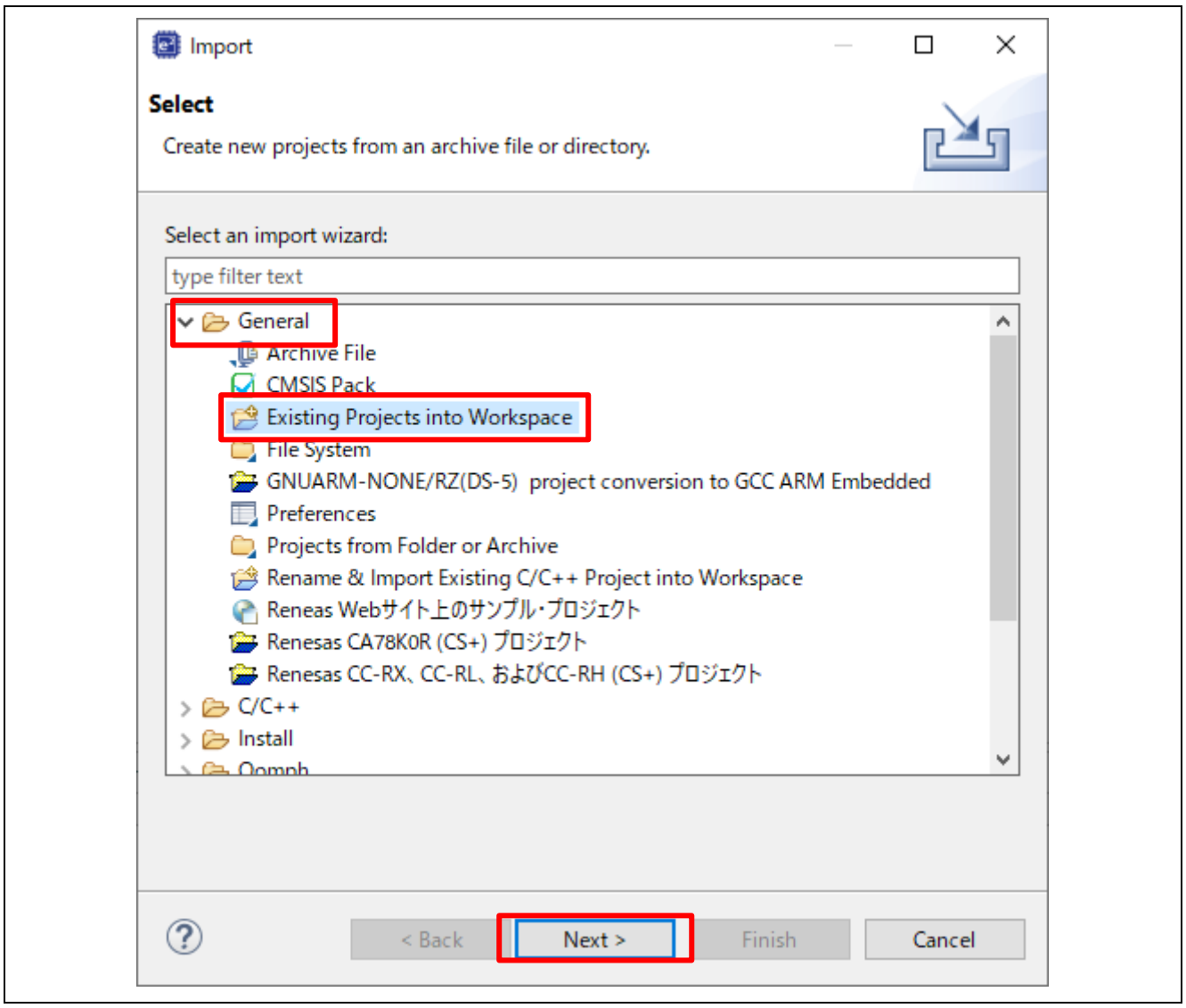


Figure 6.25 Select import type

4. Import project by the following steps.
   A) Set the root directory.
      ➔ “(Extracted zip folder path)\project\rzn2l_rsk\CR52\e2studio\RAM”
   B) Select the project.
   C) Click “Finish”

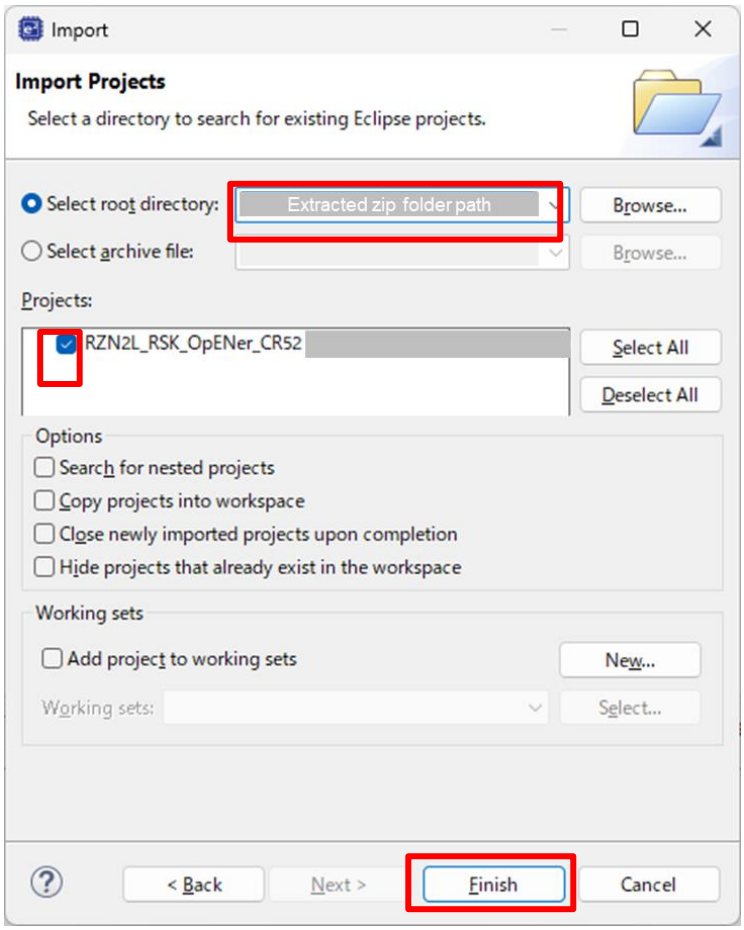


Figure 6.26 Import the project on e² studio

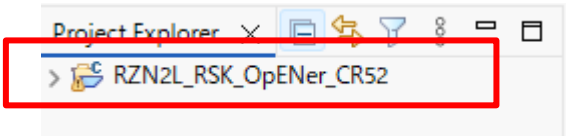


Figure 6.27 Open the project

6.3.1.2 Program execution

(1) How to generate source code and how to build

A) Click the configuration.xml.

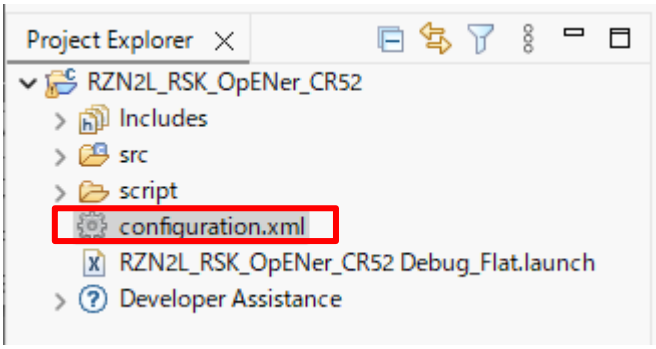


Figure 6.28 Configuration

B) Click “Generate Project Content” button then generate rz, rz_gen, rz_cfg folder.

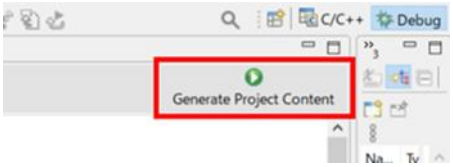


Figure 6.29 Generate Project Content

C) Click the Build button in tool bar to build the project and confirm that there is no error message in the build message log.

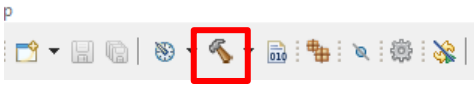


Figure 6.30 Build button

(2) Download application and run debugger

A) Click the Debug button in tool bar to download the built application program and launch the debugger.

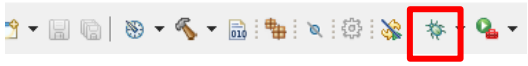


Figure 6.31 Debug button

B) Click the “Switch” button.

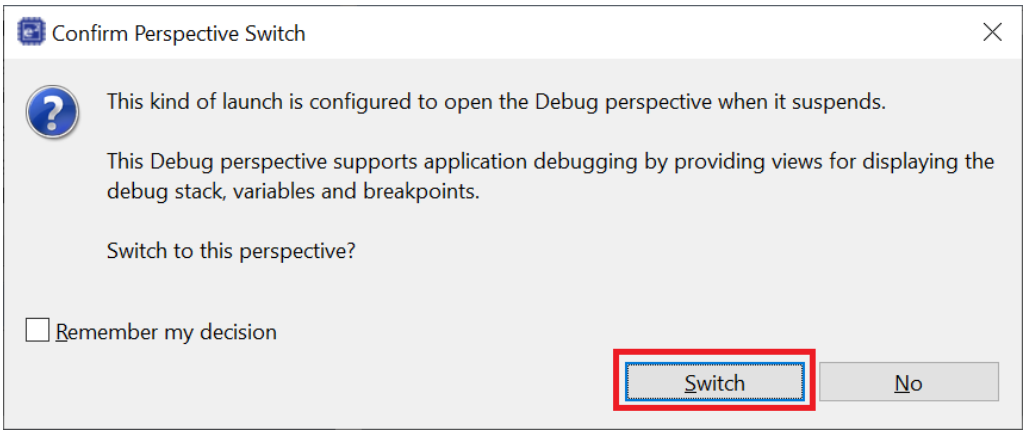


Figure 6.32 Confirm Perspective Switch

C) The program will break at “system_init” for startup.

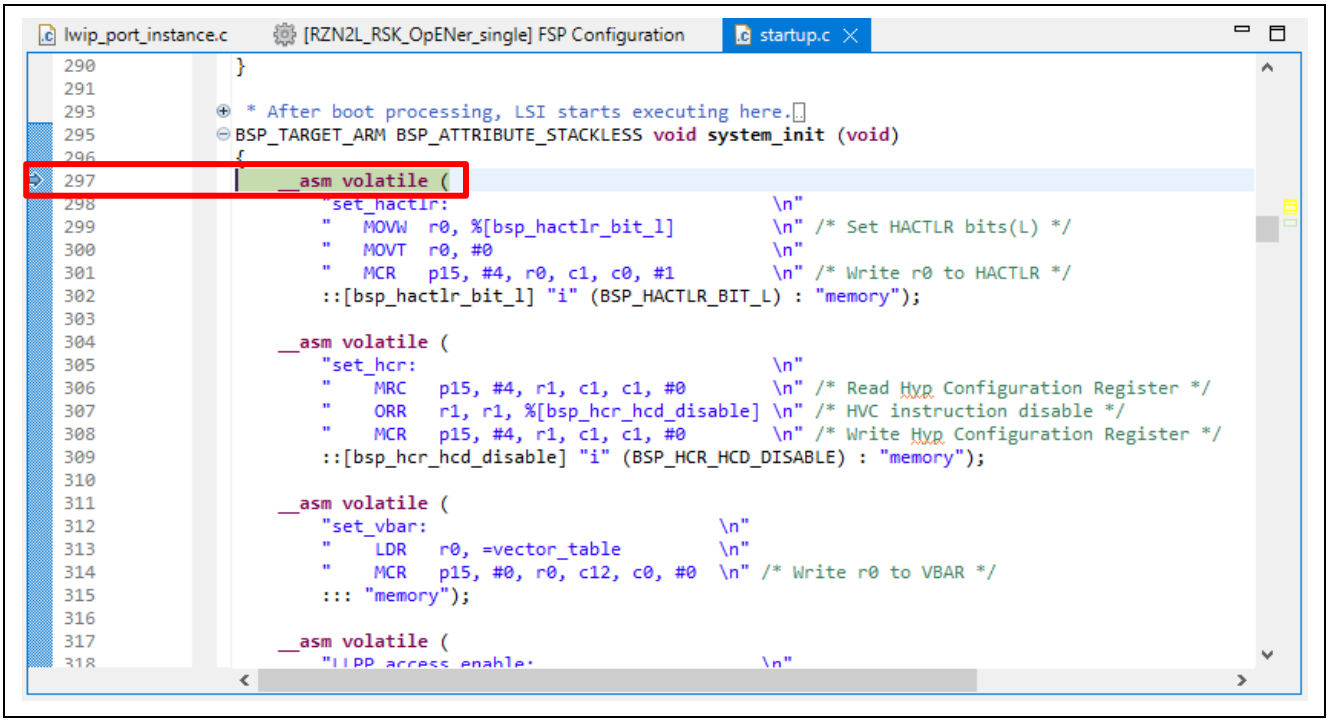


Figure 6.33 Break point 1

D) Click the Resume button. The program will break at the first of the main function.

Figure 6.34 Resume button

E) Click the Resume button again to execute the program.

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

6.3.2.1 Startup e² studio

1. Open the e² studio and select a directory as workspace.
2. Click “Import” in File tab.

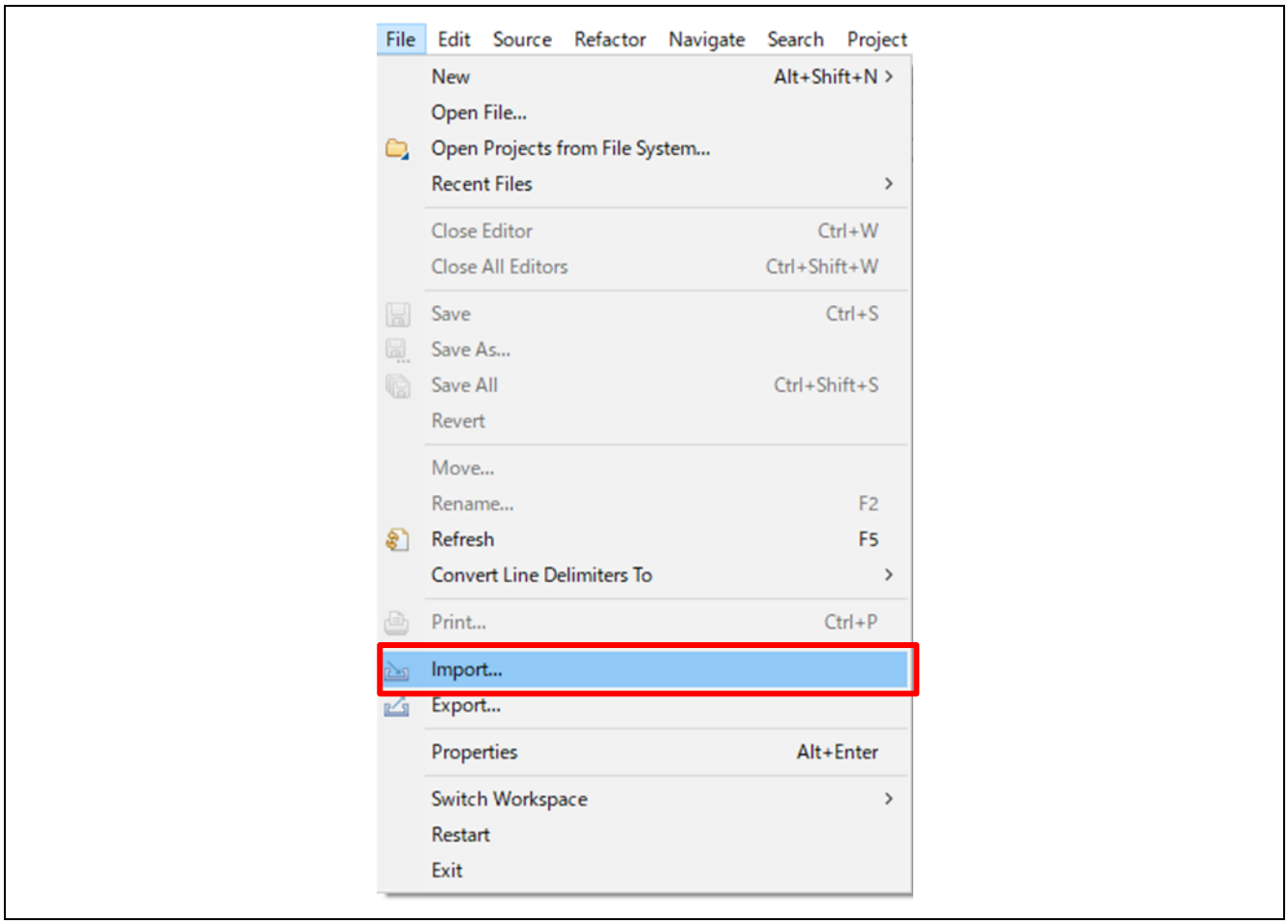


Figure 6.35 e² studio File tab

3. Click “Existing Projects into Workspace” in “General” and Click “Next”.

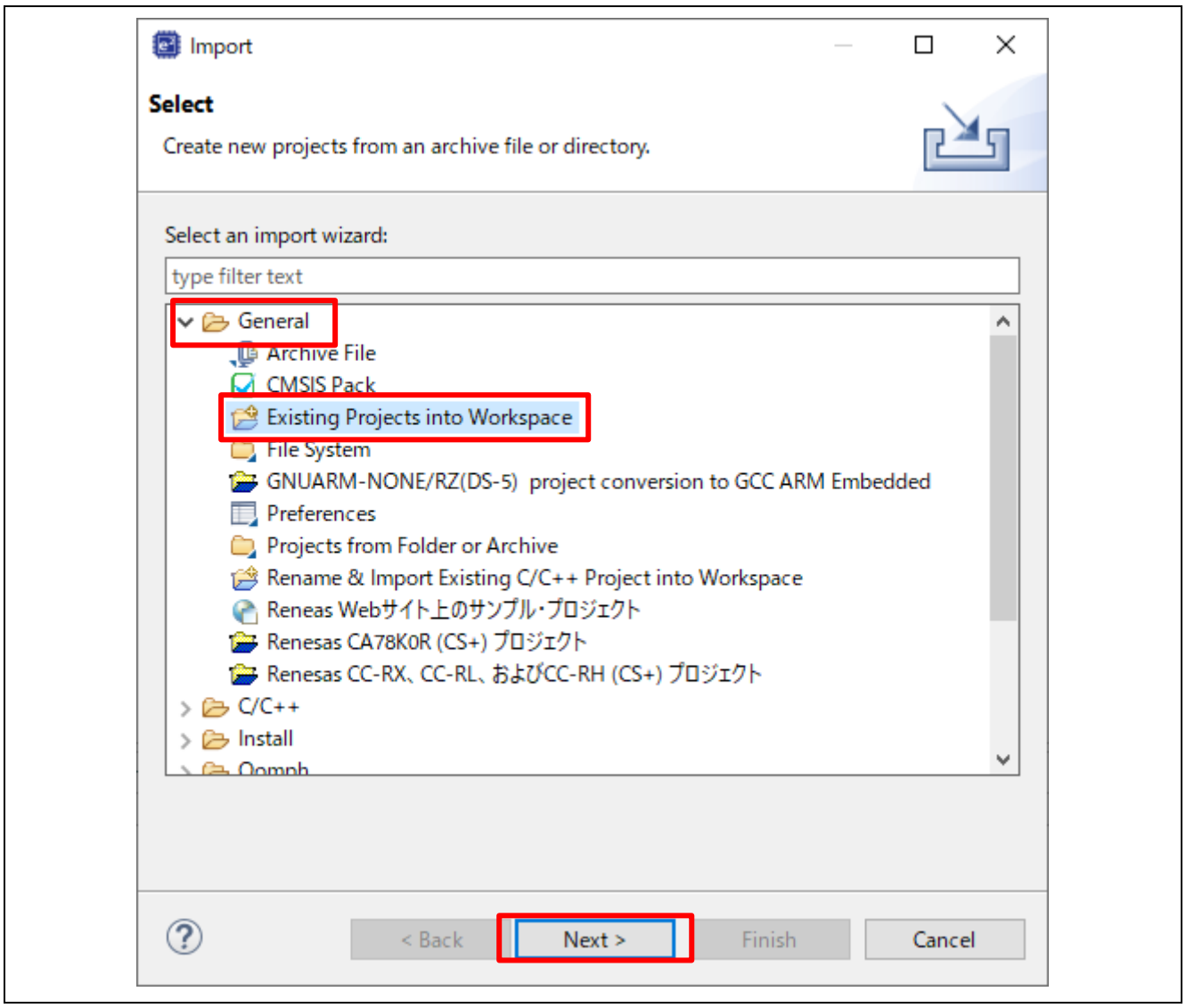


Figure 6.36 Select import type

4. Import project by the following steps.
   A) Set the root directory.
      "(Extracted zip folder path)\project\rzt2m_rsk"
   B) Select 2 projects.
   C) Click "Finish"

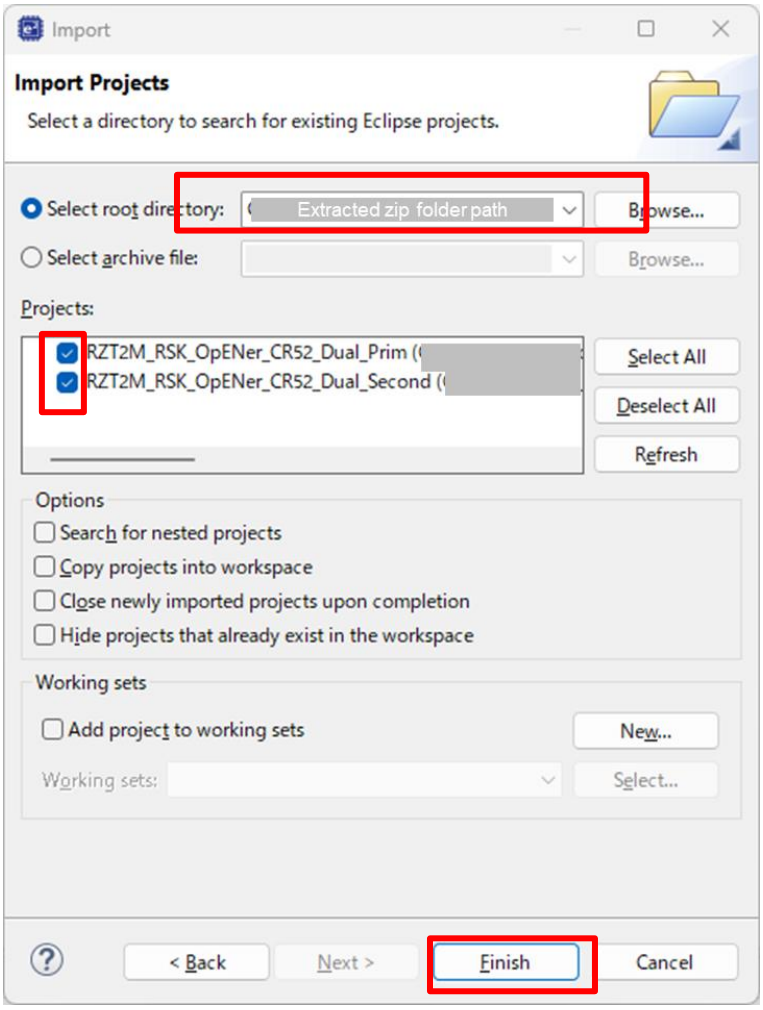


Figure 6.37 Import two projects on e² studio

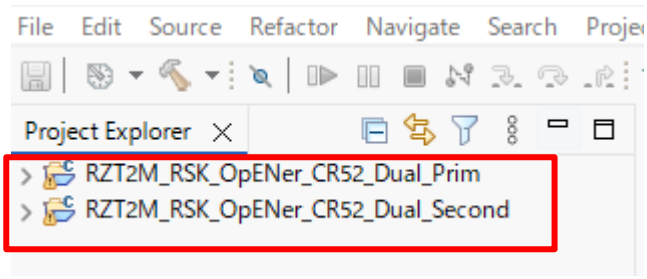


Figure 6.38 Open 2 projects

6.3.2.2 Program execution

(1) How to generate source code and how to build

A) Click the configuration.xml of the Primary project.

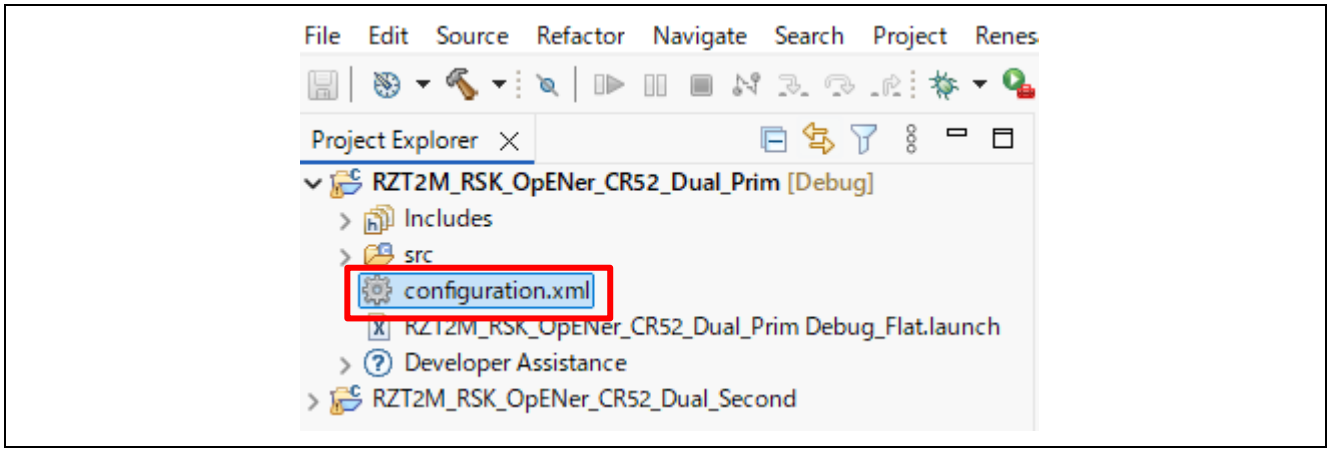


Figure 6.39 Configuration

B) Click “Generate Project Content” button then generate rz, rz_gen, rz_cfg folder.

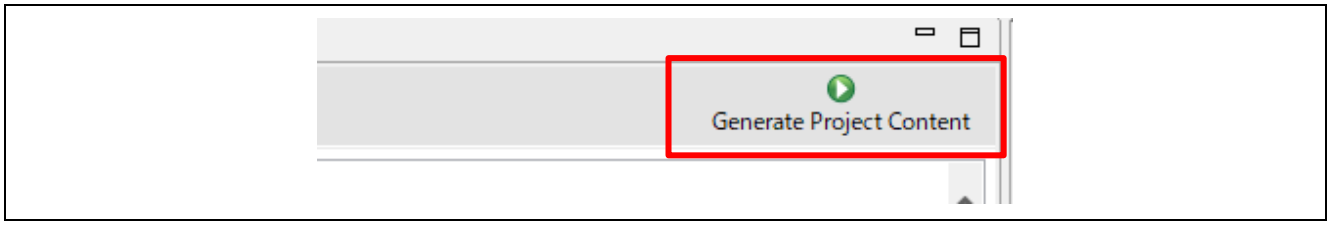


Figure 6.40 Generate Project Content

C) Click the Build button in tool bar to build the project and confirm that there is no error message in the build message log.

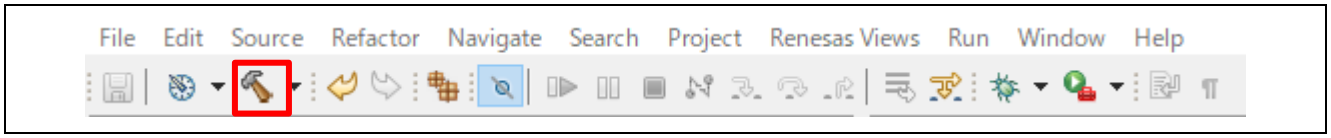


Figure 6.41 Build button

D) For the Secondary project, generate the folders and build the project.

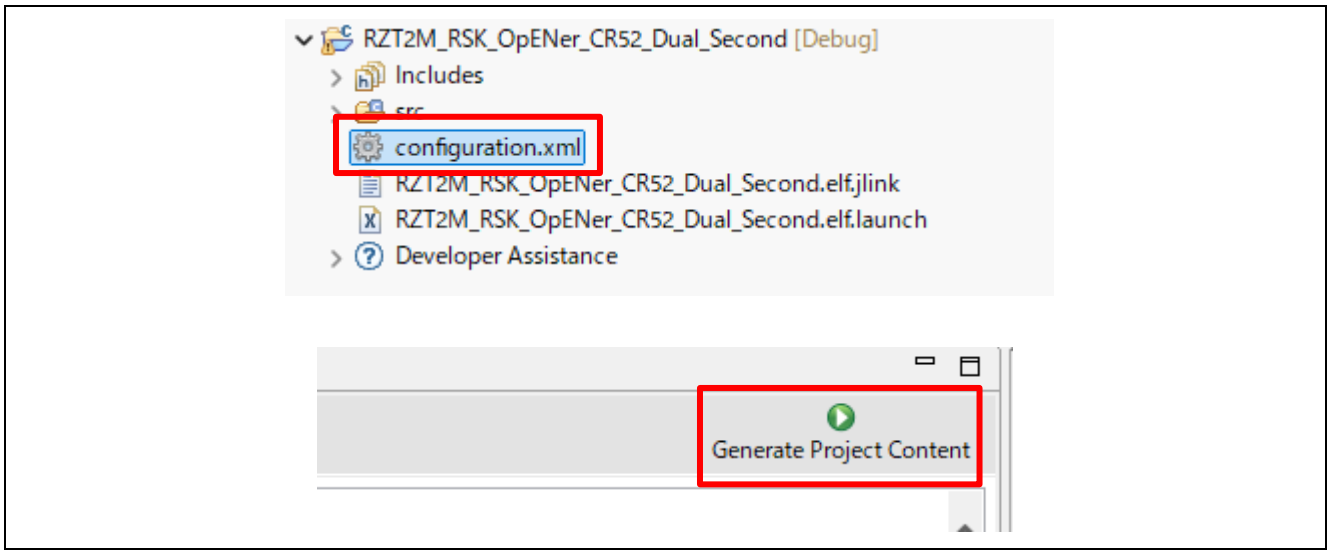


Figure 6.42 Generate folders for the Secondary project

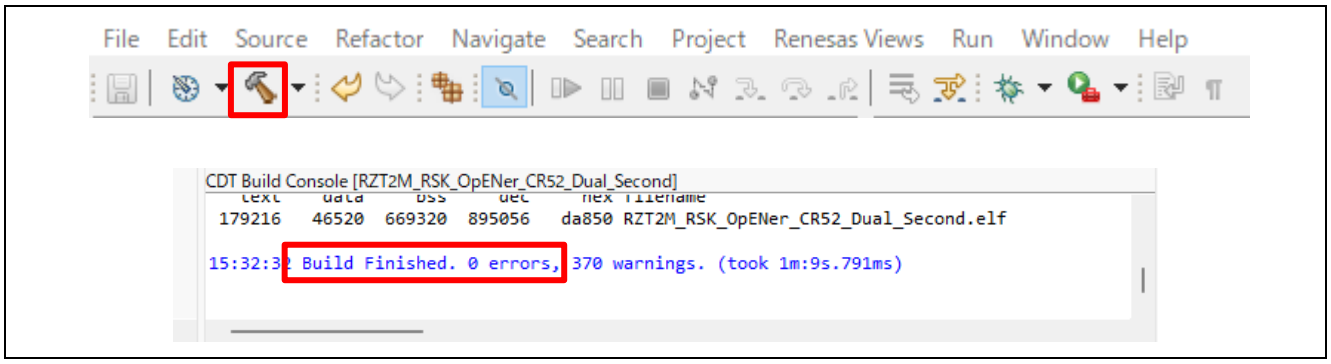


Figure 6.43 Build the Secondary project

(2) Download application and run debugger

A) Select the Primary project and click the Debug button to download the Primary program.

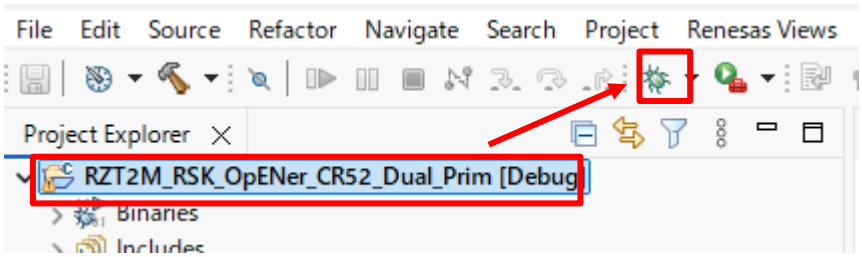


Figure 6.44 Debug button, the Primary project

B) Click the “No” button on the Perspective Switch dialog box.

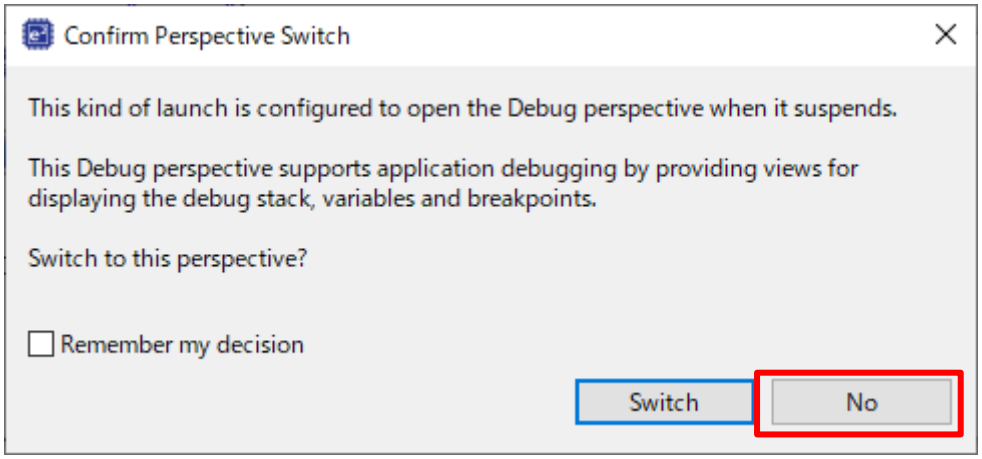


Figure 6.45 Confirm Perspective Switch

C) Select the Secondary project and click the Debug button to download the Secondary program.

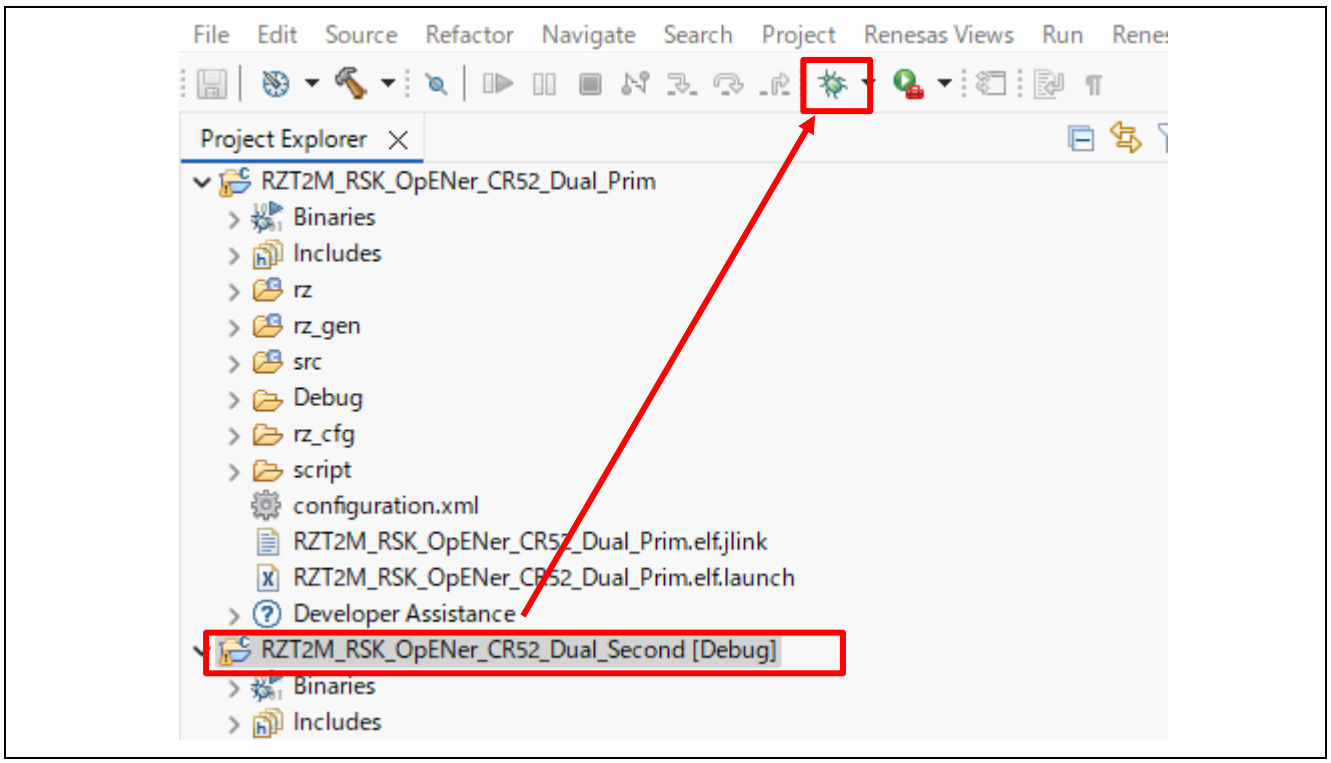


Figure 6.46 Debug button, the Secondary project

D) Click the “No” button on the Launcher dialog box.

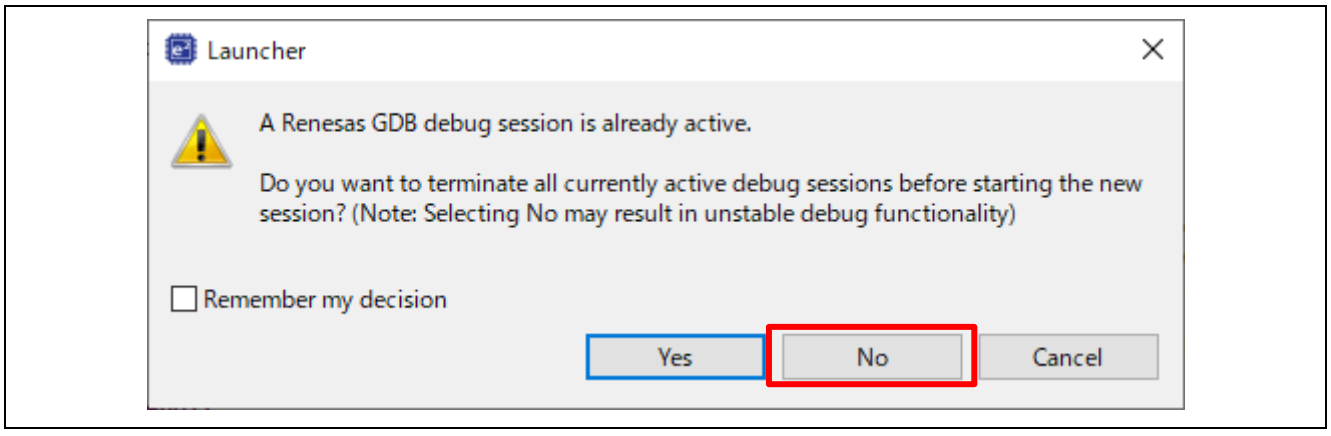


Figure 6.47 Launcher dialog box

E) Click the “Yes” button on the Proceed confirmation dialog box.

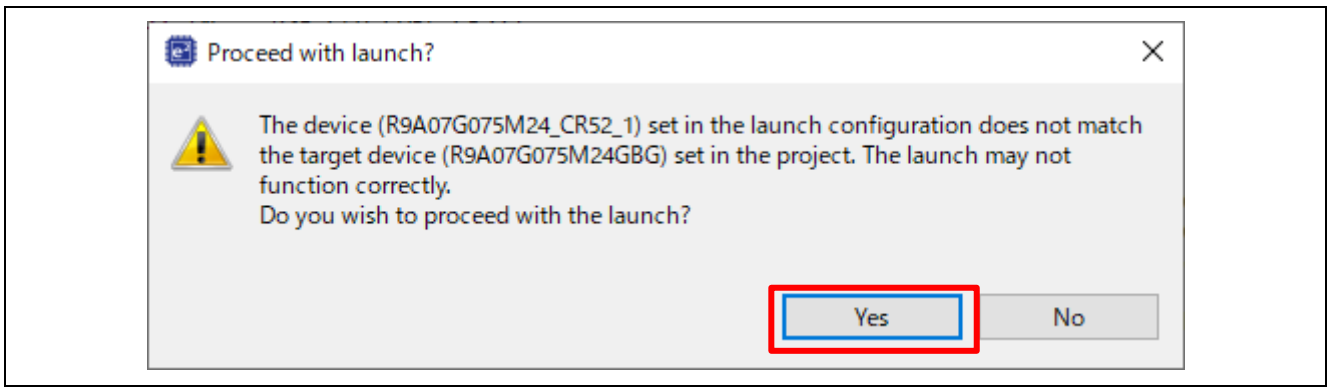


Figure 6.48 Proceed confirmation dialog box

F) Click the “Switch” button on the Perspective Switch dialog box.

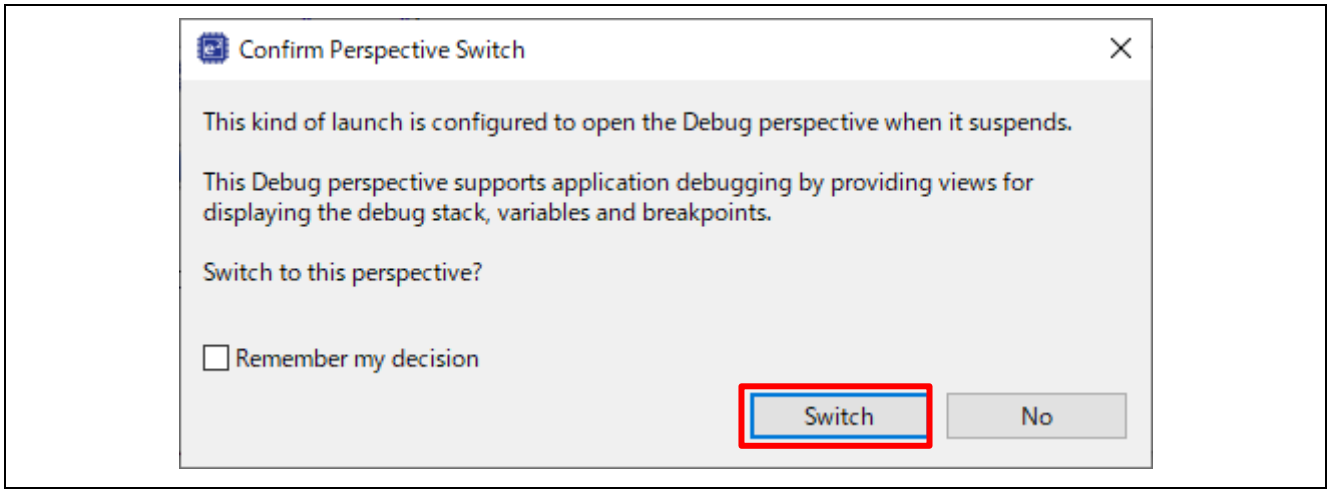


Figure 6.49 Confirm Perspective Switch

G) Run the Primary project.

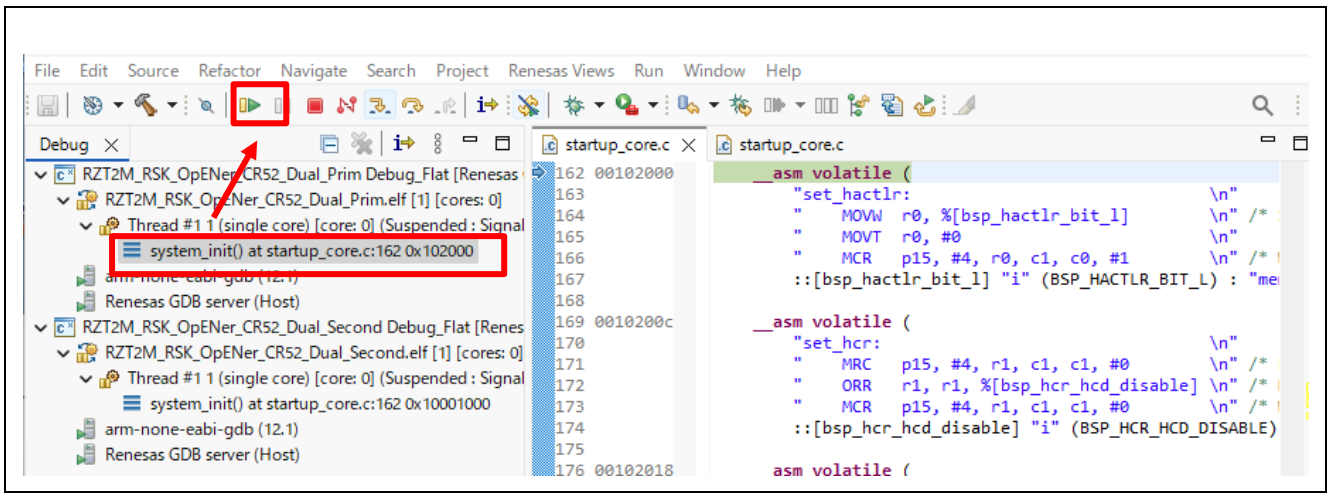


Figure 6.50 Run the Primary project

H) Run the Secondary project.

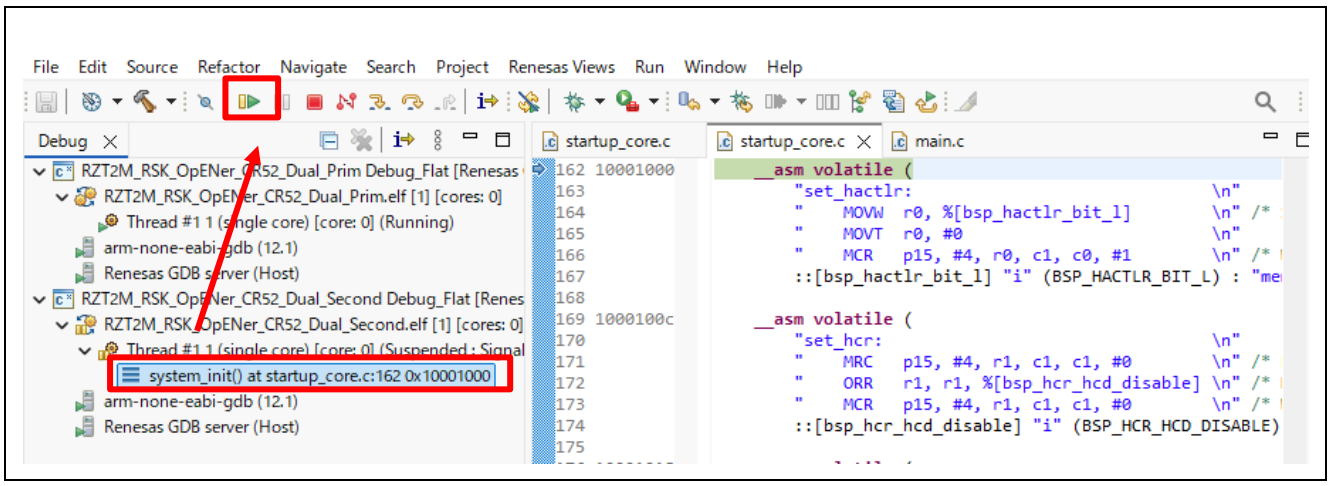


Figure 6.51 Run the Secondary project

7. Demonstration of the application with the CODESYS

The demonstration uses 4 LEDs and 4 Switches. These are named differently on each board. In the explanation in this chapter, they are uniformly referred to as LED 1 to LED 4 and SW 1 to SW 4.

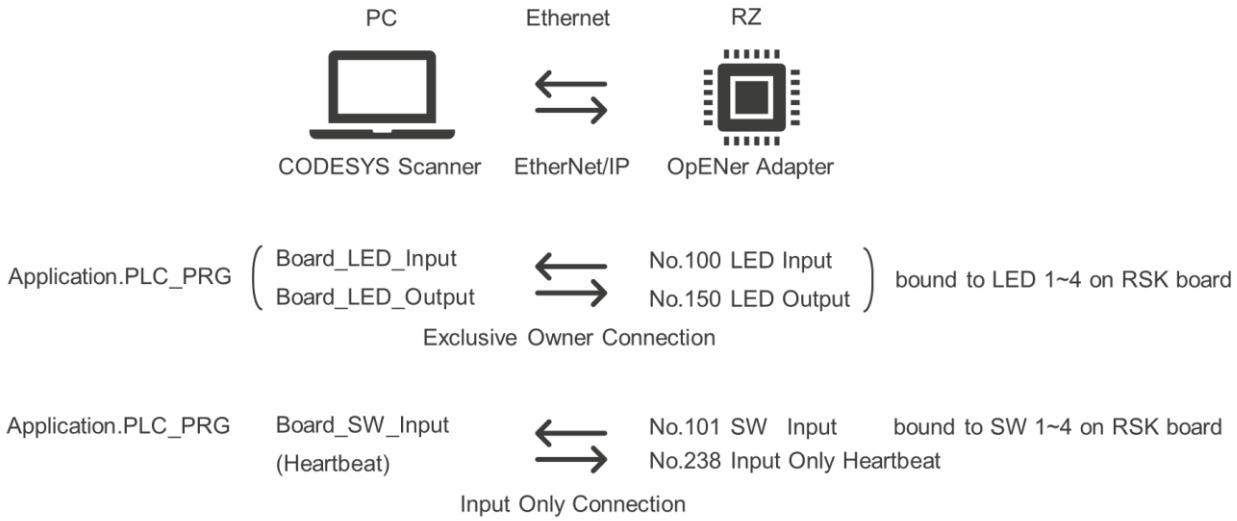
The following table shows the corresponding LEDs and Switches used on each board. Assignments are defined in the opener_port_main.c file.

Table 7.1 LEDs and Switches used for the demonstration on each board

Device Name	Board Name	Used LEDs	Used Switches
RZ/T2M RZ/T2ME	RZ/T2M RSK Board RZ/T2ME RSK Board	LED0 LED1 LED2 LED3	SW3-0 SW3-1 SW3-2 SW3-3
RZ/T2H	RZ/T2H Evaluation Board	LED0 LED1 LED7 LED8	SW12-1 SW12-2 SW12-3 SW12-4
RZ/N2L	RZ/N2L RSK Board	LED0 LED1 LED2 LED3	SW3-0 SW3-1 SW3-2 SW3-3
RZ/N2H	RZ/N2H Evaluation Board	LED5 LED6 LED7 LED8	DSW1-1 DSW1-2 DSW1-3 DSW1-4
Expression in this chapter ->		LED 1 LED 2 LED 3 LED 4	SW 1 SW 2 SW 3 SW 4

7.1 Application Behavior

For demonstration, the application of OpENer RZ/T2 and RZ/N2 port has an Exclusive Connection and an Input Only Connection. The connections between the RZ sample application and the CODESYS project application are shown below.



The PLC program of CODESYS project is shown below. The PLC program is described by ST (Structured Text) language.

```
CycleCounter := CycleCounter+Board_SW_Input;
IF CycleCounter > CyclePerTick THEN
   CycleCounter := CycleCounter-CyclePerTick;
   IF 0 < Board_LED_Input AND Board_LED_Input < 8 THEN
      Board_LED_Output := Board_LED_Input * 2;
   ELSE
      Board_LED_Output := 1;
   END_IF
END_IF
```

- The Exclusive Owner Connection is bound to LED on RSK board.
  - The PLC program makes the LEDs light by shifting every CyclePerTick.
- The Input Only Connection is bound to SW 1 ~ SW 4 on RSK board.
  - The PLC program read the SW values and change the incrementation value of CycleCounter for controlling the frequency of LED light shifting.

The CODESYS application is shown in the figure below.

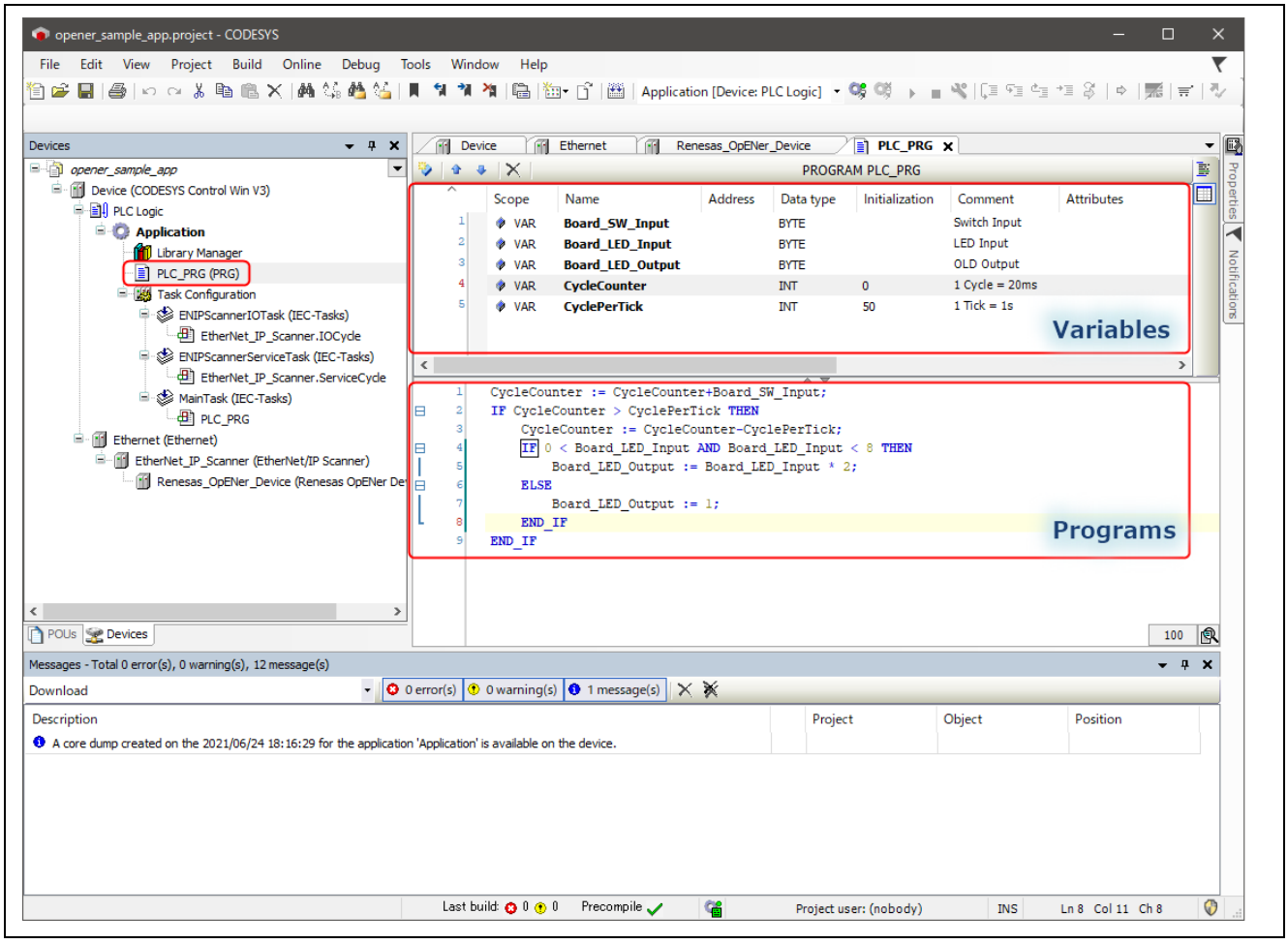


Figure 7.1 CODESYS Project included in this sample program

7.2 IP and MAC Address Configuration

7.2.1 IP Configuration

This program sets its own IP and related parameters to the following values defined statically by “g_lwip_port0_netif_cfg” variable in “common\renesas\application\opener_port_instance.c”.

If you change the IP configuration, please check the following parameters.

Table 7.2 IP Configuration

Item	Value	Variable
IP Address	192.168.1.10	g_lwip_port0_netif_cfg.ip_address
Net Mask	255.255.255.0	g_lwip_port0_netif_cfg.subnet_mask
Gateway Address	192.168.1.2	g_lwip_port0_netif_cfg.gateway_address
Host Name	OPENER_NETIF0	g_lwip_port0_netif_cfg.p_host_name
DHCP	LWIP_PORT_DHCP_DISABLE	g_lwip_port0_netif_cfg.dhcp

If using DHCP, please set “g_lwip_port0_netif_cfg.dhcp” to “LWIP_PORT_DHCP_ENABLE”. After the program starts, DHCP process is executed to get an IP address dynamically. If the program gets an IP address, EtherNet/IP communication starts.

7.2.2 MAC Address Configuration

The MAC address is set to the following values defined statically by FSP Configurator.

Table 7.3 MAC Address Configuration

Item	Value (Hexadecimal)
MAC Address	74:90:50:10:05:9A

If you change the MAC address, please change the value on FSP configuration.

Regarding FSP configuration, please see the "RZ/T2, RZ/N2 Getting Started with Flexible Software Package" (r01an6434ejxxxx-rzt2-rzn2-fsp-getting-started.pdf) document.

7.3 Startup Software PLC

7.3.1 Open CODESYS project

Please select “File” > “Open project...” in CODESYS tool bar and open “opener_sample_app.project” in “scanner/codesys”.

You may be prompted to update the versions of some libraries.

- Do not update the version of three libraries
  - Standard: 3.5.15.0
  - StringUtils: 3.5.15.0
  - SysSocket Implementation: 3.5.15.0
- If other libraries are displayed, please update them to the latest version.

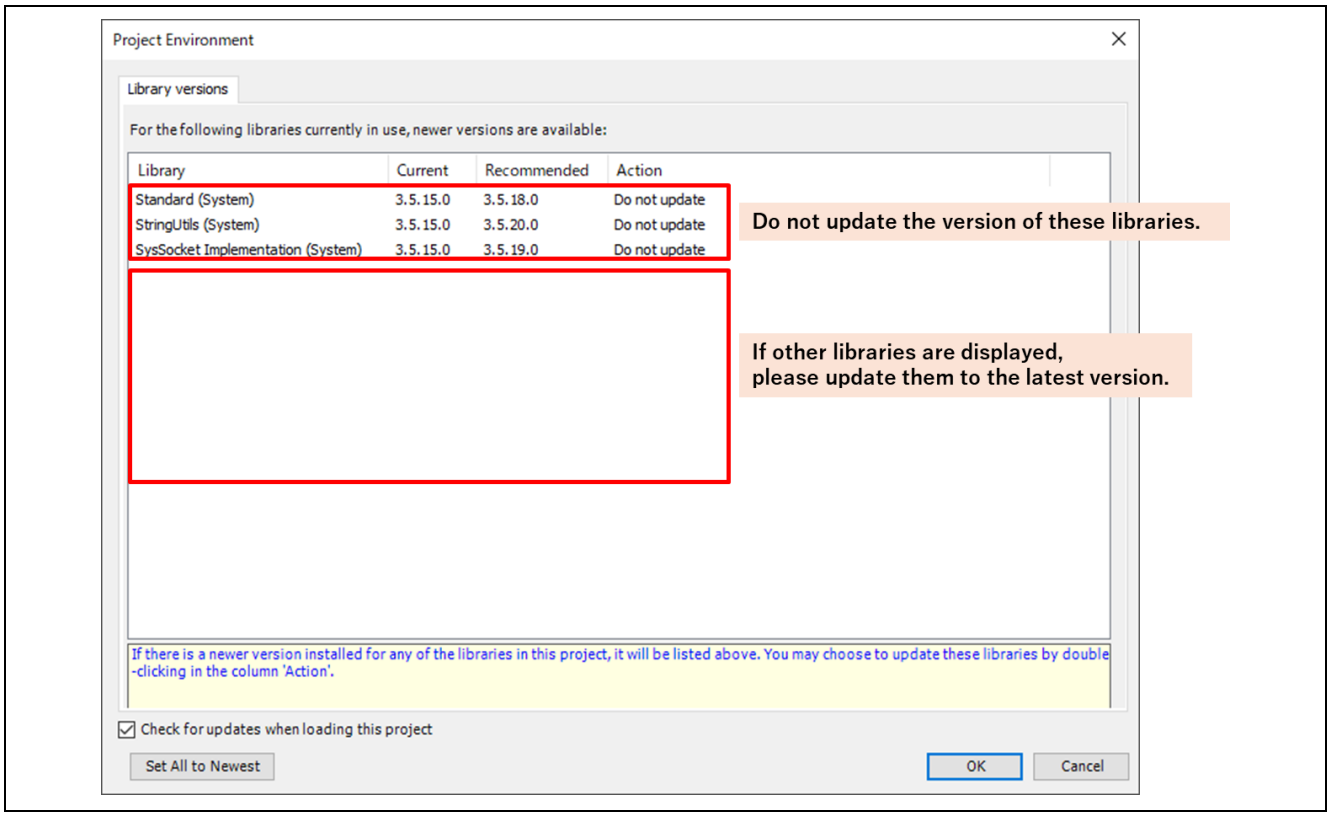


Figure 7.2 Update Codesys libraries

If the project is opened properly, the opened project is shown in “Device” section located at left in the following window.

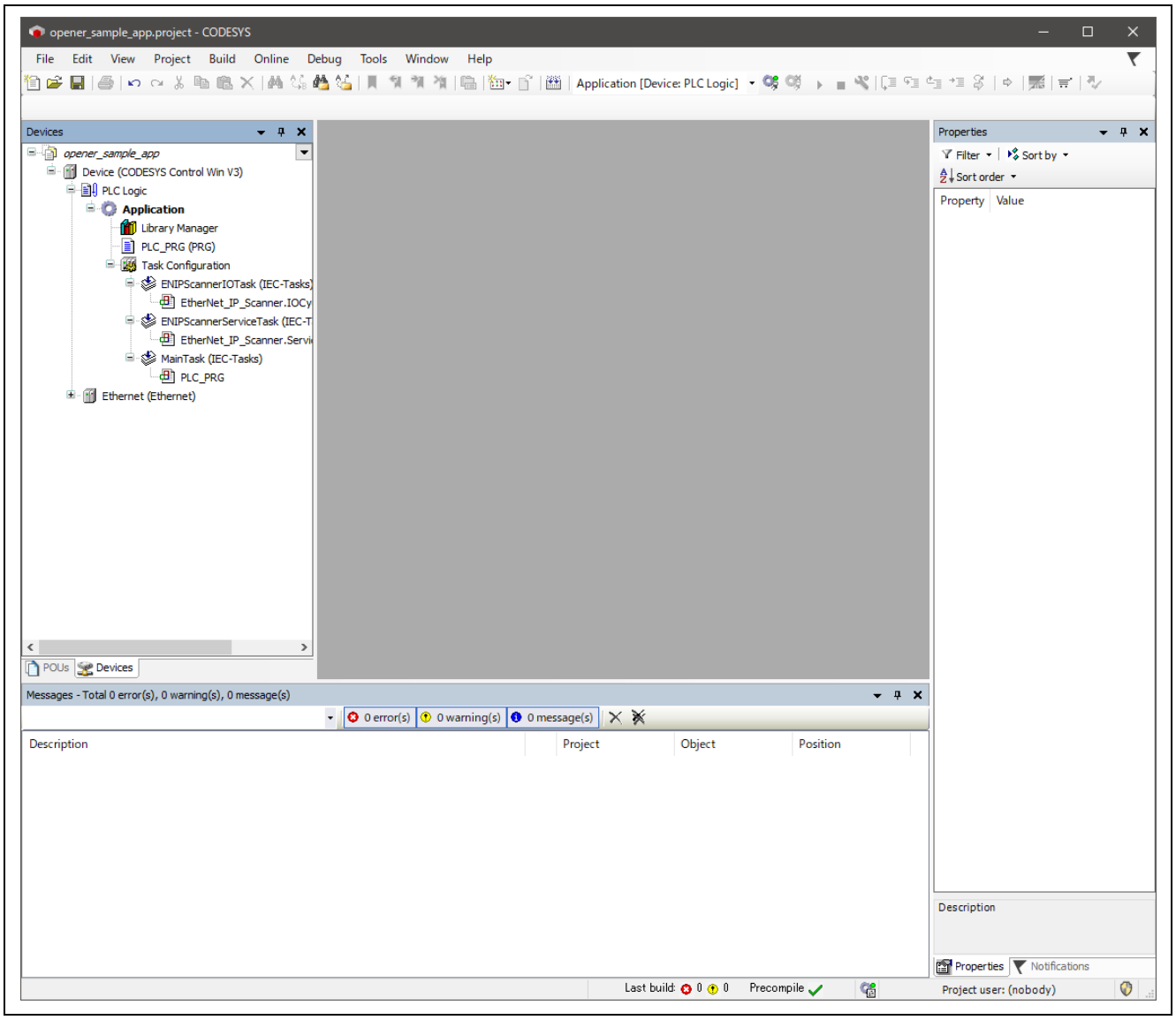


Figure 7.3 Open a CODESYS Project

7.3.2 Network Configuration

Please double-click “Device (CODESYS Control Win V3)” to open “Communication Settings” at center section, and please click “Scan Network...” to open “Select Device” window.

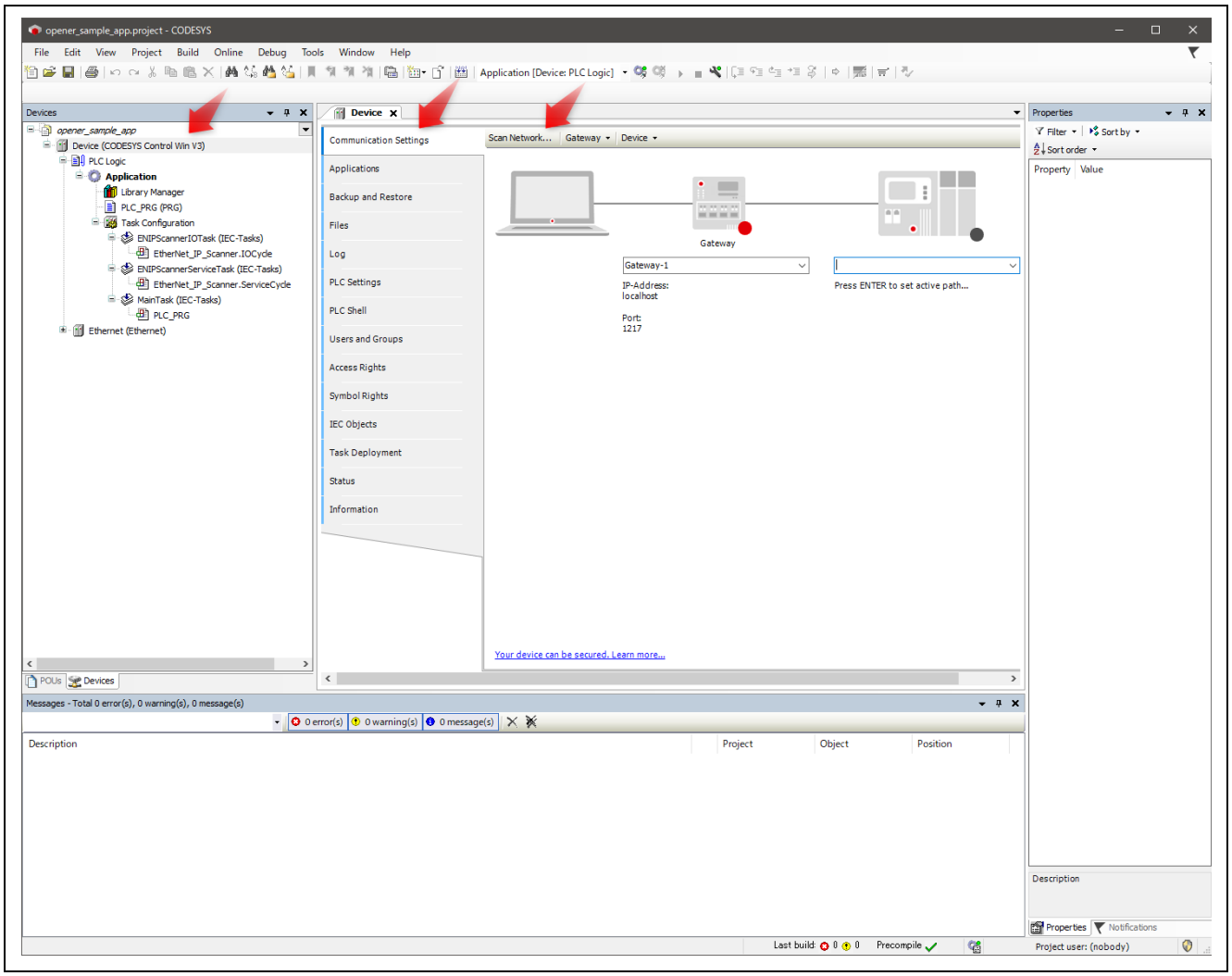


Figure 7.4 Communication Settings

If the software PLC is found after scanning the network, the device name (here, PC name) is shown under Gateway tree. Please double-click this device name (in blue portion).

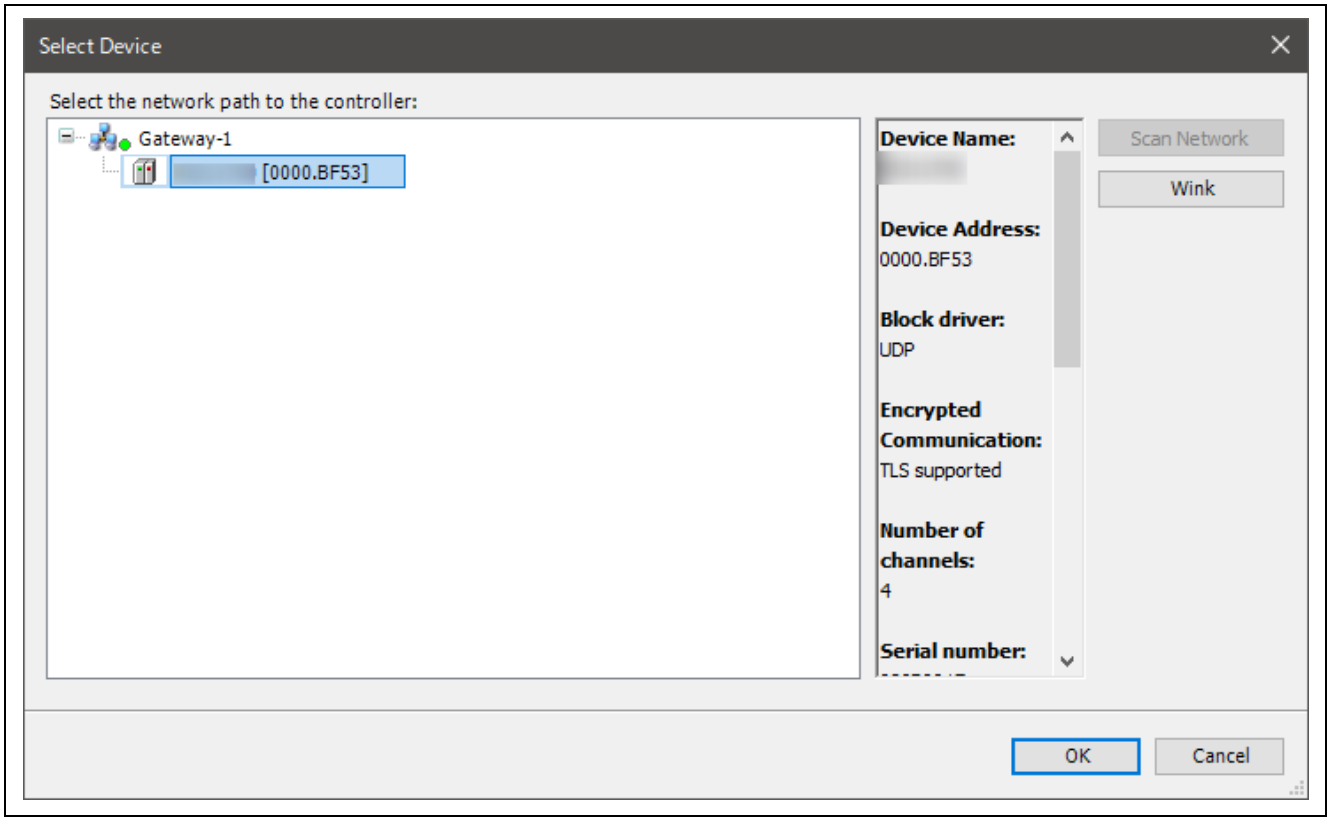


Figure 7.5 Select Device

Click the pull-down button next to the PLC name and select the device name (in this case, the PC name) displayed in Scan Network. If the network is configured properly, its configuration is shown in “Communication Settings” tab, and there are the green marks at gateway and device portions.

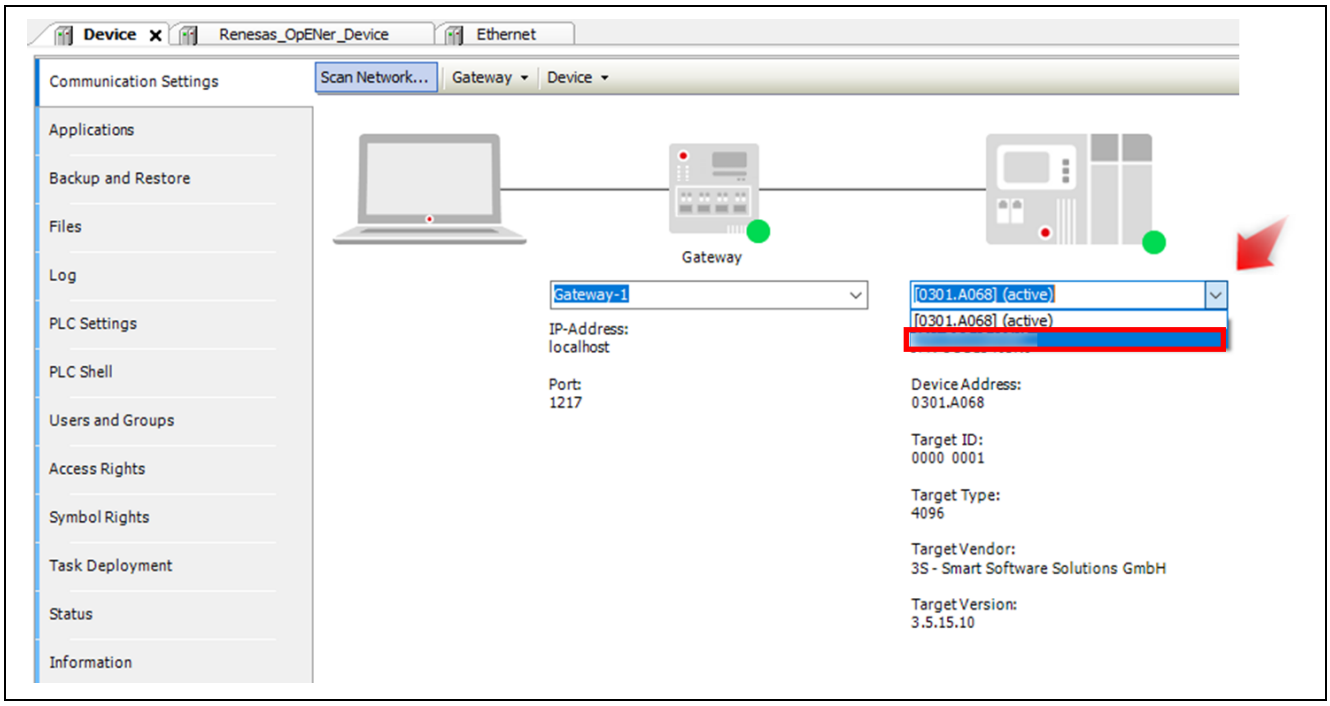


Figure 7.6 Green Marks at Gateway and Device Portions

7.3.3 Interface and IP address configuration

Please click “Ethernet (Ethernet)” in the left section to open “Ethernet” tab in center section.

After that, please select “...” button to select network interface ethernet which relates to RSK board, and please configure the IP address and related address values of the ethernet network interface.

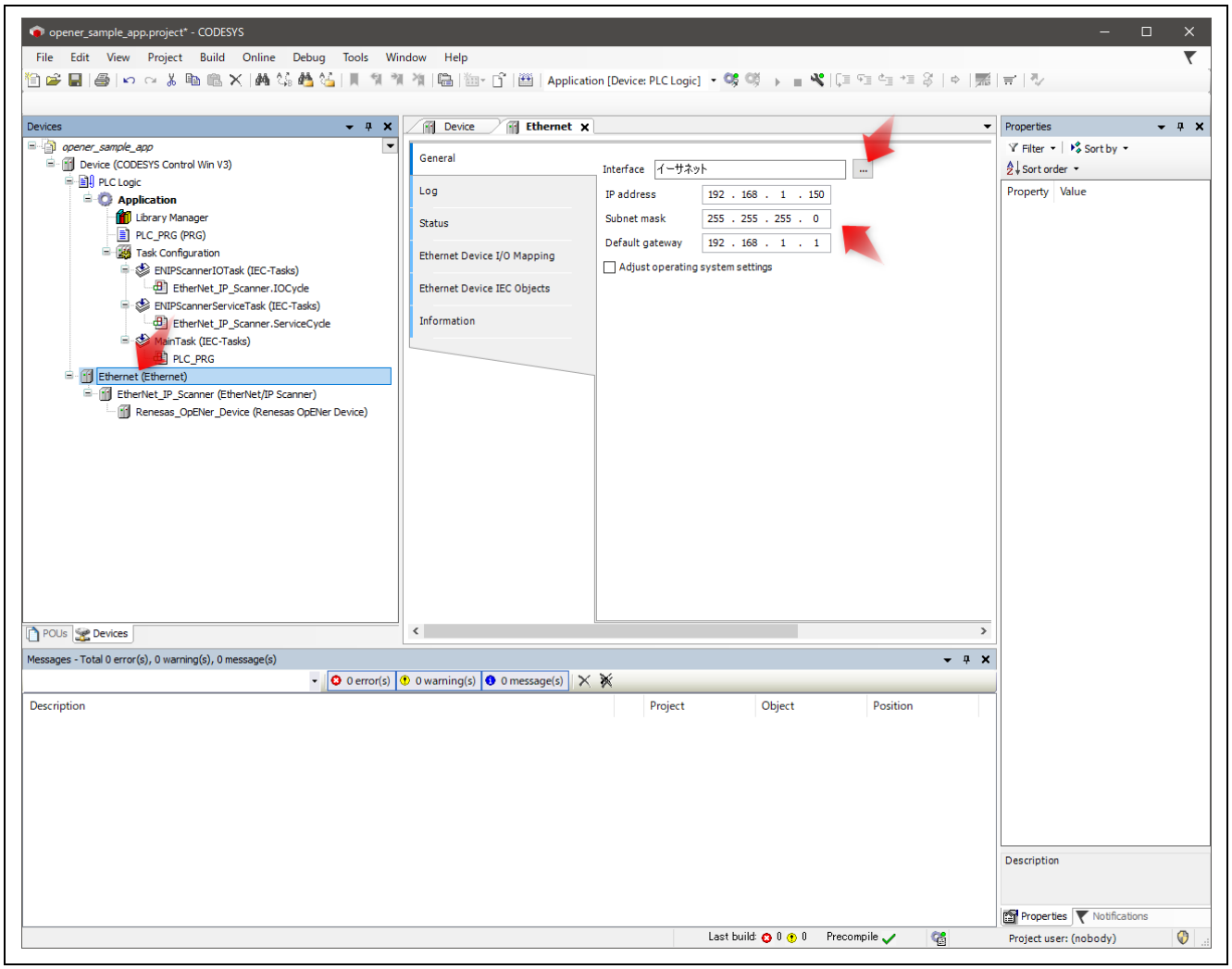


Figure 7.7 Open Ethernet tab

Next, please double-click “Renesas_OpENer_Device (Renesas OpENer Device)” in the left section to open “Renesas_OpENer_Device” tab in center section.

Enter the IP address of the sample code in the IP Address form. (Default setting is 192.168.1.10)

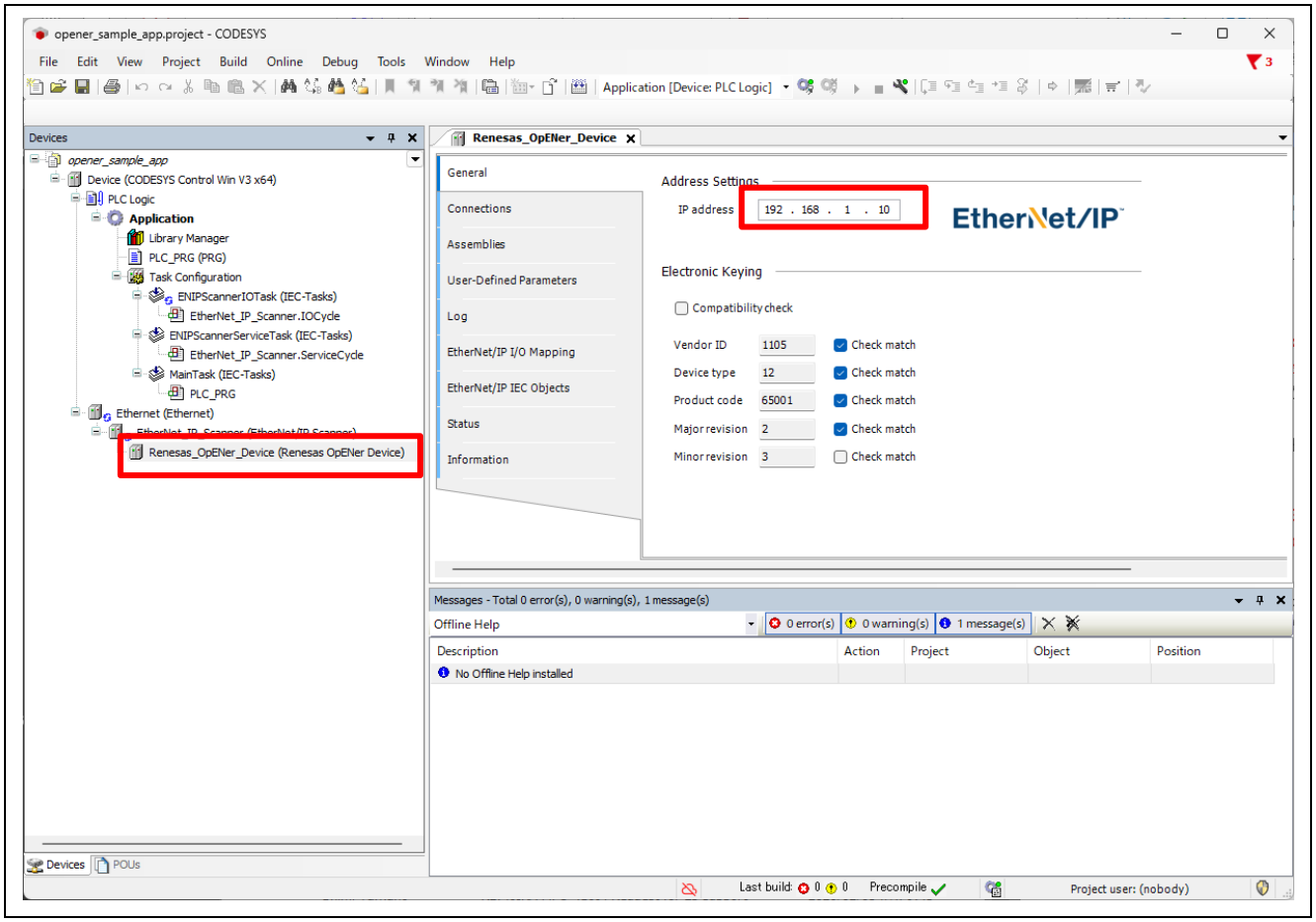


Figure 7.8 Renesas_OpENer_Device (Renesas OpENer Device)

7.4 Operation Check

7.4.1 Build Project and Start Application

Follow the following steps and figure to build the project and start the application.

1. Click “Build” button in the tool bar to build the CODESYS project.
2. Click “Login” button in the tool bar to login to the network.
3. Click “Start” button in the tool bar to run network and application.

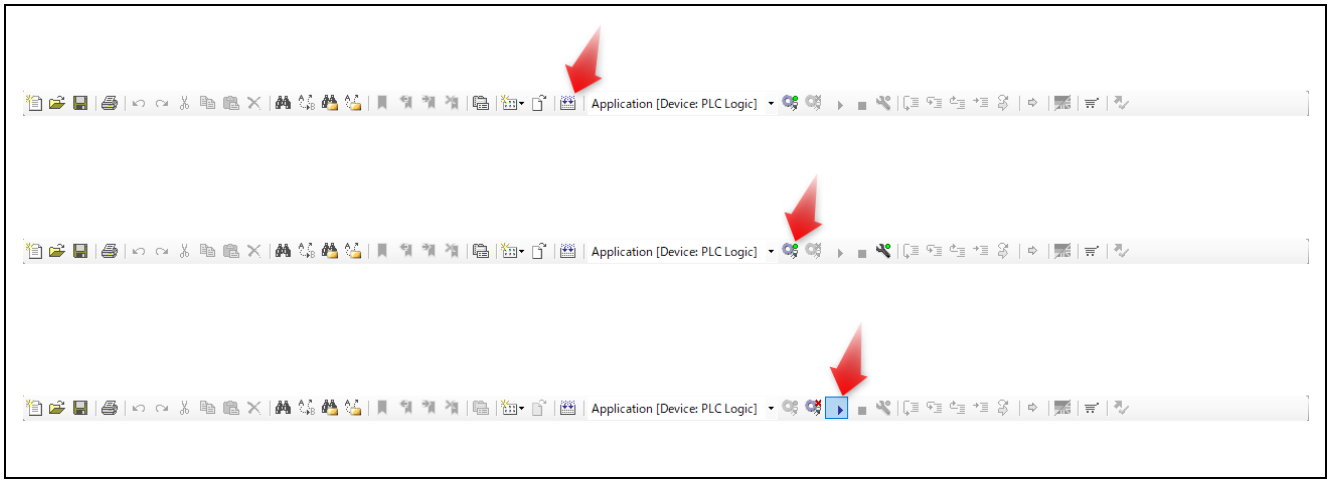


Figure 7.9 Build Project and Start Application

7.4.2 Check Network Connection

If the CODESYS application on PC connects with OpENer application properly, “Device”, “Ethernet”, “EtherNet_IP_Scanner”, and “Renesas_OpENer_Device” in left section are marked with green cycle mark.

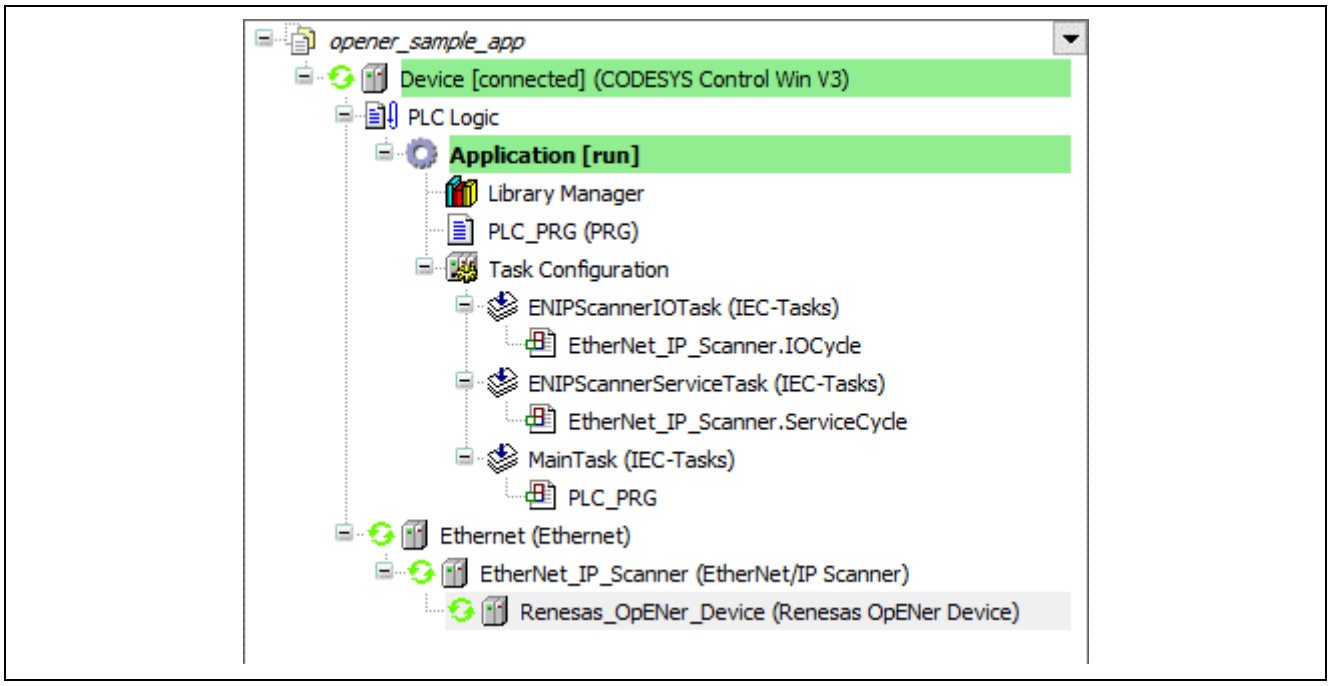


Figure 7.10 Check Network Connection

7.4.3 Check Application Behavior

Please open “EtherNet/IP I/O Mapping” page in “Renesas_OpENer_Device” tab.

This page shows the Exclusive Owner Connection and Input Only Connection mapping which indicates the connections between the CODESYS PLC application and the Assembly objects on OpENer application. The “Current Value” column indicates the LED and SW values.

- The Exclusive Owner Connection is bound to LED on RSK board.
  - The LEDs lights by shifting with [SW Value] Hz.
- The Input Only Connection is bound to SW 1 ~ SW 4 on RSK board.
  - The SW values indicate the frequency of LED light shifting.

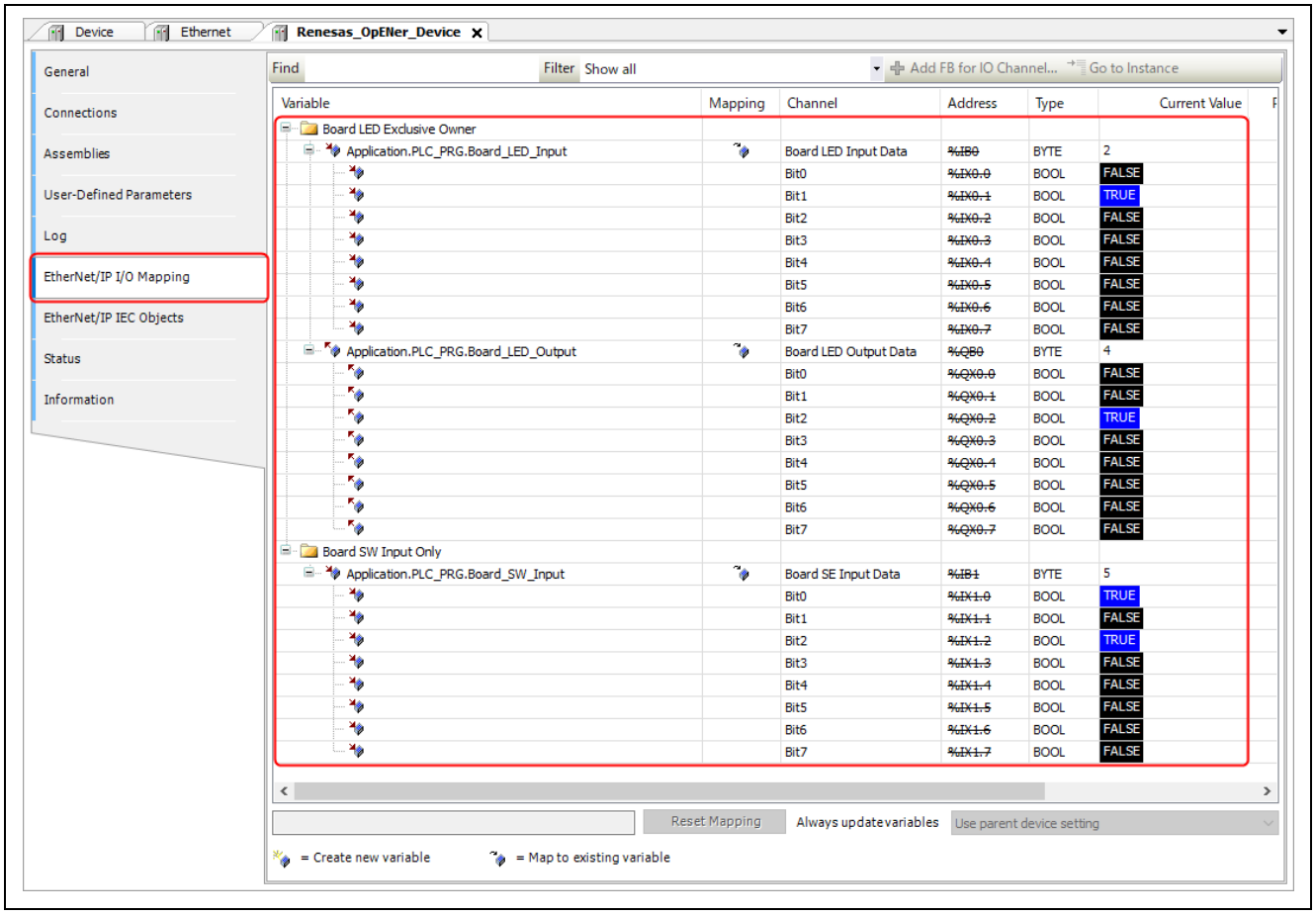


Figure 7.11 Check Application Behavior

8. Software Specification

This chapter shows the specifications of the sample code and other information.

8.1 Stack Diagram

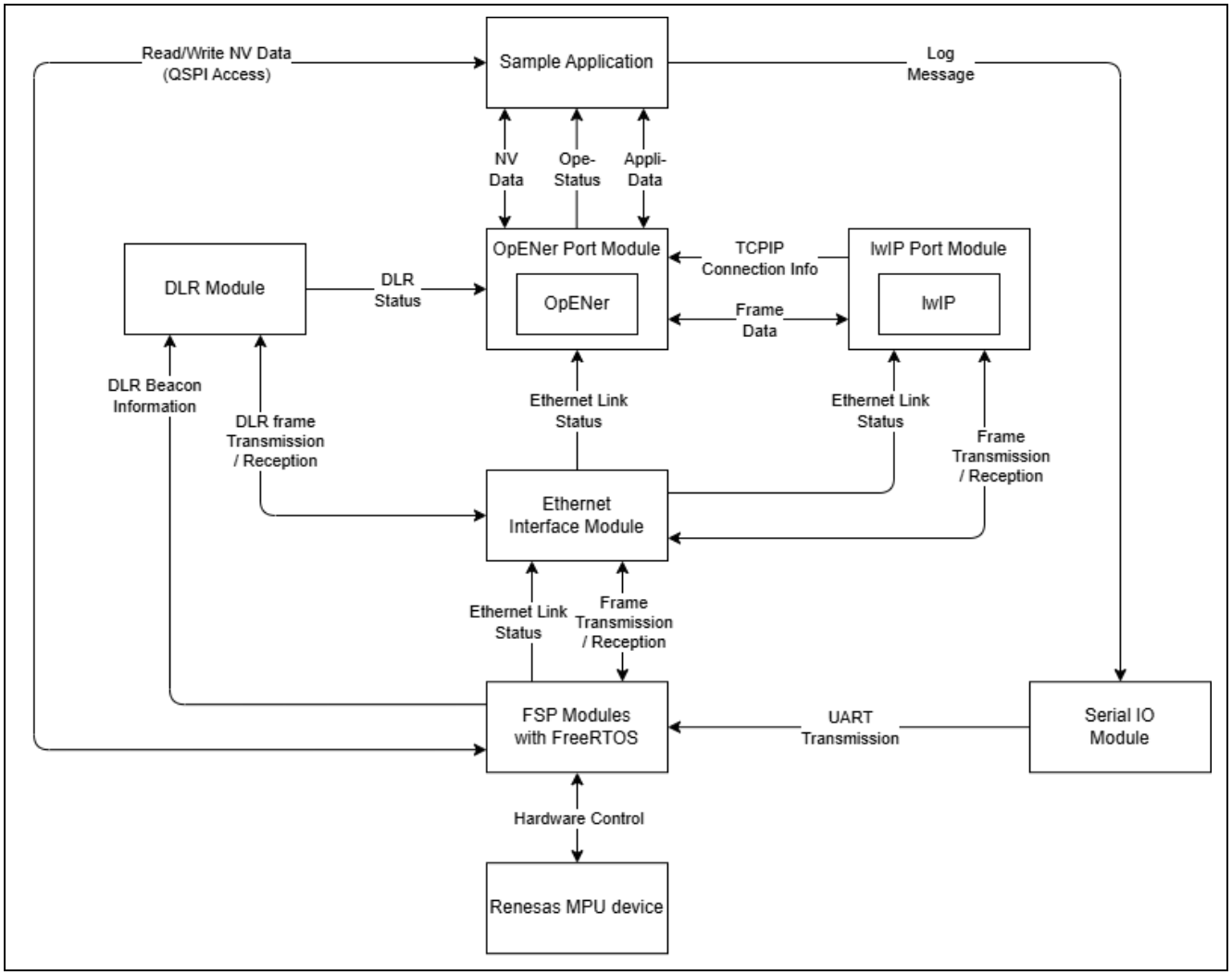


Figure 8.1 Software Stack Diagram

8.2 System Configuration Diagram

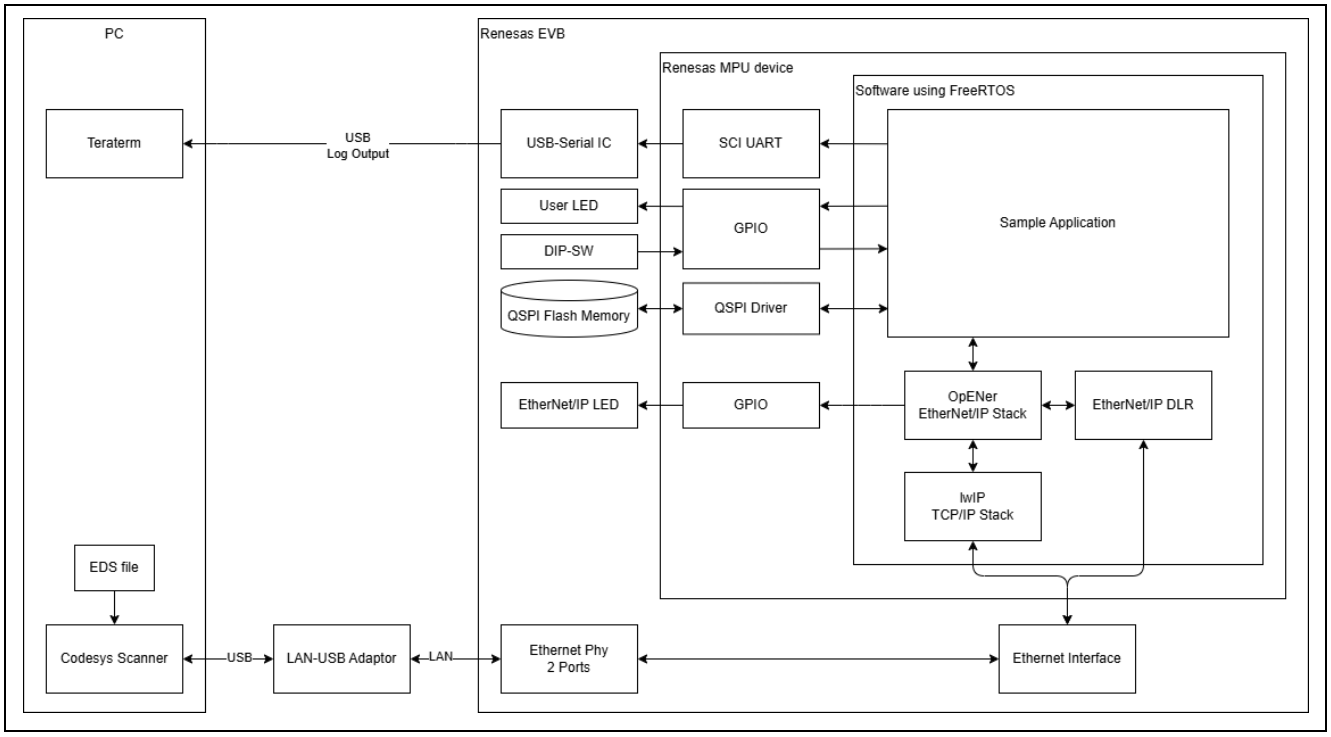


Figure 8.2 System Configuration Diagram

8.3 Support Object Classes, Services and Attributes

Object classes, Services and Attributes supported in this sample code are shown below.

Table 8.1 CIP Object Class on OpENer port

Object Class #	Object Class Name
0x01	Identity
0x02	Message Router
0x04	Assembly
0x06	Connection Manager
0x47	Device Level Ring
0xF5	TCP/IP Interface
0xF6	Ethernet Link
0x48	QoS
0x109	LLDP Management

Table 8.2 0x01: Identity

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
1	Vendor ID	○	-
2	Device Type	○	-
3	Product Code	○	-
4	Revision	○	-
5	Status	○	-
6	Serial Number	○	-
7	Product Name	○	-
Instance Service			
#	**Service Name**		
0x01	Get_Attributes_All		
0x05	Reset		
0x0E	Get_Attribute_Single		

*: The values of instance attributes #1~#4, #6, and #7 are configured by macros in “devicedata.h" in “common\renesas\oss_deps\opener” directory.

Table 8.3 0x02: Message Router

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
No instance attributes are implemented.			
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		

Table 8.4 0x04: Assembly

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
3	Data	○	○
4	Size	○	-
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		
0x10	Set_Attribute_Single		

Table 8.5 0x06: Connection Manager

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
No instance attributes are implemented.			
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		
0x4E	Forward_Close		
0x54	Forward_Open		
0x5a	Get_Connection_Owner		
0x5b	Large_Forward_Open		

Table 8.6 0x47: Device Level Ring

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
1	Network Topology	○	-
2	Network Status	○	-
10	Active Supervisor Address	○	-
12	Capability Flags	○	-
Instance Service			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		

Table 8.7 0x48: QoS

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
1	802.1Q Tag Enable	○	-
2	DSCP PTP Event	○	-
3	DSCP PTP General	○	-
4	DSCP Urgent	○	○
5	DSCP Scheduled	○	○
6	DSCP High	○	○
7	DSCP Low	○	○
8	DSCP Explicit	○	○
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		
0x10	Set_Attribute_Single		

Table 8.8 0xF5: TCP/IP Interface

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
1	Status	○	-
2	Configuration Capacity	○	-
3	Configuration Control	○	○
4	Physical Link Object	○	-
5	Interface Configuration	○	-
6	Host name	○	-
8	TTL Value	○	-
9	Mcast Config	○	-
10	Select ACD	○	○
11	LastConflictDetected	○	○
13	Encapsulation Inactivity Timeout	○	○
Instance Service			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
0x10	Set_Attribute_Single		

Table 8.9 0xF6: Ethernet Link

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
1	Interface Speed	○	-
2	Interface Flag	○	-
3	Physical Address	○	-
7	Interface Type	○	-
11	Interface Capability	○	-
Instance Service			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
0x4C	Get_and_Clear		

Table 8.10 0x109 LLDP Management

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
1	LLDP Enable	○	○
2	msgTxInterval	○	○
3	msgTxHold	○	○
4	LLDP Database	○	-
5	Last Change	○	-
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		
0x10	Set_Attribute_Single		

8.4 Device Data Configuration

Configurations that are often set for individual devices are listed in this section.

Table 8.11 Device data configuration List

Directory and file	Line	Item	Macro definition	Initial setting
common\renesas\application\opener_port_instance.c	L71	IP address	SAMPLE_STATIC_IP_ADDRESS	192.168.1.10
	L72	Subnet mask	SAMPLE_STATIC_SUBNET_MASK	255,255,255,0
	L73	Gateway address	SAMPLE_STATIC_GATEWAY_ADDRESS	192,168,1,2
	L74	Host name	SAMPLE_HOSTNAME	"OPENER_NETIF0"
	L54	Device serial number	SAMPLE_CIP_DEVICE_SERIAL_NUMBER	0x12345678
common\renesas\oss_deps\opener\devicedata.h	L15	Device vendor ID	OPENER_DEVICE_VENDOR_ID	1105
	L16	Device type	OPENER_DEVICE_TYPE	12
	L17	Device name	OPENER_DEVICE_NAME	"Renesas OpENer Device"
	L20	Product code	OPENER_DEVICE_PRODUCT_CODE	65001
	L21	Device major revision	OPENER_DEVICE_MAJOR_REVISION	2
	L22	Device minor revision	OPENER_DEVICE_MINOR_REVISION	3
common\renesas\oss_deps\lwip\lwipopts.h	L103	MIB2 system description	SNMP_LWIP_MIB2_SYSDESC	"Renesas Electronics Corporation EtherNet/IP Sample"
	L104	MIB2 system contact	SNMP_LWIP_MIB2_SYSCONTACT	"sysContact not set"
	L105	MIB2 system name	SNMP_LWIP_MIB2_SYSNAME	"sysName not set"
	L106	MIB2 system location	SNMP_LWIP_MIB2_SYSLOCATION	"sysLocation not set"

9. Appendix

Appendix A: OSS implemented in the sample code

The following three OSS are used in the sample code.

OpENer

The "OpENer" is an open-source software for I/O communication adapters of EtherNet/IP. Please refer to the following link for the details.

https://github.com/EIPStackGroup/OpENer

- Git commit ID used in this sample software: 05cdd03

This package is the RZ/T2 and RZ/N2 port of OpENer and includes OpENer source codes. Regarding the open-source license of OpENer, please see the following file.

```
common\oss\OpENer\license.txt
```

FreeRTOS

FreeRTOS is an open-source software for real-time operating system (RTOS) for microcontrollers. Please refer to the following link for the details.

https://aws.amazon.com/freertos/

https://github.com/aws/amazon-freertos

- Git commit ID used in this sample software: a038063

The RZ/T2 and RZ/N2 port of OpENer includes FreeRTOS source codes as RTOS kernel. Regarding the open-source license of OpENer, please see the following file.

```
common\oss\amazon-freertos\LICENSE
```

lwIP

The lwIP is an open-source software for a small independent implementation of the TCP/IP protocol suite. Please refer to the following link for the details.

https://savannah.nongnu.org/projects/lwip/

https://github.com/lwip-tcpip/lwip

- Git commit ID used in this sample software: 79cd89f

The RZ/T2 and RZ/N2 port of OpENer includes lwIP source codes as a TCP/IP stack. Regarding the open-source license of lwIP, please see the following file.

```
common\oss\lwip\COPYING
```

Appendix B: Assembly Objects and I/O Connections

This sample application has 7 instances of Assembly Object. 5 of these instances are bound to the following array variables.

Table 9.1 Assembly Objects and I/O Connections

Instance No.	Type	Description	Bound variables
100 (0x64)	Static Input	Input of Exclusive Owner I/O Connection bound to LED	`g_assembly_data_led_input[1]`
101 (0x65)	Static Input	Input of Input Only I/O Connection bound to SW	`g_assembly_data_sw_input[1]`
150 (0x96)	Static Output	Output of Exclusive Owner I/O Connection bound to LED	`g_assembly_data_led_output[1]`
151 (0x97)	Static Configuration	Configuration of I/O Connections	`g_assembly_data_config[4]`
154 (0x9A)	Static I/O	Accessing by explicit message connection only	`g_assembly_data_explicit[4]`
238 (0xEE)	Static Output	Heartbeat output of Input only I/O Connection bound to SW	-
237 (0xED)	Static Output	Heartbeat output of listen only I/O Connection.	-

Appendix C: CIP Safety feature

Overview

This section describes how to enable the CIP Safety feature, as well as the software structure and behavior when it is enabled.

Note: The enabled CIP Safety feature has only been tested for basic operation in the connection configurations described later. A self-test based on the official ODVA guidelines has not been conducted.

How to enable CIP Safety feature

To enable the CIP Safety feature, please do the following.

- (1): Add Define “OPENER_CIP_SAFETY”

- (2): Change the ISR function name for safety UART Receive data full callback event to “user_uart_rx_callback”

For detail

(1) Add Define “OPENER_CIP_SAFETY”

Please remove “DISABLE_OPENER_CIP_SAFETY” and add “OPENER_CIP_SAFETY” in the project's Defined Symbols.

Place of the project's Defined Symbols

- e^2 studio case
  - Project Properties -> C/C++ Build -> Settings -> Tool Settings -> Cross ARM C Compiler -> Preprocessor -> Defined Symbols
- EWARM case
  - Project’s Options -> C/C++ Compiler -> Preprocessor -> Defined symbols

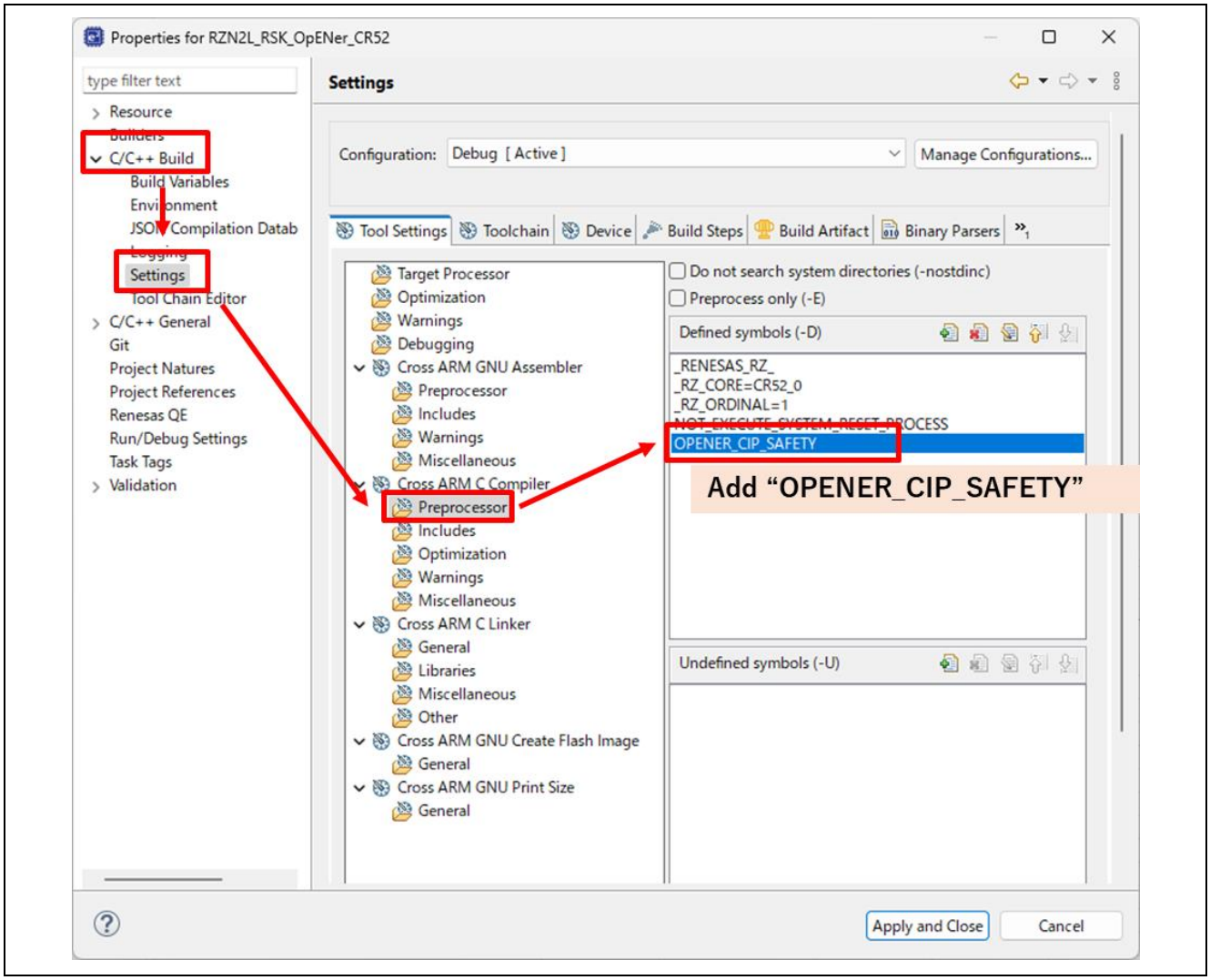


Figure 9.1 How to add define symbol for CIP Safety feature, e² studio case

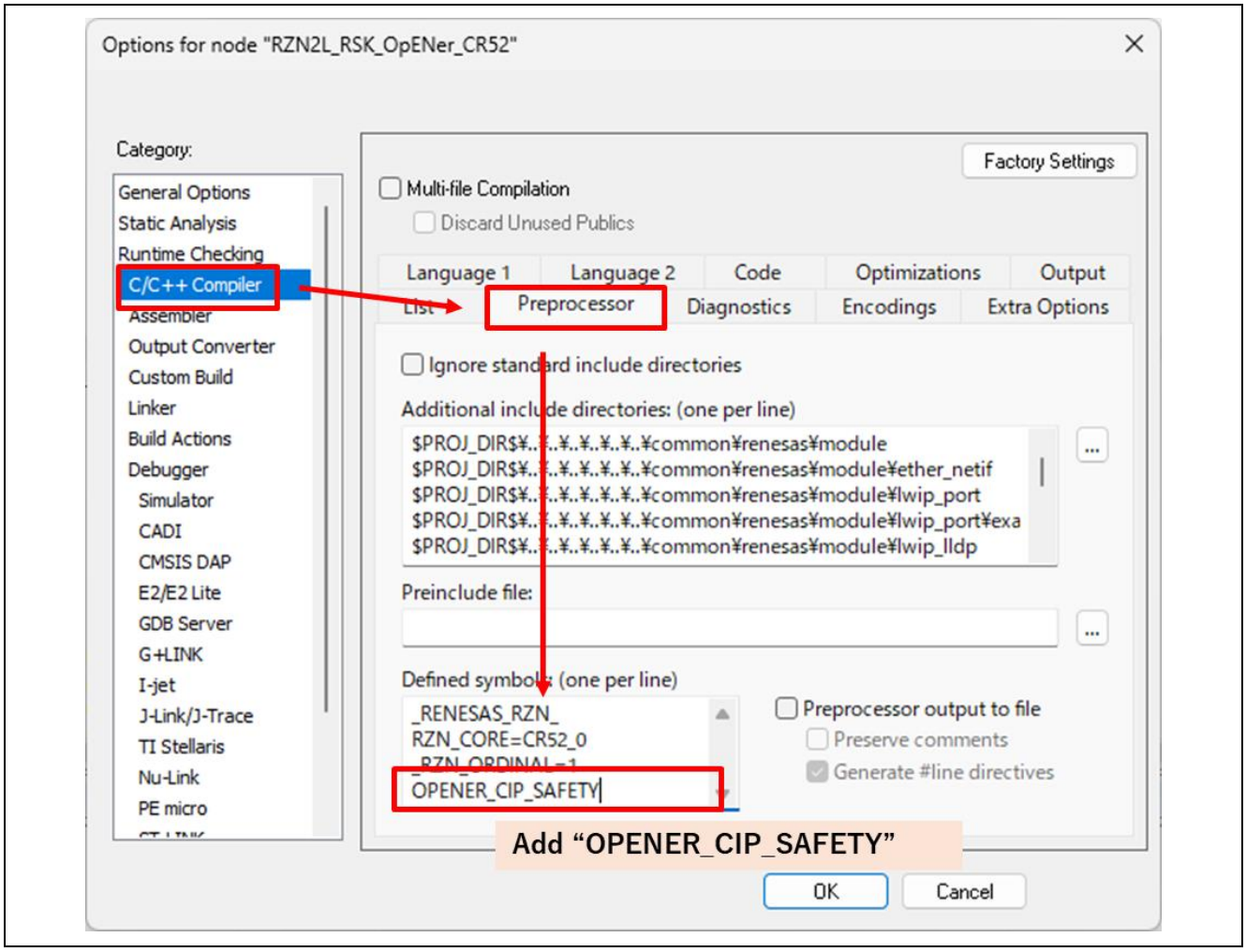


Figure 9.2 How to add define symbol for CIP Safety feature, EWARM case

(2): Change the ISR function name for safety UART Receive data full callback event to “user_uart_rx_callback”

Check the sci channel value of safety uart on the FSP Stacks window. And change the ISR function name for safety UART Receive data full callback event from “sci_uart_rxi_isr” to “user_uart_rx_callback”.

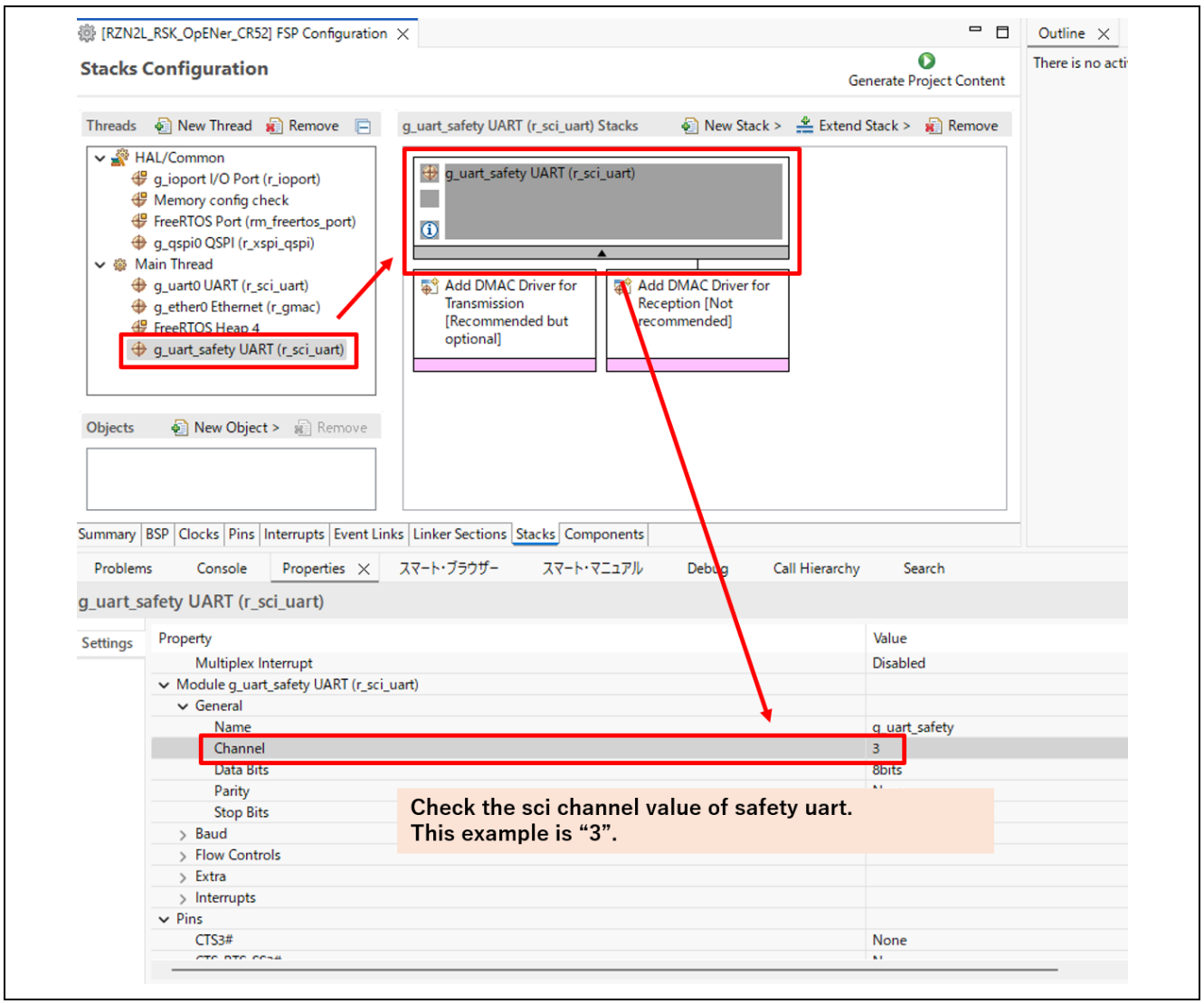


Figure 9.3 How to change ISR function name (1/2)

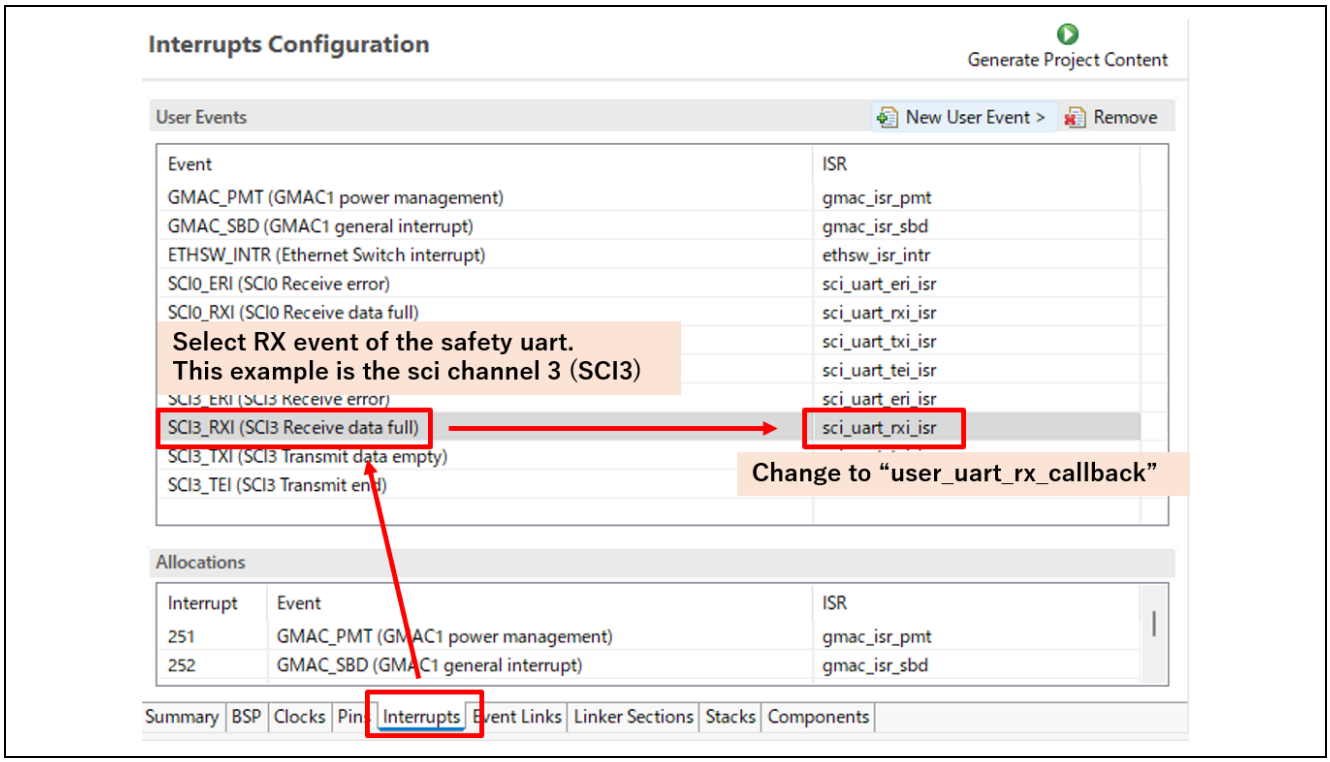


Figure 9.4 How to change ISR function name (2/2)

Hardware Setup and Start

Please follow the steps below to complete the setup.

The evaluation board with this sample code downloaded functions as a black channel for functional safety with respect to the RZ/T2L Functional Safety Reference Board on which the CIP Safety Reference Software is downloaded.

1. Please prepare the contents of Table 9.2.
   - For Renesas product and software, please refer to the Renesas website or contact a distributor.
   - For products from other manufacturers, please contact the respective manufacturer's support.

Table 9.2 Required devices and software

#	Category	Product name/ Model name	Manufacturer
1	Safety PLC (Originator)	Compact GuardLogix SIL3 0.6.0,3M Motion : 5069-L306ERMS3	Rockwell Automation
2	RZ/T2L Functional Safety Reference Board	RTK0EF0179D01001BJ	Renesas Electronics
3	RZ/T2L CIP Safety Reference Software	RTK0EF0200F01001SJ_CI	Renesas Electronics

2. Download the RZ/T2L CIP Safety Reference Software to the RZ/T2L Functional Safety Reference Board.
   - Refer to the software documentation for the download procedure.

3. Download the software to the evaluation board.
   - Enable CIP Safety feature by following the former section. And download this sample program by following the instructions in Chapter 6.

4. Connect the devices as shown in Figure 9.5 and Table 9.3.
   - About UART communication cables:
     Please select cables suitable for communication at relatively high baud rates (e.g., by short cables of around 10cm).

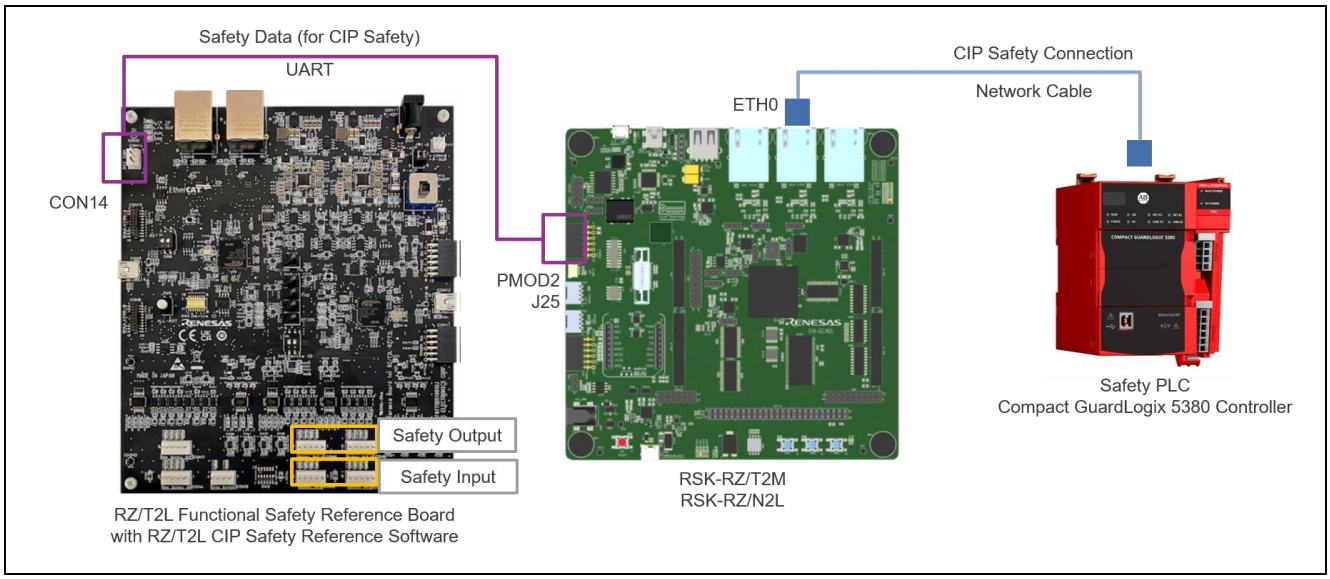


Figure 9.5 Topology when CIP Safety feature is enabled (RZ/T2M, RZ/N2L)

Table 9.3 Connection between RSK-RZ/T2M or RSK-RZ/N2L and FuSa Board

RSK-RZ/N2L		RZ/T2L Functional Safety Reference Board	
J25 PMOD2		CON14	
#	Connector Pin Name	#	Connector Pin Name
3	SCI_RXD	3	CON_TXD_A
2	SCI_TXD	2	CON_RXD_A
5	GROUND	1	DGND

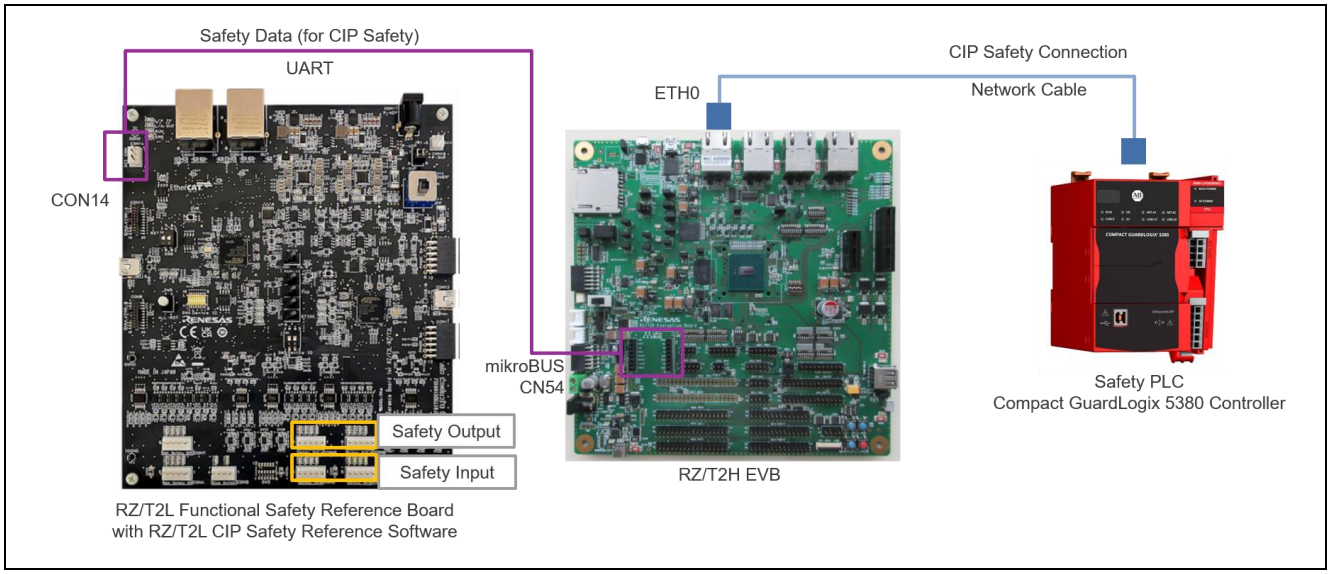


Figure 9.6 Topology when CIP Safety feature is enabled (RZ/T2H)

Table 9.4 Connection between RZ/T2H EVB and FuSa Board

RZ/T2H EVB		RZ/T2L Functional Safety Reference Board	
CN54 mikroBUS		CON14	
#	Connector Pin Name	#	Connector Pin Name
3	mikroBUS RX	3	CON_TXD_A
4	mikroBUS TX	2	CON_RXD_A
8	GROUND	1	DGND

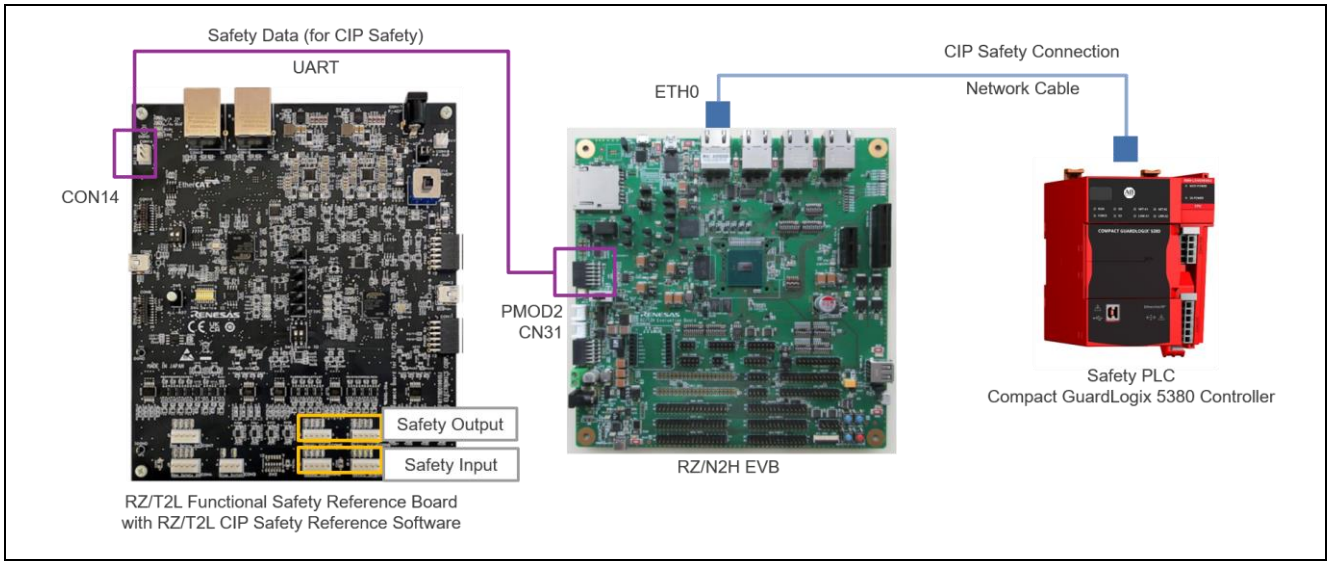


Figure 9.7 Topology when CIP Safety feature is enabled (RZ/N2H)

Table 9.5 Connection between RZ/N2H EVB and FuSa Board

RZ/N2H EVB		RZ/T2L Functional Safety Reference Board	
CN31 PMOD2		CON14	
#	Connector Pin Name	#	Connector Pin Name
3	PMOD2 RXD2	3	CON_TXD_A
2	PMOD2 TXD2	2	CON_RXD_A
5	GROUND	1	DGND

5. Power on all devices.

The evaluation board will begin EtherNet/IP operation upon receiving a wakeup request from the Functional Safety Reference Board. For the overall operation as a functional safety system, please refer to the documentation of the CIP Safety Reference Software.

Operation

The evaluation board with this sample code downloaded functions as a black channel for functional safety with respect to the RZ/T2L Functional Safety Reference Board on which the CIP Safety Reference Software is downloaded.

Stack Diagram

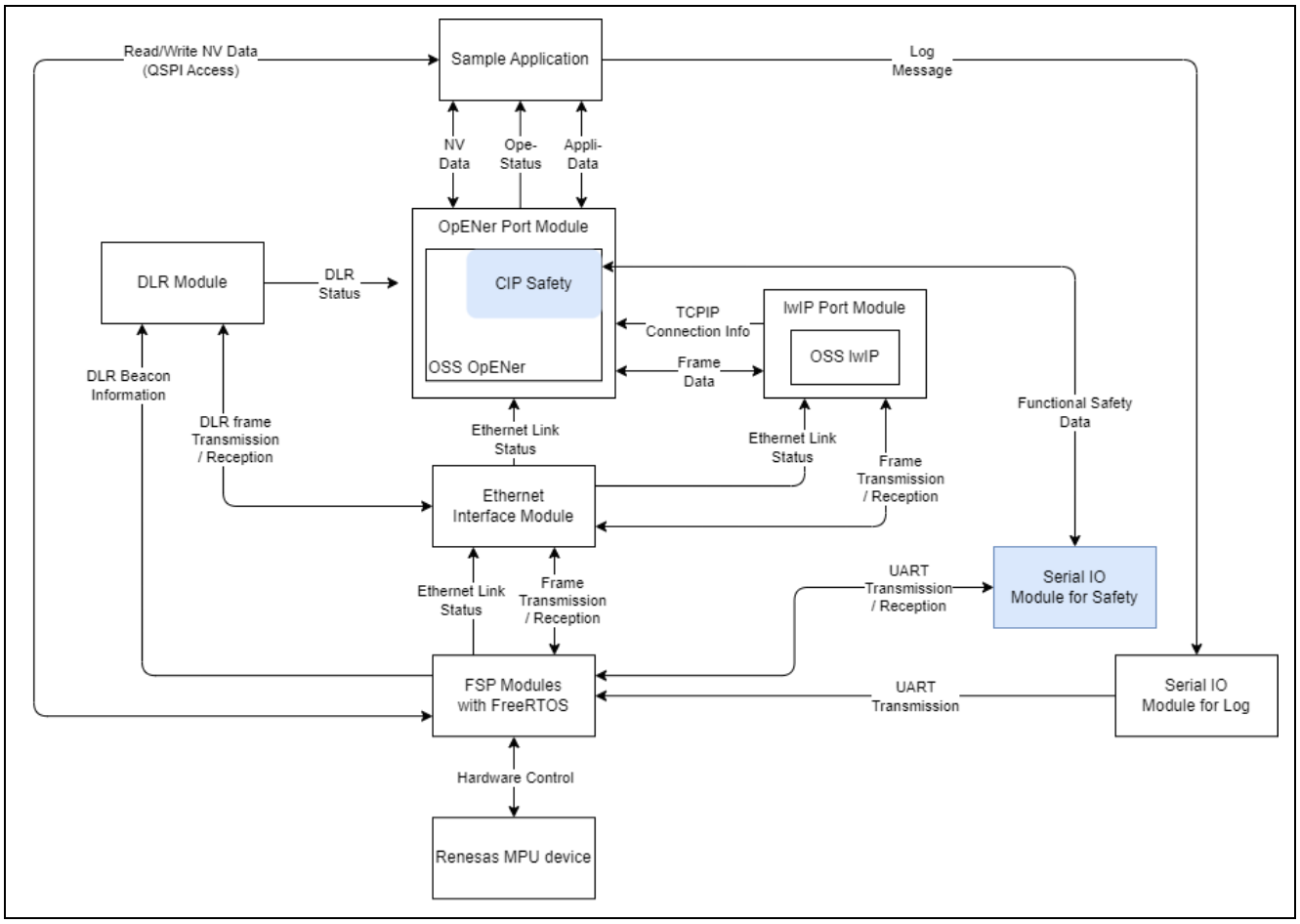


Figure 9.8 Software Stack Diagram when CIP Safety feature is enabled

System Configuration Diagram

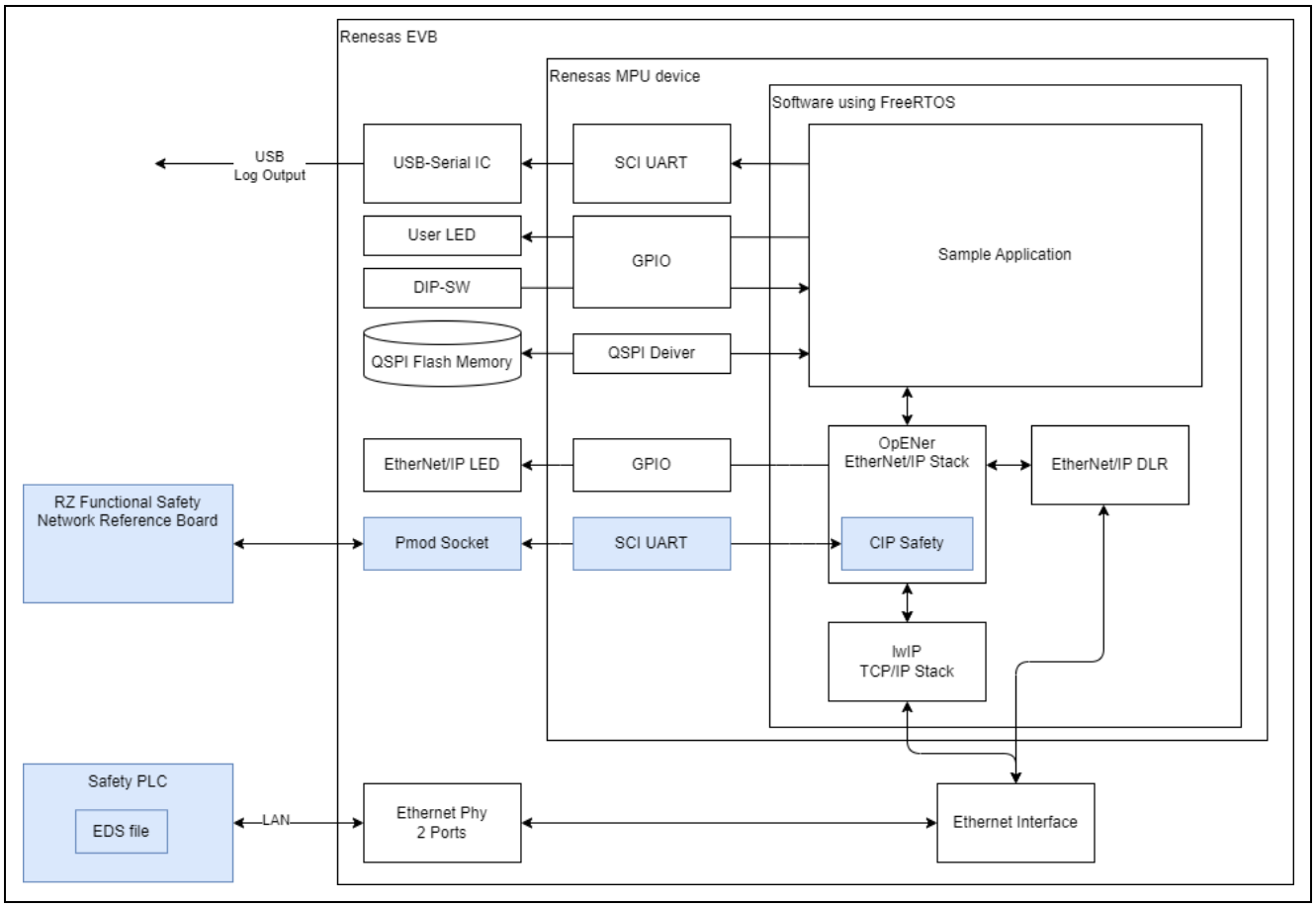


Figure 9.9 System Configuration Diagram when CIP Safety feature is enabled

Support Objects, Services and Attributes

Object classes, Services and Attributes supported in this sample code are shown below. The sections indicated in red are the ones that are modified when CIP Safety is enabled.

Table 9.6 Object Classes supported when CIP Safety feature is enabled.

Object Class #	Object Class Name
0x01	Identity
0x02	Message Router
0x04	Assembly
0x06	Connection Manager
0x39	Safety Supervisor
0x3A	Safety Validator
0x47	Device Level Ring
0x48	QoS
0x49	Safety Analog Input Point
0x64	Safety Analog Output Point
0xF5	TCP/IP Interface
0xF6	Ethernet Link
0x109	LLDP Management

Table 9.7 0x01: Identity

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
1	Vendor ID	○	-
2	Device Type	○	-
3	Product Code	○	-
4	Revision	○	-
5	Status	○	-
6	Serial Number	○	-
7	Product Name	○	-
Instance Service			
#	**Service Name**		
0x01	Get_Attributes_All		
0x05	Reset **		
0x0E	Get_Attribute_Single		

*: The values of instance attributes #1~#4, #6, and #7 are configured by macros in “devicedata.h" in “common\renesas\oss_deps\opener” directory.

**: In accordance with the SRS61 specification, the Identity Reset service will respond with error code 0x10 if even a single I/O connection is established.

Table 9.8 0x02: Message Router

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
No instance attributes are implemented.			
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		

Table 9.9 0x04: Assembly

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
3	Data	○	○
4	Size	○	-
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		
0x10	Set_Attribute_Single		

Table 9.10 0x06: Connection Manager

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
No instance attributes are implemented.			
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		
0x4E	Forward_Close		
0x54	Forward_Open		
0x5a	Get_Connection_Owner		
0x5b	Large_Forward_Open		

Table 9.11 0x39: Safety Supervisor

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
11	Device Status	○	-
12	Exception Status	○	-
13	Exception Detail Alarm	○	-
14	Exception Detail Warning	○	-
15	Alarm Enable	○	○
16	Warning Enable	○	○
24	Configuration Lock	○	-
25	Configuration UNID	○	-
26	Safety Configuration Identifier	○	-
27	Target UNID	○	-
28	Output Connection Point Owners	○	-
29	Proposed UNID	○	-
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		
0x10	Set_Attribute_Single		
0x0D	Apply_Attributes		
0x4F	Configure_Request		
0x50	Validate_Configuration		
0x51	Set_Password		
0x52	Configuration_Lock/Unlock		
0x54	Safety_Reset (Type0 Power Cycle, Type 1 Factory Reset)		
0x55	Reset_Password		
0x56	Propose_TUNID		
0x57	Apply_TUNID		

Table 9.12 0x3A: Safety Validator

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
8	Safety Connection Fault Count	○	-
Class Services			
#	**Service Name**		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
1	Safety Validator Status	○	-
2	Safety Validator Type	○	-
12	Max Data Age	○	○
13	Application Data Path	○	-
15	Producer/Consumer Fault Counters	○	-
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		

Table 9.13 0x47: Device Level Ring

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
1	Network Topology	○	-
2	Network Status	○	-
10	Active Supervisor Address	○	-
12	Capability Flags	○	-
Instance Service			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		

Table 9.14 0x48: QoS

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
1	802.1Q Tag Enable	○	-
2	DSCP PTP Event	○	-
3	DSCP PTP General	○	-
4	DSCP Urgent	○	○
5	DSCP Scheduled	○	○
6	DSCP High	○	○
7	DSCP Low	○	○
8	DSCP Explicit	○	○
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		
0x10	Set_Attribute_Single		

Table 9.15 0x49: Safety Analog Input Point

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
2	Safety Analog Input Value	○	-
3	Input Range	○	○
4	Input Channel Mode	○	○
5	Input Point Status	○	-
6	Point Fault Reason	○	-
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		
0x10	Set_Attribute_Single		

Table 9.16 0x64: Safety Analog Output Point

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
2	Safety Analog Output Value	○	-
3	Output Range	○	○
4	Output Channel Mode	○	○
5	Output Point Status	○	-
6	Point Fault Reason	○	-
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		
0x10	Set_Attribute_Single		

Table 9.17 0xF5: TCP/IP Interface

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
1	Status	○	-
2	Configuration Capacity	○	-
3	Configuration Control	○	○
4	Physical Link Object	○	-
5	Interface Configuration	○	-
6	Host name	○	-
7	Safety Network Number	○	-
8	TTL Value	○	-
9	Mcast Config	○	-
10	Select ACD	○	○
11	LastConflictDetected	○	○
13	Encapsulation Inactivity Timeout	○	○
Instance Service			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
0x10	Set_Attribute_Single		

Table 9.18 0xF6: Ethernet Link

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
1	Interface Speed	○	-
2	Interface Flag	○	-
3	Physical Address	○	-
7	Interface Type	○	-
11	Interface Capability	○	-
Instance Service			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
0x4C	Get_and_Clear		

Table 9.19 0x109 LLDP Management

Class Attributes			
#	**Attribute Name**	**Get**	**Set**
1	Revision	○	-
2	Max Instance	○	-
3	Number of Instance	○	-
4	Optional Attribute List	-	-
5	Optional Service List	-	-
6	Max Number Class Attributes	○	-
7	Max Number Instance Attributes	○	-
Class Services			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
Instance Attributes*			
#	**Attribute Name**	**Get**	**Set**
1	LLDP Enable	○	○
2	msgTxInterval	○	○
3	msgTxHold	○	○
4	LLDP Database	○	-
5	Last Change	○	-
Instance Service			
#	**Service Name**		
0x0E	Get_Attribute_Single		
0x10	Set_Attribute_Single		

Device Data Configuration

Configurations that are often set for individual devices are listed in this section when CIP Safety feature is enabled.

Table 9.20 Device data configuration List

Directory and file	Line	Item	Macro definition	Initial setting
common\renesas\application\opener_port_instance.c	L71	IP address	SAMPLE_STATIC_IP_ADDRESS	192.168.1.10
	L72	Subnet mask	SAMPLE_STATIC_SUBNET_MASK	255,255,255,0
	L73	Gateway address	SAMPLE_STATIC_GATEWAY_ADDRESS	192,168,1,2
	L74	Host name	SAMPLE_HOSTNAME	"OPENER_NETIF0"
	L54	Device serial number	SAMPLE_CIP_DEVICE_SERIAL_NUMBER	0x12345678
common\renesas\oss_deps\opener\devicedata.h	L15	Device vendor ID	OPENER_DEVICE_VENDOR_ID	1105 (0x0451, Renesas Electronics)
	L16	Device type	OPENER_DEVICE_TYPE	42 (0x002A, Safety Analog I/O Device)
	L17	Device name	OPENER_DEVICE_NAME	“RZ Adaptor Safety”
	L20	Product code	OPENER_DEVICE_PRODUCT_CODE	21258 (0x530A)
	L21	Device major revision	OPENER_DEVICE_MAJOR_REVISION	3
	L22	Device minor revision	OPENER_DEVICE_MINOR_REVISION	7
common\renesas\oss_deps\lwip\lwipopts.h	L103	MIB2 system description	SNMP_LWIP_MIB2_SYSDESC	"Renesas Electronics Corporation EtherNet/IP Sample"
	L104	MIB2 system contact	SNMP_LWIP_MIB2_SYSCONTACT	"sysContact not set"
	L105	MIB2 system name	SNMP_LWIP_MIB2_SYSNAME	"sysName not set"
	L106	MIB2 system location	SNMP_LWIP_MIB2_SYSLOCATION	"sysLocation not set"

Note: The sections indicated in red are the ones that are modified when CIP Safety is enabled.

Appendix D: DLR Operation Check

This section describes how to check DLR operation.

The sample code implements the function as a DLR node only (not Supervisor or Redundant Gateway). Refer to Section 8.3 for supported Attributes and Services.

Check Procedure

1. Prepare Table 9.21 devices and Table 9.22 software tools.
2. Establish the Figure 9.10 topology and power up.
3. Download the program to the RSK board and run it on e^2 studio or EWARM. (Please refer to Chapter 6)
4. Monitor DLR Instance Attribute values and DLR operation with software tools.

Table 9.21 Used Devices

Product Name	Catalog #:	Vendor	Comment
Allen Bradley Compact Logic 384KB DI/O Controller	1769-L16ER-BB1B	Rockwell Automation	Used as a DLR Supervisor
Allen Bradley 3 port Ethernet Tap	1783-ETAP	Rockwell Automation	Used as a DLR node (Sub-supervisor)

Table 9.22 Used Software Tools

Software Name	Released by
EIP Tools	Molex
RSLinx	Rockwell Automation
DLR Tool	Rockwell Automation

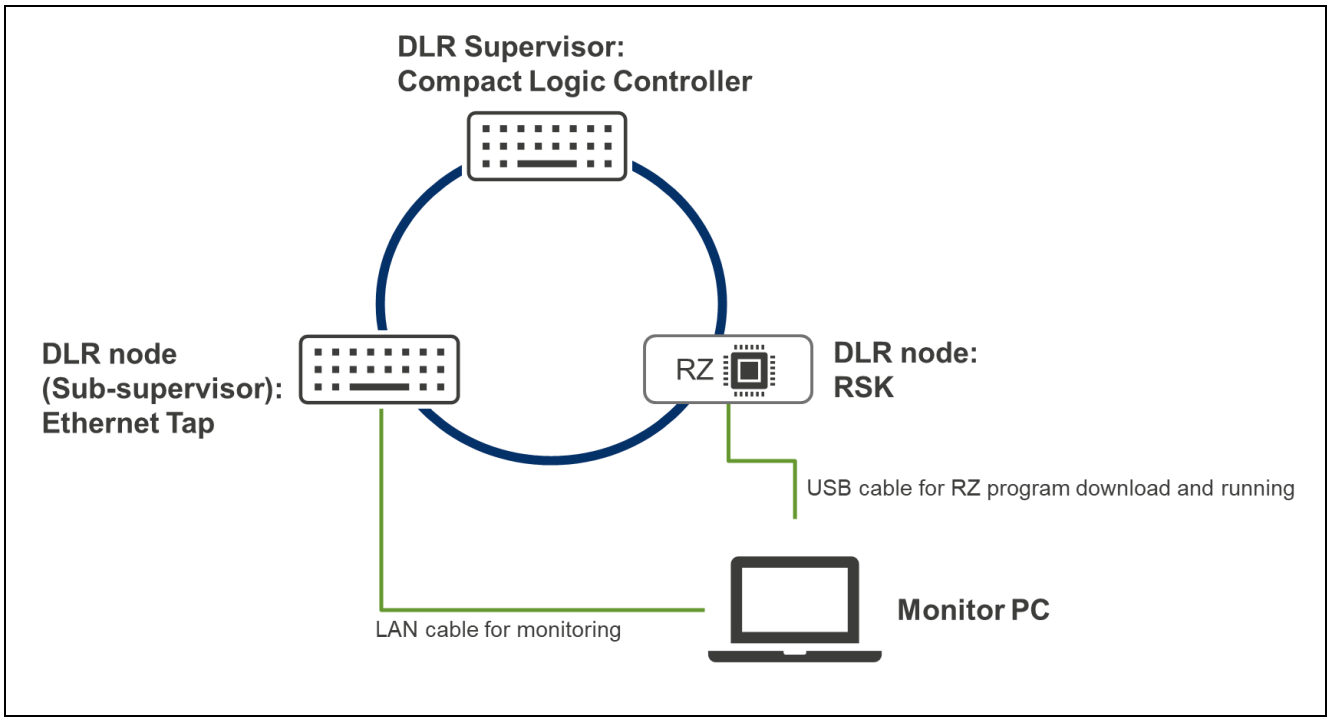


Figure 9.10 Topology for check

Check DLR Instance Attribute values by EIP Tools

The Instance Attribute value can be checked using the EIP Tools. The procedure is described below.

Open EIP Tools and enter the IP address of the target device. The default setting for the IP address written in the sample code is 192.168.1.10.

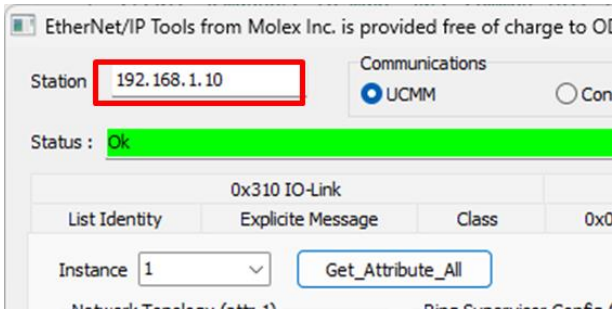
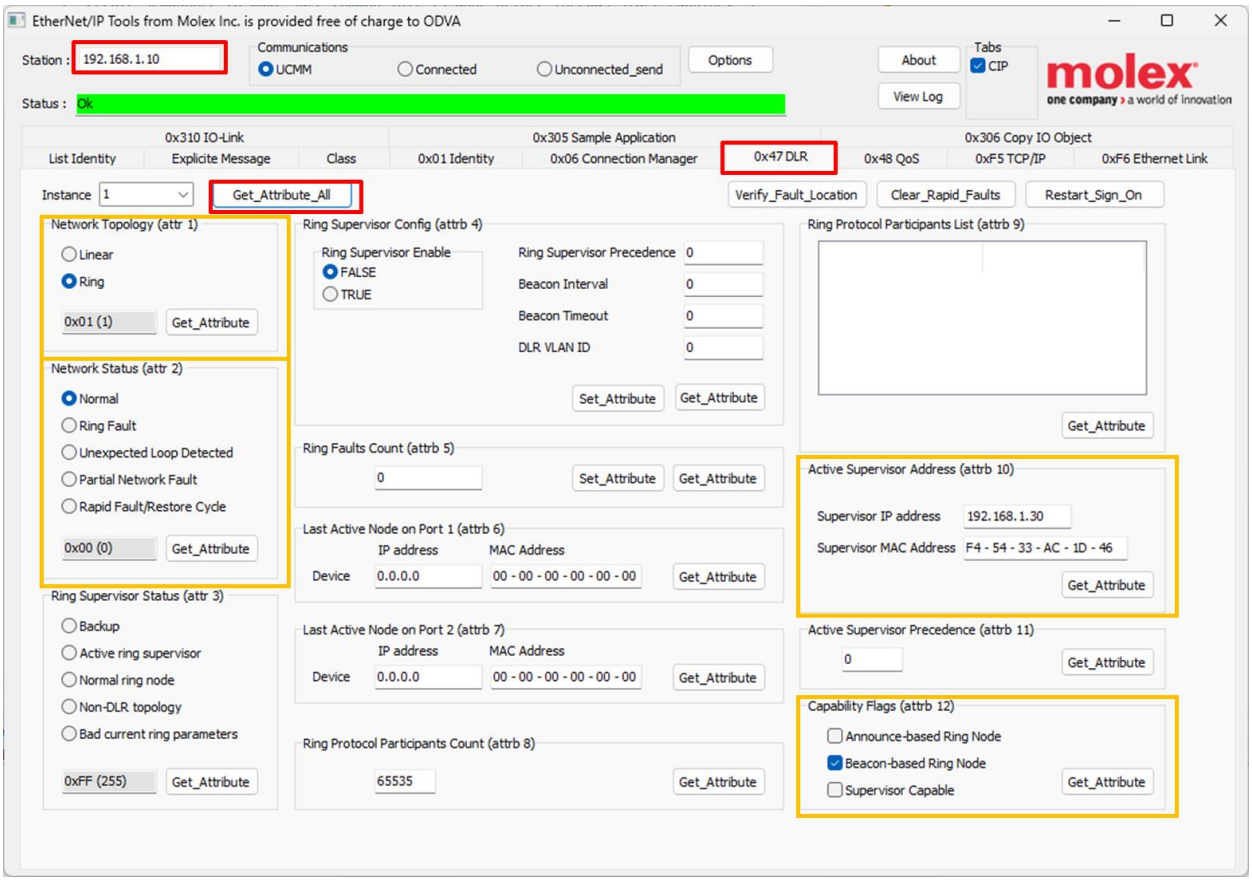


Figure 9.11 Enter the IP address of the target device in EIP Tools

Select the 0x47 DLR tab and click Get_Attributes_All or Get_Attribute_Single. You can see the value of each.



Note: The sample code supports Instance Attribute #1, 2, 10, and 12. (The yellow circled area)

Figure 9.12 Execute Get_Attributes_All in EIP Tools

Check DLR operation by RSLinx and DLR Tool

DLR operation can be checked with the DLR Tool. RSLinx driver settings are required as a preliminary step.

In the following images, please read the IP address 192.168.1.170 as 192.168.1.10.

■RSLinx driver settings

Launch RSLinx. If the RSWho window is not open, click RSWho from the Communication tab.

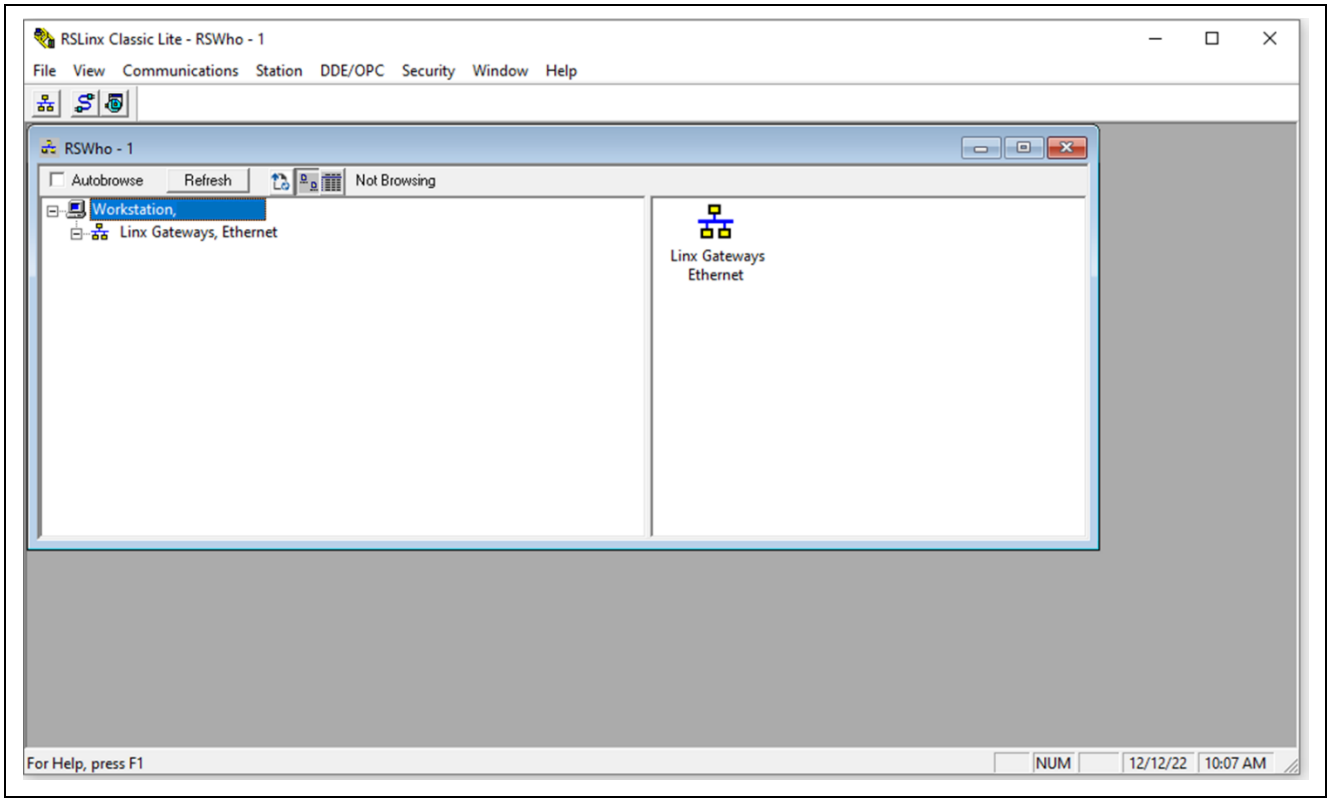


Figure 9.13 Start RSLinx

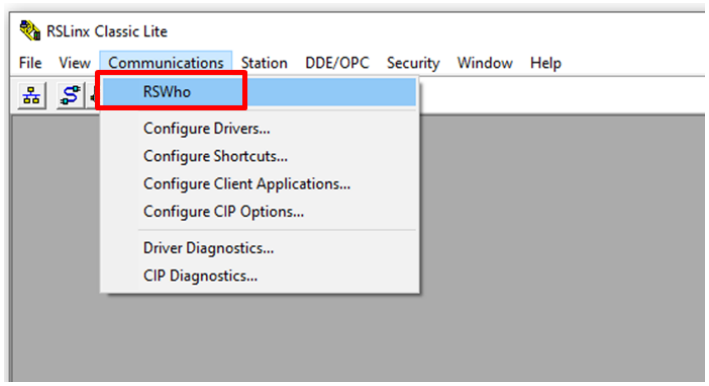


Figure 9.14 Open RSWho window

Create a new driver to monitor the DLR status. Click Configure Driver from the Communication tab.

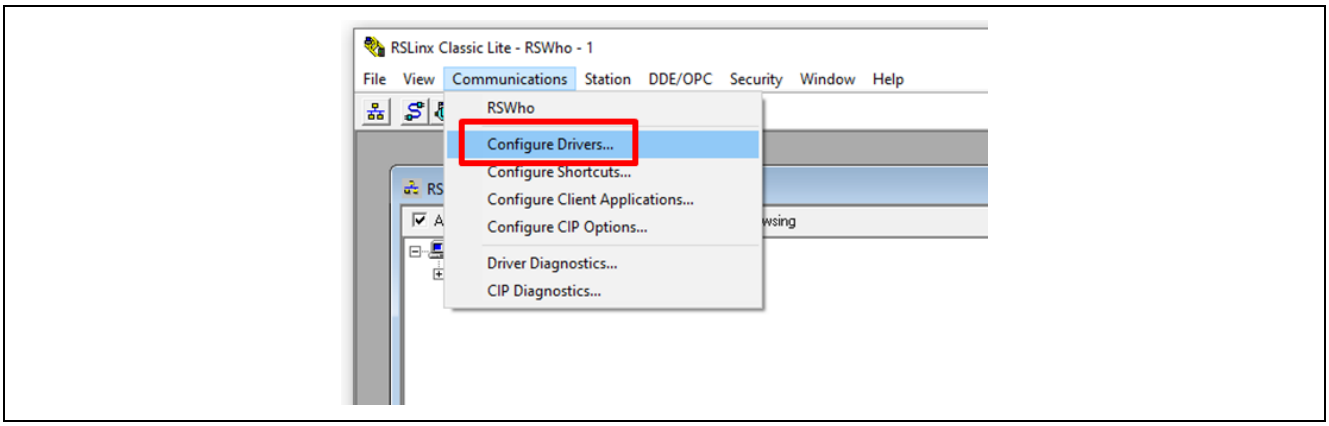


Figure 9.15 Open Configure Drivers window

Select EtherNet/IP Driver from “Available Driver Types” and click “Add New”.

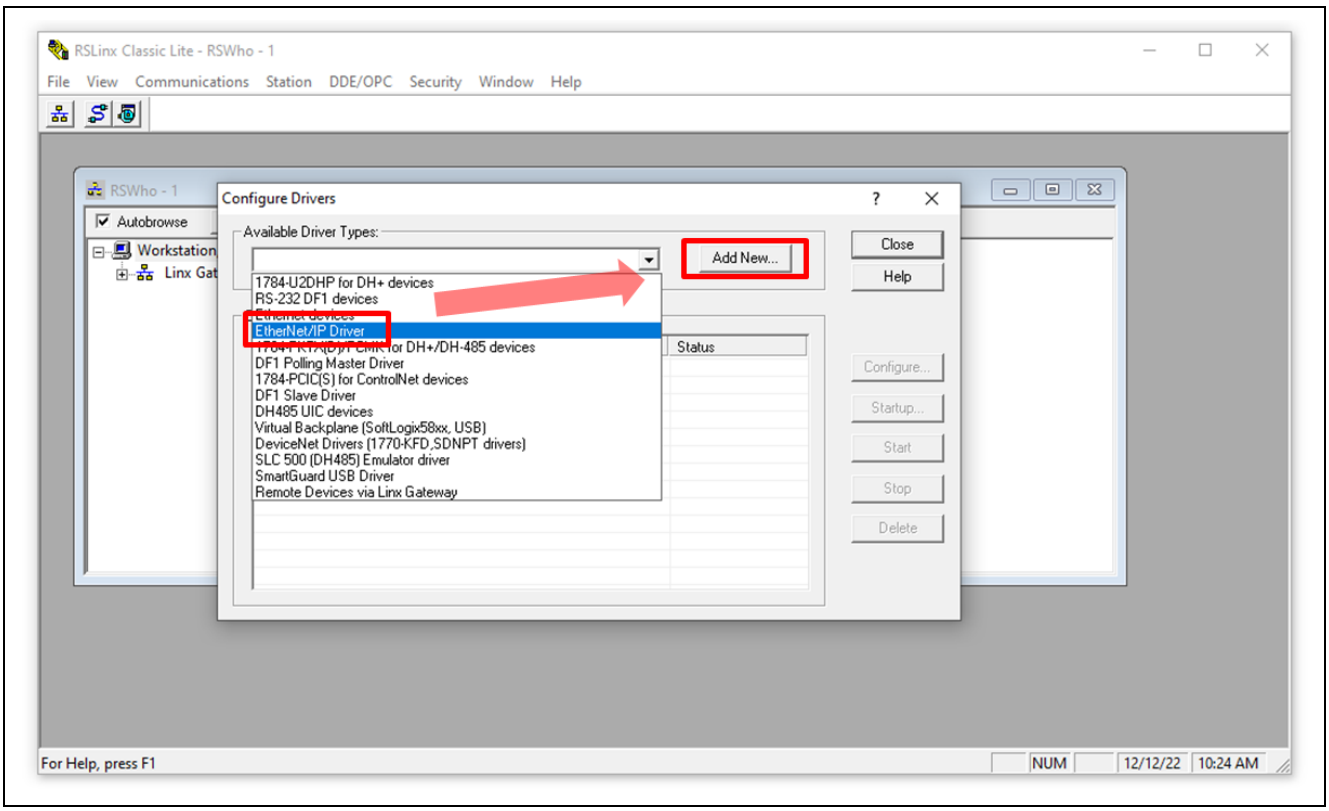


Figure 9.16 Create new EtherNet/IP Driver

Enter any driver name.

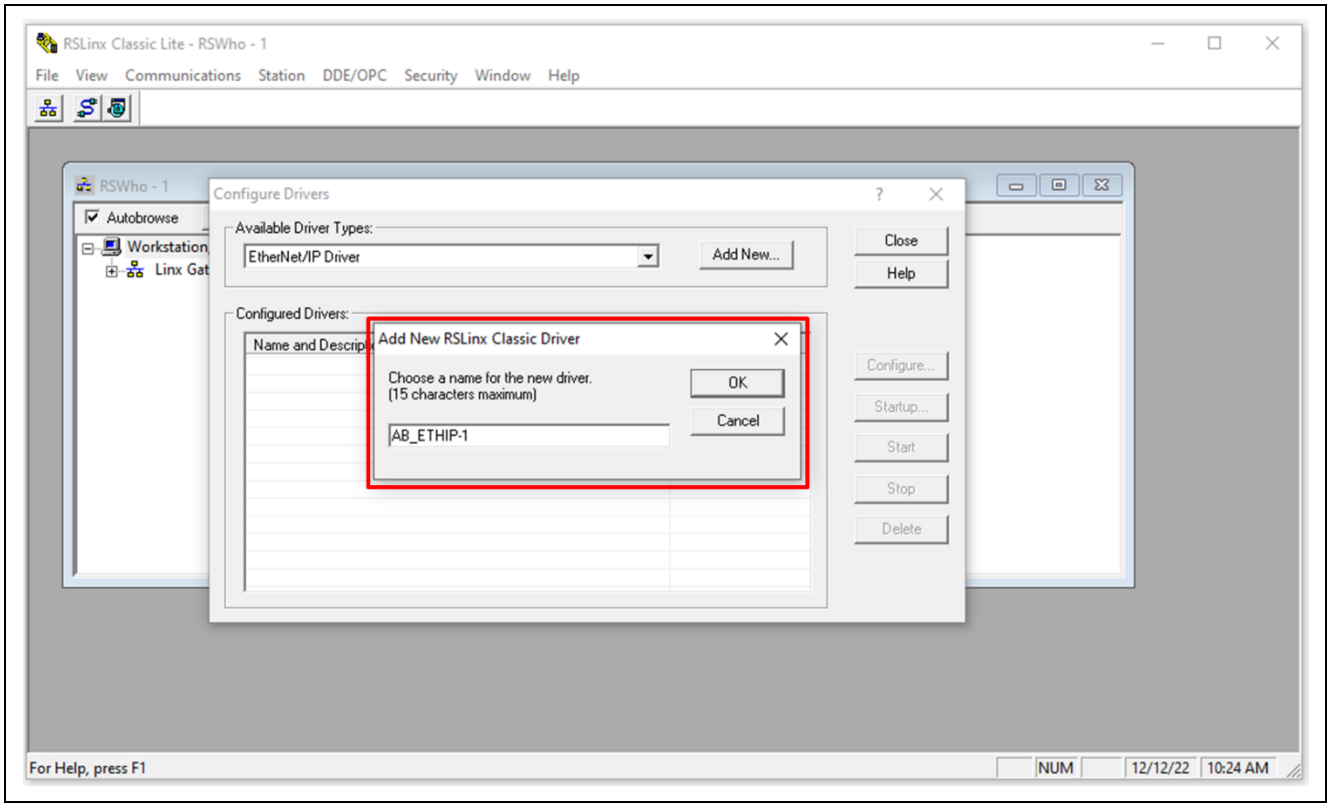


Figure 9.17 Enter new driver name

Select the Ethernet adapter connected to the DLR device, click Apply, and then click OK.

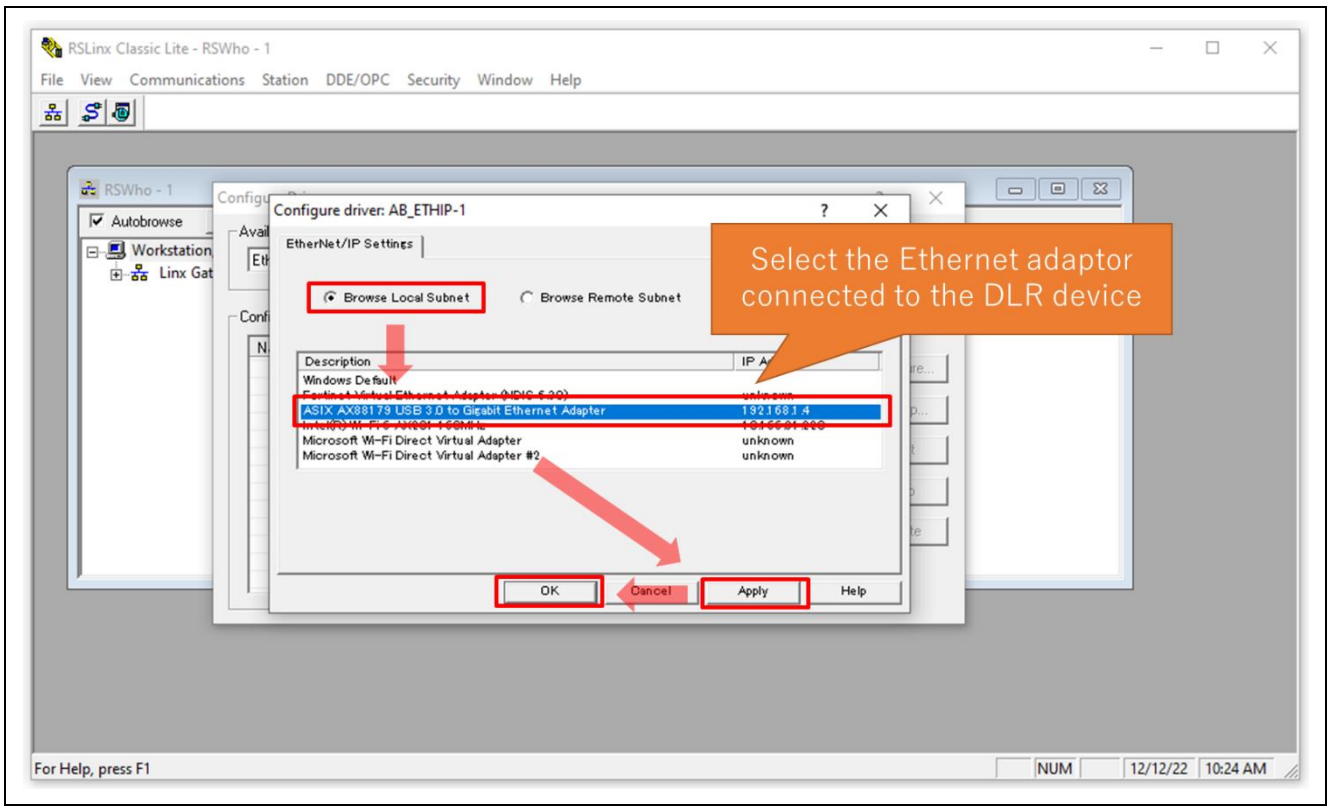


Figure 9.18 Select the Ethernet adaptor connected to the DLR device

Confirm that the driver has been added and Close the Configure Drivers window.

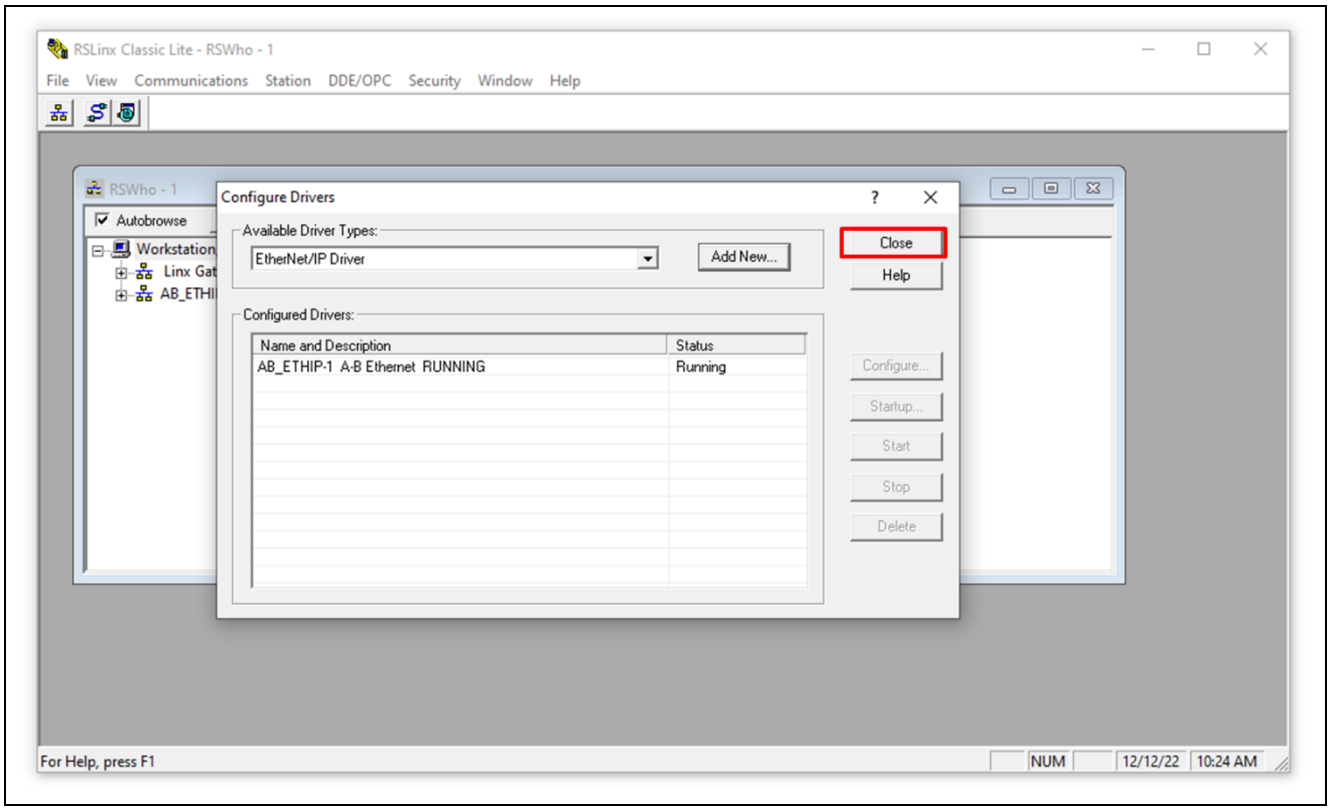


Figure 9.19 Close Configure Drives window

Make sure Autobrowse is enabled and expand the driver you have created. Confirm that the device is recognized as shown in Figure 9.20.

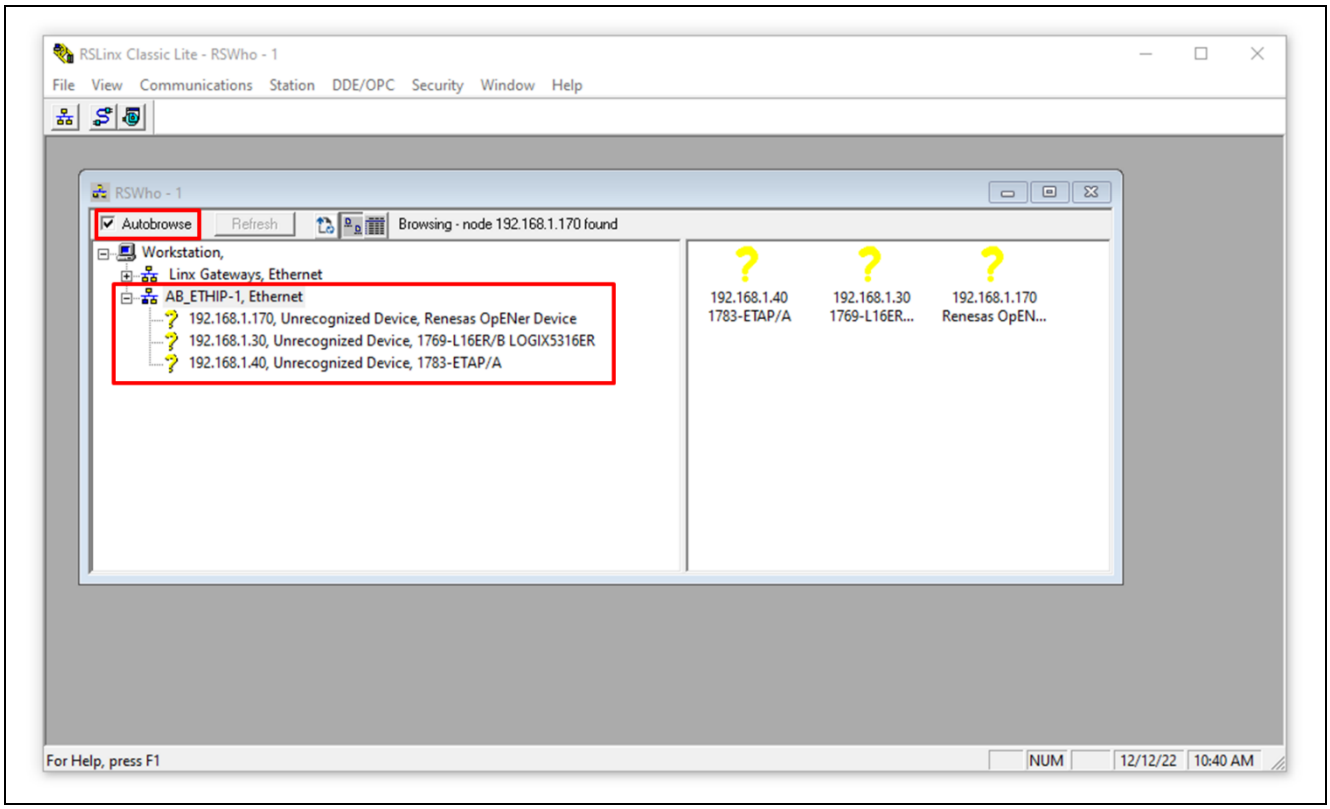


Figure 9.20 Check Autobrowse setting and devices

Note:

Yellow question marks are resolved by registering the EDS files of each device. See "Register EDS files of devices" in RSLinx Classic Help for details.
The EDS file for the sample code is stored in the scanner folder (top level of the sample code zip file directory).

It is now ready to use the DLR Tool.

■Monitor DLR operation with the DLR Tool

Launch the DLR Tool; click the Connect button.

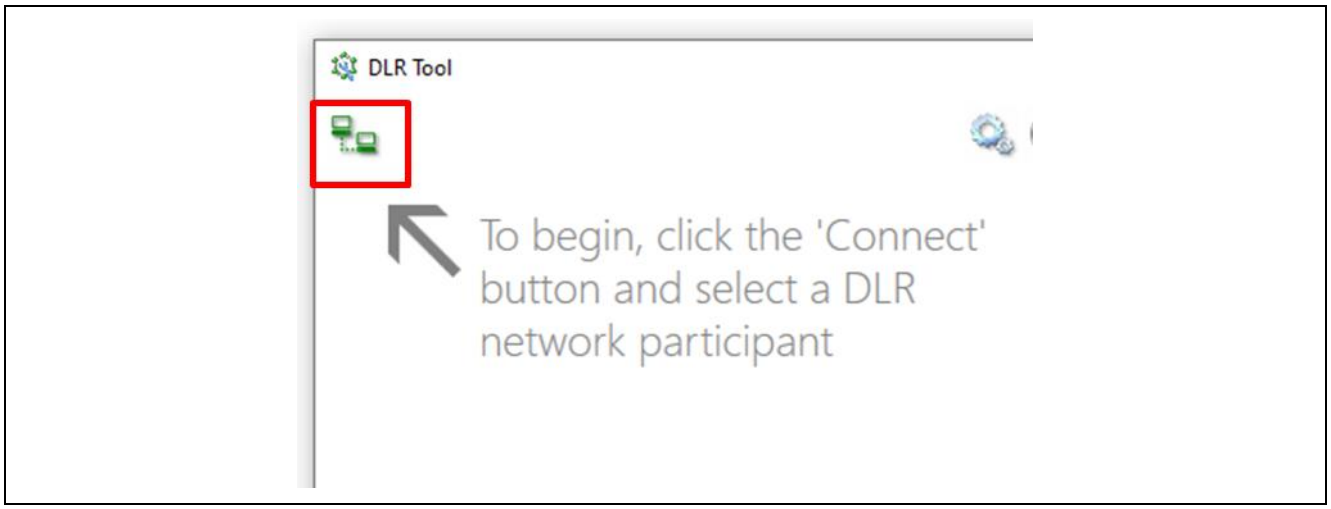


Figure 9.21 Launch DLR Tool and Click Connect button

Expand the added drivers, select one of the devices, and click the Select button.

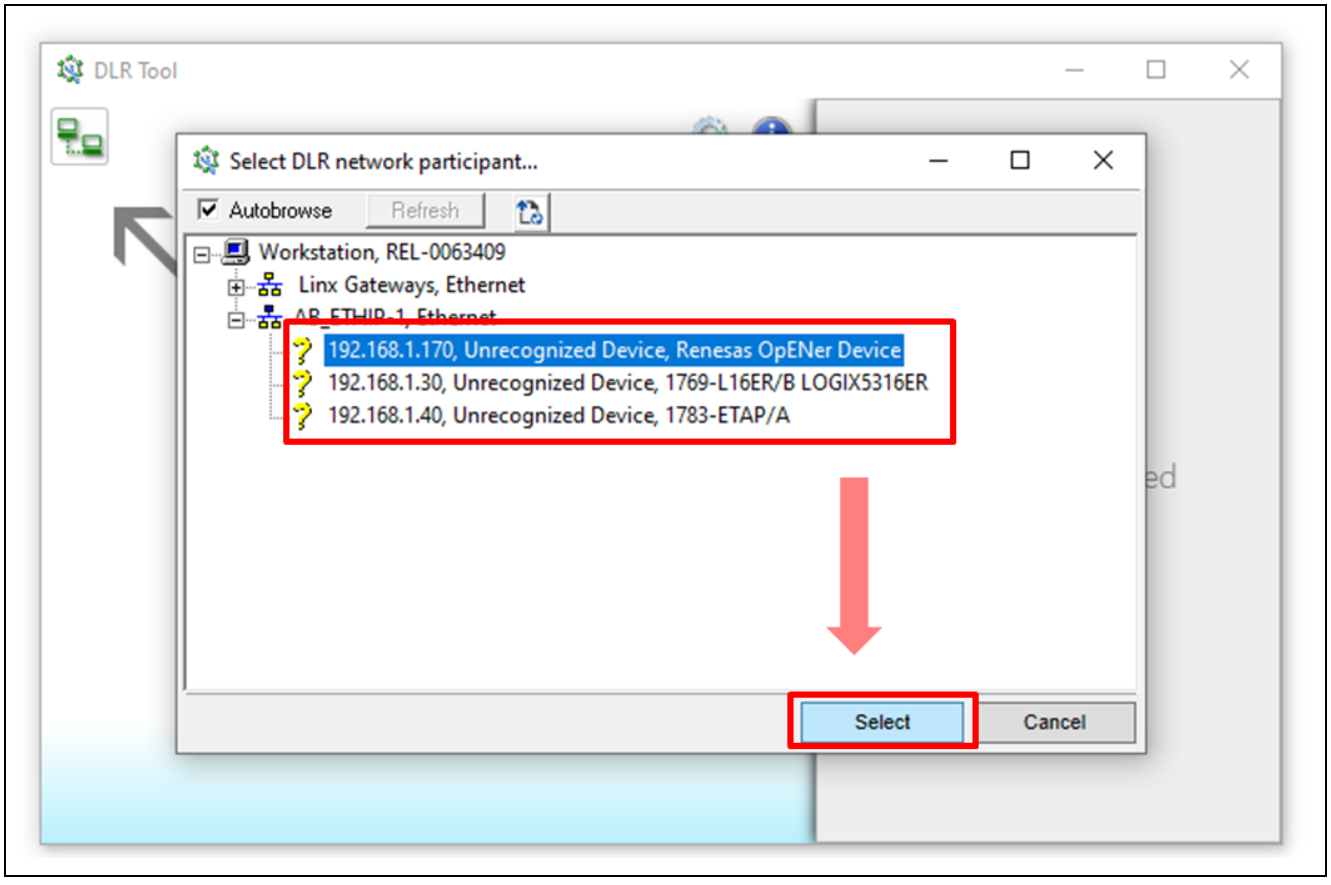


Figure 9.22 Click one of DLR devices and Select button

The status of the DLR operation is graphically displayed and can be monitored.

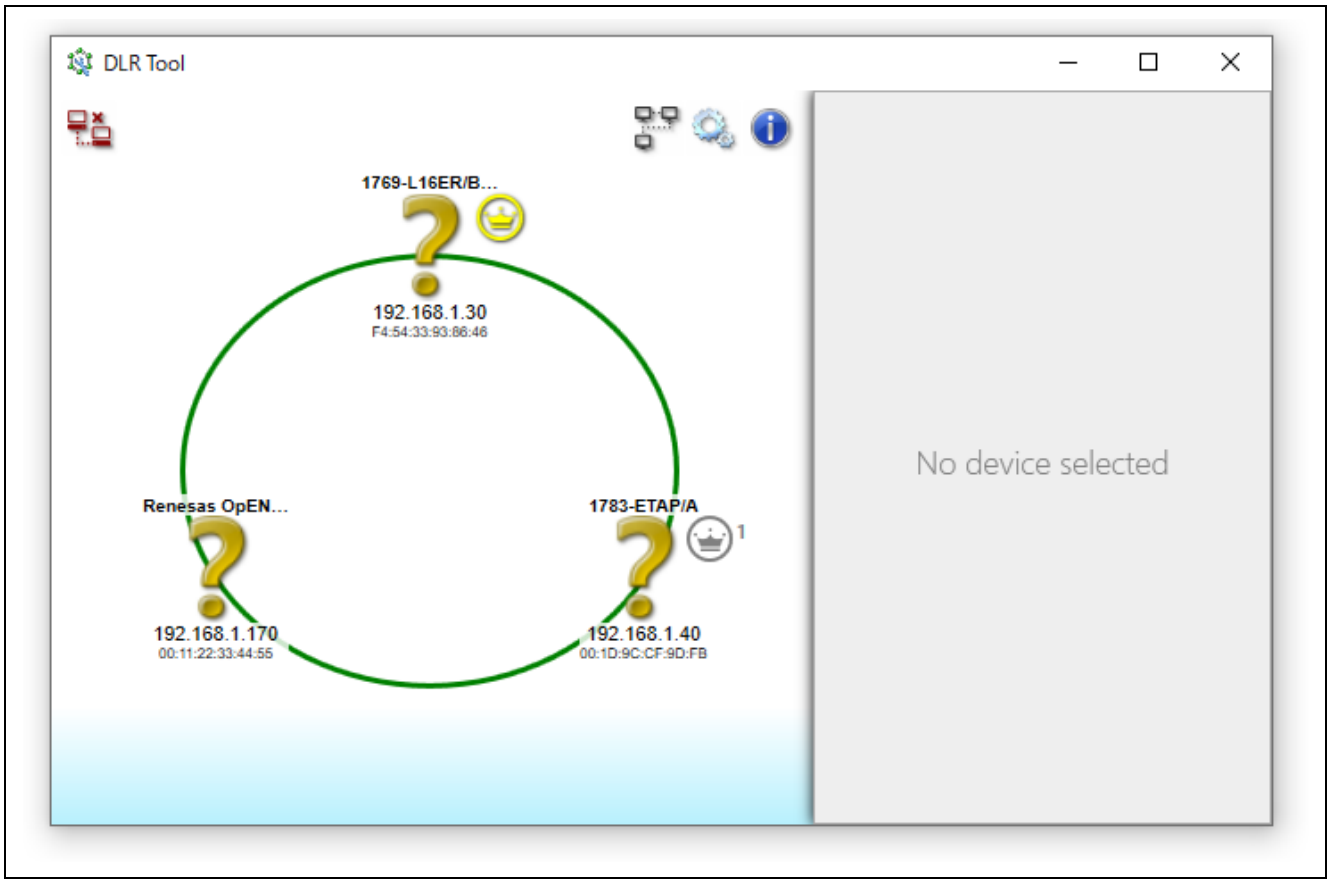


Figure 9.23 DLR operation is graphically displayed on DLR Tool

If any of the topology route is disconnected, the area is highlighted in red. It will be restored when reconnecting.

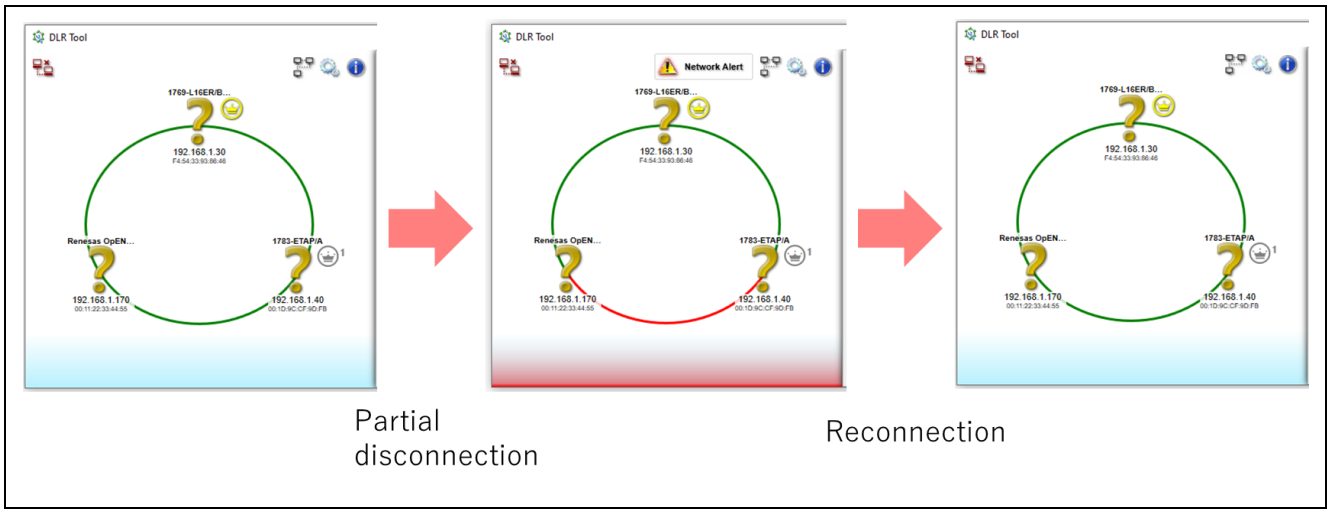


Figure 9.24 Partial disconnection indication and Reconnection

By clicking on a device, information about the device is displayed.

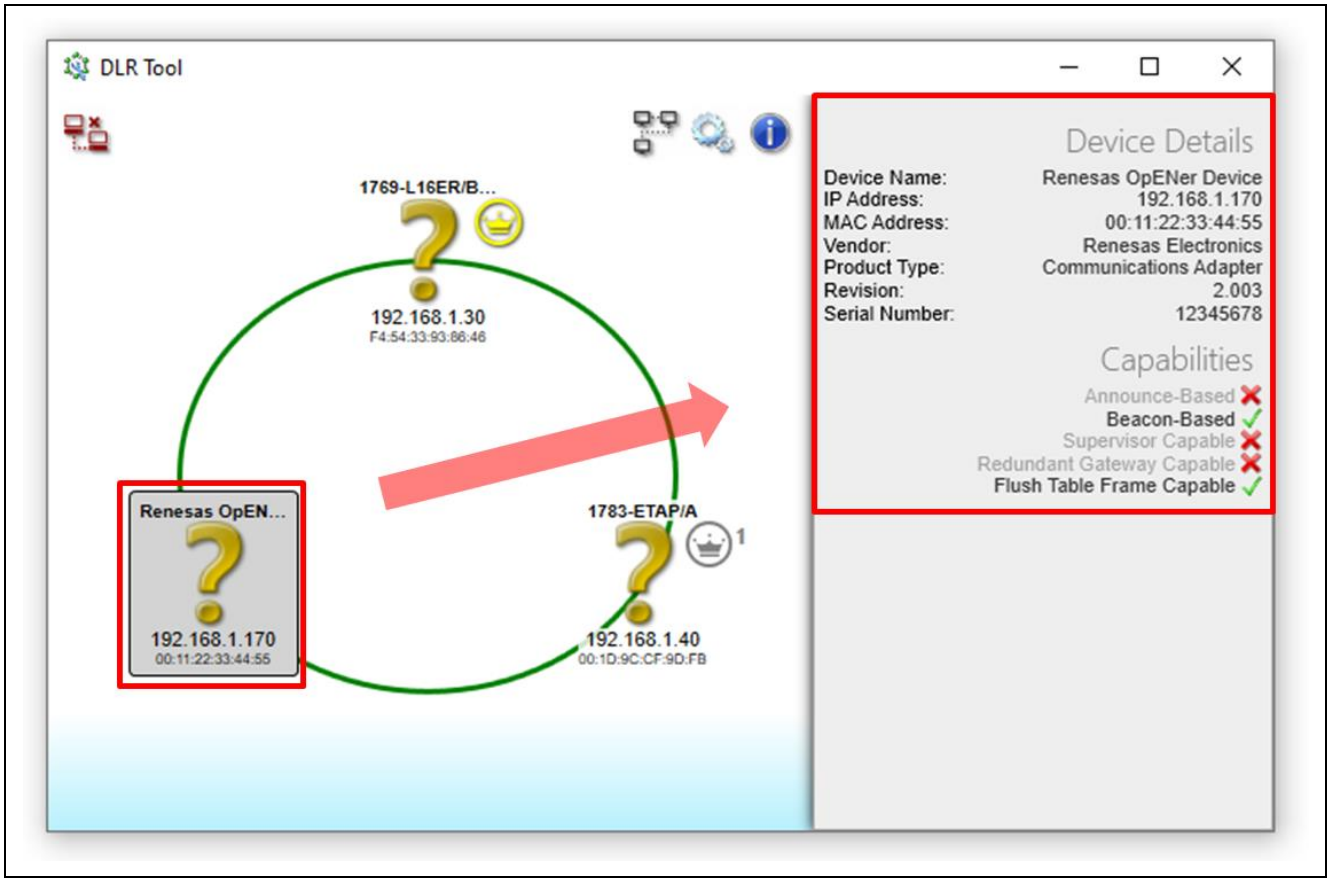


Figure 9.25 Device Details field

Appendix E: How to set fixed Interface Speed 100Mbps Full-Duplex

In testing EtherNet/IP devices that support DLR, Auto-Negotiation may have to be disabled, and the interface speed fixed to 100 Mbps Full-Duplex.

This section describes how to set fixed Interface Speed.

1. Open FSP Configuration Window

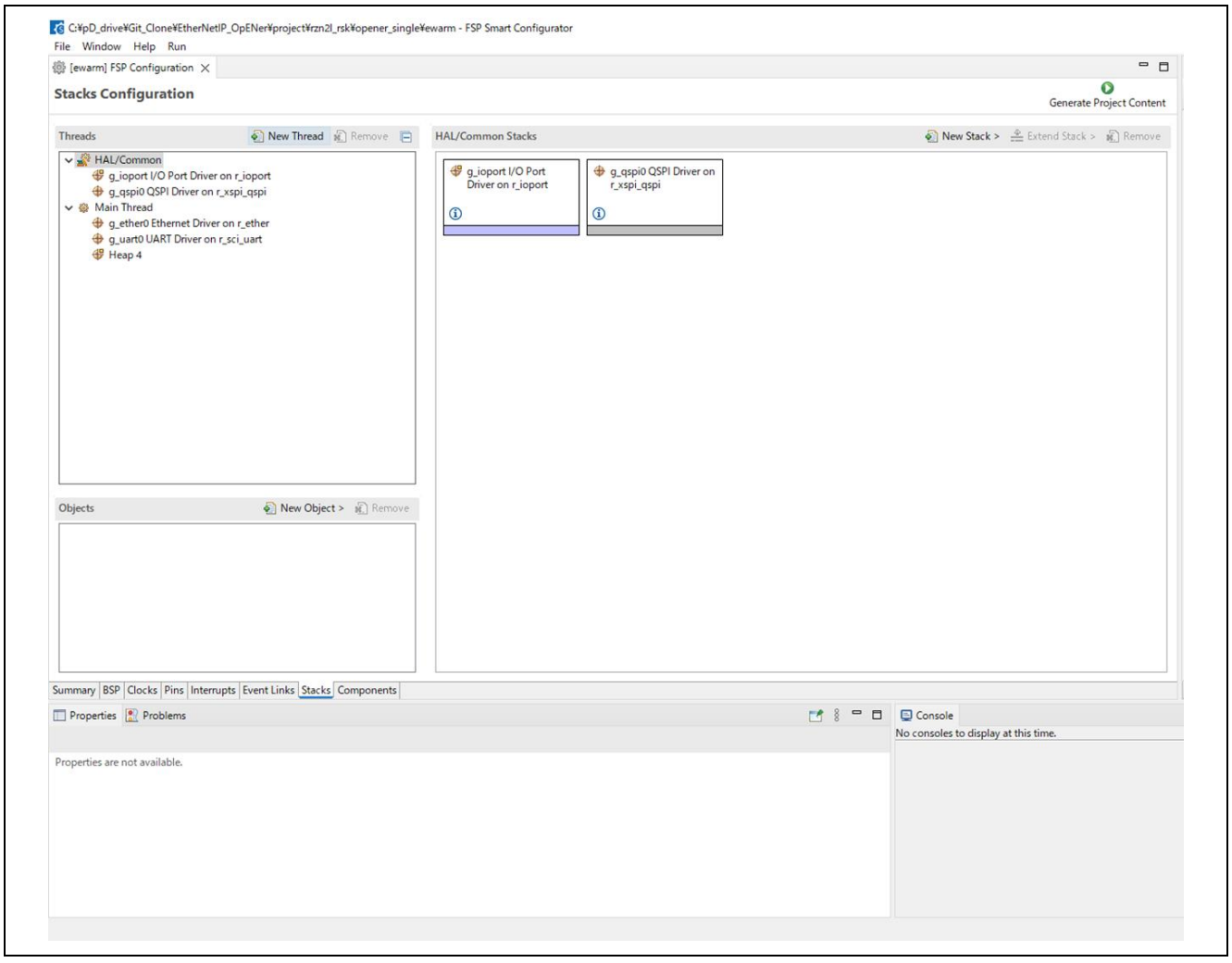


Figure 9.26 FSP Configuration Window, in the case of Smart Configurator

2. Please select “g_ether0 Ethernet Driver” -> “g_ether_phy0 Ethernet Driver on r_ether_phy” (ETH 0 case), and check Properties tab.

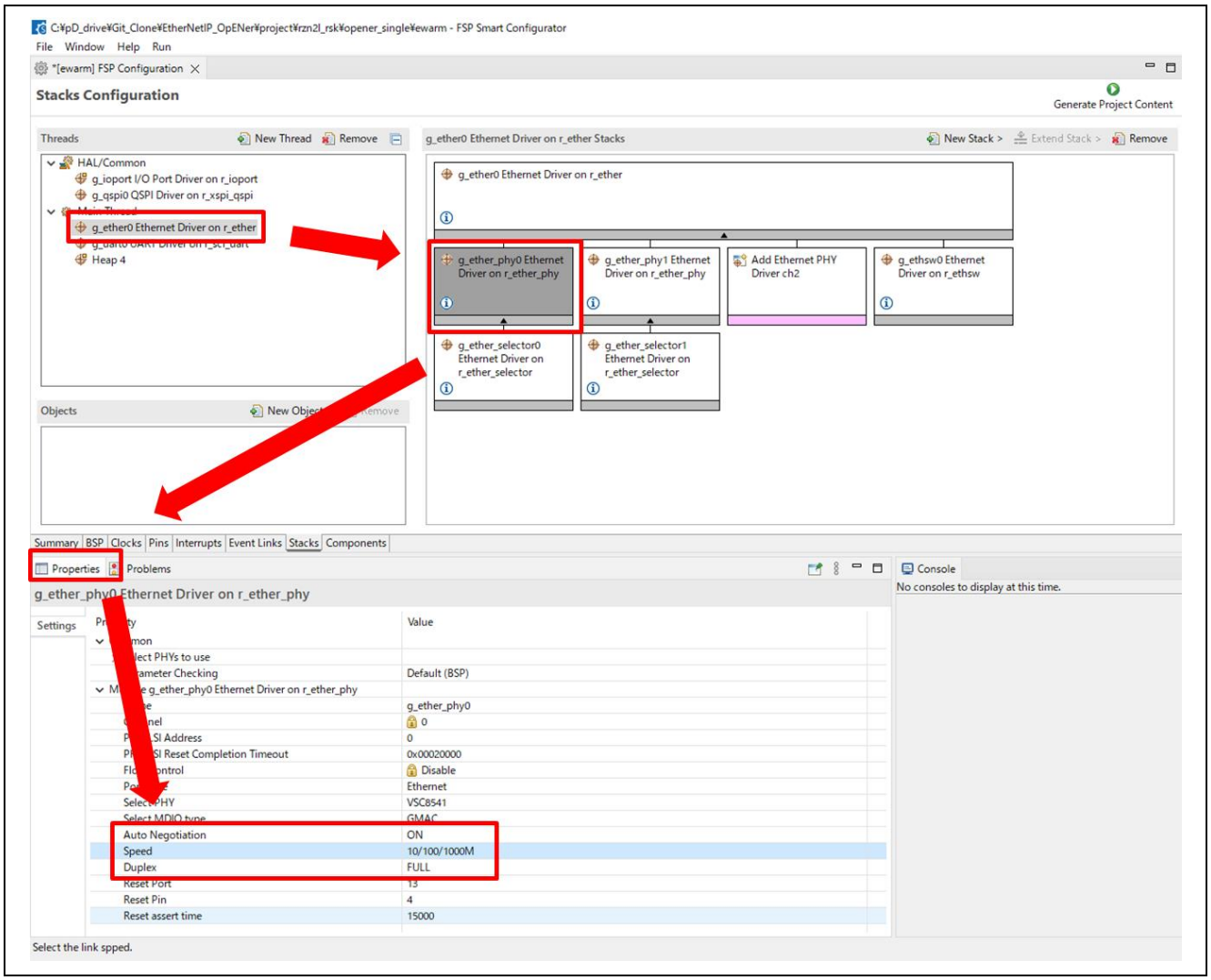


Figure 9.27 Property tab in r_ether_phy

3. Change Interface settings.
   - Auto Negotiation: ON -> OFF
   - Speed: 10/100/1000M -> 100M
   - Duplex: Full (No need to change)

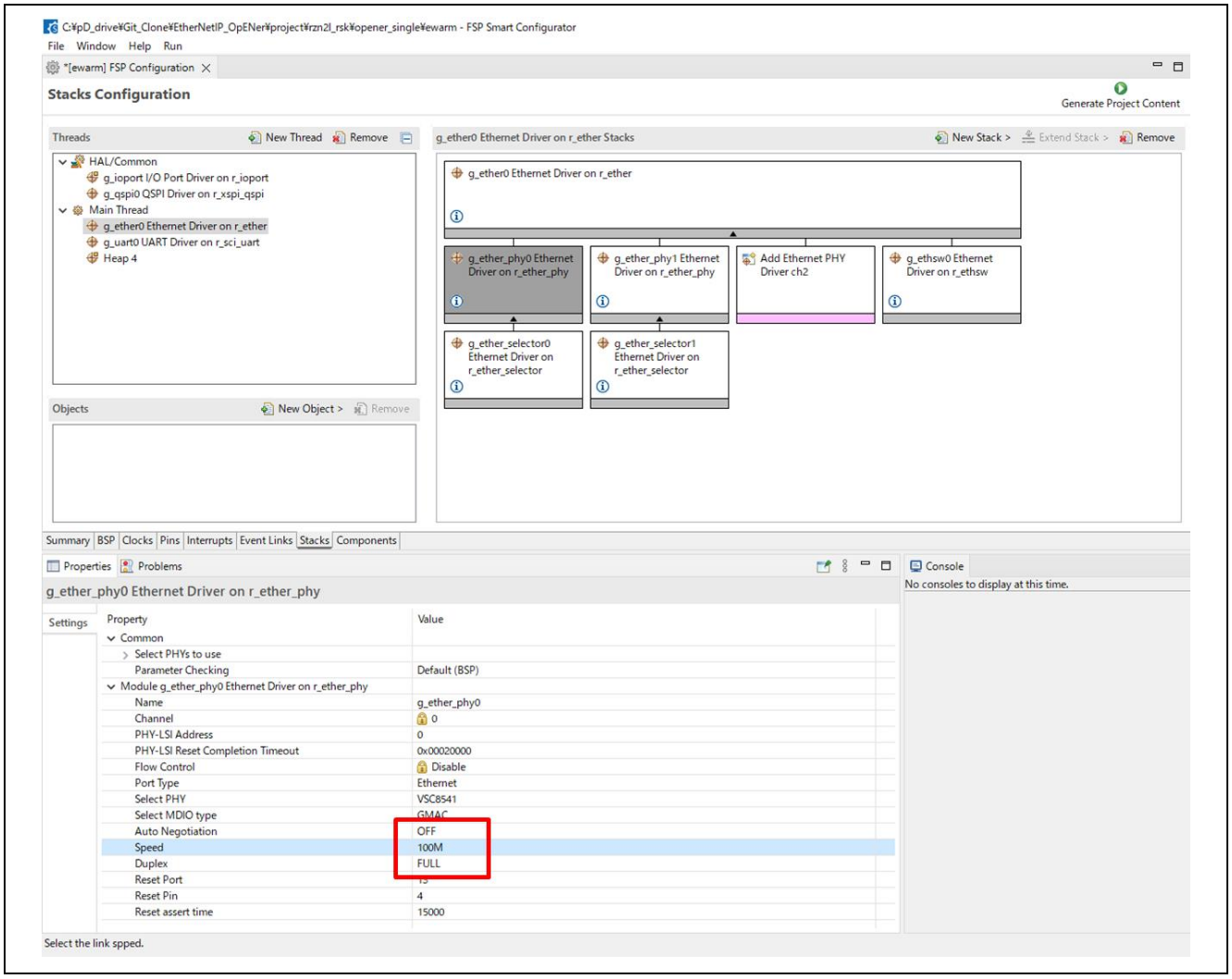


Figure 9.28 Change values in Interface Speed settings

4. Re-generate FSP files and build.

Appendix F: How to resolve issues with building or running sample program with FSP v4.1.0 or later v4.x.x

If you encounter issues with building or running sample program with FSP v.4.1.0 or later v4.x.x, resolve them by following the steps below.

1. Before generating the project content, back up the file/folder shown below.

Table 9.23 the file/folder to back up

Project	File/folder to back up
EWARM	script\fsp_ram_execution.icf
e² studio	script

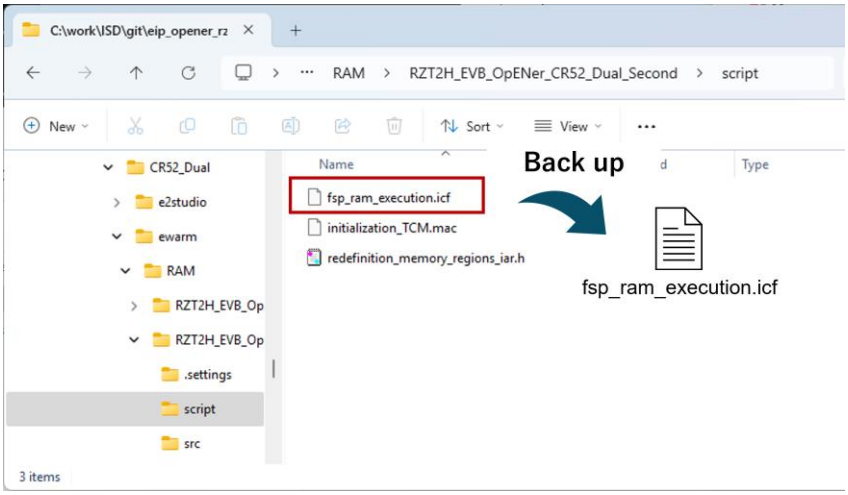


Figure 9.29 Back up file/folder (EWARM project)

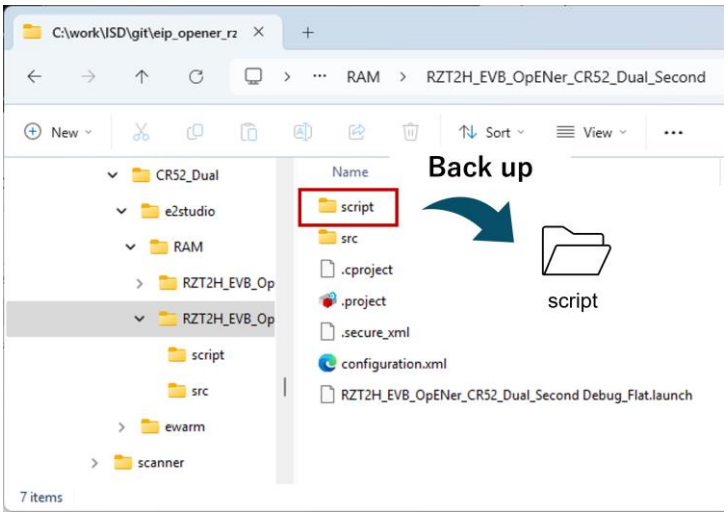


Figure 9.30 Back up file/folder (e² studio project)

2. Generate FSP files.

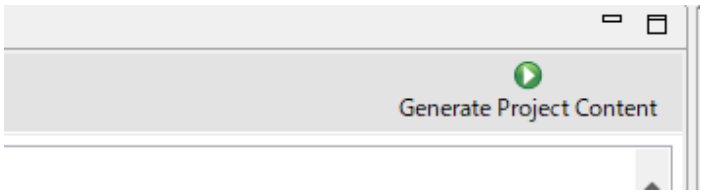


Figure 9.31 Generate FSP files

3. Overwrite the file/folder you backed up in step 1 and build the project.

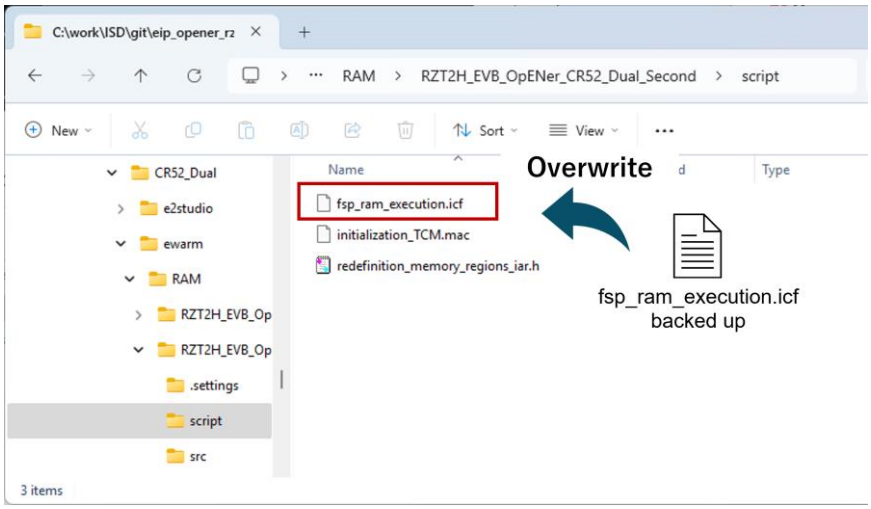


Figure 9.32 Overwrite the file/folder (EWARM project)

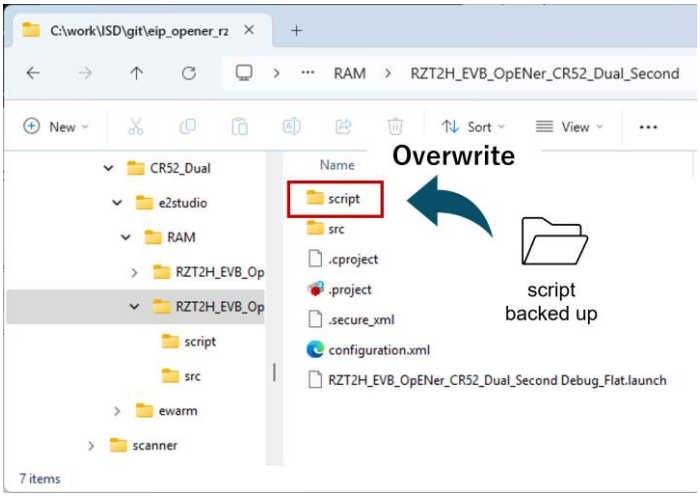


Figure 9.33 Overwrite the file/folder (e² studio project)

Licenses

- OpENer

Regarding the license of OpENer, please see the following file.

```
common\oss\OpENer\license.txt
```

- FreeRTOS

Regarding the license of FreeRTOS, please see the following file.

```
common\oss\amazon-freertos\LICENSE
```

- lwIP

Regarding the license of lwIP, please see the following file.

```
common\oss\lwip\COPYING
```

Revision History

Rev.	Date	Description	
		Page	Summary
Older	-	-	Older history omitted.
2.00	Jun. 13, 2025	-	Implement the CIP Safety feature and ACD for RZ/N2L
		6	Delete the restriction regarding Flash writing because it is resolved. Delete the sentence regarding the icf file replacing.
		7	Add explanation about CIP Safety feature to Chapter 2 Features
		8-10	Remove annotations related to the CODESYS project
		11-23	Exchange the section order in Chapter 4 for easy to read.
		24	Remove the instructions for downloading a specific version, as well as any references to non-existent appendices.
		133	Delete guidance regarding replacing the e^2 studio linker file as a Note.
		137	Add guidance regarding replacing the ewarm linker file as a Note.
		170	Add a guide regarding library versions.
		178-187	Create a new chapter which describes software specifications. Move some appendixes to this chapter.
		184	Update support Attributes and Services for the implementation of ACD
		190-210	Create a new appendix which describes CIP Safety feature
3.00	Feb. 6, 2026	All	Update for RZ/T2M (and RZ/T2ME) Update FSP version from FSP v2.1.0 to v3.0.0 Implement following - ACD Improve functionality equivalent to that required by CT21. - LLDP function - Connection state transition Remove Flash boot projects.
		All	Update for RZ/N2L Update FSP version from v2.1.0 to v3.0.0. Disable support for CIP Safety feature temporarily. Improve functionality equivalent to that required by CT21. - LLDP - Connection state transition Remove Flash boot projects.
		5	Change expressions of reference list.
		6	Update the limitation about linker scripts for RZ/N2L projects.
		7, 158	Remove explanation about CIP Safety.
		8-9	Update requirements of RZ/t2M, RZ/T2ME and RZ/N2L.
		31-64	Remove guide of Flash boot setup for RZ/T2M and RZ/T2ME.
		95-102	Remove guide of Flash boot setup for RZ/N2L.
		137, All	Unify MAC Address into only 74:90:50:10:05:9A
		146	Update Software stack diagram.
		147	Update Configuration diagram.
		24,30,136,155	Unify IP Address into 192.168.1.10.
Later	-	-	Please find it on the next page.

Rev.	Date	Description	
		Page	Summary
4.00	Apr. 13, 2026	All	Support FSP v4.0.0 for all devices
		All	Update for RZ/T2H, RZ/N2H Implement following - ACD Improve functionality equivalent to that required by CT21. - LLDP function - Connection state transition Remove Flash boot projects.
		6	Update Restriction and Limitation. Memory allocation changes are not treated as limitations.
		17	Update RZ/T2H EVB setup configuration. Change SW14-4 setting from ON to OFF CPU0 1 wait
		31-57	Update build and download procedures. The build and download procedures that were previously described separately for each device have been unified into a single common procedure.
4.10	Aug. 21, 2026	7	Restore the description of the CIP Safety feature in Chapter 2 Features
		82-106	Restore the description of the CIP Safety feature in Appendix chapter C.
		87	Add explanation about connecting the UART communication cable for CIP safety.
		31	Add a guide regarding trouble shooting for FSP v4.1.0 or later v4.x.x.
		120-121	Create a new appendix which describes trouble shooting for FSP v4.1.0 or later v4.x.x.
-			

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

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

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

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

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

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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