RZ/N2L Group

R01AN6798EJ0110
Rev.1.10
Aug 7, 2023

RZ/N2L Industrial Network SOM Kit Application Note: EtherNet/IP OpENer Sample Program

Introduction

This document describes the setup procedure of the sample program for RZ/N2L port of EtherNet/IP™ “OpENer”.

Target Device

RZ/N2L Group

Contents

1. Overview

This document describes the setup procedure of the sample program for RZ/N2L 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 package.

1.1 Abbreviations / Definitions

Table 1-1 Abbreviations/Definitions

Index	Abbreviations /Definitions	Description
1	IP	Internet Protocol
2	TCP	Transmission Control Protocol
3	USB	Universal Serial Bus
4	PC	Personal Computer
5	SW	Switch
6	EWARM	Embedded Workbench® for ARM
7	lwIP	lightweight IP
8	CIP	Common Industrial Protocol
9	OSS	Open-Source Software

1.2 Reference

Technical information about RZ/N2L is available via Renesas.

Table 1-2 Technical Inputs for RZ/N2L

Index	Technical Inputs
1	r01uh0955ejxxxx-rzn2l.pdf (RZ/N2L User's Manual: Hardware)
2	r01an6434ejxxxx-rzt2-rzn2-fsp-getting-started.pdf (Getting started with Flexible Software Package)
3	r12ut0020edxxxx-rzn2l-som-kit-hw.pdf (RZ/N2L Industrial Network SOM Kit Use's Manual)

1.3 Limitation / Known Issue

None

2. 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/N2L 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
```

The Class Objects implemented in this sample software are as follows. For details, please see Chapter 7 Appendix C Table 7-3 ~ Table 7-9.

Table 2-1 CIP Object Classes supported on this sample software

Object Class #	Object Class Name
0x01	Identity
0x02	Massage Router
0x04	Assembly
0x06	Connection Manager
0xF5	TCP/IP Interface
0xF6	Ethernet Link
0x48	QoS

3. Project Setup

3.1 Requirements

This RZ/N2L lwIP protocol stack project has been developed and tested on these environments using the following boards and tools.

Table 3-1 RZ/N2L Requirements

Item	Vender	Description
Board	Renesas Electronics	RZ/N2L Industrial Network SOM Kit
IDE	IAR Systems	● Embedded Workbench® for ARM Version 9.30.1 Please apply patch (EWARM_Patch_for_RZN2L) which is available in http://www.renesas.com/rzn2l. Regarding how to apply the patch, please read the readme file in patch file.
	Renesas Electronics	● e² studio 2023-04 ● FSP Smart Configurator 2023-04 ● RZ/N2L Flexible Software Package (FSP) v1.2.0 Please download from the link below. https://github.com/renesas/rzn-fsp/releases/tag/v1.2.0
Emulator	IAR Systems	I-jet
	SEGGER	Hardware: J-Link Software: J-Link Commander V7.82f *1
Evaluation Software	CODESYS GmbH	CODESYS v3.5.15.10 32-bit *2

*1: J-Link Commander is used for erasing flash memory.
J-Link Commander is included in “J-Link Software and Documentation Pack” on the following site.
https://www.segger.com/downloads/jlink/

*2: Please use 32bit version, the 3.5.15.10 64-bit version and other versions may not work.

3.2 Hardware Settings

This document describes the major hardware. Refer to RZ/N2L Industrial Network SOM Kit user's manual and schematic for more board details.

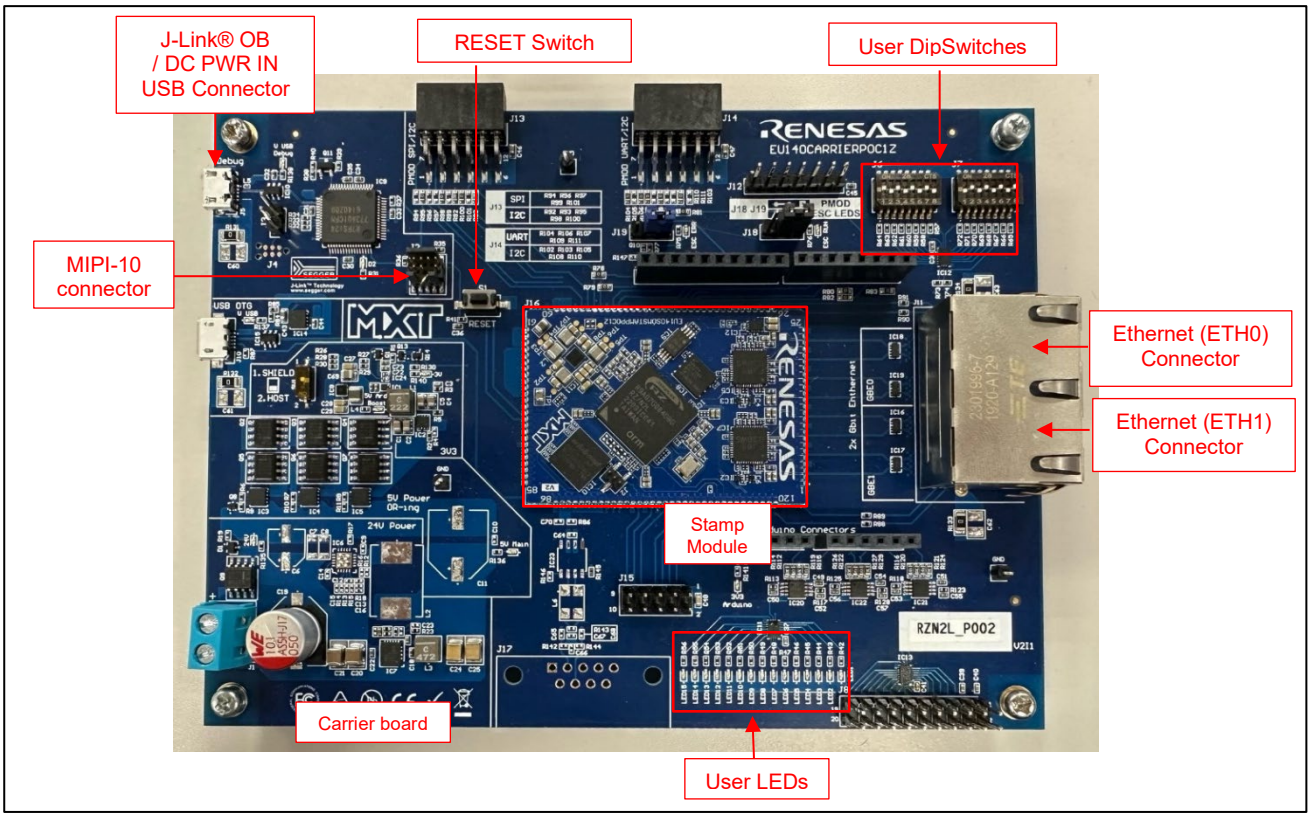


Figure 3-1 RZ/N2L Industrial Network SOM Kit

3.3 Note about Ethernet PHY driver using FSP

This SOM Kit has VSC8531 that is not compatible with FSP as PHY chip. Therefore, we have modified the PHY driver for VSC8531. For details, see "Appendix D: FSP Configuration for VSC8531".

3.4 Setup the Board

Setting the board for running sample program is shown below.

1. Connect the I-jet to J2 or the USB cable to J5 for J-link OB on Carrier board.

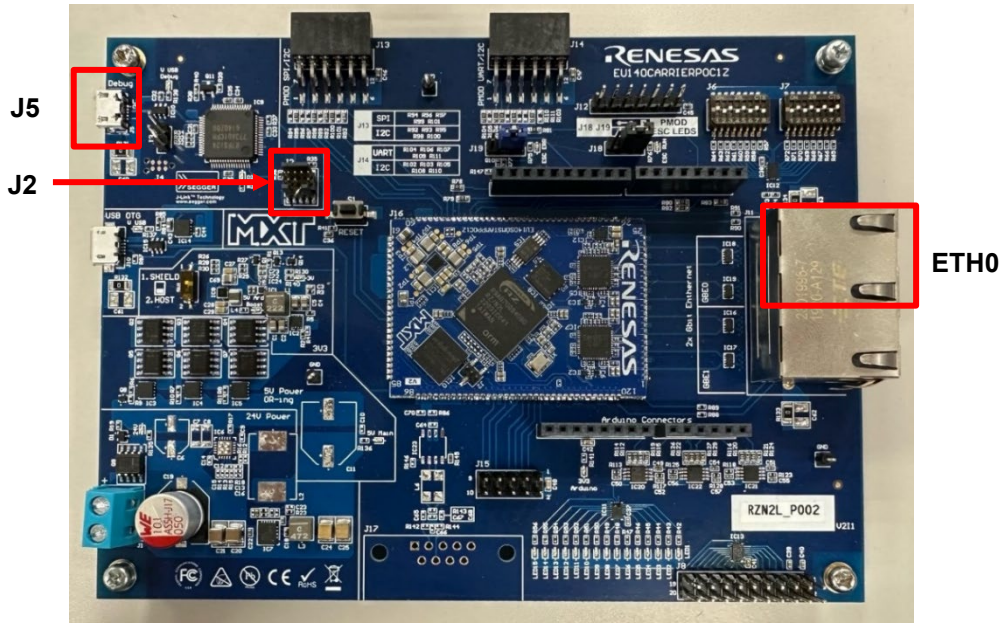


Figure 3-2 Setup the SOM Kit

2. Power is supplied by connecting USB Micro-B cable to the USB connector “J5) of the Carrier board.
3. Connect Ethernet Cable to the Ethernet Connector “ETH0”.

4. Setup the Host Device

4.1 Configuration 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 192.168.1.170 and subnet mask 255.255.255.0 for the device by default. Therefore, for example, set as follows on the PC side.

- IP address: 192.168.1.100
- Subnet mask: 255.255.255.0

4.2 Setup the CODESYS Software

This chapter describes the setup of the CODESYS software.

4.2.1 How to get CODESYS

CODESYS Development system is available from the following web sites. Please get version 3.5.15.10 32-bit*.

*: Please note that the 3.5.15.10 64-bit version or other versions may not work.

- CODESYS Store
  - ➢ To create your account and login is required to download the CODESYS tools.
    - ✧ When you create the 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.
  - ➢ Clicks "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".
  - ➢ Clicks "Versions" tab to get the specified version.

- LINX (Distributer in Japan)
  - ➢ To create your account and login is required to download the CODESYS tools.
    - ✧ Only Japanese companies can create the account.
  - ➢ The latest version only is available.
    - ✧ If you use the latest version, please try updating the CODESYS project included in this package by referring the section Appendix A: How to update CODESYS project.

4.2.2 Startup CODESYS Tools

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

Table 4-1 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.

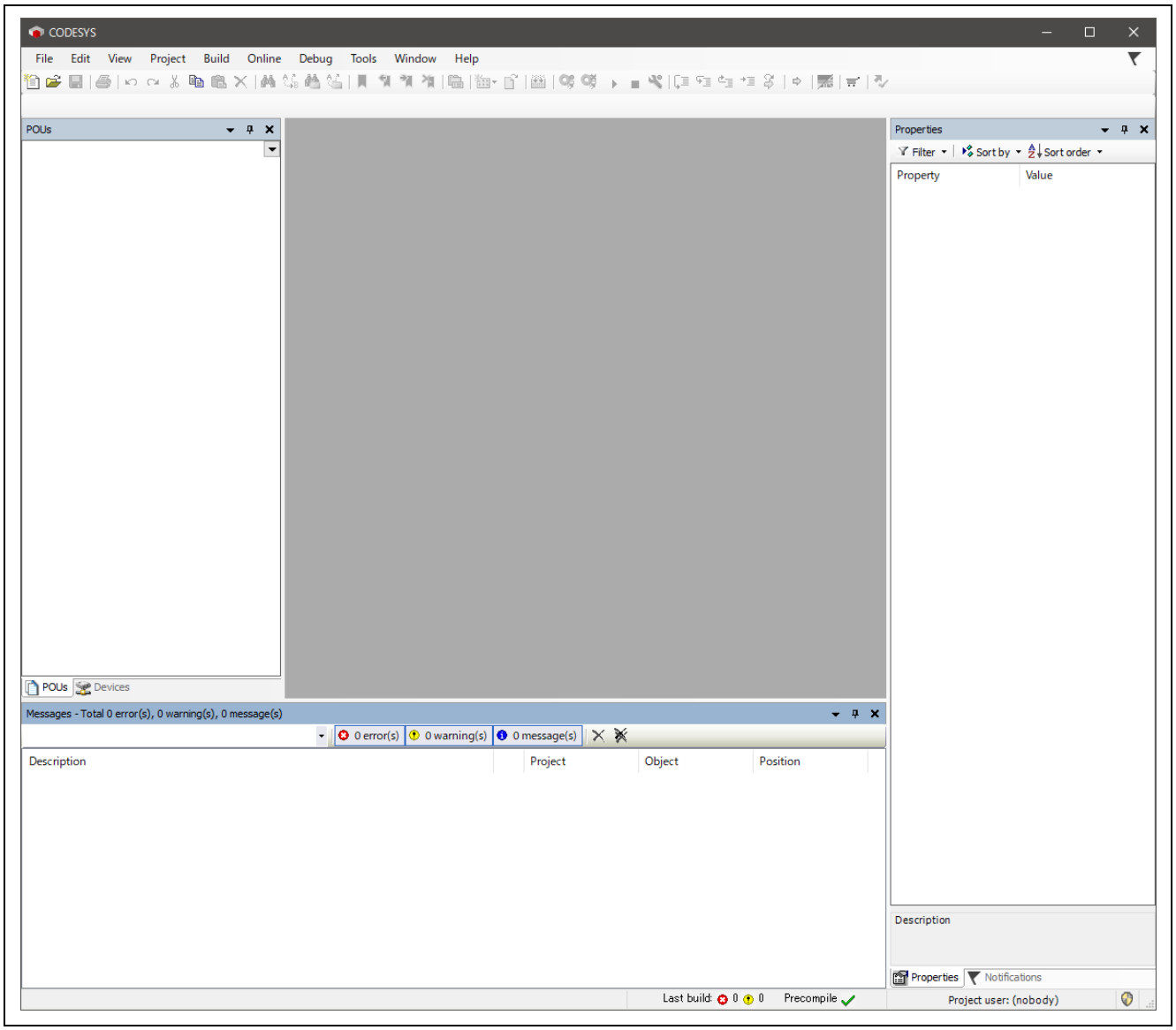


Figure 4-1 CODESYS Initial Window

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 4-2 CODESYS Icons

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

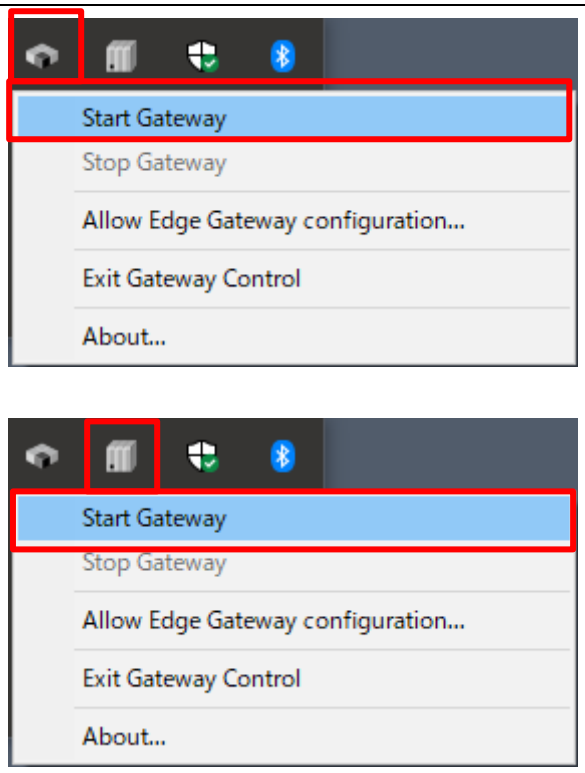


Figure 4-3 Start Gateway and PLC

If the Gateway and PLC is started properly, the icons pigment like the following image.

Figure 4-4 Icons with A and B successfully started

4.2.3 Install EDS File into CODESYS

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

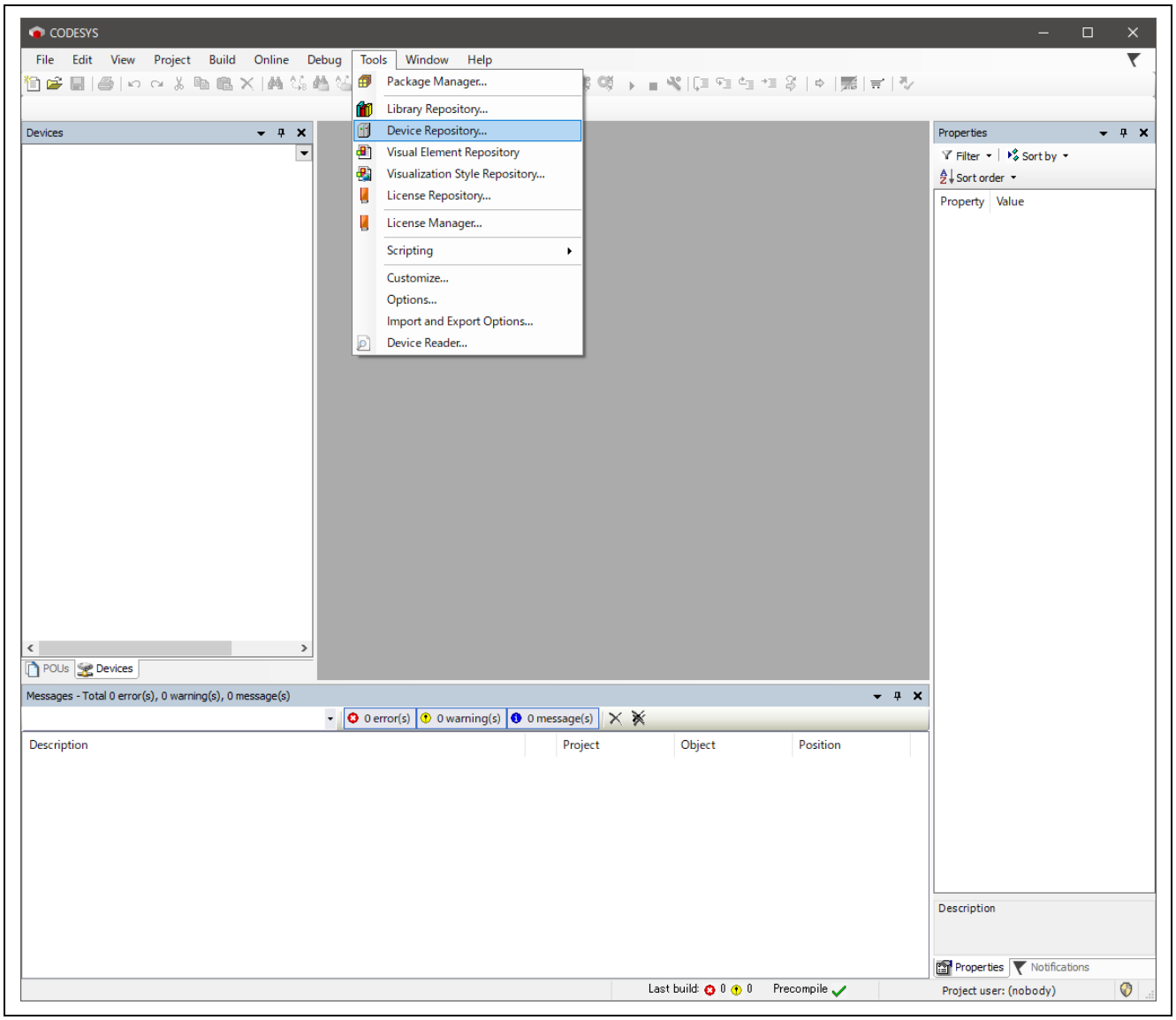


Figure 4-5 Device Repository in CODESYS

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

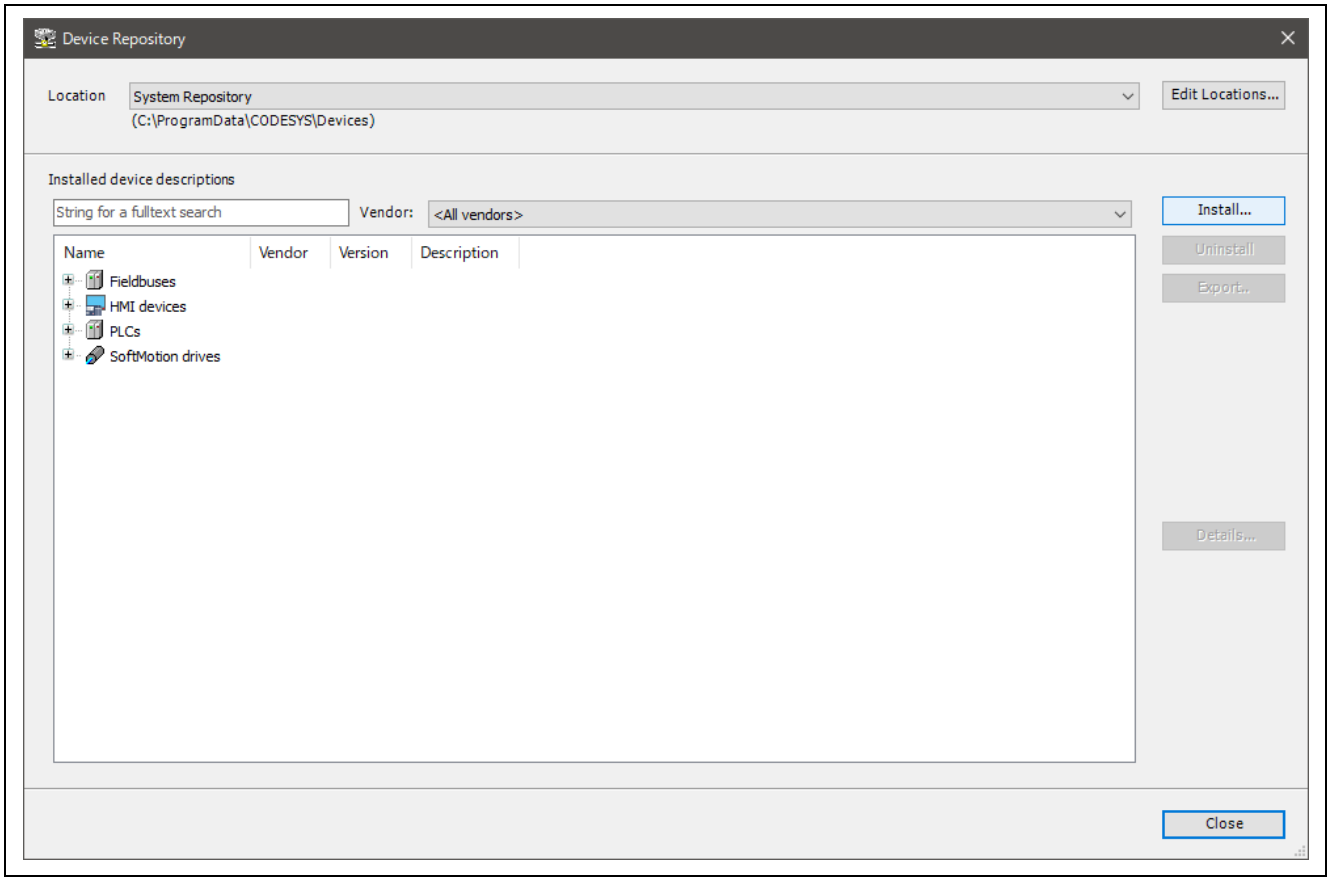


Figure 4-6 Click "Install" in Device Repository Window

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

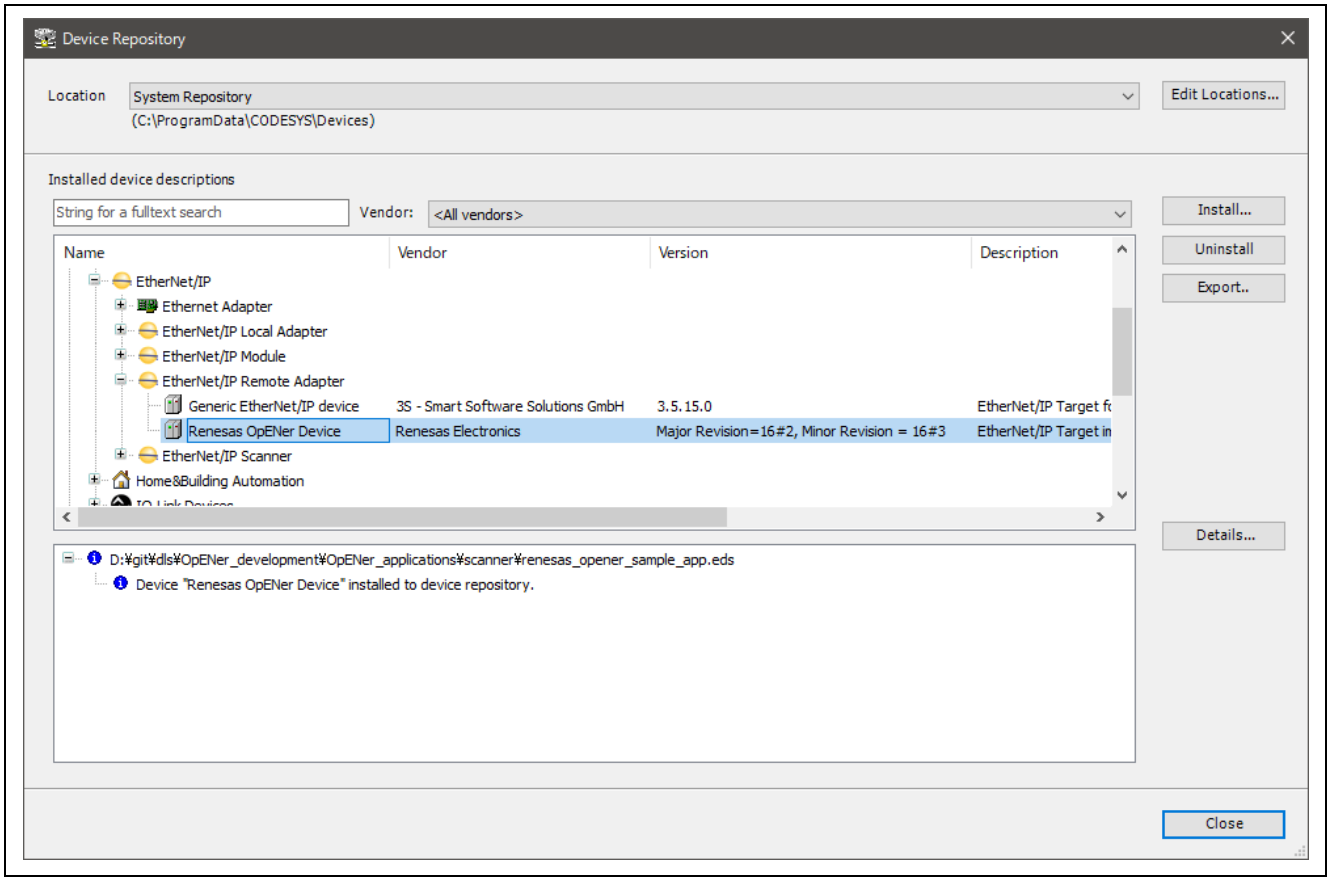


Figure 4-7 EtherNet/IP Remote Adapter

Preparation for using CODESYS is now complete. Continue with the program operation in Chapter 5 and then operate CODESYS again in Chapter 6.

5. Running the Sample Application

Before following this chapter, please look at Section 3.2 and 3.4 for board setup.

The setup differs depending on the IDE.

- When using e² studio, refer to section 5.1.
- When using EWARM, refer to section 5.2.

5.1 Setup sample project for e² studio

Replace the project name in the figure with the project name of this sample project.

5.1.1 Startup e² studio

1. Open the e² studio and select a directory as workspace.
2. Click “Open Projects from File System…” in File tab.

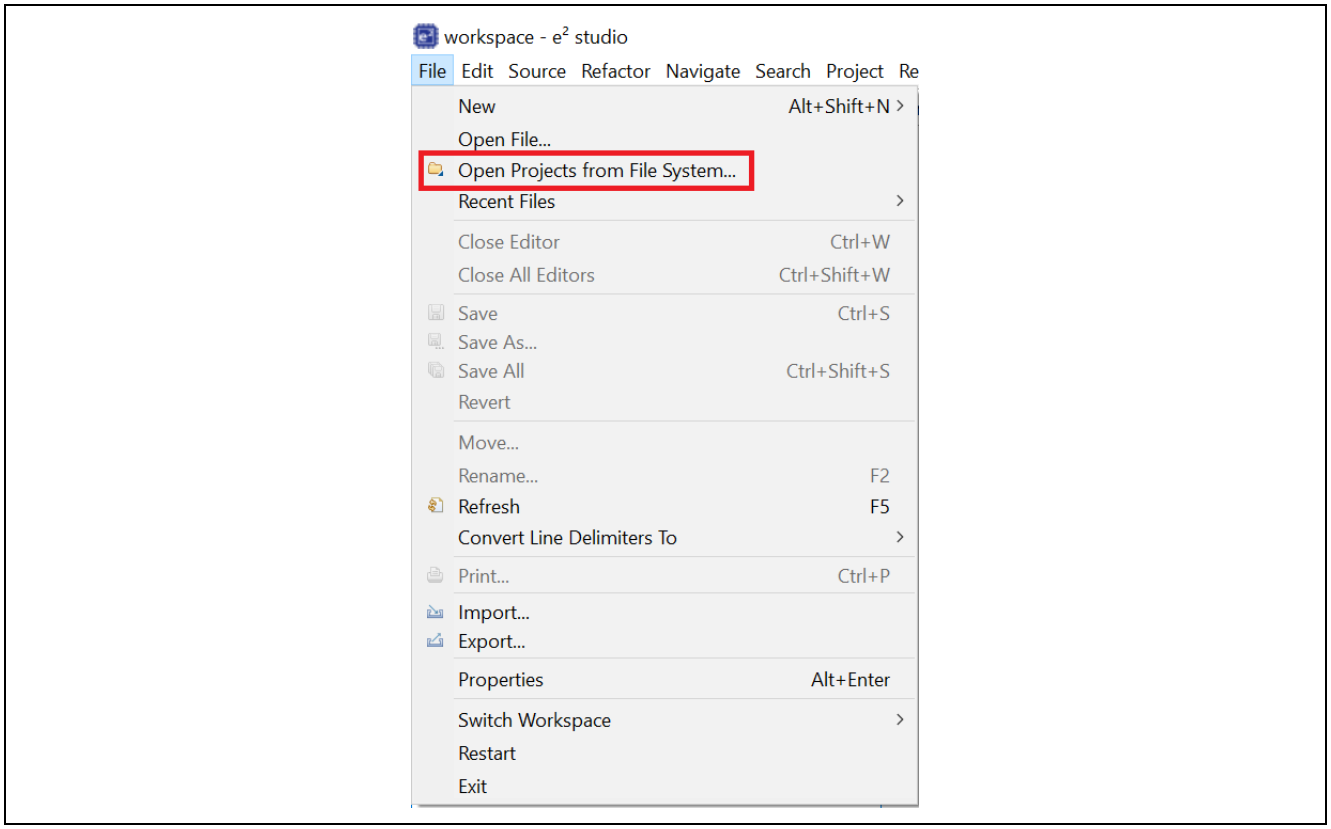


Figure 5-1 e² studio File tab

3. Import the project folder.

Import “\project\rzn2l_som\opener_single\e2studio”.

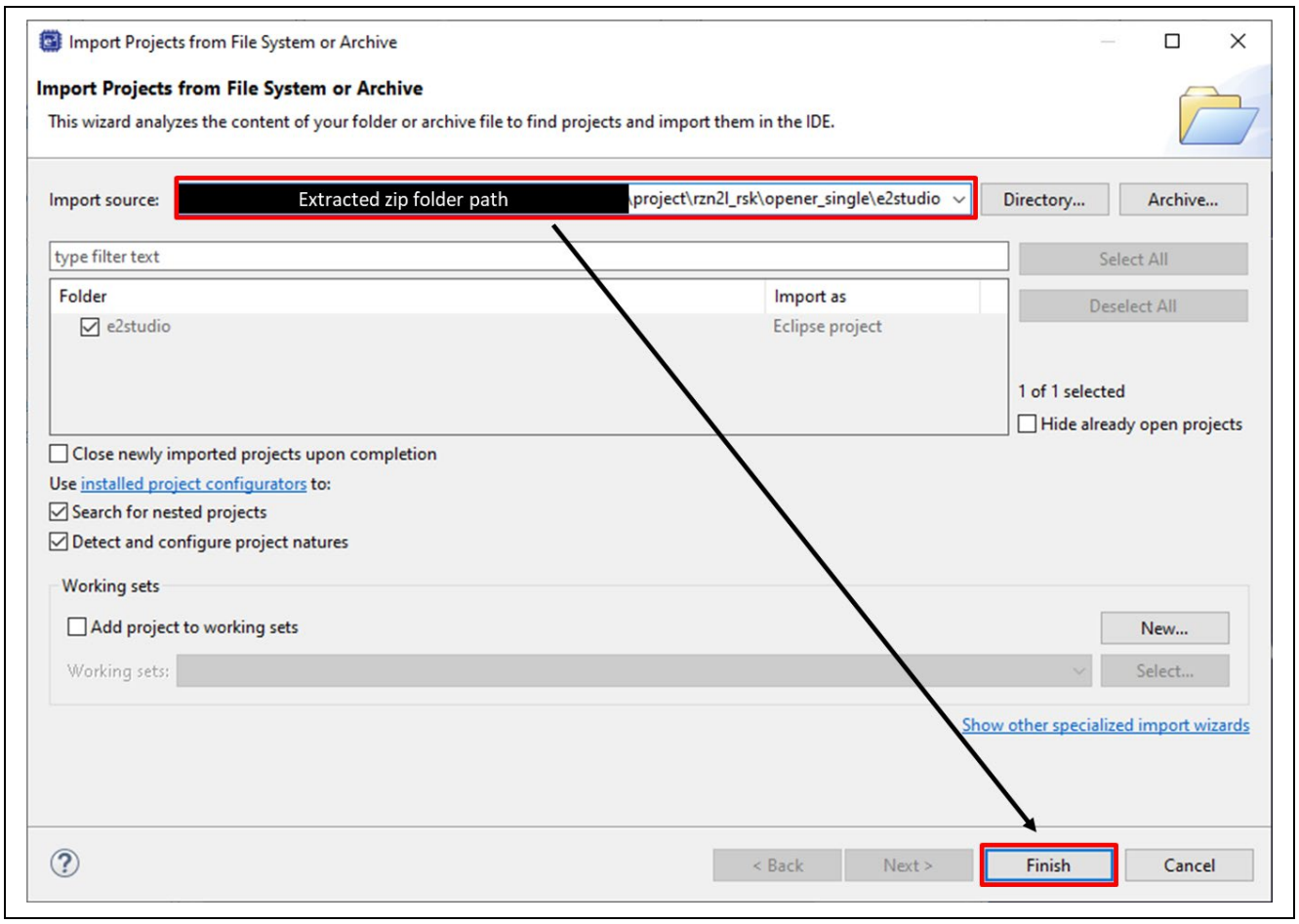


Figure 5-2 Import project on e² studio

5.1.2 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 : 192.168.1.170

Subnet mask : 255.255.255.0

Default gateway : 192.168.1.1

Host Name : OPENER_NETIF0

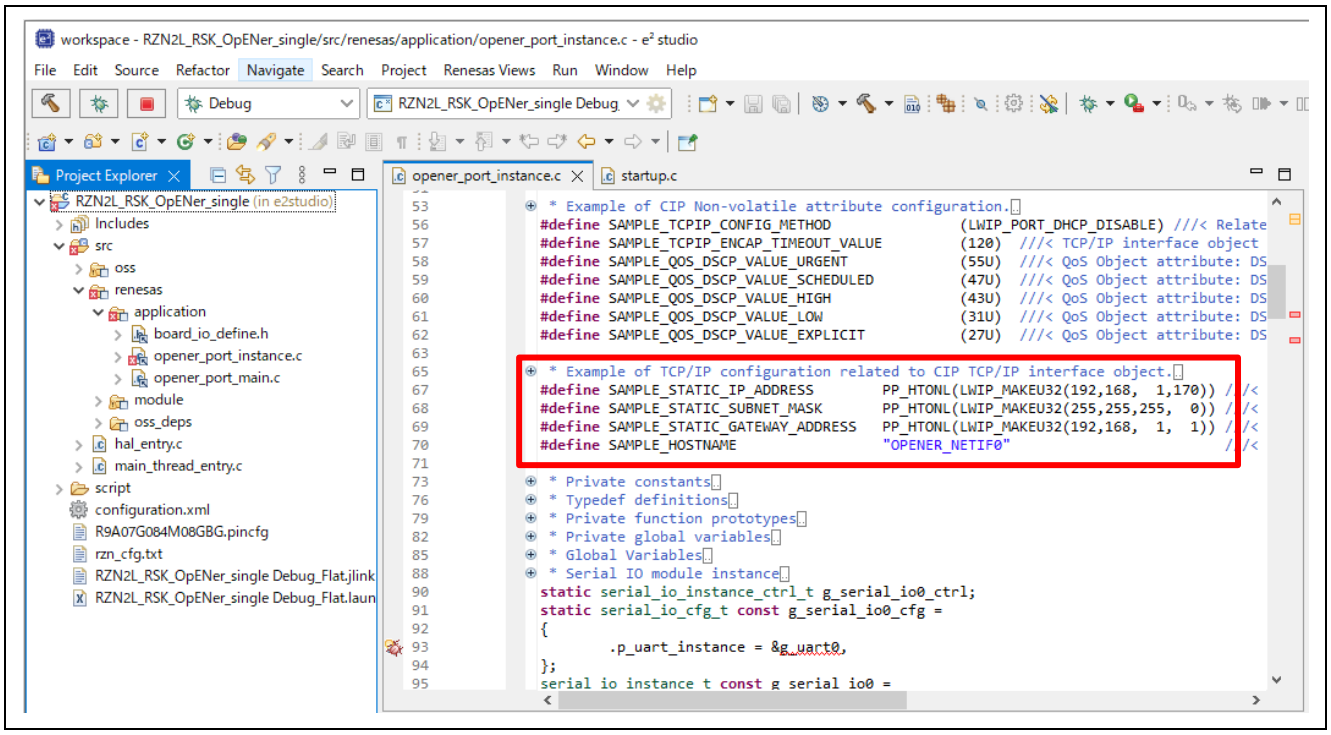


Figure 5-3 Static IP address

5.1.3 How to generate source code and how to build

1. Click the Configuration.xml.

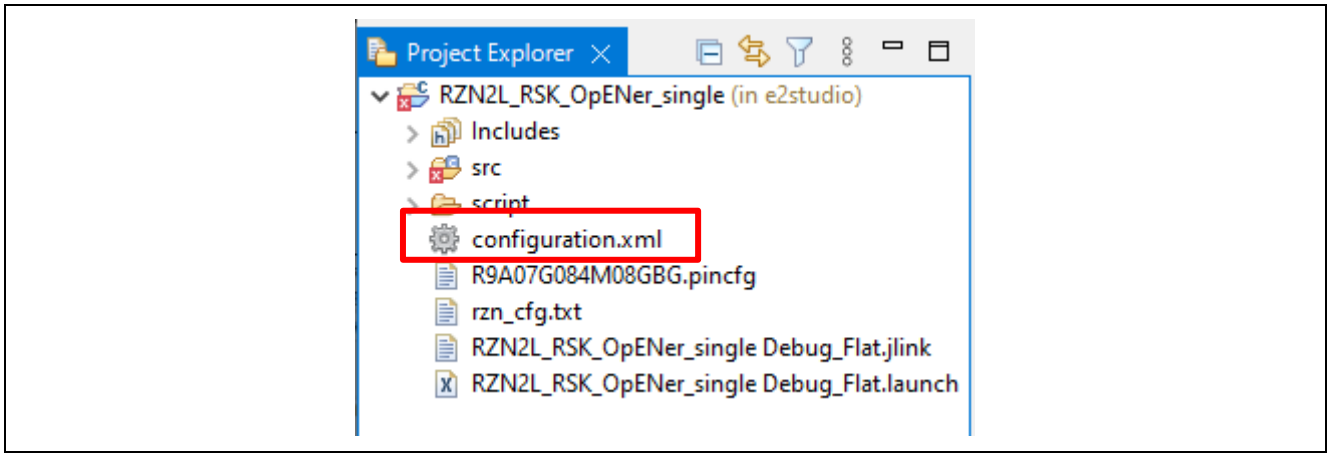


Figure 5-4 Configuration

2. Click ‘Generate Project Content’ button then generate rzn, rzn_gen, rzn_cfg folder.

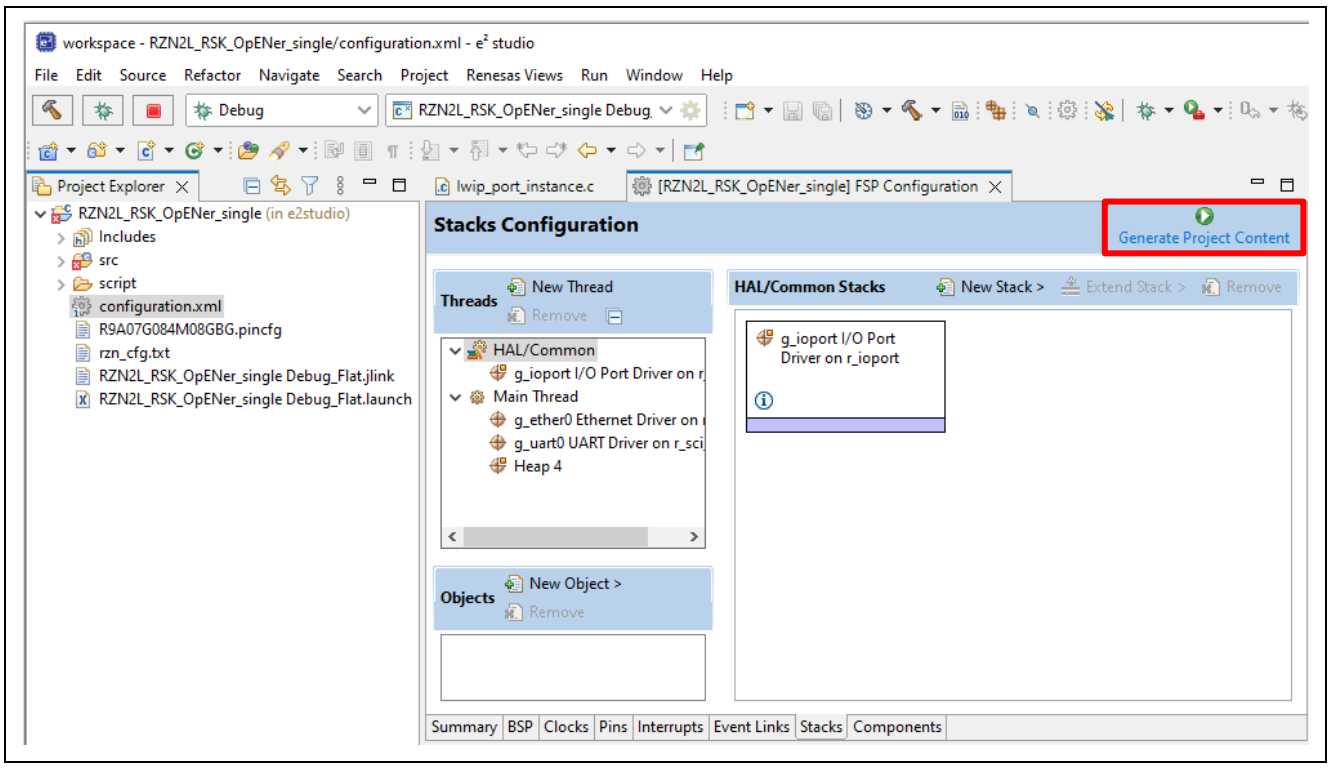


Figure 5-5 Generate Project Content

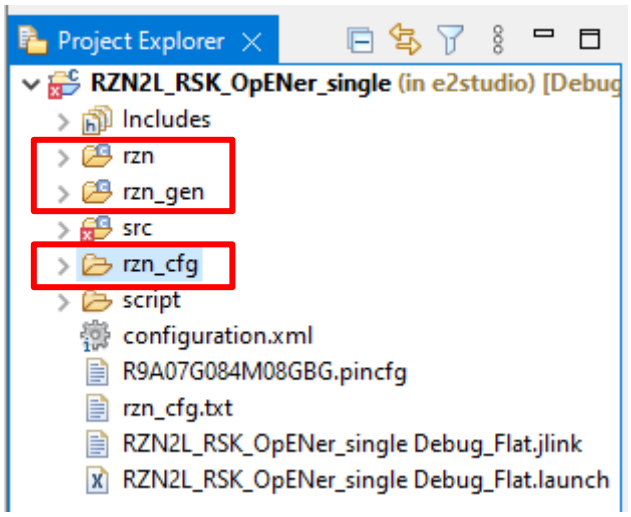


Figure 5-6 Generate project folder

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

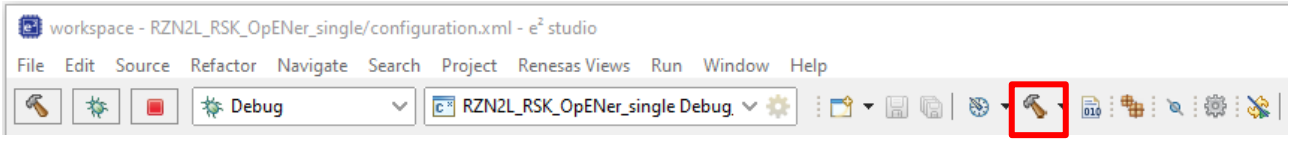


Figure 5-7 Build button

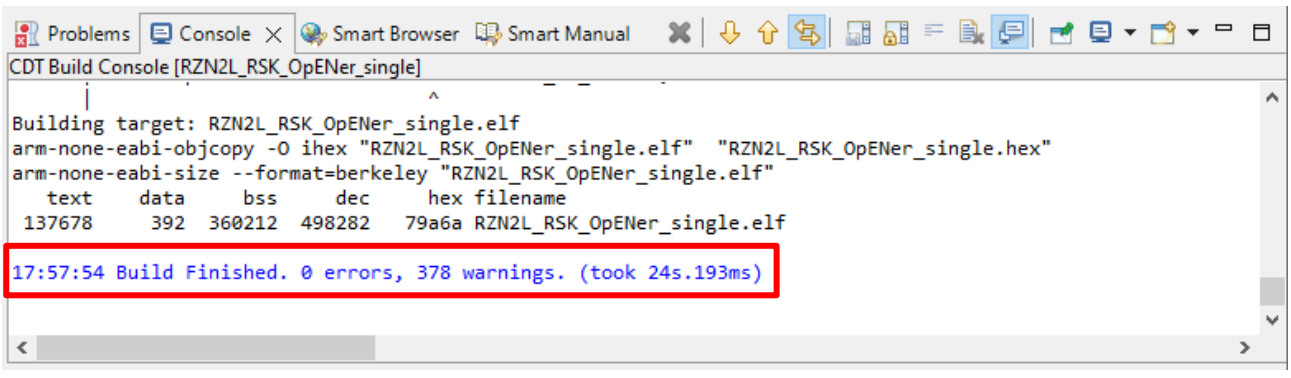


Figure 5-8 Build message

5.1.4 Download application and run debugger

1. First, erase the flash memory by following the steps below. This step can be skipped after erasing the flash memory.

Open the J-Link Commander.

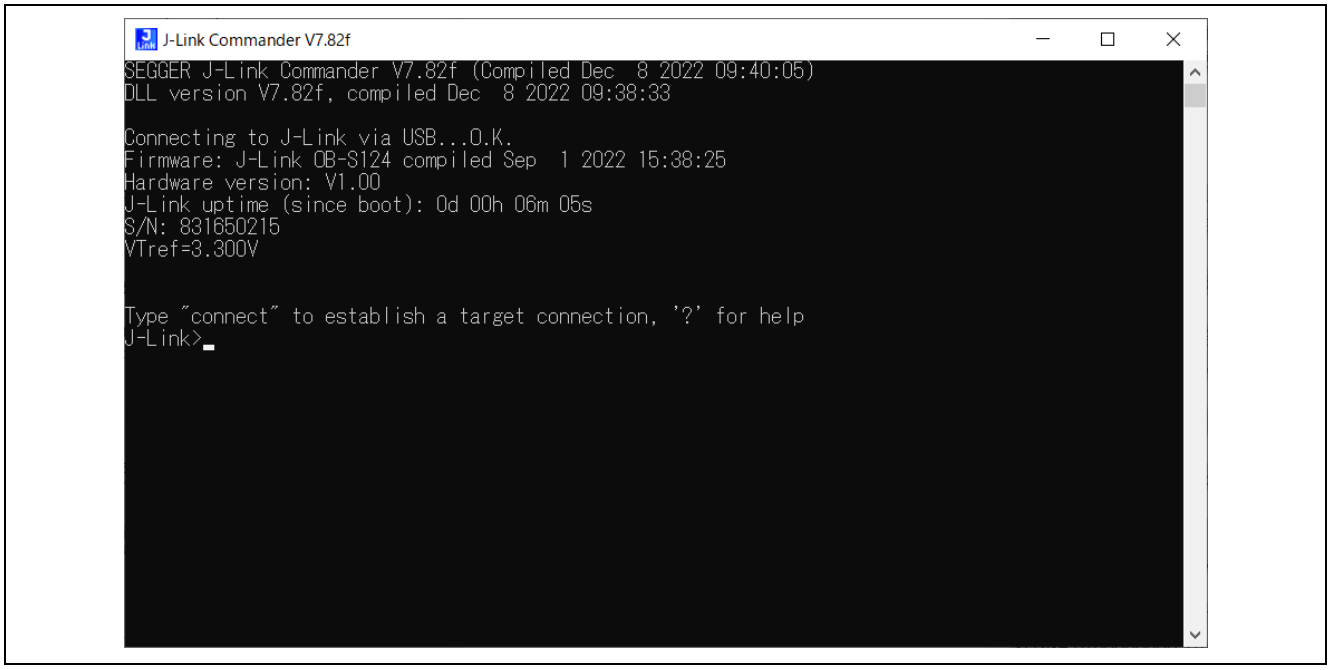


Figure 5-9 Open J-Link Commander

First, type “connect” to establish a target connection and press enter.

Next, specify the connection conditions as follows.

- Device> (Default = press enter)
- TIF>S
- Speed> (Default = press enter)

After that, confirm the message “Cortex-R52 identified.” Is displayed.

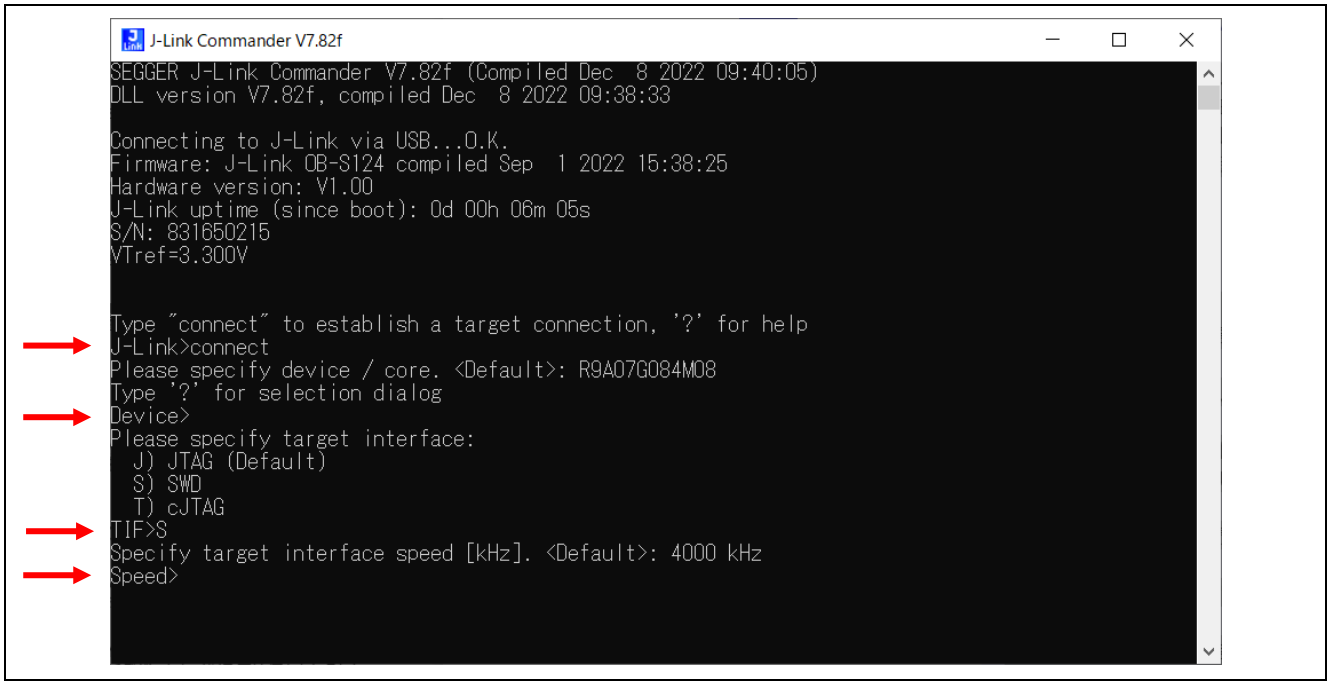


Figure 5-10 Connection conditions (1/2)

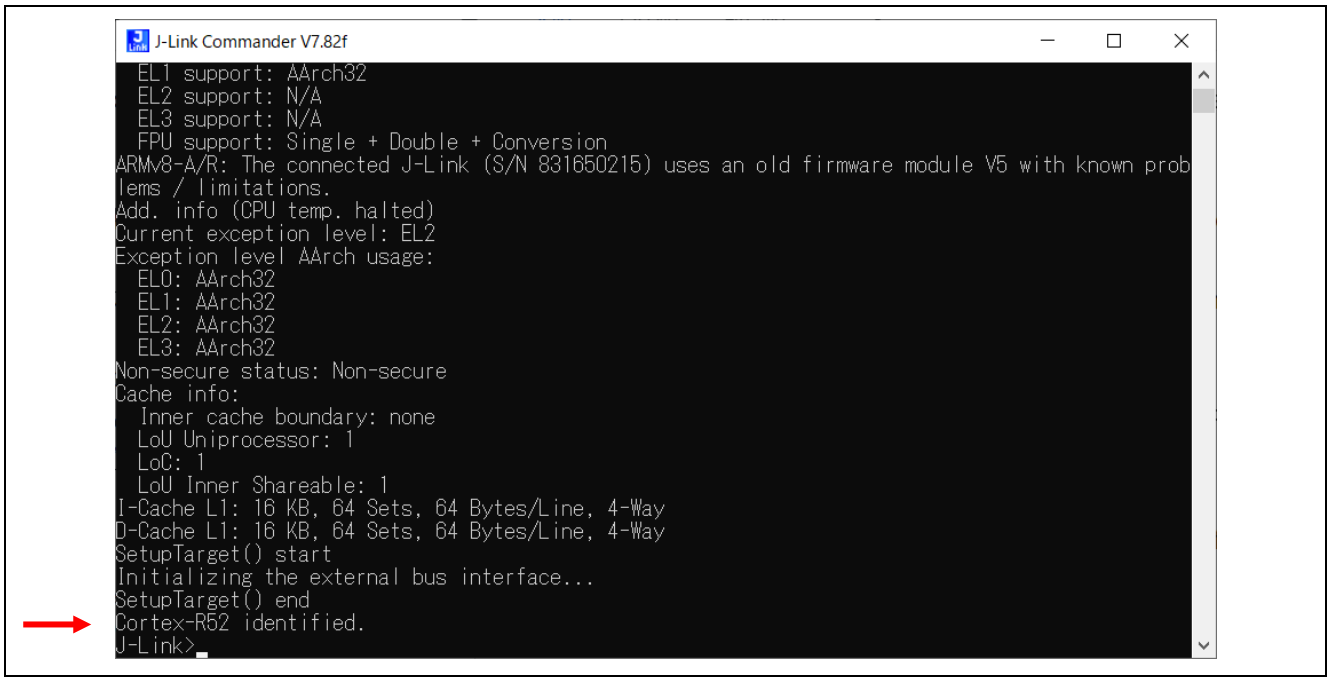


Figure 5-11 Connection conditions (2/2)

Use the commands below to enable flash erase and erase the flash memory.

- >exec EnableEraseAllFlashBanks
- >Erase 0x60000000, 0x60100000

After that, confirm the message “Erasing done.” Is displayed.

Enter “q” to exit J-Link Commander.

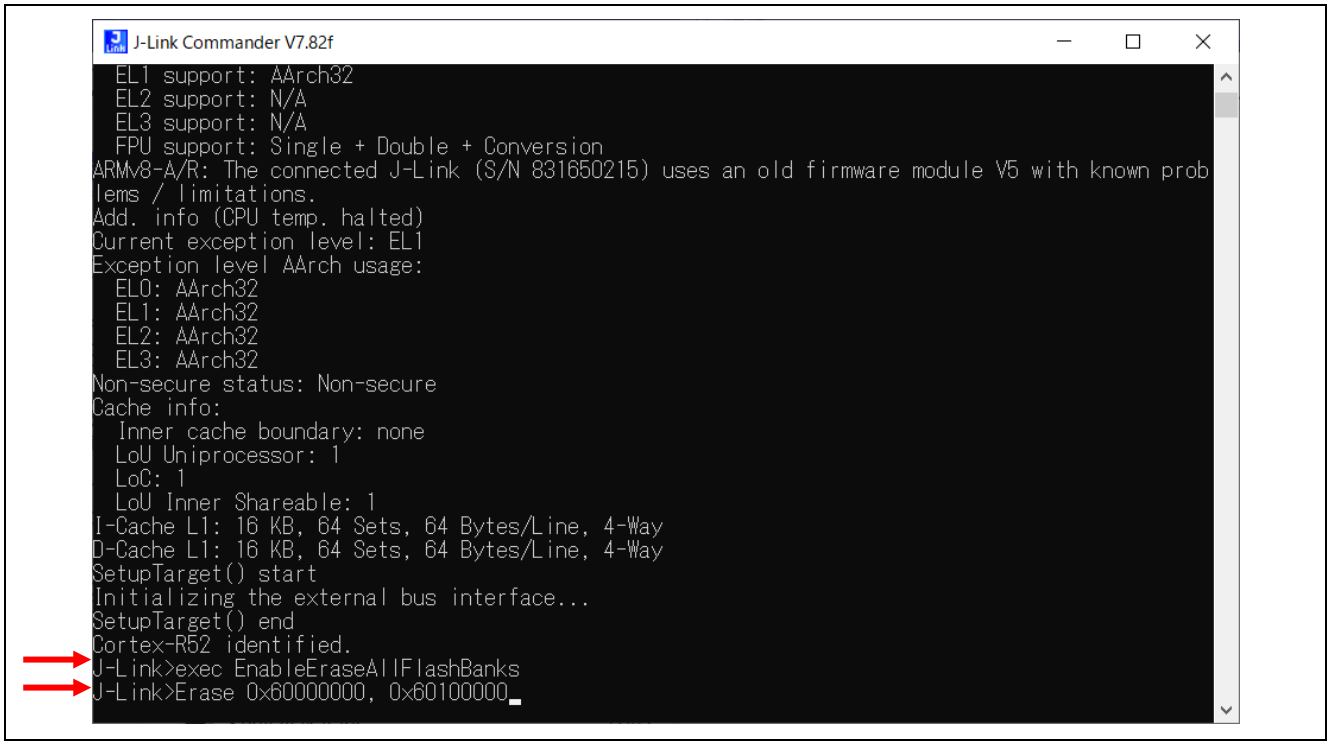


Figure 5-12 Erase flash memory (1/2)

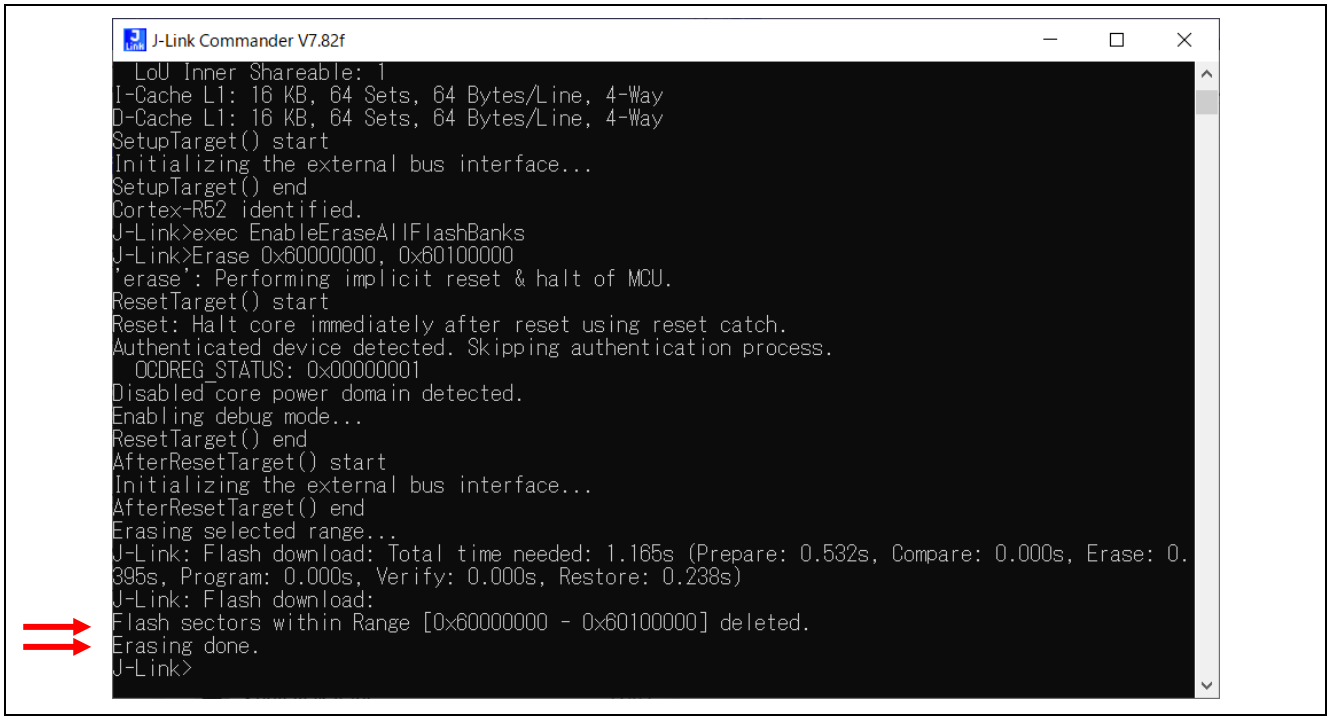


Figure 5-13 Erase flash memory (2/2)

2. Return to e2 studio. Click the Debug button in tool bar to download the built application program and launch the debugger.

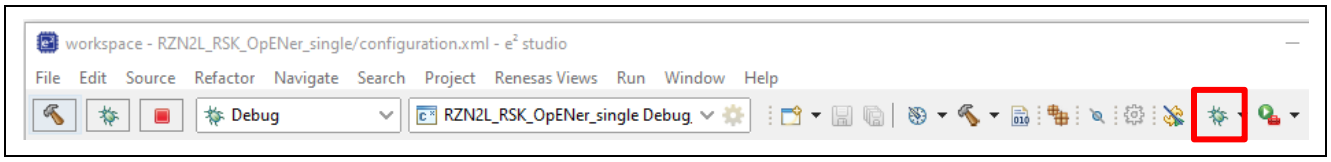


Figure 5-14 Debug button

3. Click to “Switch” button.

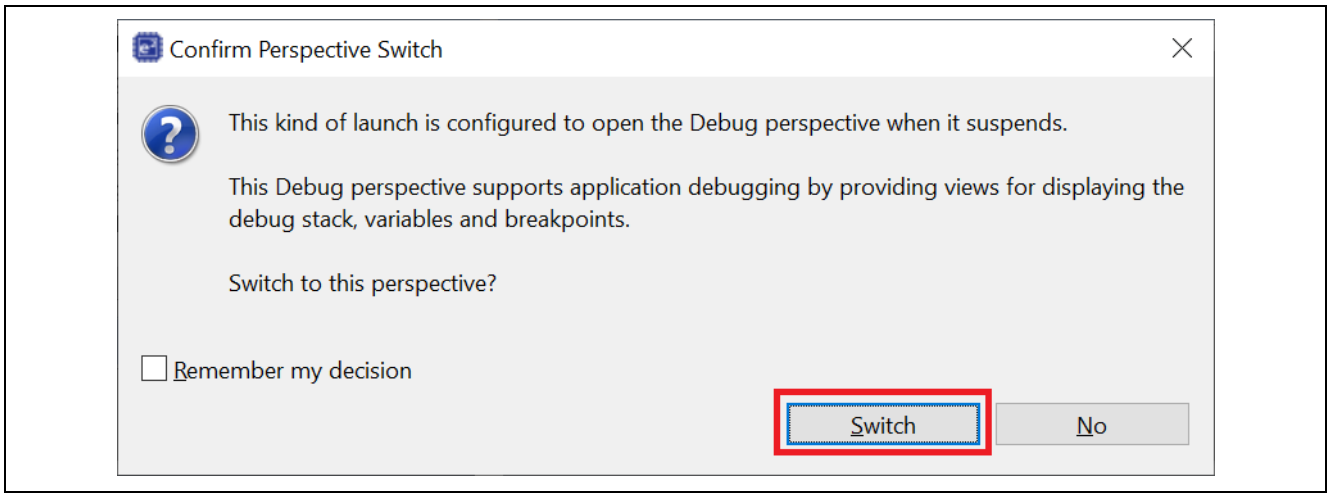


Figure 5-15 Confirm Perspective Switch

4. The program will break at “system_init” for startup.

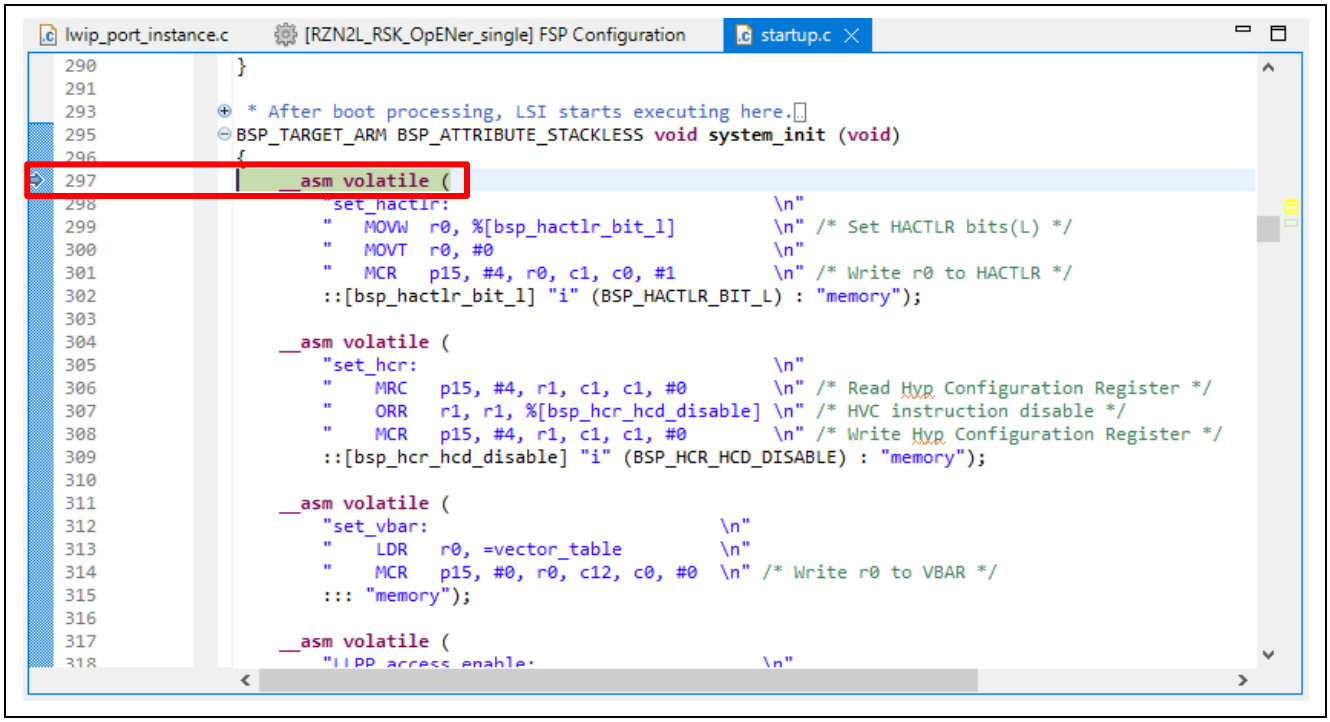


Figure 5-16 Break point 1

5. Before running the loaded program, please change the CPSR register of CR52 general register on Registers tabs.

- Change the “T” register bit (bit 5 in CPSR register), which is Thumb execution state bit, from “1” to “0” to switch the instruction mode from “Thumb” to “Arm”.

  ➢ When the register value is “0x000001**f**a”, set it to “0x000001**d**a”.

If the value of “T” bit in CPSR register is not changed, please note that the program does not work properly when running.

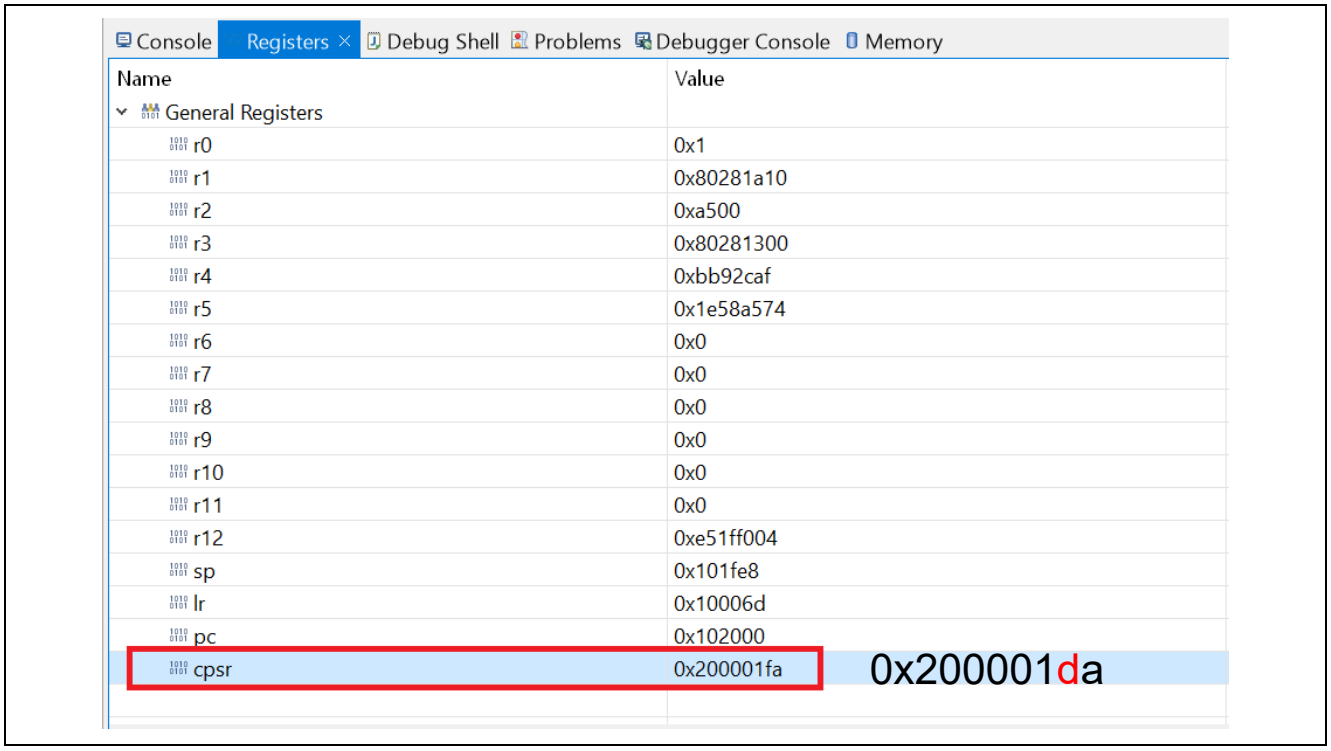


Figure 5-17 CPSR register of CR52 generic register on Registers tab

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

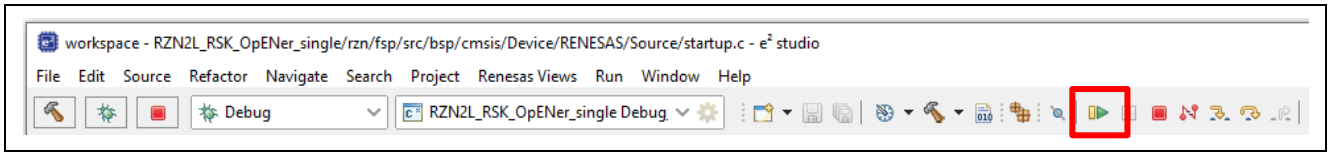


Figure 5-18 Resume button

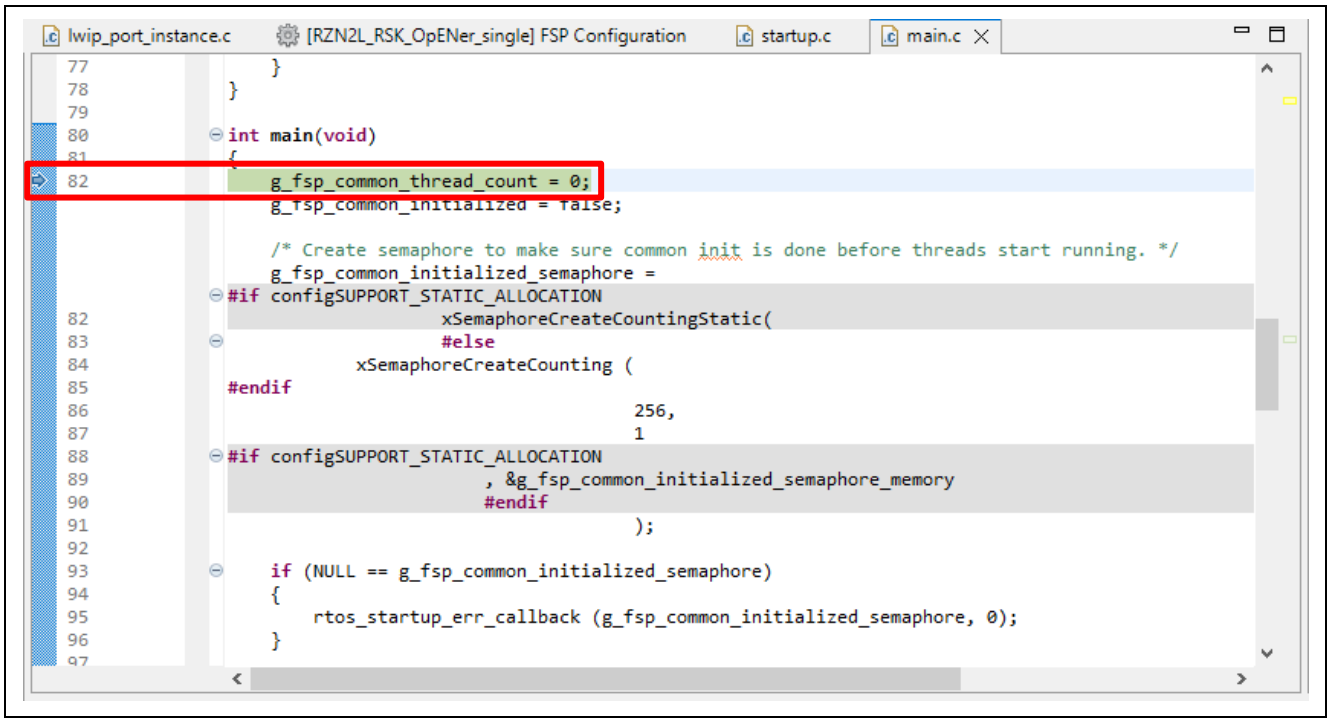


Figure 5-19 Break point 2

7. Click the Resume button again to execute the program.

5.2 Setup sample project for EWARM

Replace the project name in the figure with the project name of this sample project.

5.2.1 Startup EWARM

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

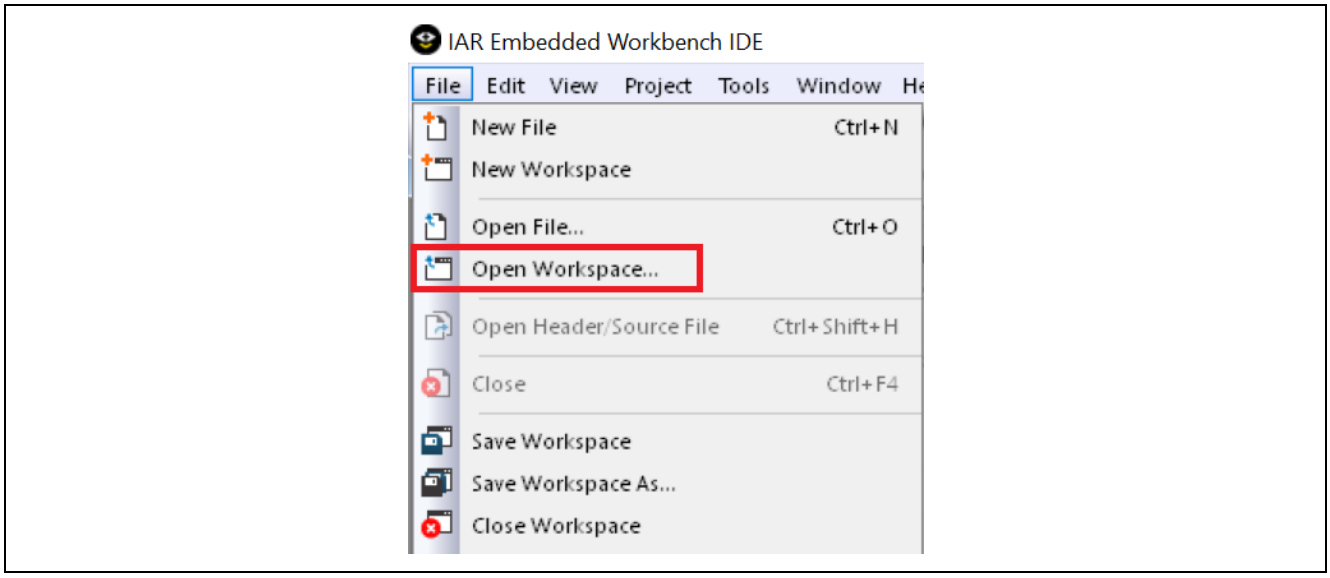


Figure 5-20 EWARM file tab

3. Select the Workspace File(.eww) and click the Open button.
   “\project\rzn2l_som\opener_single\ewarm\RZN2L_SOM_EIP_OpENer.eww”

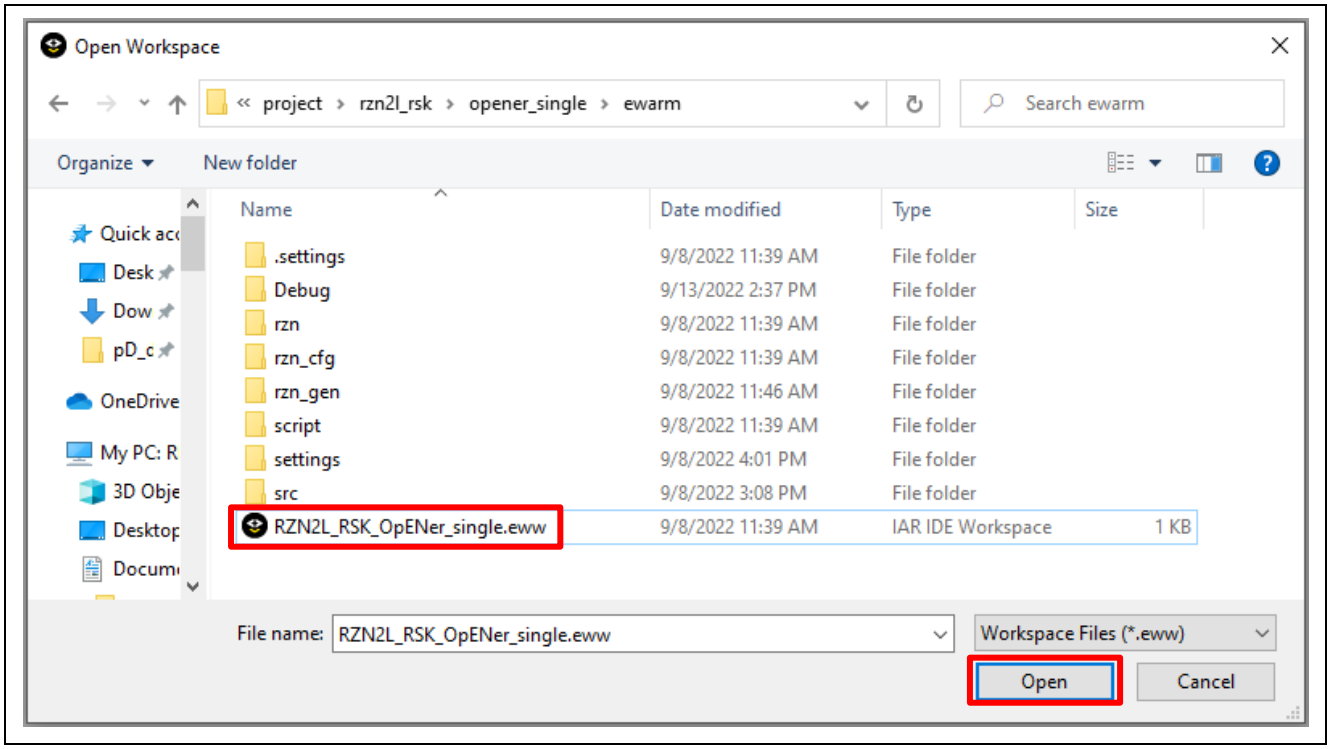


Figure 5-21 Open project file

5.2.2 Board IP Address Setting

The IP address is set in \common\renesas\application\opener_port_instance.c.

The following addresses are used in the example:

IP address : 192.168.1.170

Subnet mask : 255.255.255.0

Default gateway : 192.168.1.1

Host Name : OPENER_NETIF0

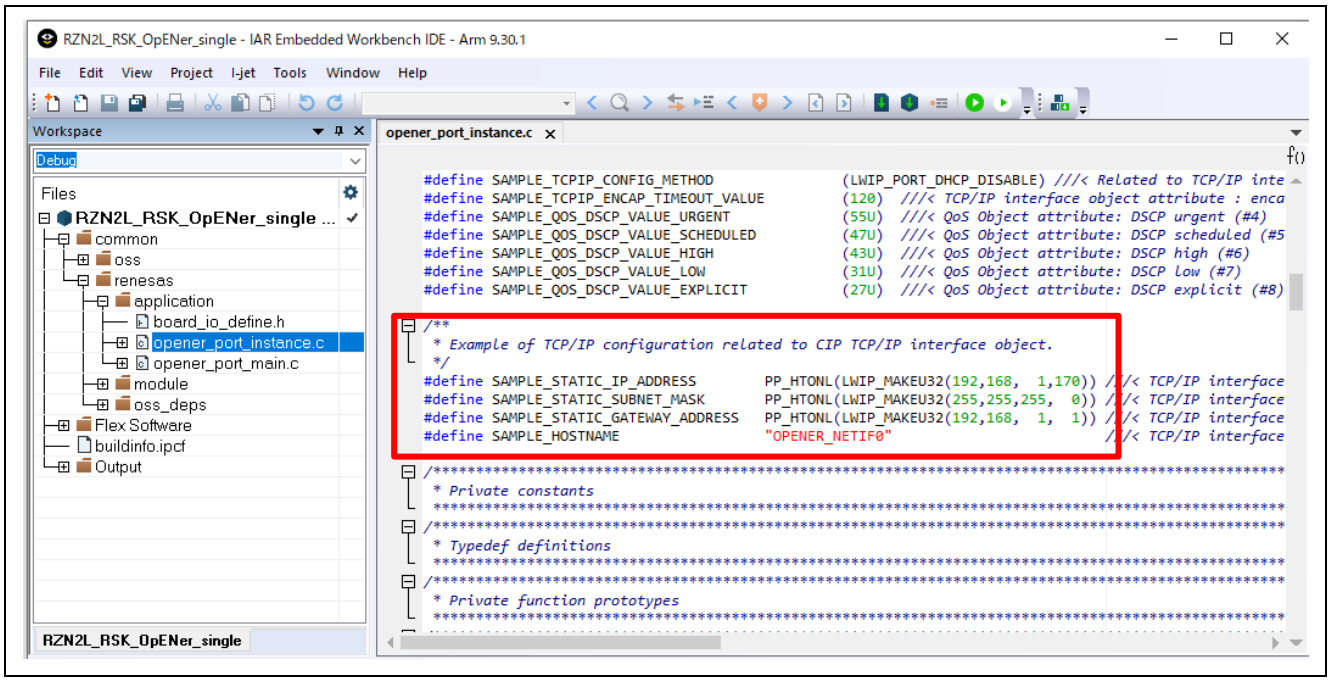


Figure 5-22 Static IP address

5.2.3 How to generate source code and how to build

1. Click the “Tool” -> “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 up it.

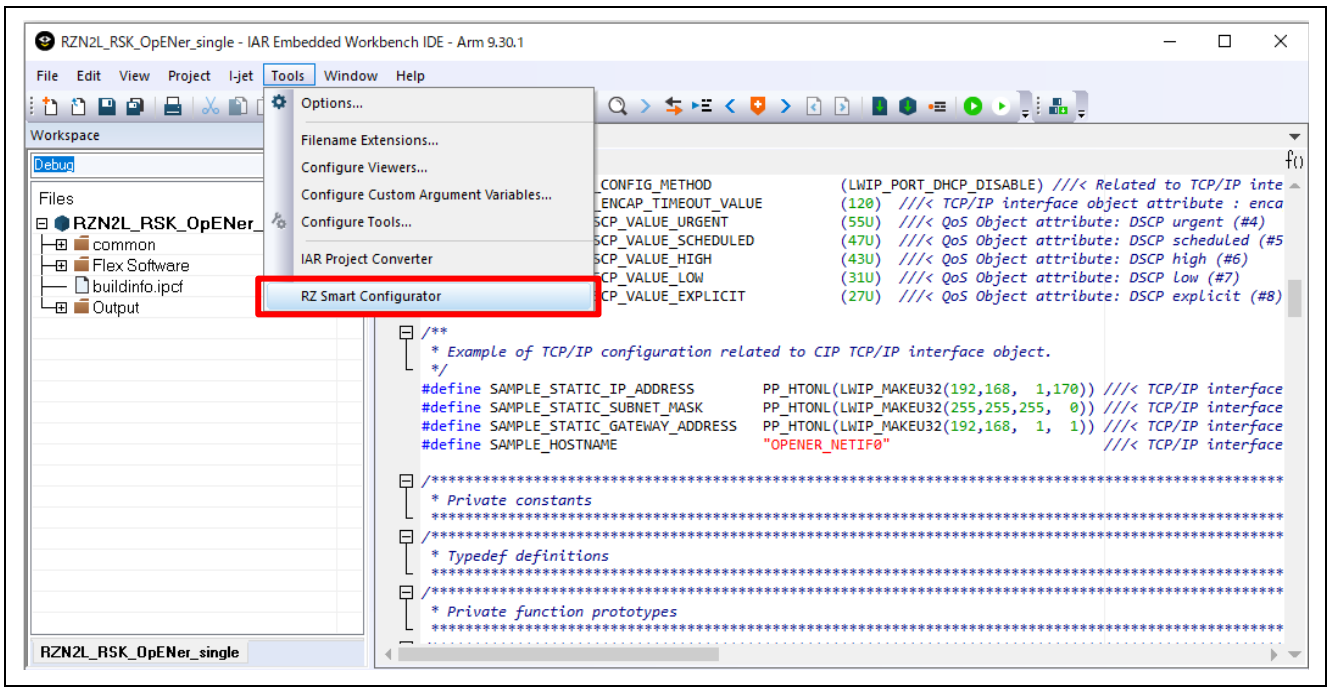


Figure 5-23 Tools tab

2. Click ‘Generate Project Content’ button then will be generate rzn, rzn_gen, rzn_cfg folder.

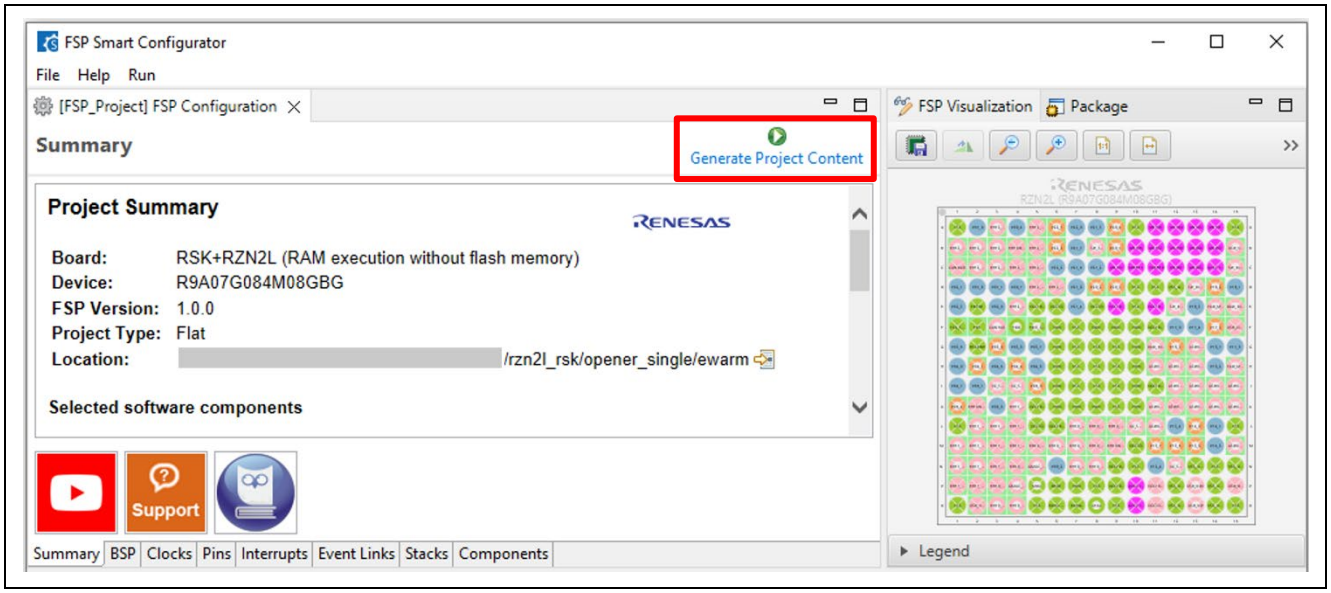


Figure 5-24 FSP SC Smart Configurator

3. Click the Make button on 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.

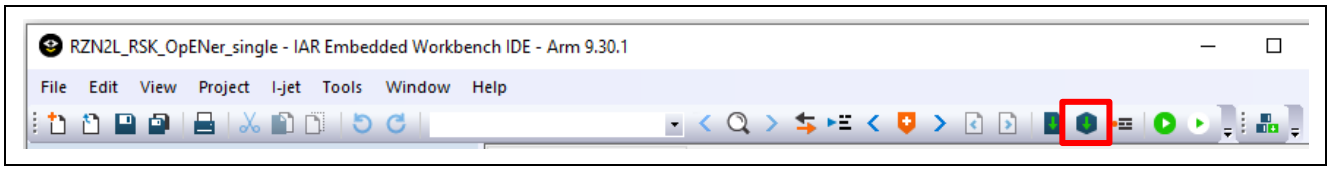


Figure 5-25 Make button 1

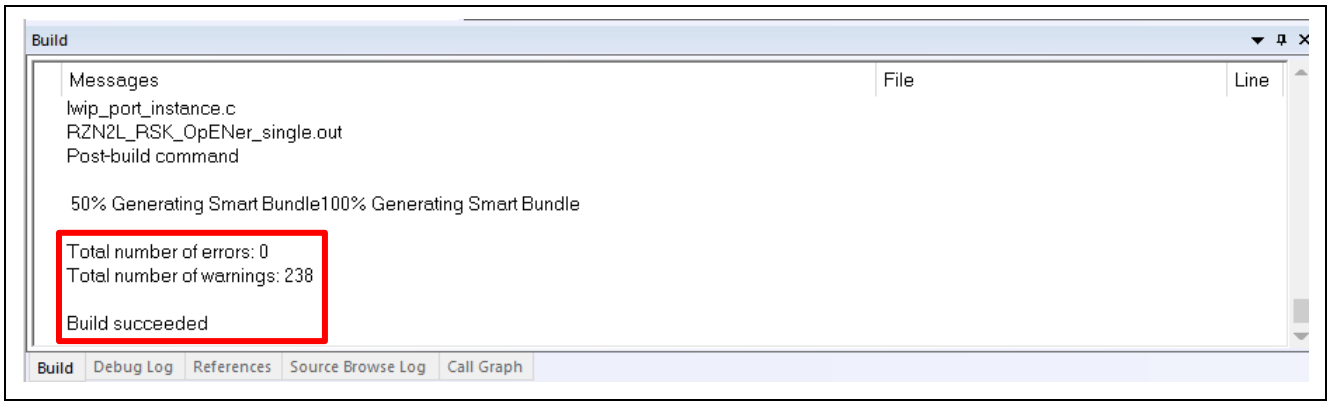


Figure 5-26 Make button 2 and Build console

5.2.4 Download application and run debugger

1. 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.

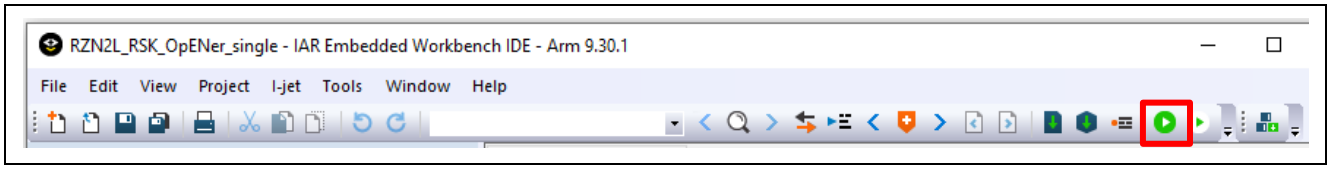


Figure 5-27 Debug button

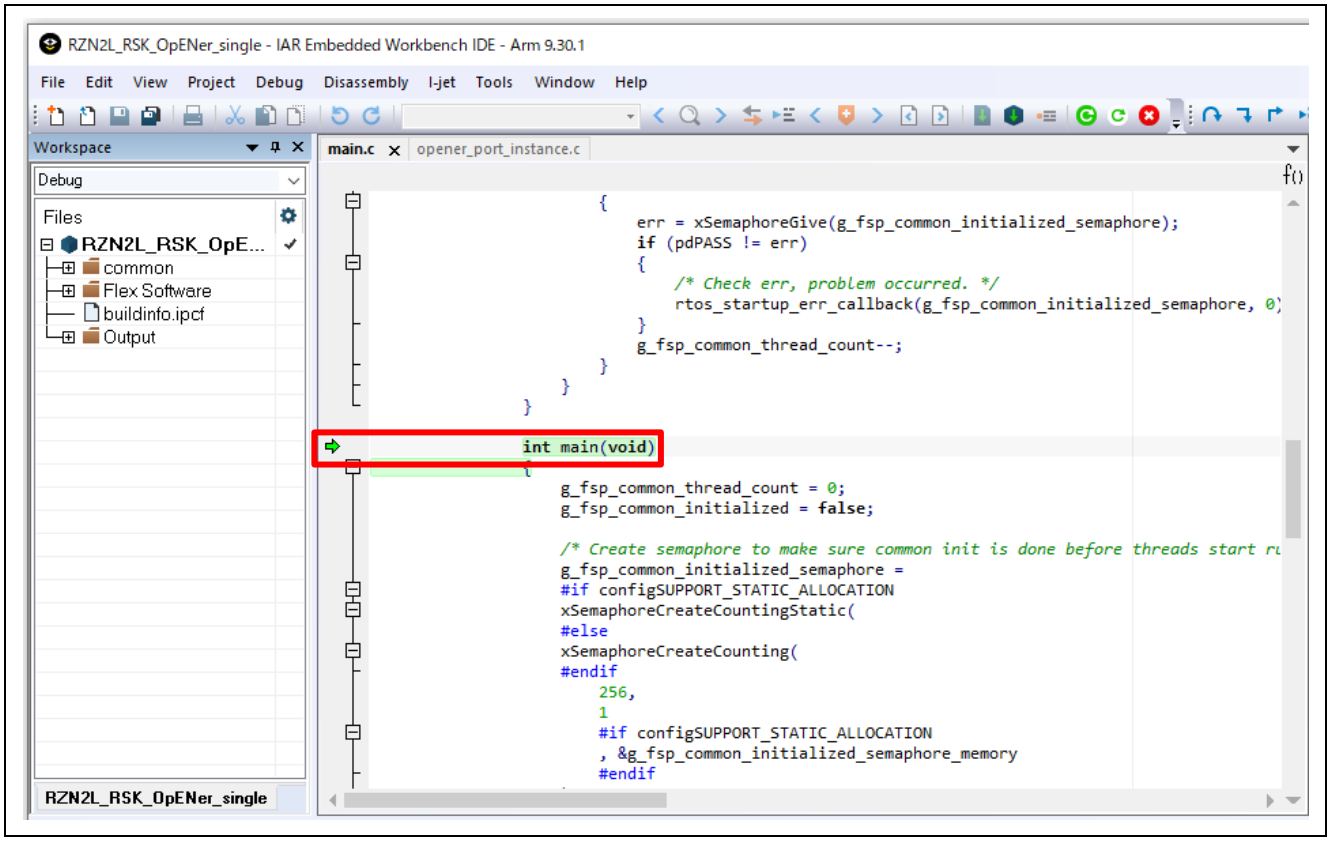


Figure 5-28 Break point

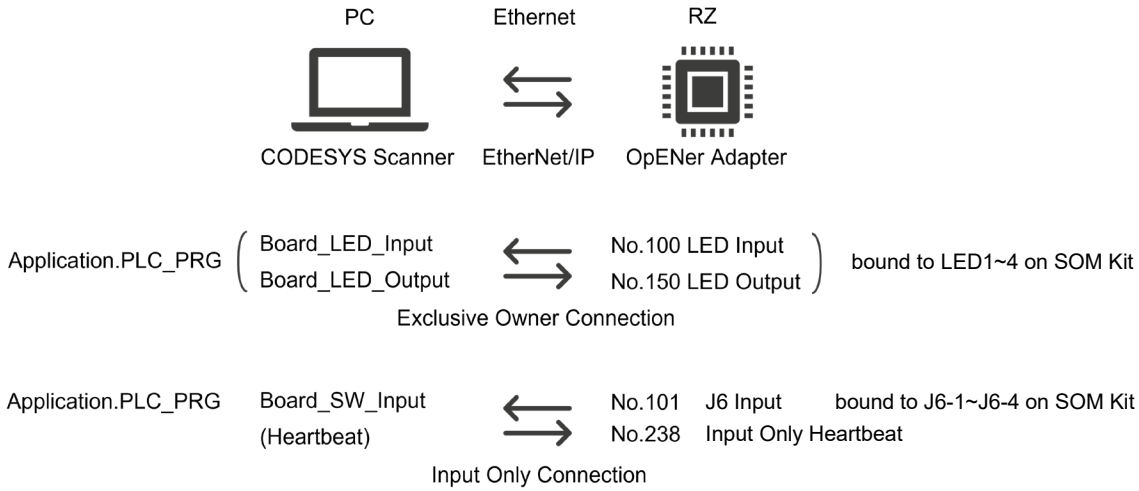
2. Click the Go button.

Figure 5-29 Go button

6. Demonstration of the application with the CODESYS

6.1 Application Behavior

For demonstration, the application of OpENer RZ/N2L port has an Exclusive Connection and an Input Only Connection. The connections between the RZ/N2L application and the CODESYS project application is 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 of RZ/N2L Industrial Network SOM Kit.
  - The PLC program make the LEDs light by shifting every CyclePerTick .
- The Input Only Connection is bound to J6-1 ~ J6-4 of RZ/N2L Industrial Network SOM Kit.
  - The PLC program read the SW values and change the incrementation value of CycleCounter for controlling the frequency of LED light shifting.

CODESYS application can be shown page.

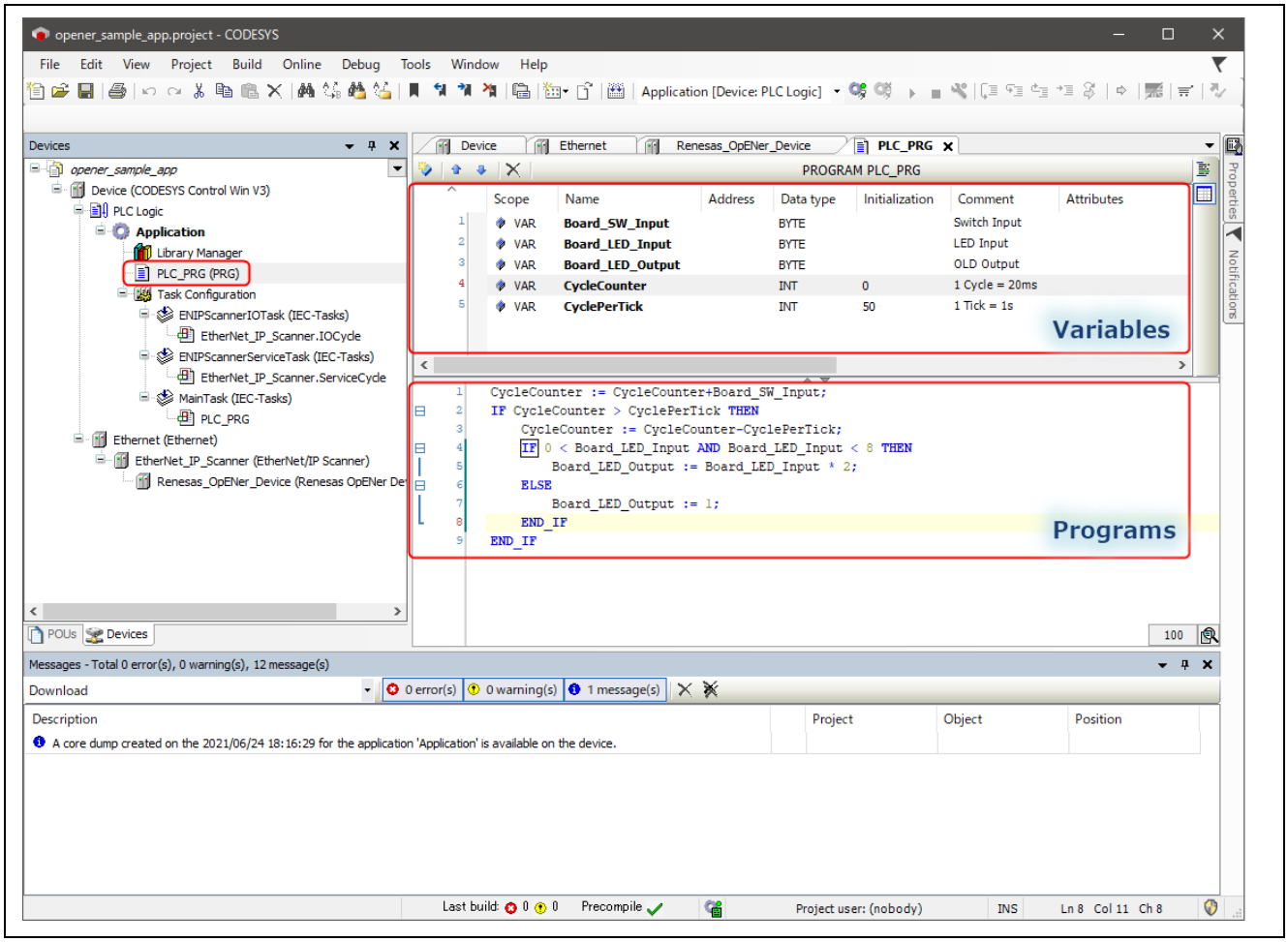


Figure 6-1 CODESYS Project included in this sample program

6.2 IP and MAC Address Configuration

6.2.1 IP Configuration

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

If changing the IP configuration, please change the following parameters.

Table 6-1 IP Configuration

Item	Value	Variable
IP Address	192.168.1.170	g_lwip_port0_netif_cfg.ip_address
Net Mask	255.255.255.0	g_lwip_port0_netif_cfg.subnet_mask
Gateway Address	192.168.1.1	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.

6.2.2 MAC Address Configuration

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

Table 6-2 MAC Address Configuration

Item	Value (Decimal)
MAC Address	00:11:22:33:44:55

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

Regarding FSP configuration, please see the "RZ/T2M, RZ/N2L Getting Started with Flexible Software Package" (r01an6434ejxxxx-rzt2-rzn2-fsp-getting-started.pdf) document.

6.3 Startup Software PLC

6.3.1 Open CODESYS project

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

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

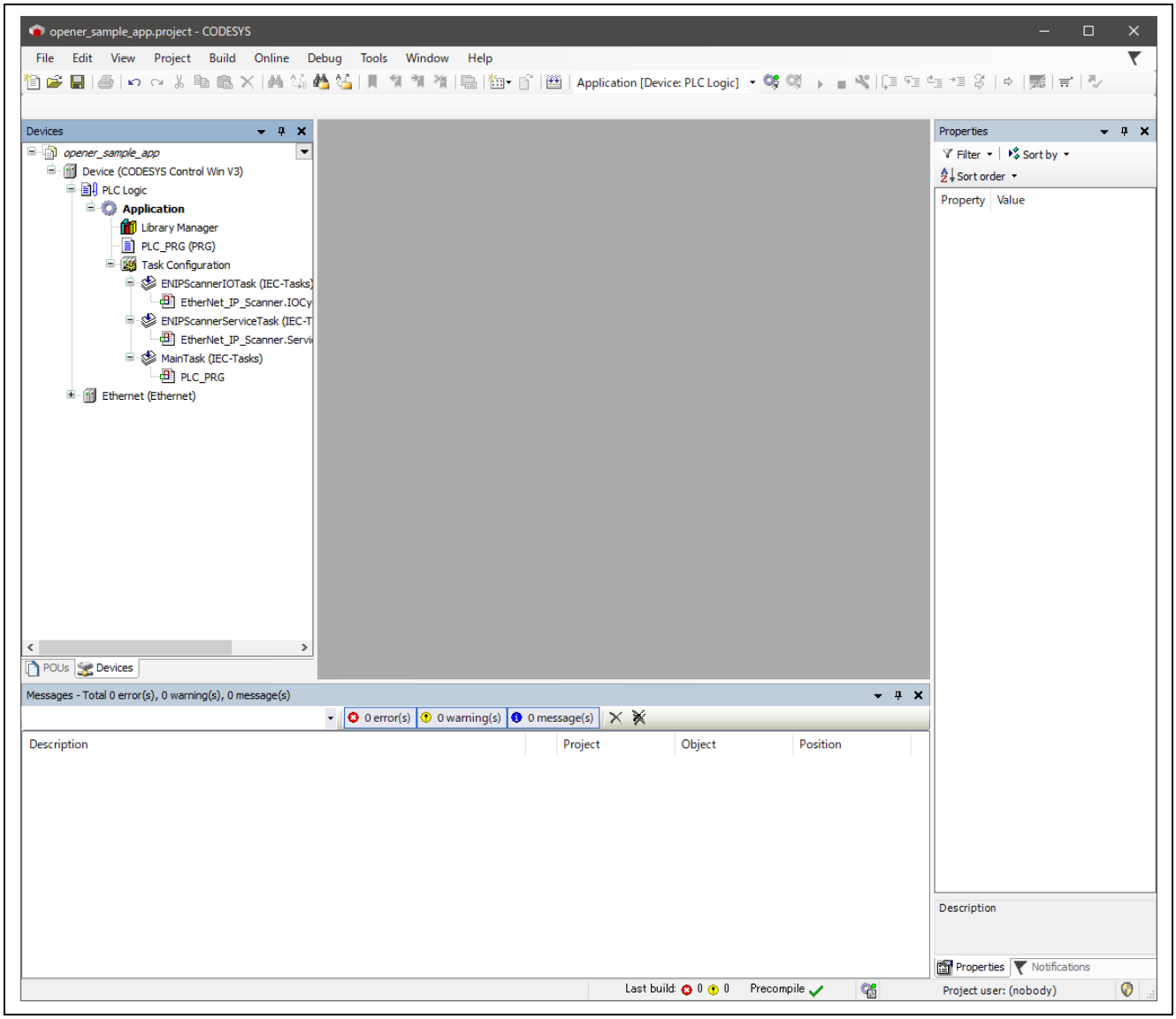


Figure 6-2 Open a CODESYS Project

6.3.2 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.

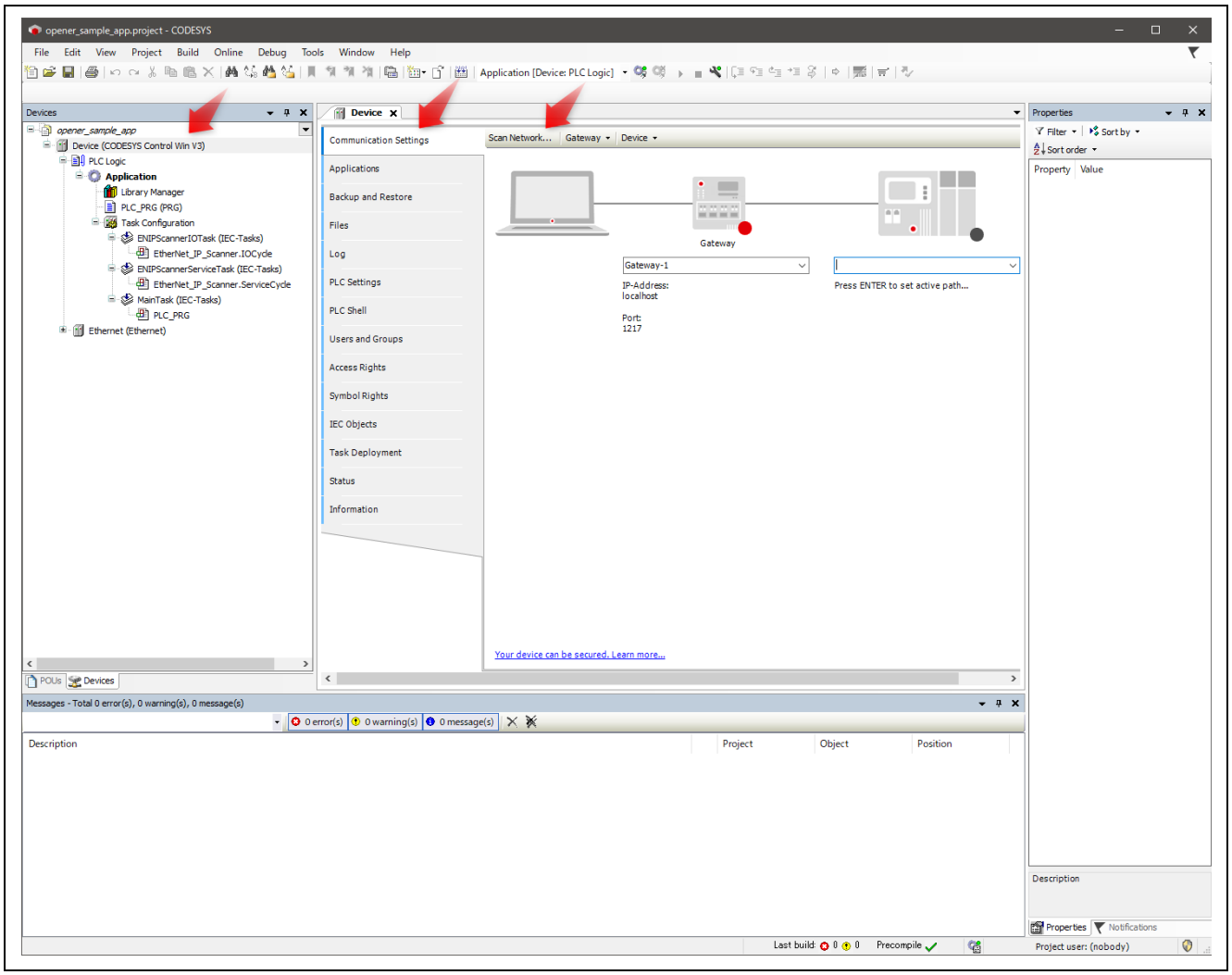


Figure 6-3 Communication Settings

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

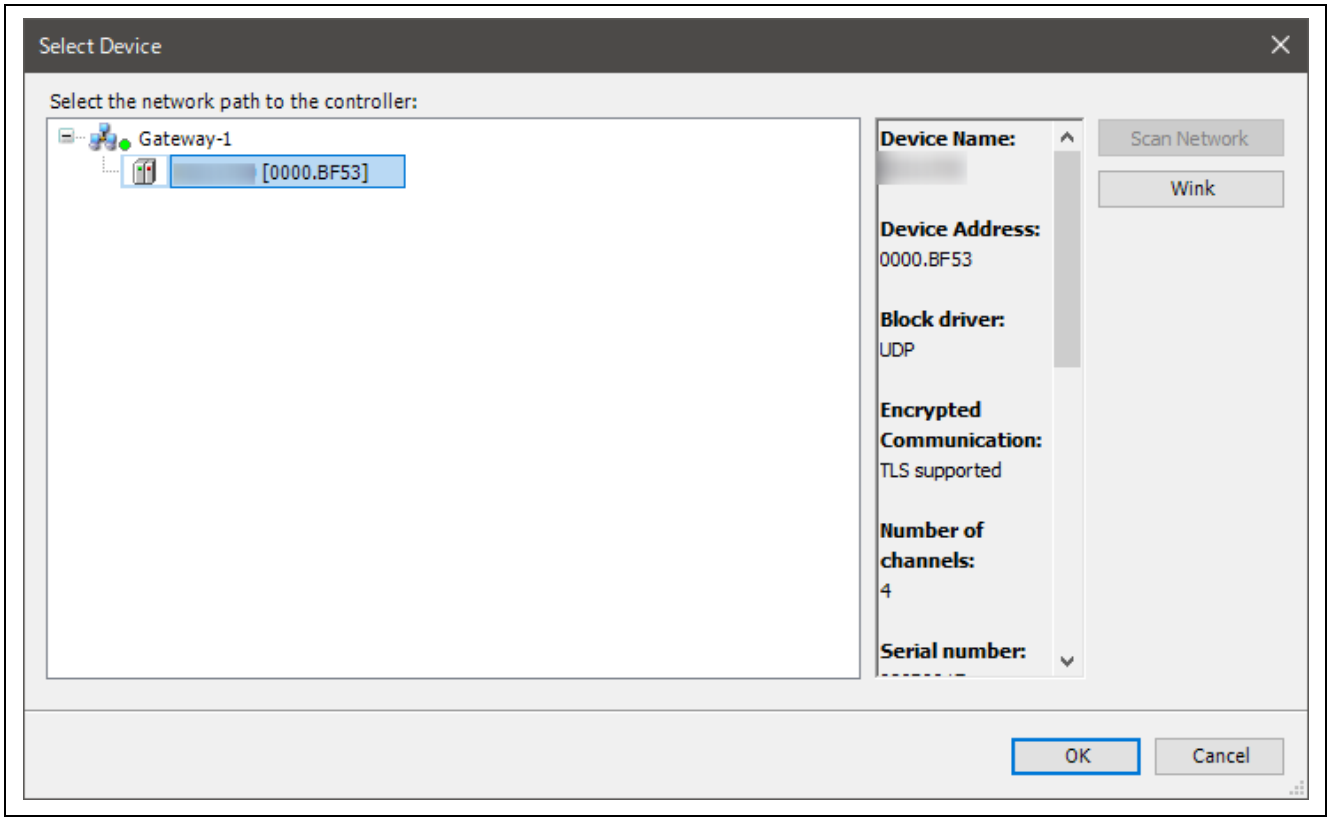


Figure 6-4 Select Device

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.

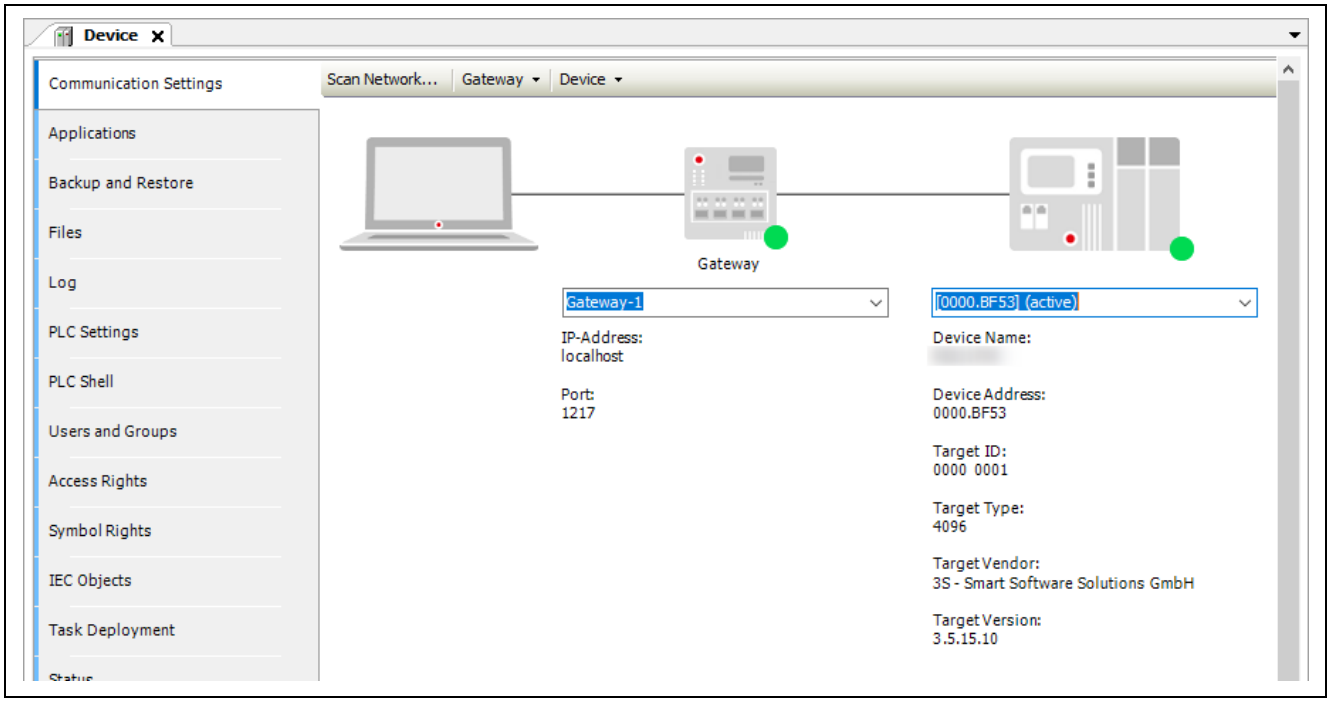


Figure 6-5 Green Marks at Gateway and Device Portions

6.3.3 Interface and IP address configuration

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

After that, please select “...” button to select network interface ethernet which is connected withRZ/N2L Industtrial Network SOM Kit, and please configure the IP address and related address values of the ethernet network interface.

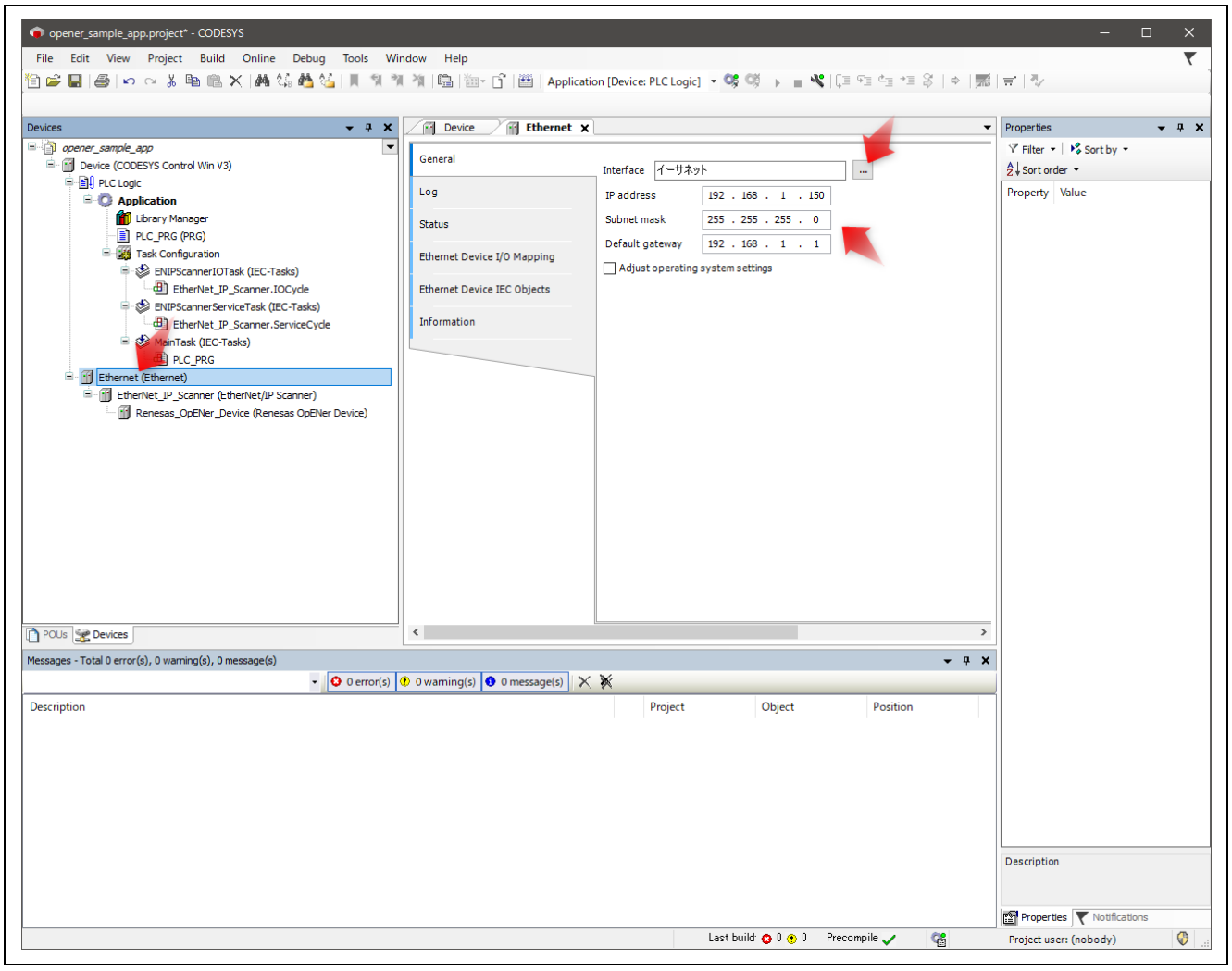


Figure 6-6 Open Ethernet tab

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

Finally, please set IP address form to the RZ/N2L Industrial Network SOM Kit device IP address.

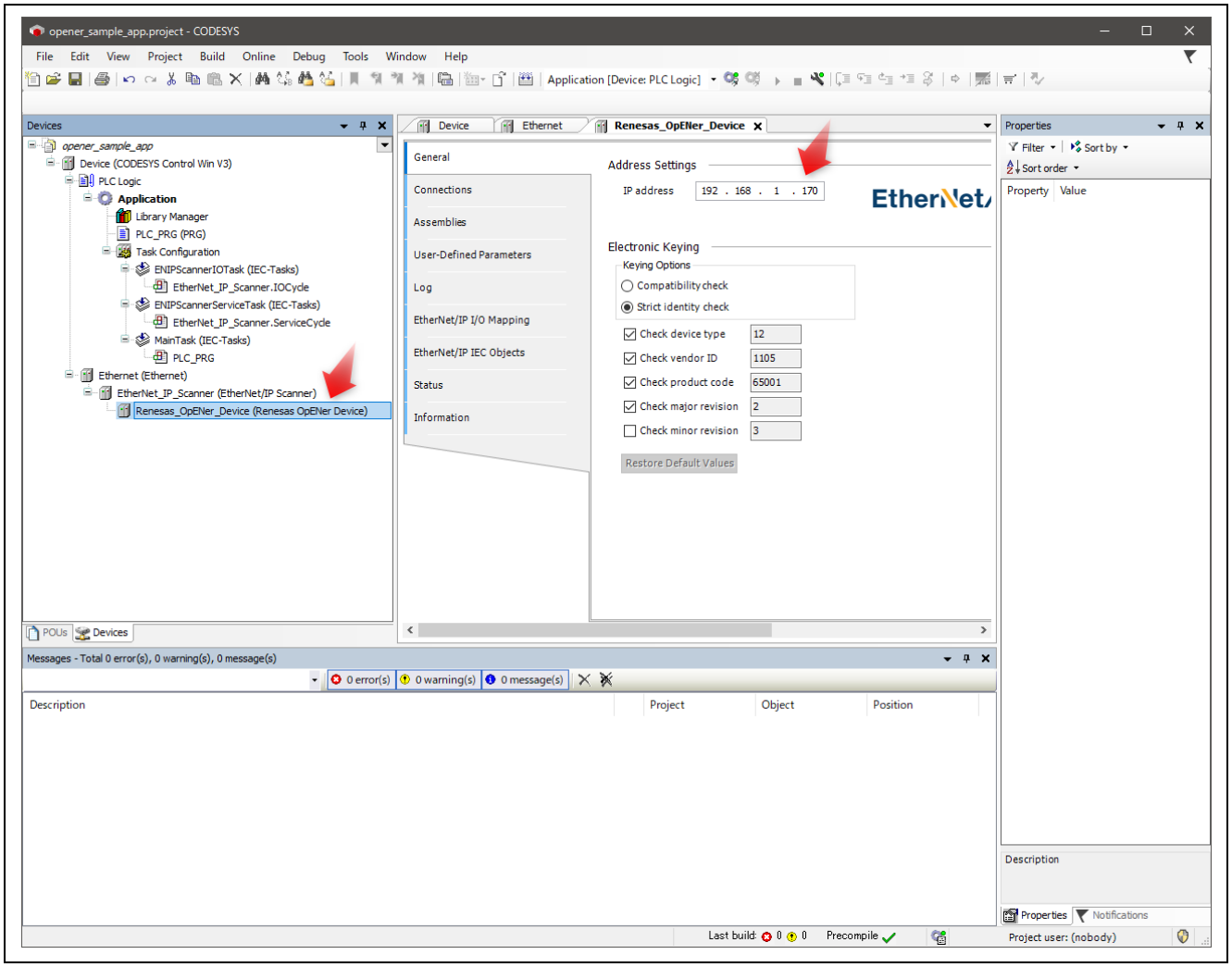


Figure 6-7 Renesas_OpENer_Device (Renesas OpENer Device)

6.4 Operation Check

6.4.1 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 the network.
3. Click “Start” button in the tool bar to run network and application.

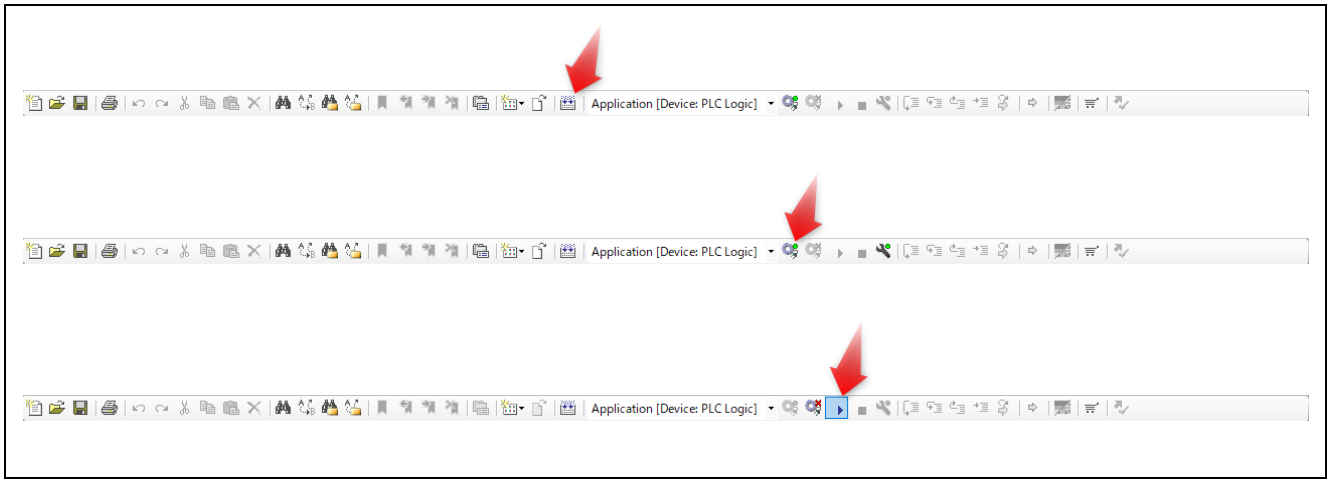


Figure 6-8 Build Project and Start Application

6.4.2 Check Network Connection

If the CODESYS application on PC connects with OpENer application on RZ/N2L properly, “Device”, “Ethernet”, “EtherNet_IP_Scanner”, and “Renesas_OpENer_Device” in left section are marked with green cycle mark.

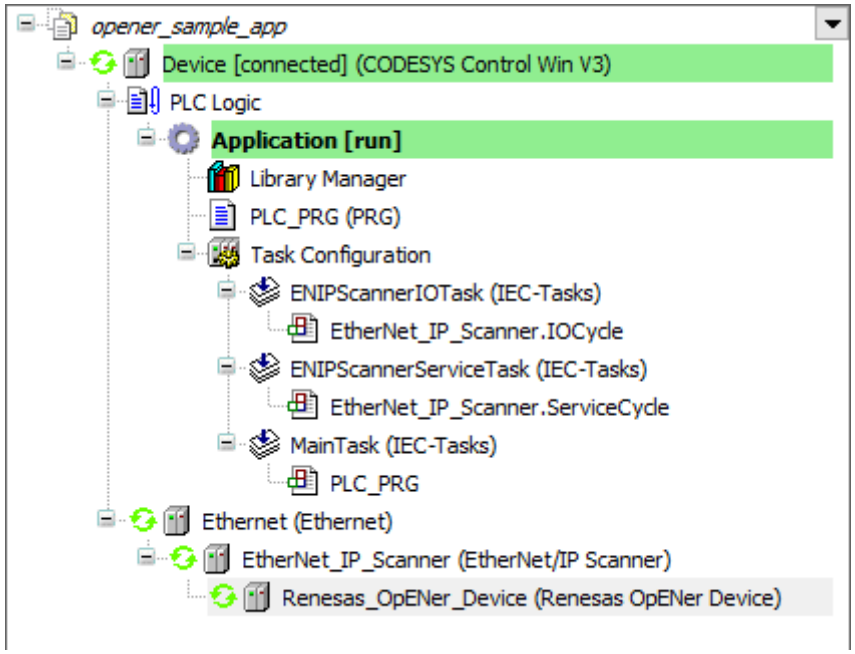


Figure 6-9 Check Network Connection

6.4.3 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 RZ/N2L OpENer application. The “Current Value” column indicates the LED and SW3 values.

- The Exclusive Owner Connection is bound to LED of RZ/N2L Industrial Network SOM Kit.
  - The LEDs lights by shifting with [SW Value] Hz.
- The Input Only Connection is bound to J6-1 ~ J6-4 of RZ/N2L Industrial Network SOM Kit.
  - The SW values indicates the frequency of LED light shifting.

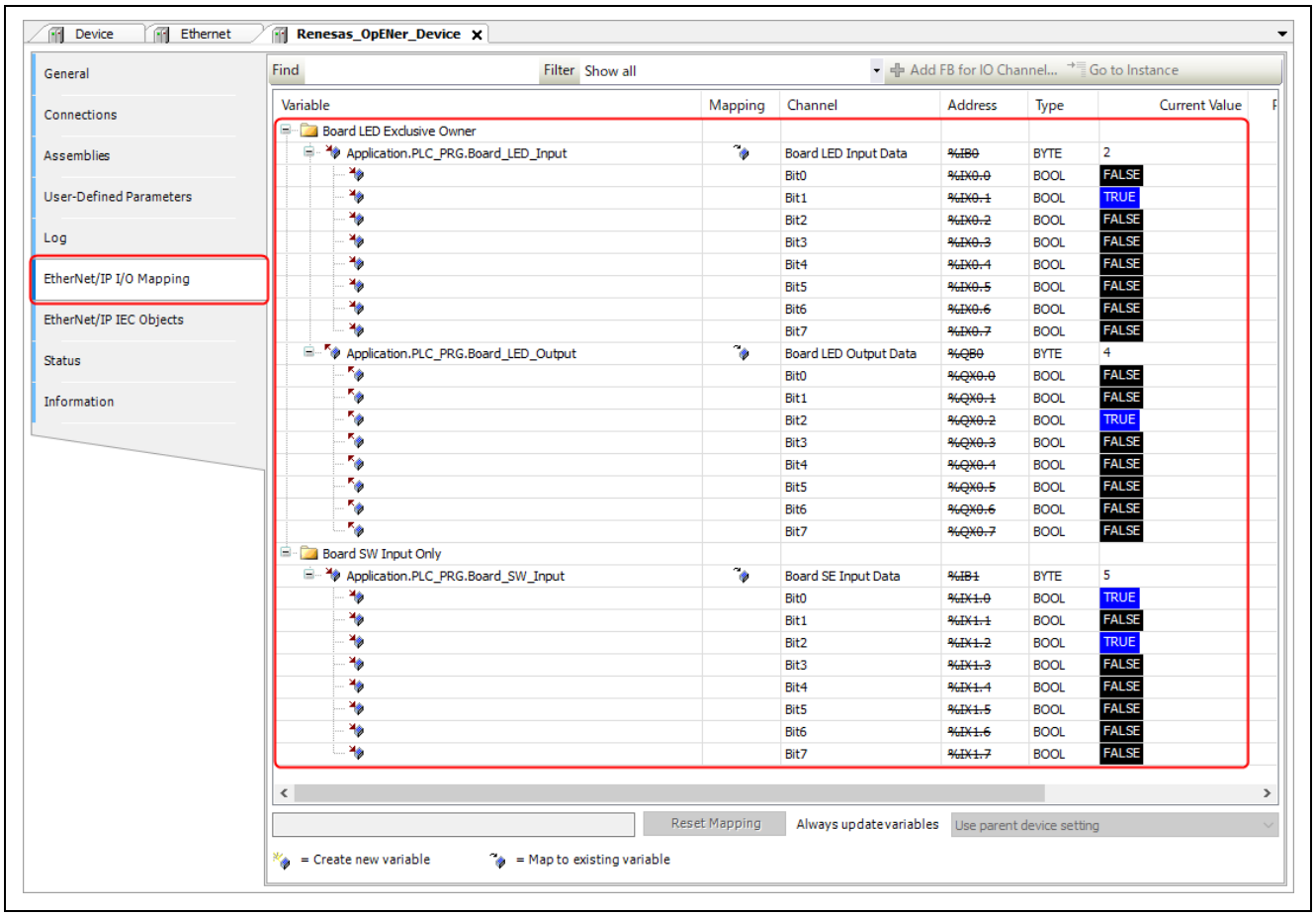


Figure 6-10 Check Application Behavior

7. 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 the following link for the detail.

https://github.com/EIPStackGroup/OpENer

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

This package is the RZ/N2L 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

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

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

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

- Git comment ID used in this sample software: a038063

The RZ/N2L port of OpENer includes FreeRTOS source codes as a 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 the following link for the details.

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

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

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

The RZ/N2L 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 the 7 instances of Assembly object. 5 of these instances are bound to the following array variables.

Table 7-1 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 SW3	`g_assembly_data_sw_input[1]`
150 (0x96)	Static Output	Input 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 SW3	-
237 (0xED)	Static Output	Heartbeat output of listen only I/O Connection.	-

Appendix C: Support CIP Object Classes

The CIP object classes on OpENer RZ/N2L port are shown below.

Table 7-2 CIP Object Class on OpENer RZ/N2L port

Object Class #	Object Class Name
0x01	Identity
0x02	Massage Router
0x04	Assembly
0x06	Connection Manager
0xF5	TCP/IP Interface
0xF6	Ethernet Link
0x48	QoS

Table 7-3 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 “deveicedata.h" in “common\renesas\oss_deps\opener” directory.

Table 7-4 0x02: Massage 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 7-5 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 7-6 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 7-7 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	○	-
13	Encapsulation Inactivity Timeout	○	○
Instance Service			
#	**Service Name**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		
0x10	Set_Attribute_Single		

Table 7-8 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		

Table 7-9 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		
0x0E	Get_Attribute_Single		
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**		
0x01	Get_Attributes_All		
0x0E	Get_Attribute_Single		

Appendix D: FSP Configuration for VSC8531

RZ/N2L Industrial Network SOM Kit has VSC8531 as PHY chip.
If reconfiguring by latest FSP, FSP configuration and source code needs to change from default.

(1) Regenerate source files by lates FSP

Remove the following four folders. After that, open the project according to section 5.

- When using e2studio, \project\rzn2l_som\opener_single\e2studio
- When using EWARM, \project\rzn2l_som\opener_single\ewarm

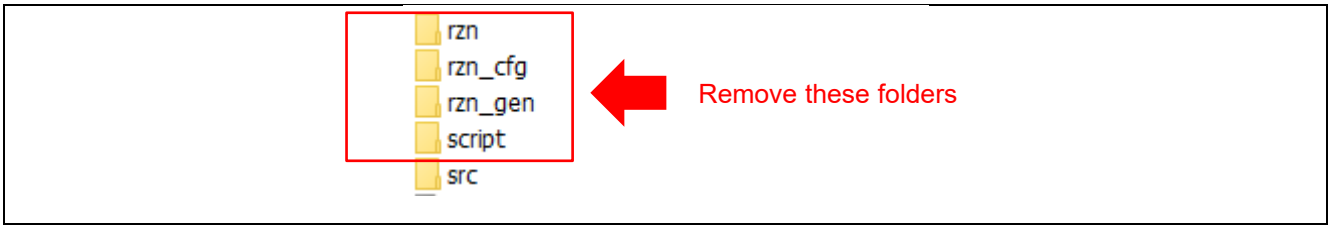


Figure 7-1 Remove folder generated by FSP

(2) Change ethernet driver configuration for VSC8531

Configure g_ether_phy0 Ethernet Driver on r_ether_phy for VSC8531 as shown in Figure 7-2.
Configuration value for VSC8531 shows in Table 7-10.

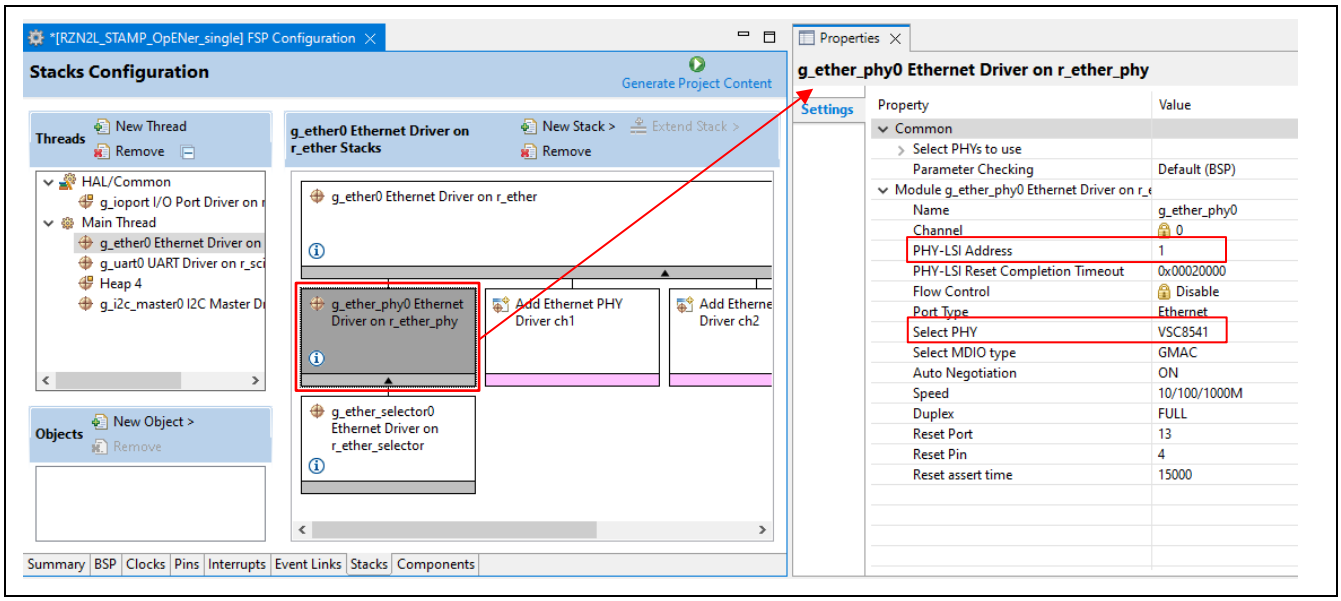


Figure 7-2 Ethernet Driver Configuration for VSC8531 (e.g. ETH0)

Table 7-10 FSP Configuration Value for VSC8531

Items	Default value	Config value for VSC8531	
		ETH0	ETH1
PHY-LSI Address	0	0	1
Select PHY	Default	VSC8541	VSC8541

(3) Add initialization code for VSC8531

The following code for VSC8531 initialization should be added to “ether_phy_targets_initialize_vsc8541” function in rzn/fsp/src/r_ether_phy/r_ether_phy.c.

The inclusion of “board_som.h” is aiso required for code activation.

```
#include "board_som.h"
                                        ~~ Omission ~~

void ether_phy_targets_initialize_vsc8541 (ether_phy_instance_ctrl_t * p_instance_ctrl)
{

                                        ~~ Omission ~~

    /* LED Behavior */
    reg  = ether_phy_read(p_instance_ctrl, ETHER_PHY_REG_LED_BEHAVIOR);
    reg &= ~(1U << ETHER_PHY_REG_LED0_FEATURE_DISABLE_OFFSET);
    reg |= 1U << ETHER_PHY_REG_LED1_FEATURE_DISABLE_OFFSET;
    ether_phy_write(p_instance_ctrl, ETHER_PHY_REG_LED_BEHAVIOR, reg);
#if defined(BOARD_RZN2L_SOM_KIT)  /* for VSC8531 */
    /* select extended page 2 register */
    ether_phy_write(p_instance_ctrl,   ETHER_PHY_REG_EXTEND_GPIO_PAGE, 0x02);

    /* read WoL and MAC Interface Control */
    reg  = ether_phy_read(p_instance_ctrl, 0x1b);

    /* set control to slow */
    reg &= 0xFF9F;
    ether_phy_write(p_instance_ctrl, 0x1b, reg);

    /* Configure RX_CLK delay and TX_CLK delay to 2.0ns */
    ether_phy_write(p_instance_ctrl, ETHER_PHY_REG_EXPAGE2_RGMII_CTRL, 0x0044);

    /* select extended page 0 register */
    ether_phy_write(p_instance_ctrl,   ETHER_PHY_REG_EXTEND_GPIO_PAGE, 0x00);
#endif

}                                       /* End of function ether_phy_targets_initialize() */
```

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
1.00	Feb. 7, 2023	-	First issued
1.10	Aug. 7, 2023	-	Support RZ/N2L FSP v1.2.0

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/.