R01AN6796EJ0110
Rev.1.10
Aug 7, 2023

RZ/N2L Group

RZ/N2L Industrial Network SOM Kit Application Note: EtherCAT CiA402 Slave Software

Introduction

This document explains Sample Program setup procedures for EtherCAT® slave functionalities with the adapted EtherCAT Stack Code for Renesas RZ/N2L platform. This describes steps to confirm slave behavior and stack features using TwinCAT®Master Configuration tool.

Target Device

RZ/N2L

Contents

1. Overview 3
1.1 Abbreviations/Definitions 3
1.2 Reference 3

2. Features 4

3. Project Setup 5
3.1 Requirements 5
3.2 Hardware 6
3.3 Note about Ethernet PHY driver using FSP 6
3.4 Setup the Board 7
3.5 Generating the Slave Stack Code 8

4. Setting up a TwinCAT3 12
4.1 Copying the ESI Files 12
4.2 Add Driver 12

5. Running the sample application 13
5.1 Setting sample code for EWARM 13
5.2 Setting sample code for GCC 16
5.2.1 Erasing the flash memory 16
5.2.2 Setting sample code for GCC 19

6. Connecting to TwinCAT3 23
6.1 Scanning I/O Devices 23
6.2 Updating EEPROM Data 25
6.3 CiA402 Drive Profile check 27
6.3.1 Rescanning the Device 27
6.3.2 Checking the Operation Mode 27
6.3.3 CiA402 State Transition 27

6.3.4 csp Mode 30
6.3.5 csv Mode 31

7. CiA402 Drive Profile 33
7.1 Operation Modes 33
7.2 State Transition 34
7.3 Object Dictionary 34
7.4 Implementing the Motor Control Program 36

8. Appendix: FSP Configuration for VSC8531 42

Revision History 44

1. Overview

This document describes how to run EtherCAT on the RZ/N2L Group. Run the standalone variant using only one core.

EtherCAT(Ethernet for Control Automation Technology) is an Ethernet based fieldbus system, developed by Beckhoff Automation. Development of EtherCAT was to apply Ethernet for automation applications (e.g., for motion control, I/O, sensors) requiring short data update times with low communication jitter and reduced hardware costs.

Tool to generate EtherCAT Slave Stack Code (SSC Tool) is available to the ETG members free of charge. This can be downloaded from the ETG website. SSC tool can be used to generate customized stack, device description files (ESI) and individual source code documentation to suit the developer's own needs.

This document describes the procedure for testing the EtherCAT slave function using EtherCAT stack code compatible with the Renesas RZ/N2L platform. Scope of the documentation is limited to explaining how to use the SSC tool for EtherCAT slave stack code generation and testing its behavior against TwinCAT masters and test applications.

1.1 Abbreviations/Definitions

Table 1. Abbreviations/Definitions

Index	Abbreviations /Definitions	Description
1	CoE	CAN application protocol over EtherCAT
2	EEPROM	Electrically Erasable Programmable Read-Only Memory
3	ESC	EtherCAT Slave Controller
4	ESI	EtherCAT Slave Information
5	FoE	File Access Over EtherCAT
6	I2C	Inter-Integrated Circuit
7	MB	Mail Box
8	PDO	Process Data Object
9	SSC	Slave Stack Code
10	EoE	Ethernet Over EtherCAT

1.2 Reference

Technical information about EtherCAT is available via ETG member site, and information about RZ/N2L is available via Renesas.

Table 2. Technical Inputs

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)

2. Features

EtherCAT slave stack code generated by SSC Tool provides the functionality of EtherCAT slave controller.

Includes the following features :

- ESM (EtherCAT State Machine)
- Mailbox protocols:
  - CoE (CAN application protocol over EtherCAT)

- Synchronization Modes:
  - Free Run
  - Sync Manager Synchronization
  - DC Synchronization

- CiA402 Drive Profile:
  - csp Mode
  - csv Mode

3. Project Setup

3.1 Requirements

Table 3. 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
SSC Tool	Beckhoff Automation	Slave Stack Code (SSC) Tool Version 5.13
Software PLC	Beckhoff Automation	TwinCAT3

*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/

3.2 Hardware

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

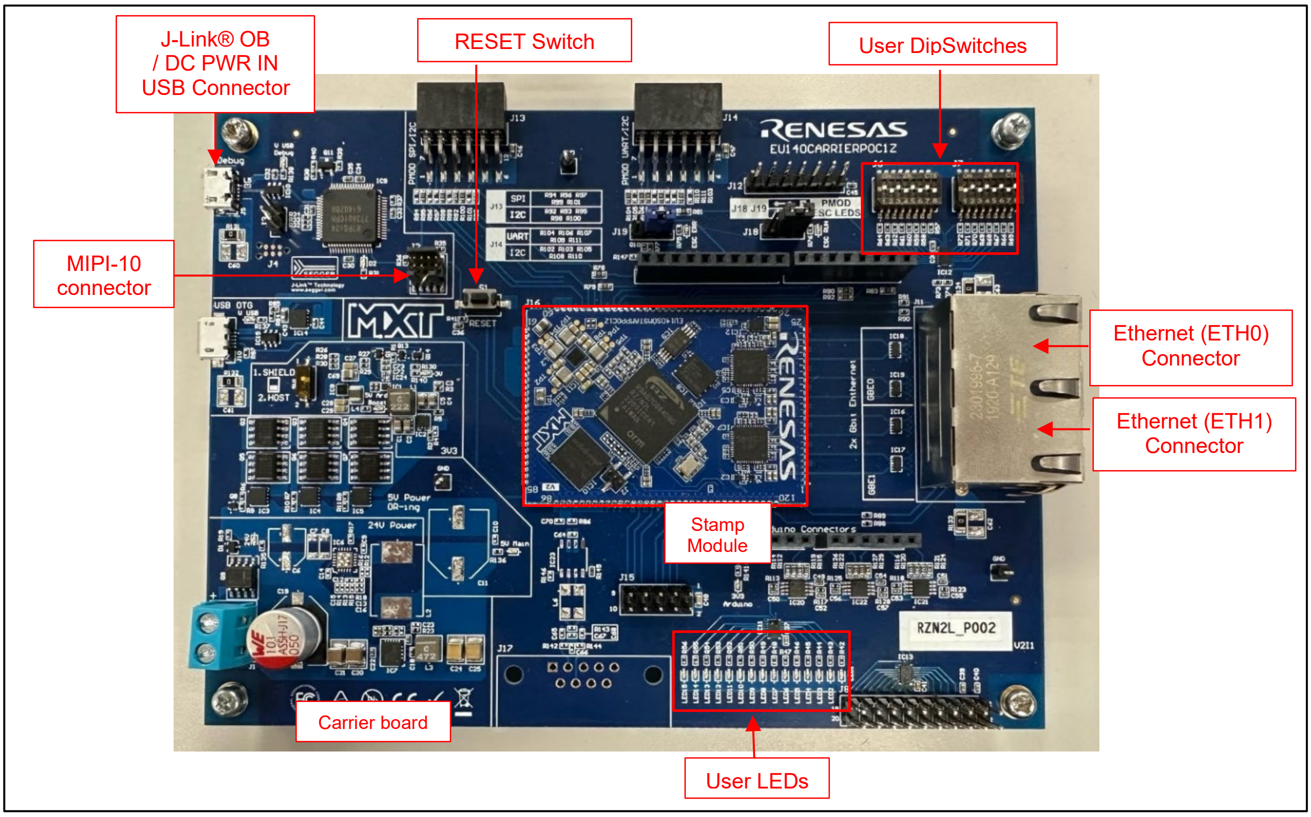


Figure 3-1 RZ/N2L Industrial Network SOM Kit

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: 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.

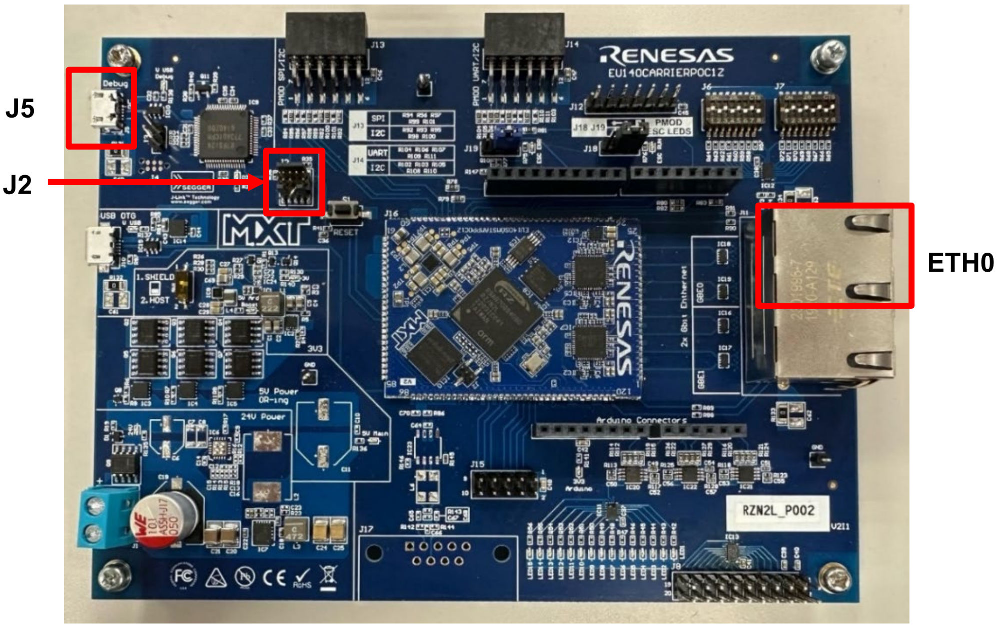


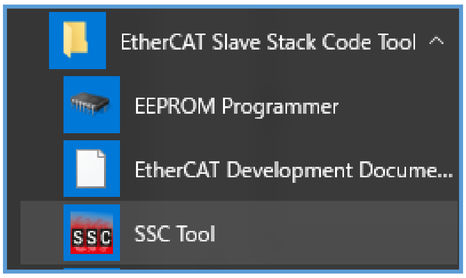
Figure 3-2 Setup the SOM Kit

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”.

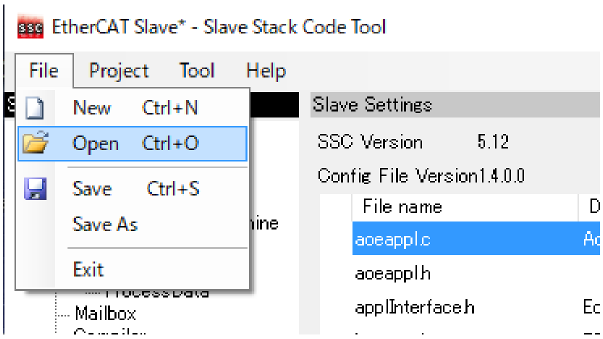
3.5 Generating the Slave Stack Code

SSC Tool is used for generating the slave stack code.
Note). Replace the folder name in the following description according to the tool to be used.

1. Start the SSC Tool from the Window Start menu.


EtherCAT Slave Stack Code Tool
EEPROM Programmer
EtherCAT Development Docume...
SSC Tool

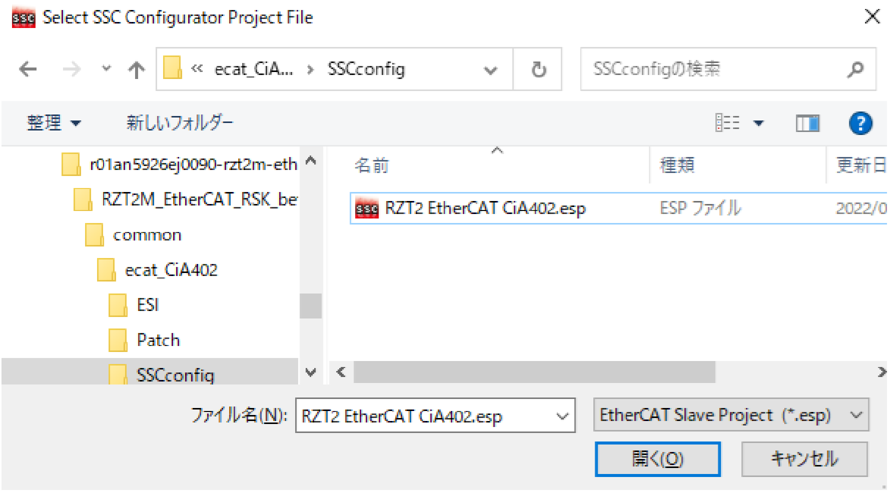
2. Select File > Open.



SSC Version 5.12
Config File Version1.4.0.0

File name	D
aoeappl.c	Ac
aoeappl.h	
applInterface.h	Ec

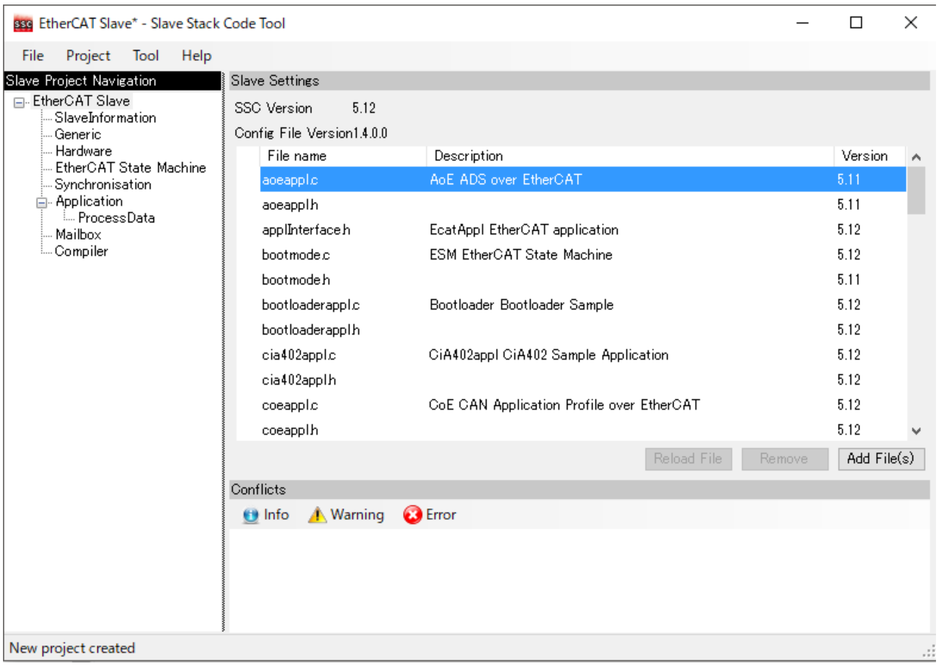
1. Select the following file,
"common\ecat_CiA402\SSCconfig\ RZN2 EtherCAT CiA402.esp"



名前	種類	更新日
RZT2 EtherCAT CiA402.esp	ESP ファイル	2022/0

ファイル名(N): RZT2 EtherCAT CiA402.esp
EtherCAT Slave Project (*.esp)

2. After clicking the [OK] button, the following window opens.



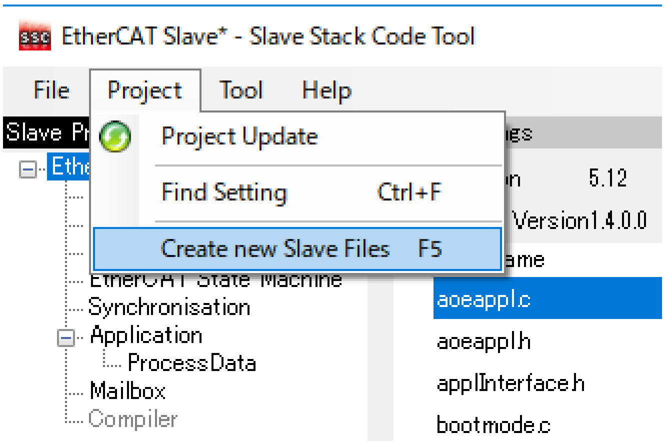
SSC Version 5.12

Config File Version1.4.0.0

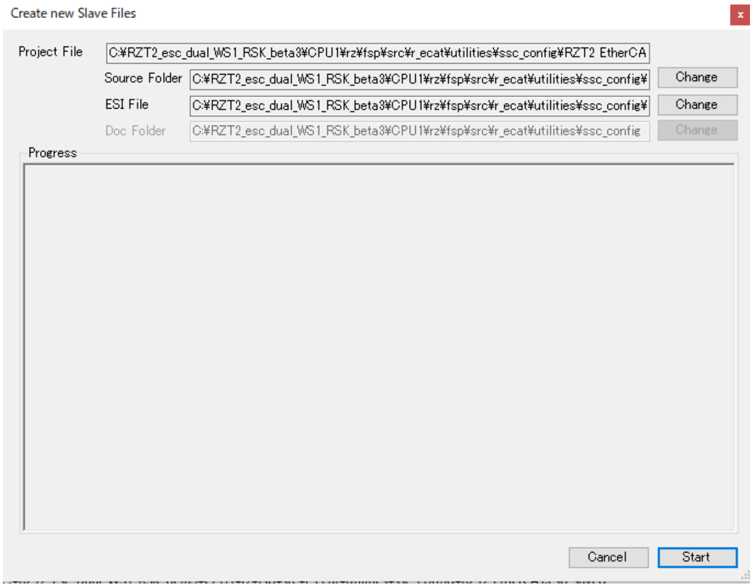
File name	Description	Version
aoeappl.c	AoE ADS over EtherCAT	5.11
aoeappl.h		5.11
applInterface.h	EcatAppl EtherCAT application	5.12
bootmode.c	ESM EtherCAT State Machine	5.12
bootmode.h		5.11
bootloaderappl.c	Bootloader Bootloader Sample	5.12
bootloaderappl.h		5.12
cia402appl.c	CiA402appl CiA402 Sample Application	5.12
cia402appl.h		5.12
coeappl.c	CoE CAN Application Profile over EtherCAT	5.12
coeappl.h		5.12

New project created

3. Select Project > Create new Slave Files.



4. Click the [Start] button to start creating the EtherCAT Slave Stack Code.



Create new Slave Files

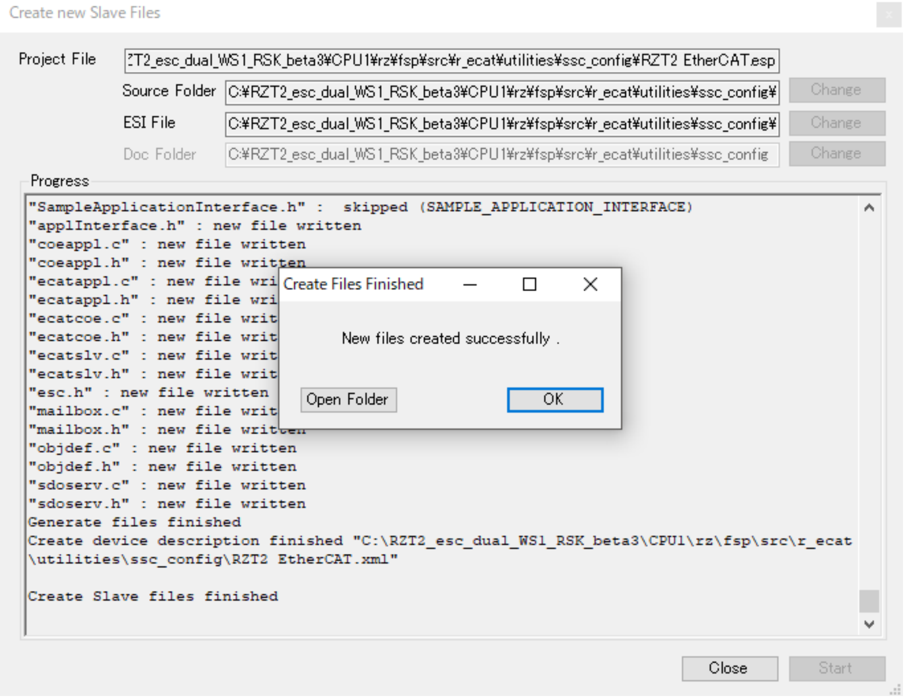
Project File C:¥RZT2_esc_dual_WS1_RSK_beta3¥CPU1¥rz¥fsp¥src¥r_ecat¥utilities¥ssc_config¥RZT2 EtherCA

Source Folder C:¥RZT2_esc_dual_WS1_RSK_beta3¥CPU1¥rz¥fsp¥src¥r_ecat¥utilities¥ssc_config¥

ESI File C:¥RZT2_esc_dual_WS1_RSK_beta3¥CPU1¥rz¥fsp¥src¥r_ecat¥utilities¥ssc_config¥

Doc Folder C:¥RZT2_esc_dual_WS1_RSK_beta3¥CPU1¥rz¥fsp¥src¥r_ecat¥utilities¥ssc_config

5. When a message “New file created successfully” appears, the creation processing is completed, and the source files are located in the following folder.
“common\ecat_CiA402\SSCconfig\Src”



Create new Slave Files

Project File ?T2_esc_dual_WS1_RSK_beta3¥CPU1¥rz¥fsp¥src¥r_ecat¥utilities¥ssc_config¥RZT2 EtherCAT.esp

Source Folder C:¥RZT2_esc_dual_WS1_RSK_beta3¥CPU1¥rz¥fsp¥src¥r_ecat¥utilities¥ssc_config¥

ESI File C:¥RZT2_esc_dual_WS1_RSK_beta3¥CPU1¥rz¥fsp¥src¥r_ecat¥utilities¥ssc_config¥

Doc Folder C:¥RZT2_esc_dual_WS1_RSK_beta3¥CPU1¥rz¥fsp¥src¥r_ecat¥utilities¥ssc_config

Progress

```
"SampleApplicationInterface.h" :  skipped (SAMPLE_APPLICATION_INTERFACE)
"applInterface.h" : new file written
"coeappl.c" : new file written
"coeappl.h" : new file written
"ecatappl.c" : new file wri
"ecatappl.h" : new file wri
"ecatcoe.c" : new file writ
"ecatcoe.h" : new file writ
"ecatslv.c" : new file writ
"ecatslv.h" : new file writ
"esc.h" : new file written
"mailbox.c" : new file writ
"mailbox.h" : new file written
"objdef.c" : new file written
"objdef.h" : new file written
"sdoserv.c" : new file written
"sdoserv.h" : new file written
Generate files finished
Create device description finished "C:\RZT2_esc_dual_WS1_RSK_beta3\CPU1\rz\fsp\src\r_ecat
\utilities\ssc_config\RZT2 EtherCAT.xml"

Create Slave files finished
```

Create Files Finished

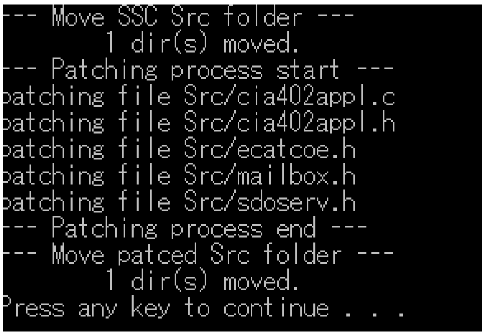
New files created successfully .

Note), If an error occurs during generation, uncheck "Create device description" in [Tool]-> [Option]-> [Create Files] of the SSC tool.

6. Right-click on the apply_patch.bat file and select [Run as an administrator] → [Yes]
   The patch file contains modifications to make the SSC source file suitable for the RZ/N2L.

   [ewarm]
   "common\ecat_CiA402\Patch\apply_patch_ewarm.bat"

   [e2studio]
   "common\ecat_CiA402\Patch\apply_patch_e2studio.bat"



```
--- Move SSC Src folder ---
        1 dir(s) moved.
--- Patching process start ---
patching file Src/cia402appl.c
patching file Src/cia402appl.h
patching file Src/ecatcoe.h
patching file Src/mailbox.h
patching file Src/sdoserv.h
--- Patching process end ---
--- Move patced Src folder ---
        1 dir(s) moved.
Press any key to continue . . .
```

After execution of the patch, the modified source file is stored in the following folder.

[ewarm]
"project\rzn2l_som\ecat_CiA402\ewarm\src\ethercat\beckhoff"

[e2studio]
"project\rzn2l_som\ecat_CiA402\e2studio\src\ethercat\beckhoff"

Note) If the patch command is not installed on your PC, you will need to install ver. 2.5.9 or a later version of GNU patch.
If it is already installed, skip this step.
Download the patch command (currently ver. 2.5.9) from the following Web page and store "patch.exe" in a folder on a path that makes the file executable from the command prompt.
http://gnuwin32.sourceforge.net/packages/patch.htm

4. Setting up a TwinCAT3

4.1 Copying the ESI Files

Before starting TwinCAT, copy the ESI files that are included in the release folder to TwinCAT destination "\TwinCAT\3.x\Config\IO\EtherCAT"

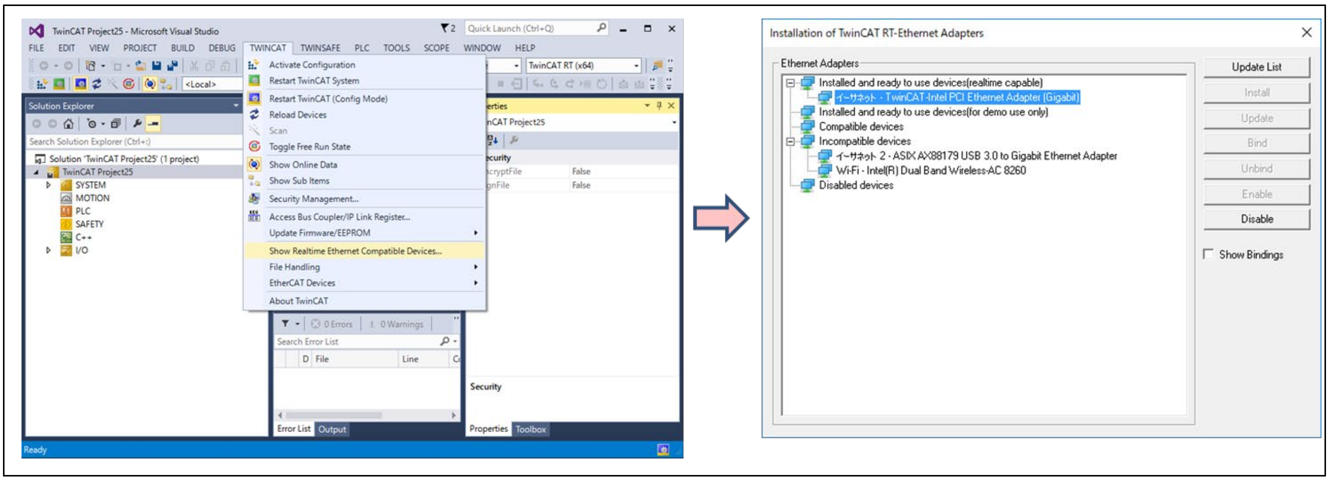
ESI file for current release available at,
"*common\ecat_CiA402\ESI\ Renesas EtherCAT RZN2 CiA402.xml*"

4.2 Add Driver

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

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

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


TwinCAT Project25 - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TWINCAT TWINSAFE PLC TOOLS SCOPE WINDOW HELP
Activate Configuration
Restart TwinCAT System
Restart TwinCAT (Config Mode)
Reload Devices
Scan
Toggle Free Run State
Show Online Data
Show Sub Items
Security Management...
Access Bus Coupler/IP Link Register...
Update Firmware/EEPROM
Show Realtime Ethernet Compatible Devices...
File Handling
EtherCAT Devices
About TwinCAT
Solution Explorer
Solution 'TwinCAT Project25' (1 project)
TwinCAT Project25
SYSTEM
MOTION
PLC
SAFETY
C++
I/O
Error List
Output
Properties
Toolbox
Ready
Installation of TwinCAT RT-Ethernet Adapters
Ethernet Adapters
Installed and ready to use devices(realtime capable)
イーサネット - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
Installed and ready to use devices(for demo use only)
Compatible devices
Incompatible devices
イーサネット 2 - ASIX AX88179 USB 3.0 to Gigabit Ethernet Adapter
Wi-Fi - Intel(R) Dual Band Wireless-AC 8260
Disabled devices
Update List
Install
Update
Bind
Unbind
Enable
Disable
□ Show Bindings

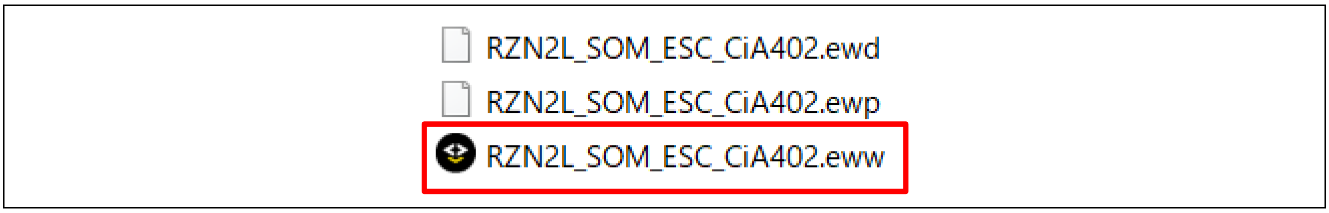
5. Running the sample application

5.1 Setting sample code for EWARM

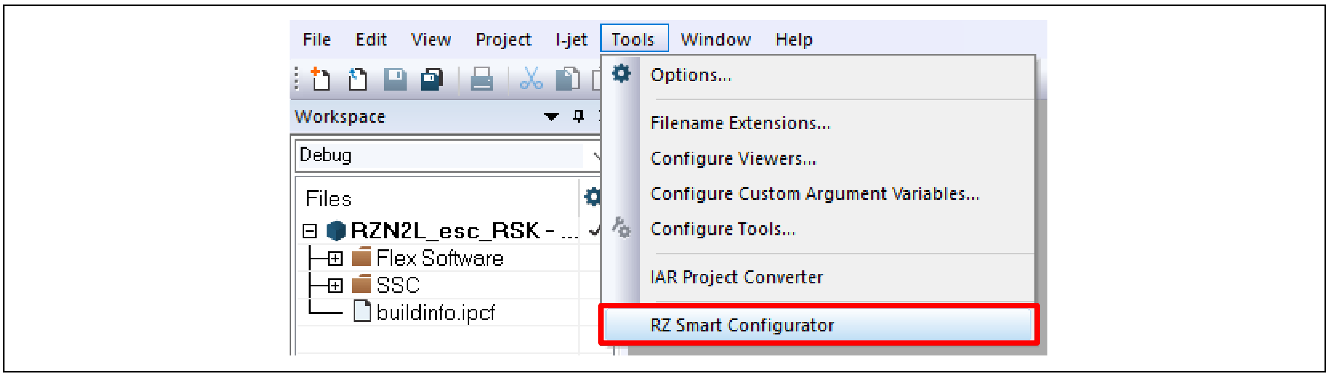
Build the sample code and load it into RAM using IAR Embedded Workbench.

Note). Please install FSP Smart Configurator in advance.
Refer to the latest getting started guide.(R01an6434ejxxxx-rzt2-rzn2.pdf)
Replace the project name in the figure with the project name of this sample project.

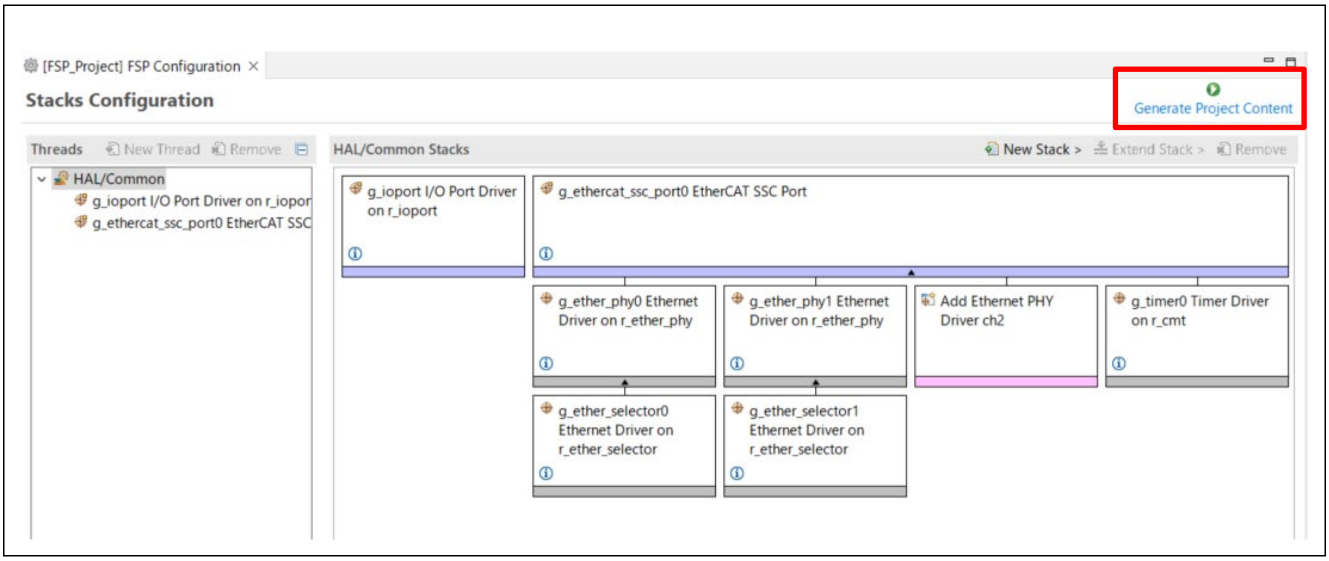
1. Open the sample project. **“project\rzn2l_som\ ecat_CiA402\ewarm \RZN2L_SOM_ESC_CiA402.eww”**



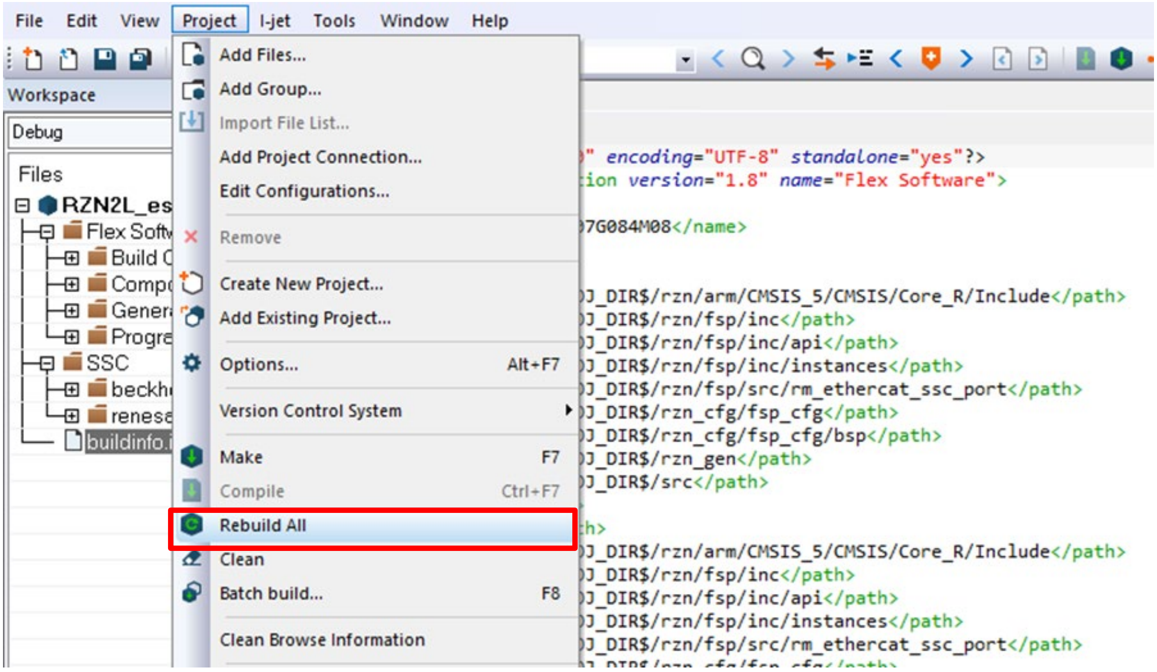
2. Open the “RZ Smart Configurator”



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


[FSP_Project] FSP Configuration
Stacks Configuration
Generate Project Content
Threads
New Thread
Remove
HAL/Common
g_ioport I/O Port Driver on r_iopor
g_ethercat_ssc_port0 EtherCAT SSC
HAL/Common Stacks
New Stack >
Extend Stack >
Remove
g_ioport I/O Port Driver on r_ioport
g_ethercat_ssc_port0 EtherCAT SSC Port
g_ether_phy0 Ethernet Driver on r_ether_phy
g_ether_phy1 Ethernet Driver on r_ether_phy
Add Ethernet PHY Driver ch2
g_timer0 Timer Driver on r_cmt
g_ether_selector0 Ethernet Driver on r_ether_selector
g_ether_selector1 Ethernet Driver on r_ether_selector

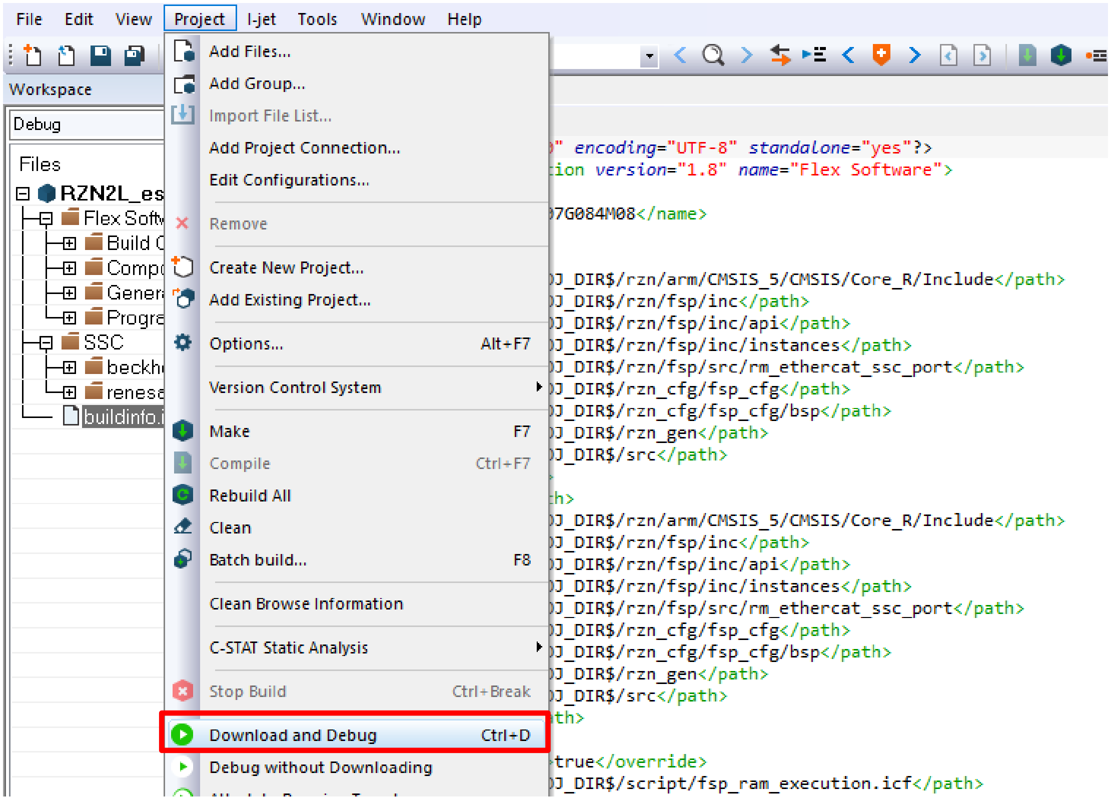
4. Select the “Rebuild All” item from the “Project” menu to rebuild the project.



```
" encoding="UTF-8" standalone="yes"?>
ion version="1.8" name="Flex Software">
7G084M08</name>
J_DIR$/rzn/arm/CMSIS_5/CMSIS/Core_R/Include</path>
J_DIR$/rzn/fsp/inc</path>
J_DIR$/rzn/fsp/inc/api</path>
J_DIR$/rzn/fsp/inc/instances</path>
J_DIR$/rzn/fsp/src/rm_ethercat_ssc_port</path>
J_DIR$/rzn_cfg/fsp_cfg</path>
J_DIR$/rzn_cfg/fsp_cfg/bsp</path>
J_DIR$/rzn_gen</path>
J_DIR$/src</path>
h>
J_DIR$/rzn/arm/CMSIS_5/CMSIS/Core_R/Include</path>
J_DIR$/rzn/fsp/inc</path>
J_DIR$/rzn/fsp/inc/api</path>
J_DIR$/rzn/fsp/inc/instances</path>
J_DIR$/rzn/fsp/src/rm_ethercat_ssc_port</path>
```

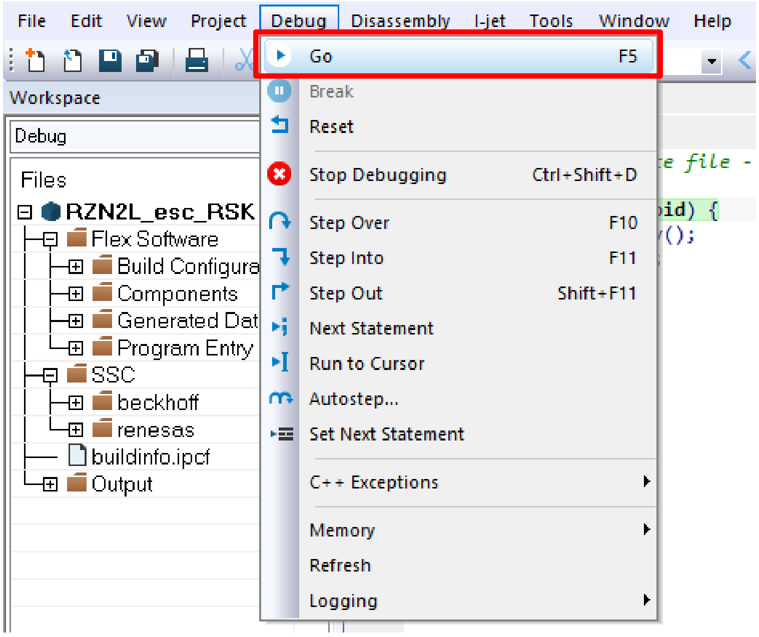
5. Press the “RESET” switch of the RZ/N2L Industrial Network SOM Kit.

6. After connecting the board and I-jet, click the Download and Debug button on the Project toolbar.



```
" encoding="UTF-8" standalone="yes"?>
ion version="1.8" name="Flex Software">
7G084M08</name>
J_DIR$/rzn/arm/CMSIS_5/CMSIS/Core_R/Include</path>
J_DIR$/rzn/fsp/inc</path>
J_DIR$/rzn/fsp/inc/api</path>
J_DIR$/rzn/fsp/inc/instances</path>
J_DIR$/rzn/fsp/src/rm_ethercat_ssc_port</path>
J_DIR$/rzn_cfg/fsp_cfg</path>
J_DIR$/rzn_cfg/fsp_cfg/bsp</path>
J_DIR$/rzn_gen</path>
J_DIR$/src</path>
h>
J_DIR$/rzn/arm/CMSIS_5/CMSIS/Core_R/Include</path>
J_DIR$/rzn/fsp/inc</path>
J_DIR$/rzn/fsp/inc/api</path>
J_DIR$/rzn/fsp/inc/instances</path>
J_DIR$/rzn/fsp/src/rm_ethercat_ssc_port</path>
J_DIR$/rzn_cfg/fsp_cfg</path>
J_DIR$/rzn_cfg/fsp_cfg/bsp</path>
J_DIR$/rzn_gen</path>
J_DIR$/src</path>
th>
true</override>
J_DIR$/script/fsp_ram_execution.icf</path>
```

7. Press the "Resume" button for the project. Program will run.

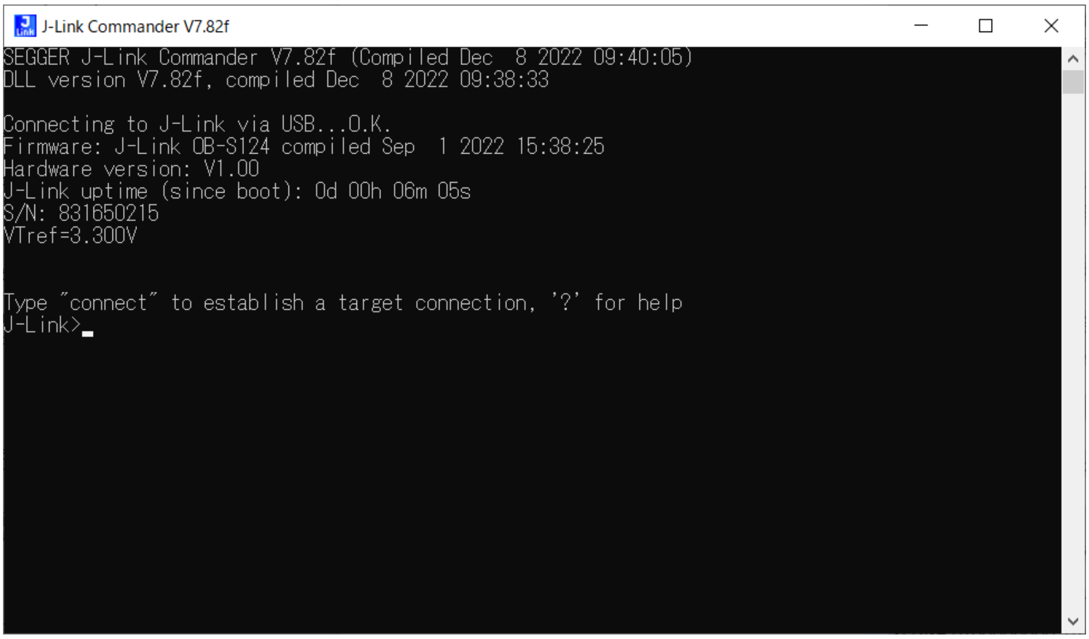


5.2 Setting sample code for GCC

5.2.1 Erasing the flash memory

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.



```
SEGGER J-Link Commander V7.82f (Compiled Dec  8 2022 09:40:05)
DLL version V7.82f, compiled Dec  8 2022 09:38:33

Connecting to J-Link via USB...O.K.
Firmware: J-Link OB-S124 compiled Sep  1 2022 15:38:25
Hardware version: V1.00
J-Link uptime (since boot): 0d 00h 06m 05s
S/N: 831650215
VTref=3.300V


Type "connect" to establish a target connection, '?' for help
J-Link>_
```

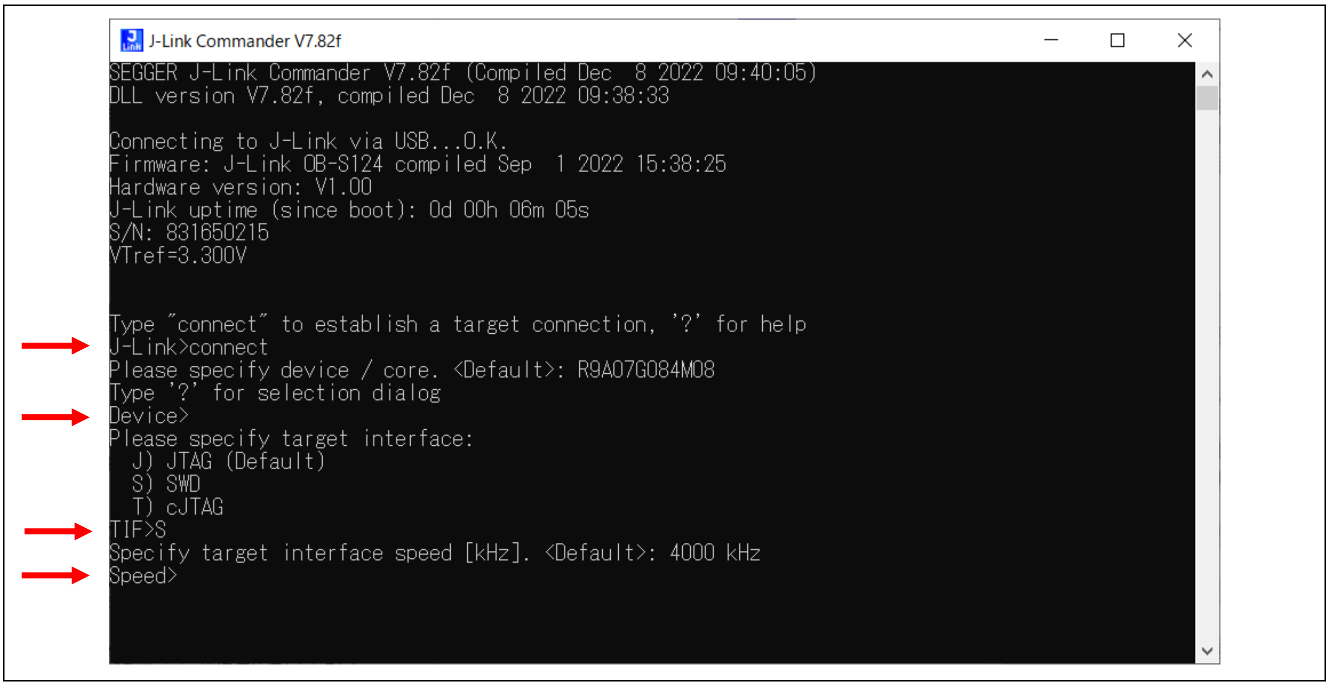
Figure 5-1 Open J-Link Commander

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.

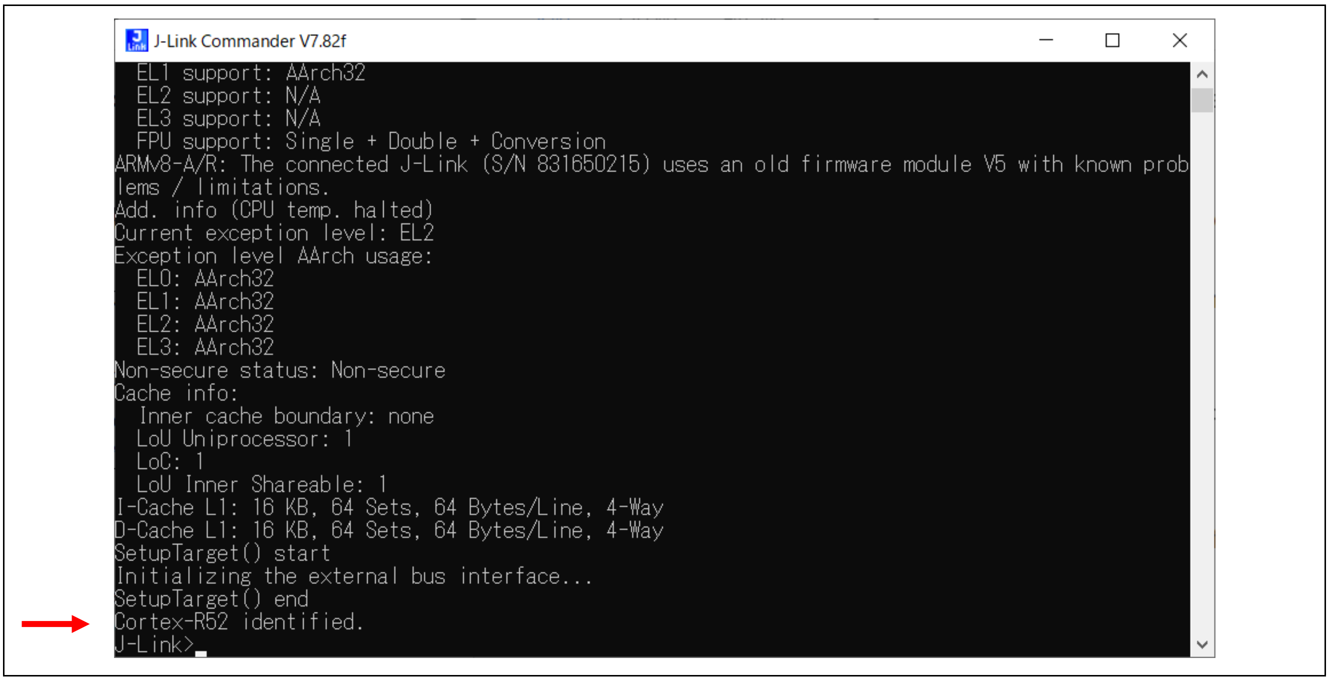


```
SEGGER J-Link Commander V7.82f (Compiled Dec  8 2022 09:40:05)
DLL version V7.82f, compiled Dec  8 2022 09:38:33

Connecting to J-Link via USB...O.K.
Firmware: J-Link OB-S124 compiled Sep  1 2022 15:38:25
Hardware version: V1.00
J-Link uptime (since boot): 0d 00h 06m 05s
S/N: 831650215
VTref=3.300V


Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. <Default>: R9A07G084M08
Type '?' for selection dialog
Device>
Please specify target interface:
  J) JTAG (Default)
  S) SWD
  T) cJTAG
TIF>S
Specify target interface speed [kHz]. <Default>: 4000 kHz
Speed>
```

Figure 5-2 Connection conditions (1/2)



```
  EL1 support: AArch32
  EL2 support: N/A
  EL3 support: N/A
  FPU support: Single + Double + Conversion
ARMv8-A/R: The connected J-Link (S/N 831650215) uses an old firmware module V5 with known prob
lems / limitations.
Add. info (CPU temp. halted)
Current exception level: EL2
Exception level AArch usage:
  EL0: AArch32
  EL1: AArch32
  EL2: AArch32
  EL3: AArch32
Non-secure status: Non-secure
Cache info:
  Inner cache boundary: none
  LoU Uniprocessor: 1
  LoC: 1
  LoU Inner Shareable: 1
I-Cache L1: 16 KB, 64 Sets, 64 Bytes/Line, 4-Way
D-Cache L1: 16 KB, 64 Sets, 64 Bytes/Line, 4-Way
SetupTarget() start
Initializing the external bus interface...
SetupTarget() end
Cortex-R52 identified.
J-Link>
```

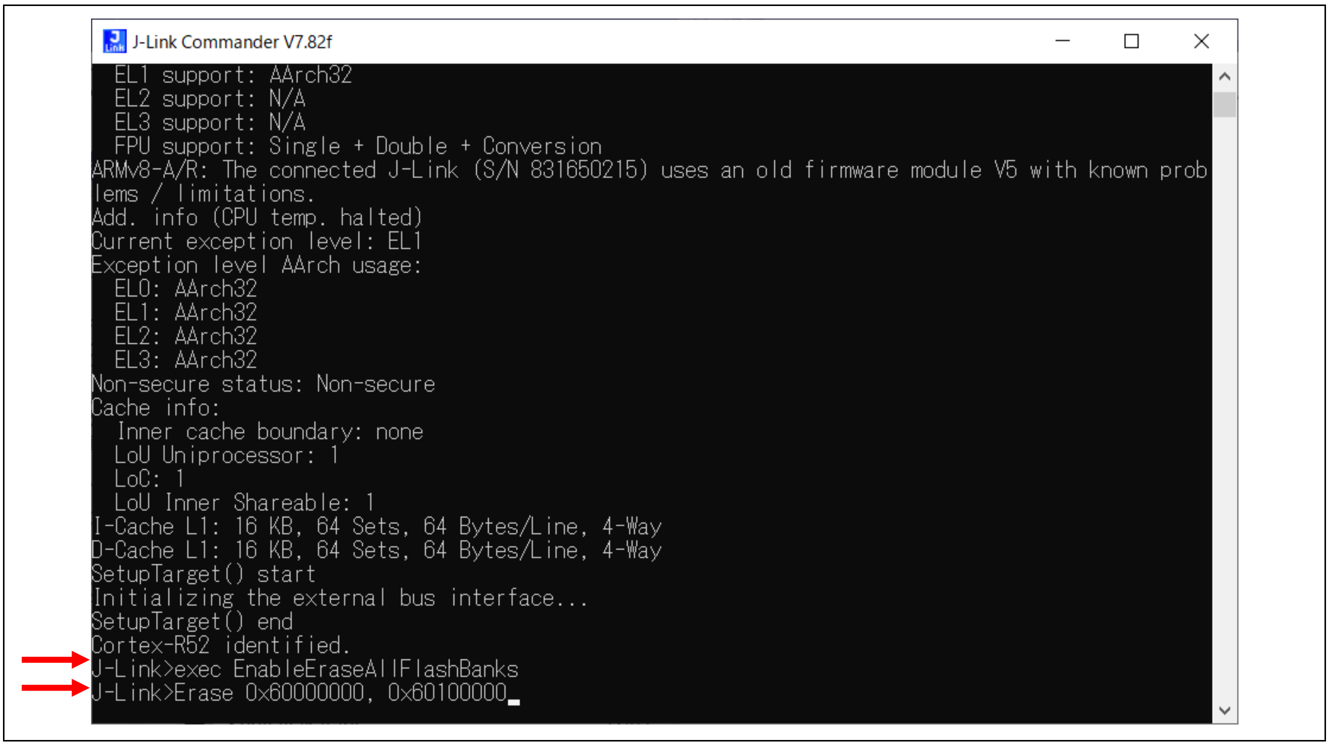
Figure 5-3 Connection conditions (2/2)

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

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

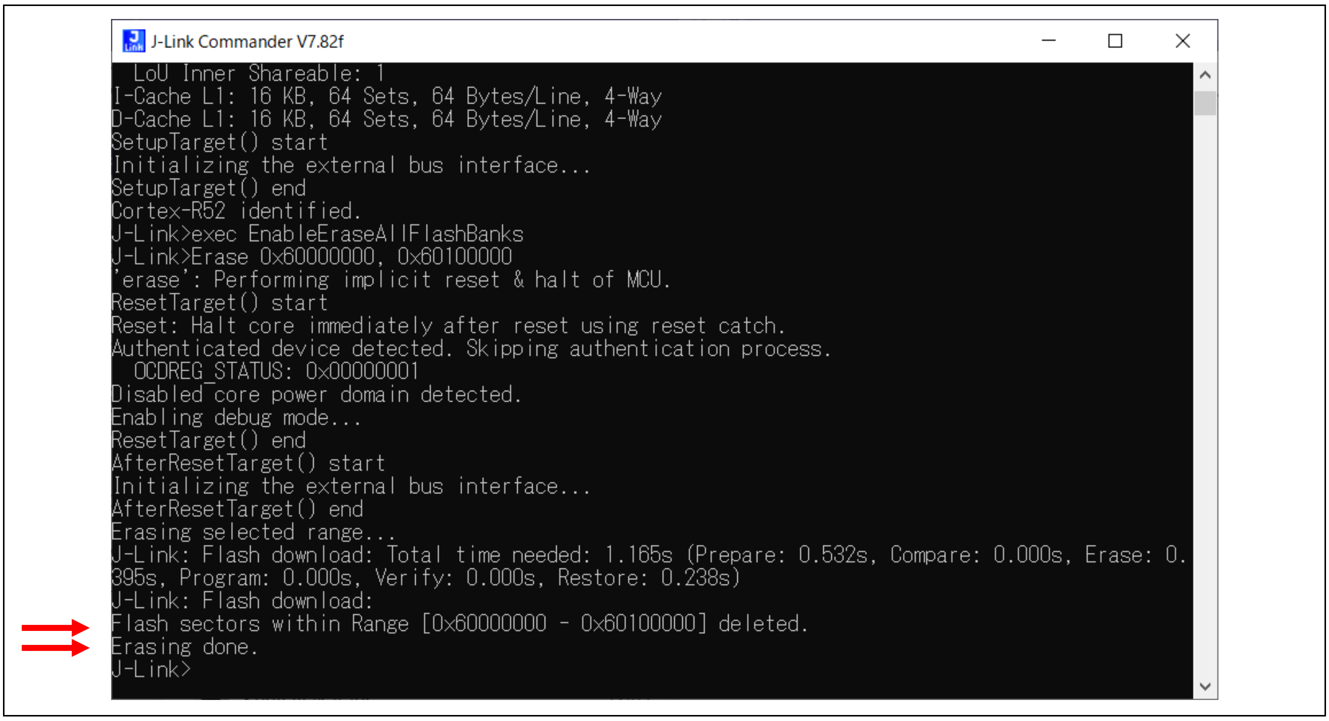
After that, confirm the message “Erasing done.” Is displayed.

Enter “q” to exit J-Link Commander.



```
  EL1 support: AArch32
  EL2 support: N/A
  EL3 support: N/A
  FPU support: Single + Double + Conversion
ARMv8-A/R: The connected J-Link (S/N 831650215) uses an old firmware module V5 with known prob
lems / limitations.
Add. info (CPU temp. halted)
Current exception level: EL1
Exception level AArch usage:
  EL0: AArch32
  EL1: AArch32
  EL2: AArch32
  EL3: AArch32
Non-secure status: Non-secure
Cache info:
  Inner cache boundary: none
  LoU Uniprocessor: 1
  LoC: 1
  LoU Inner Shareable: 1
I-Cache L1: 16 KB, 64 Sets, 64 Bytes/Line, 4-Way
D-Cache L1: 16 KB, 64 Sets, 64 Bytes/Line, 4-Way
SetupTarget() start
Initializing the external bus interface...
SetupTarget() end
Cortex-R52 identified.
J-Link>exec EnableEraseAllFlashBanks
J-Link>Erase 0x60000000, 0x60100000_
```

Figure 5-4 Erase flash memory (1/2)



```
  LoU Inner Shareable: 1
I-Cache L1: 16 KB, 64 Sets, 64 Bytes/Line, 4-Way
D-Cache L1: 16 KB, 64 Sets, 64 Bytes/Line, 4-Way
SetupTarget() start
Initializing the external bus interface...
SetupTarget() end
Cortex-R52 identified.
J-Link>exec EnableEraseAllFlashBanks
J-Link>Erase 0x60000000, 0x60100000
'erase': Performing implicit reset & halt of MCU.
ResetTarget() start
Reset: Halt core immediately after reset using reset catch.
Authenticated device detected. Skipping authentication process.
  OCDREG_STATUS: 0x00000001
Disabled core power domain detected.
Enabling debug mode...
ResetTarget() end
AfterResetTarget() start
Initializing the external bus interface...
AfterResetTarget() end
Erasing selected range...
J-Link: Flash download: Total time needed: 1.165s (Prepare: 0.532s, Compare: 0.000s, Erase: 0.
395s, Program: 0.000s, Verify: 0.000s, Restore: 0.238s)
J-Link: Flash download:
Flash sectors within Range [0x60000000 - 0x60100000] deleted.
Erasing done.
J-Link>
```

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

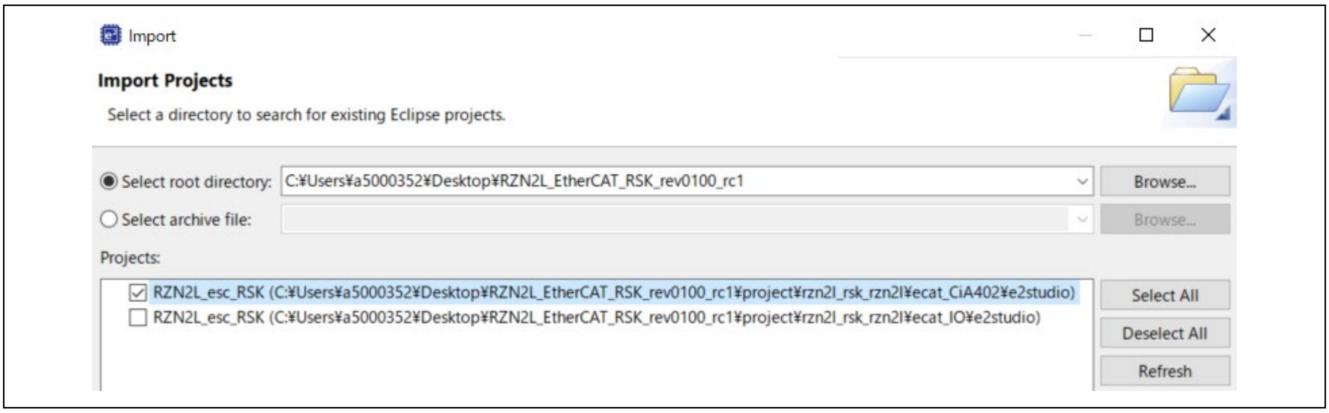
5.2.2 Setting sample code for GCC

Build the sample code and load it into RAM using Renesas Electronics e² studio.

Note). Please install e2studio and adapt the FSP_Packs_v1.0.0 in advance.
Refer to the latest getting started guide.(R01an6434ejxxxx- rzt2-rzn2.pdf)

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

1. Import the sample project. After the program is started, by selecting [File] → [Import] → [Existing Projects into Workspace]. Check the "select root directory" and select “project\rzn2l_som\ecat_CiA402\e2studio” folder →[Finish].



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:¥Users¥a5000352¥Desktop¥RZN2L_EtherCAT_RSK_rev0100_rc1

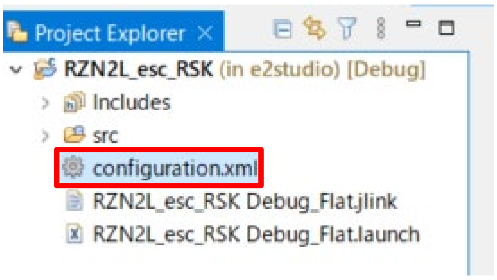
○ Select archive file:

Projects:

☑ RZN2L_esc_RSK (C:¥Users¥a5000352¥Desktop¥RZN2L_EtherCAT_RSK_rev0100_rc1¥project¥rzn2l_rsk_rzn2l¥ecat_CiA402¥e2studio)

□ RZN2L_esc_RSK (C:¥Users¥a5000352¥Desktop¥RZN2L_EtherCAT_RSK_rev0100_rc1¥project¥rzn2l_rsk_rzn2l¥ecat_IO¥e2studio)

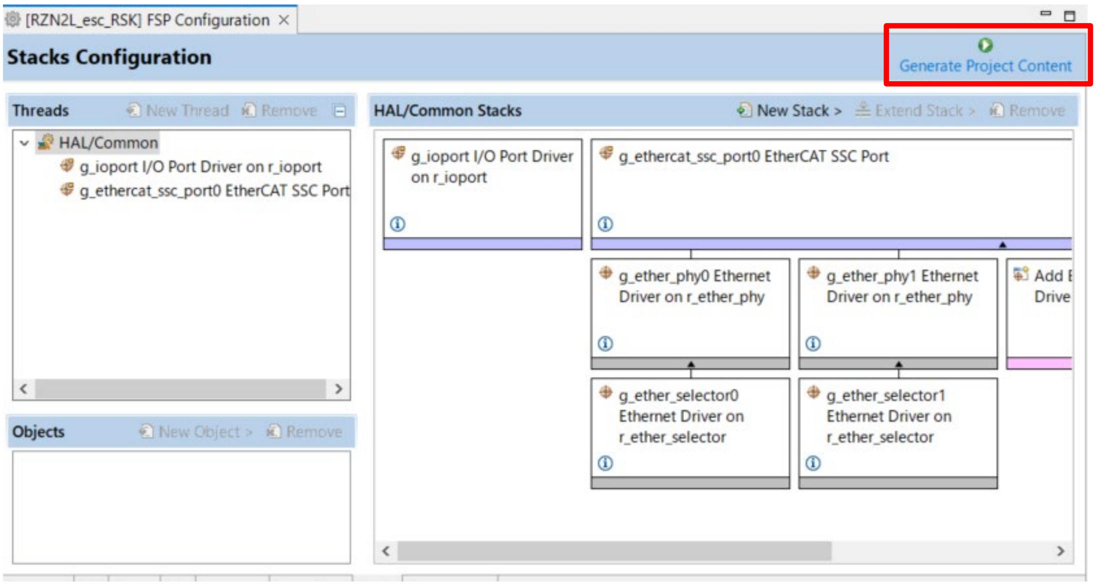
2. Open "cofiguration.xml" in the "RZN2L_SOM_ESC_CiA402" project



Project Explorer

RZN2L_esc_RSK (in e2studio) [Debug]
- Includes
- src
- configuration.xml
- RZN2L_esc_RSK Debug_Flat.jlink
- RZN2L_esc_RSK Debug_Flat.launch

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

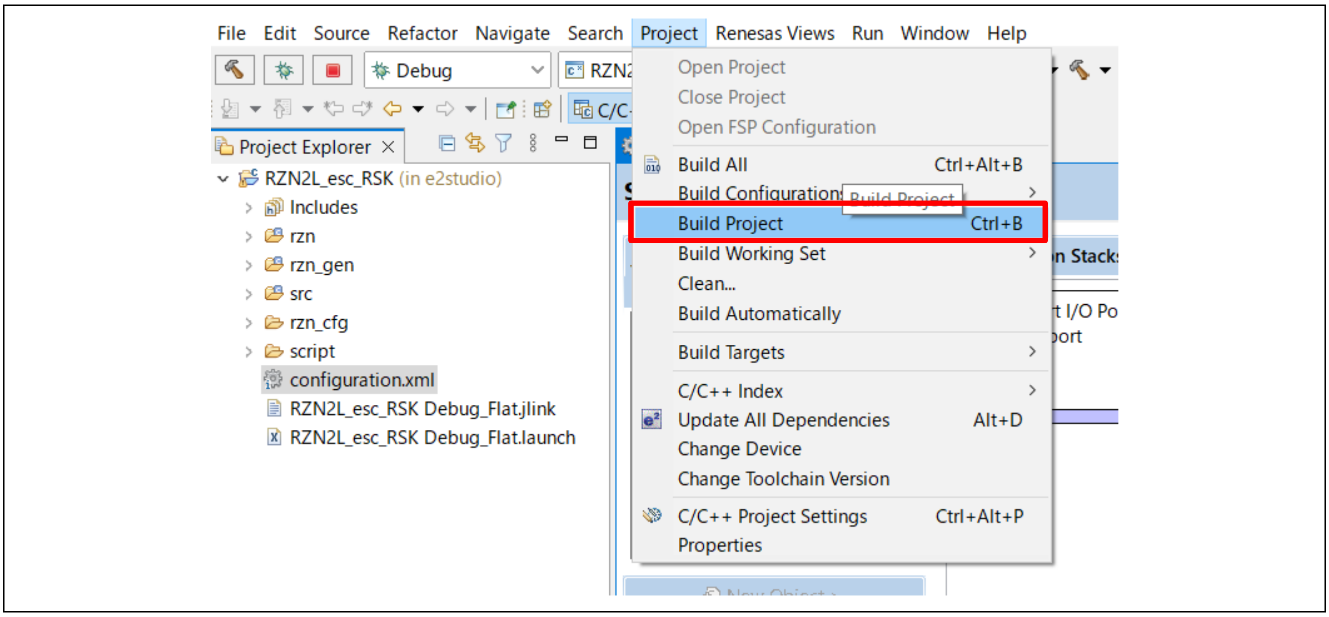


[RZN2L_esc_RSK] FSP Configuration

Stacks Configuration

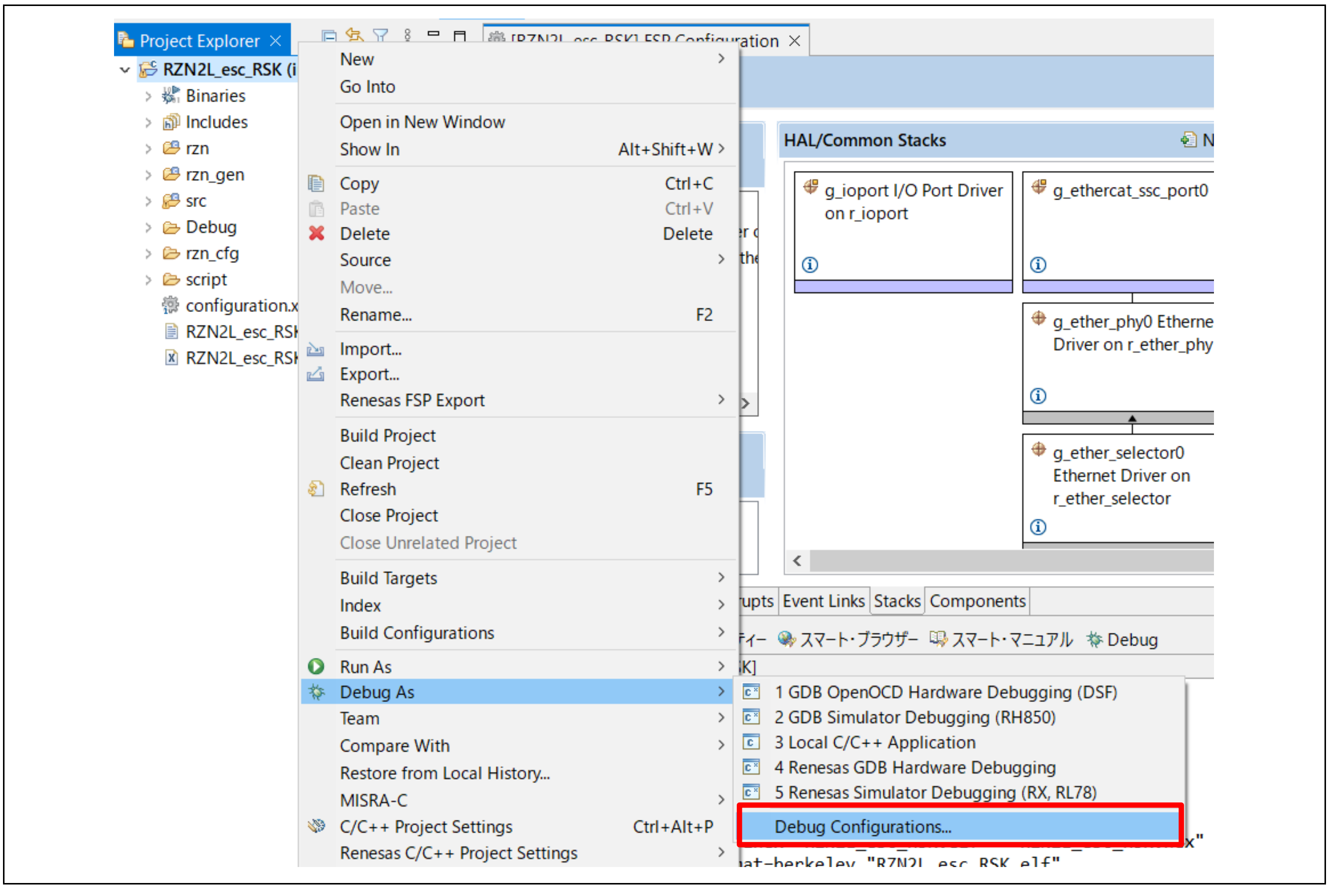
Generate Project Content
Threads
New Thread
Remove
HAL/Common
g_ioport I/O Port Driver on r_ioport
g_ethercat_ssc_port0 EtherCAT SSC Port
Objects
New Object >
Remove
HAL/Common Stacks
New Stack >
Extend Stack >
Remove
g_ioport I/O Port Driver on r_ioport
g_ethercat_ssc_port0 EtherCAT SSC Port
g_ether_phy0 Ethernet Driver on r_ether_phy
g_ether_phy1 Ethernet Driver on r_ether_phy
Add E Drive
g_ether_selector0 Ethernet Driver on r_ether_selector
g_ether_selector1 Ethernet Driver on r_ether_selector

4. Select and build the "RZN2L_SOM_ESC_CiA402" project.

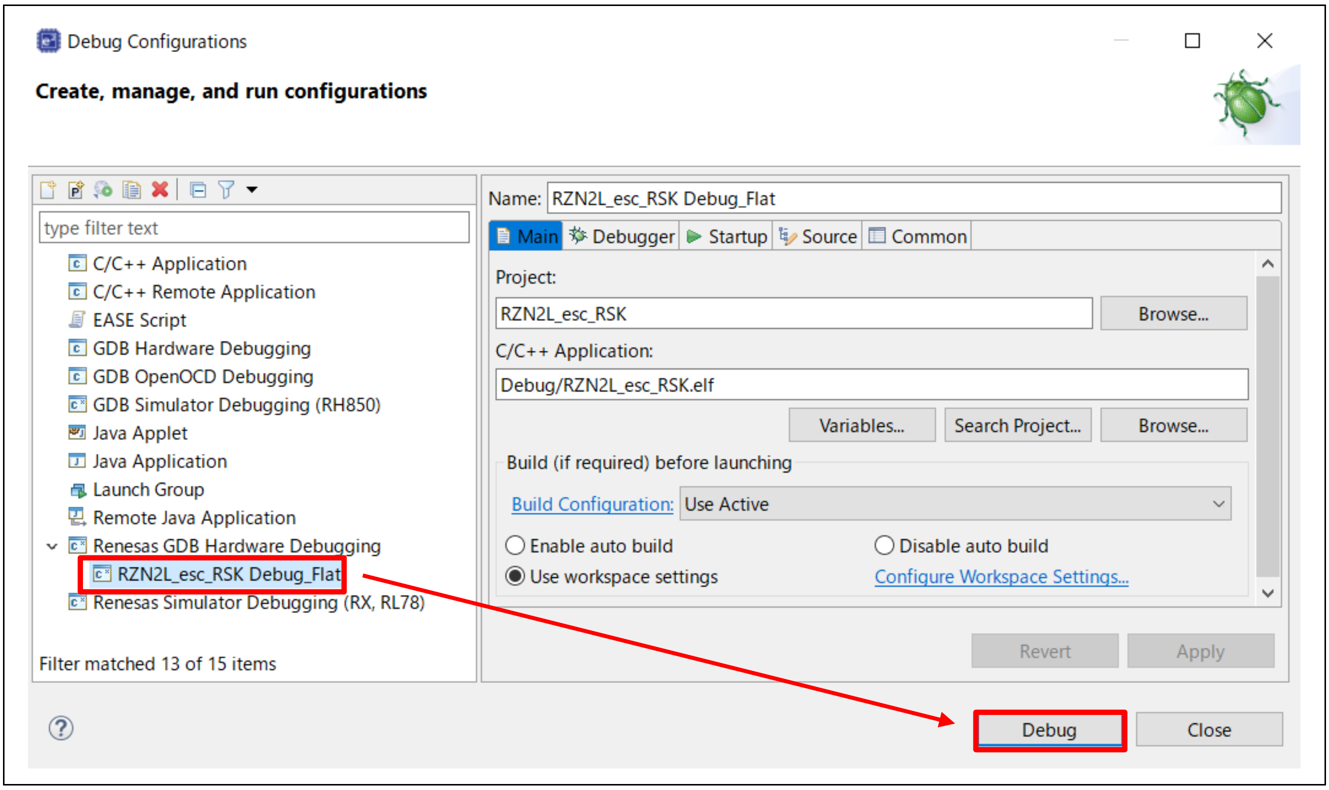


5. Press the “RESET” switch of the RZ/N2L Industrial Network SOM Kit.

6. Connect J-Link to the SOM Kit, start debugging in the following procedure.
In [Project Explorer] view, right click the node of project to be debugged and select [Debug As] → [Debug Configurations].


Project Explorer
RZN2L_esc_RSK
Binaries
Includes
rzn
rzn_gen
src
Debug
rzn_cfg
script
configuration.x
[RZN2L_esc_RSK] FSP Configuration
New
Go Into
Open in New Window
Show In Alt+Shift+W
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Source
Move...
Rename... F2
Import...
Export...
Renesas FSP Export
Build Project
Clean Project
Refresh F5
Close Project
Close Unrelated Project
Build Targets
Index
Build Configurations
Run As
Debug As
Team
Compare With
Restore from Local History...
MISRA-C
C/C++ Project Settings Ctrl+Alt+P
Renesas C/C++ Project Settings
HAL/Common Stacks
g_ioport I/O Port Driver on r_ioport
g_ethercat_ssc_port0
g_ether_phy0 Ethernet Driver on r_ether_phy
g_ether_selector0 Ethernet Driver on r_ether_selector
Event Links Stacks Components
スマート・ブラウザー スマート・マニュアル Debug
1 GDB OpenOCD Hardware Debugging (DSF)
2 GDB Simulator Debugging (RH850)
3 Local C/C++ Application
4 Renesas GDB Hardware Debugging
5 Renesas Simulator Debugging (RX, RL78)
Debug Configurations...

[Renesas DBG Hardware Debugging] → [RZN2L_SOM_ESC_CiA402 Debug_Flat] item, then press [Debug]



Debug Configurations

Create, manage, and run configurations

Name: RZN2L_esc_RSK Debug_Flat

Project: RZN2L_esc_RSK

C/C++ Application: Debug/RZN2L_esc_RSK.elf

Build (if required) before launching

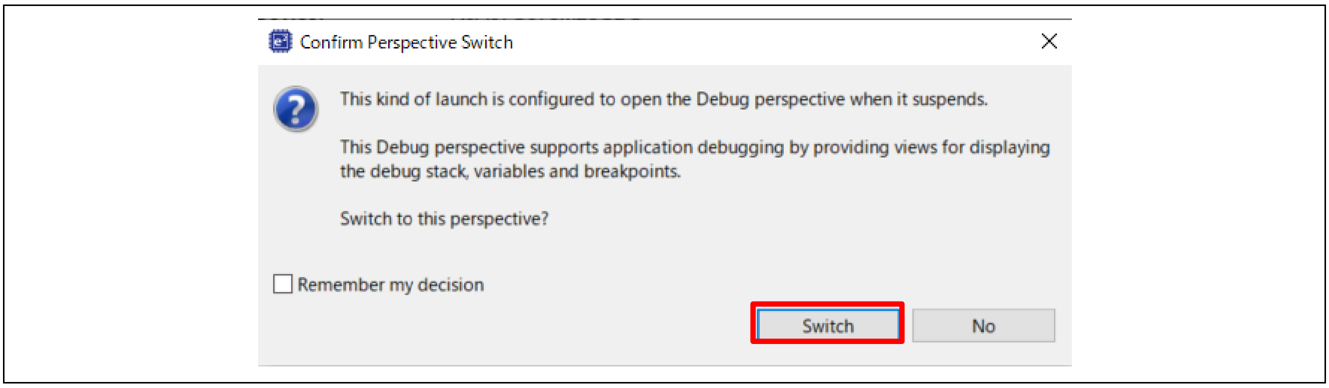
Build Configuration: Use Active

○ Enable auto build ○ Disable auto build

● Use workspace settings

Filter matched 13 of 15 items

Following dialog will appear, so switch to the debug screen.



Confirm Perspective Switch

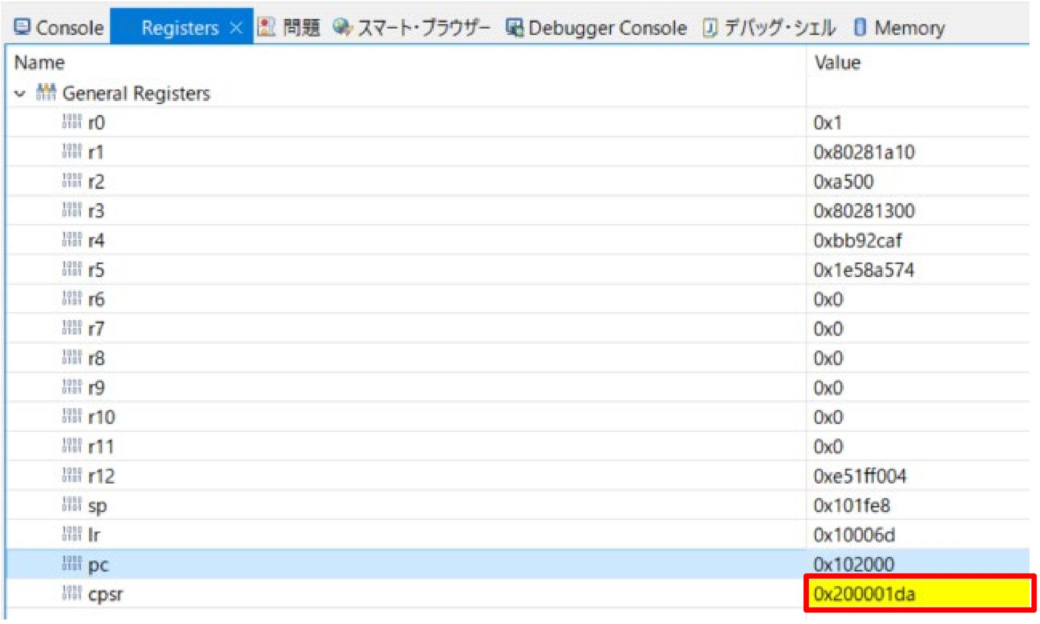
This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective supports application debugging by providing views for displaying the debug stack, variables and breakpoints.

Switch to this perspective?

□ Remember my decision

7. Before running the loaded program, change the CPSR register of CR52 general register on Registers tabs.
   Change the register value from “0x200001fa” to “**0x200001da**”.
   If the CPSR register value has not changed, program will stop at Default_Handler () at run time.



Name	Value
General Registers	
r0	0x1
r1	0x80281a10
r2	0xa500
r3	0x80281300
r4	0xbb92caf
r5	0x1e58a574
r6	0x0
r7	0x0
r8	0x0
r9	0x0
r10	0x0
r11	0x0
r12	0xe51ff004
sp	0x101fe8
lr	0x10006d
pc	0x102000
cpsr	0x200001da

8. Press the "Resume" button for the project. Program will stop at hal_entry ().
   Press the "Resume" button again. Program is running.

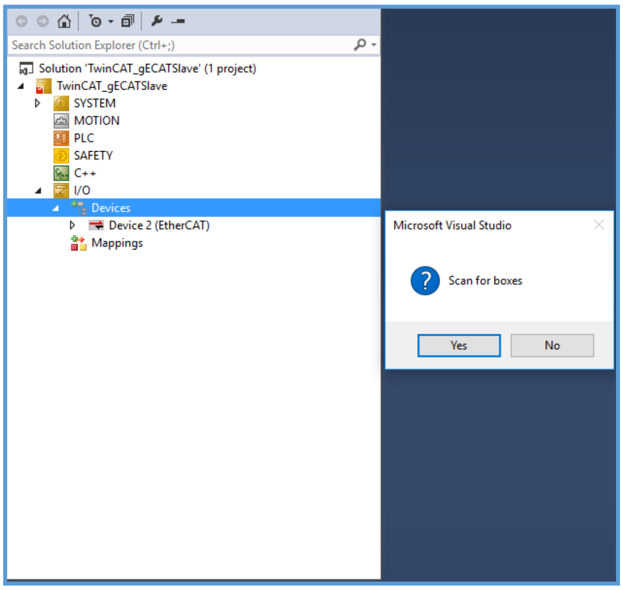
6. Connecting to TwinCAT3

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

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

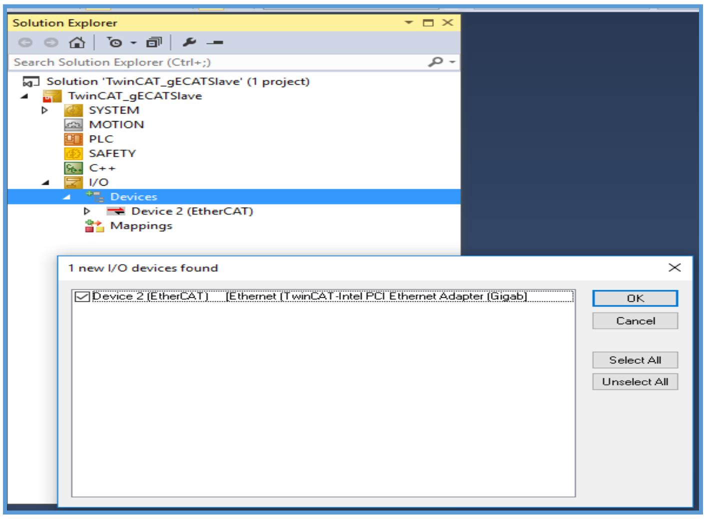
6.1 Scanning I/O Devices

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



Scan for boxes

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

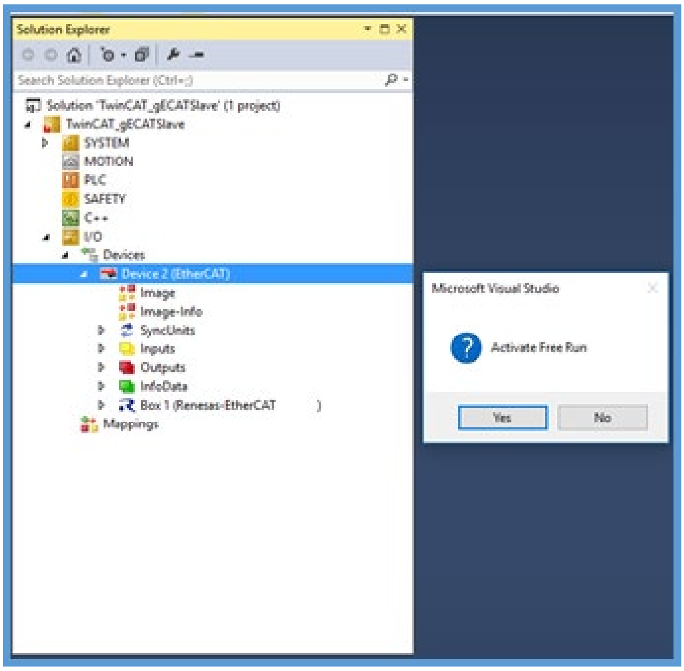


1 new I/O devices found

☑ Device 2 (EtherCAT) [Ethernet (TwinCAT-Intel PCI Ethernet Adapter (Gigab)

Note). This will list EtherCAT master if a valid slave is present in the network.

3. (Activate slave): The slave is listed in the boxes, in our case “Renesas EtherCAT” in box1 shown in figure below. Press activate free run.



Activate Free Run

6.2 Updating EEPROM Data

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

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

2. Select the [EtherCAT] tab.

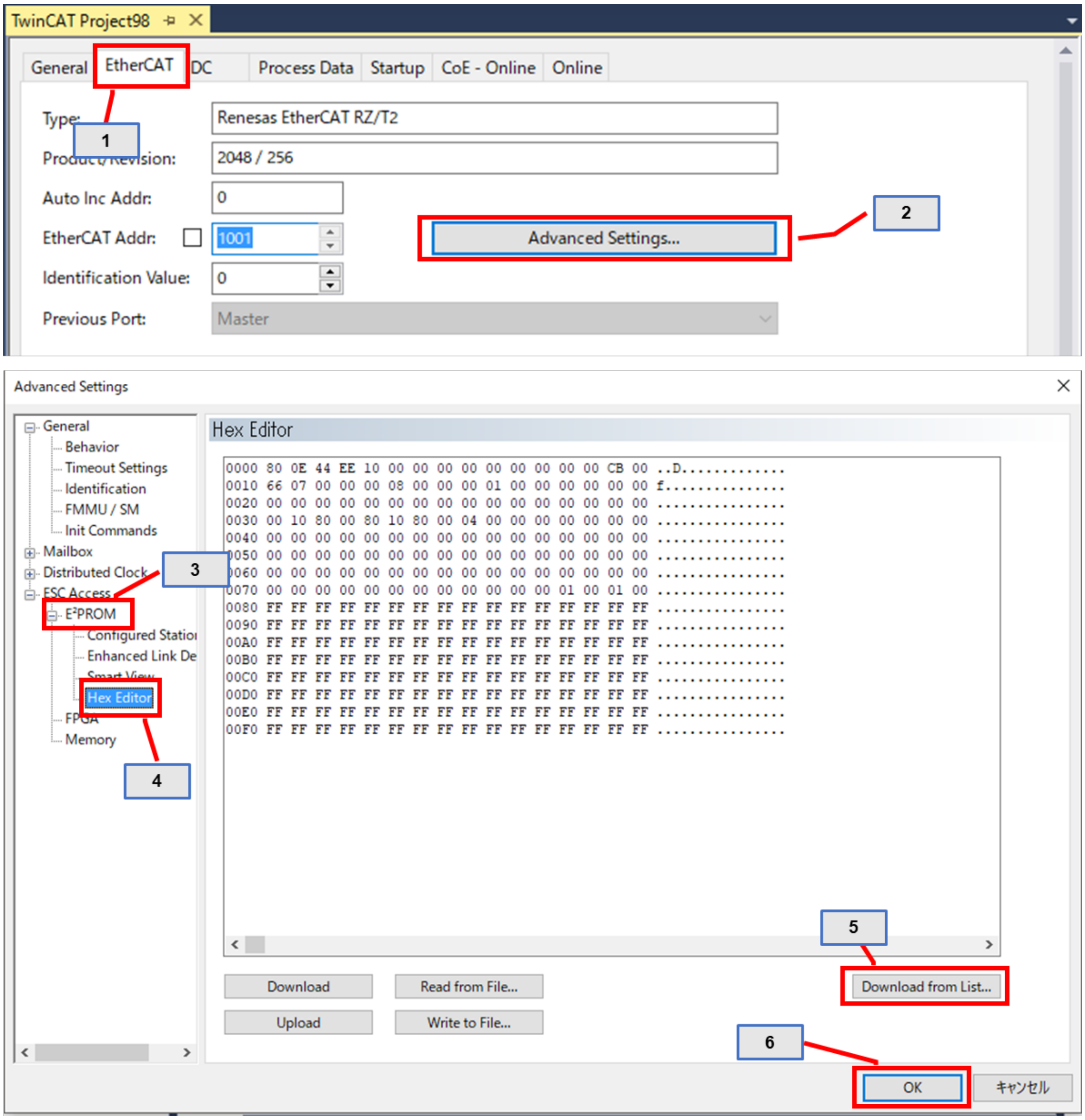
3. Click the [Advanced Setting] button.

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

5. Select [Download from List] → Select ESI File
   "*common\ecat_CiA402\ESI\Renesas EtherCAT RZN2 CiA402.xml*"

6. Select "Renesas EtherCAT RZ/N2 Cia402 2port" or "Renesas EtherCAT RZ/N2 CiA402 3port"

7. OK and Download.



Type: Renesas EtherCAT RZ/T2

Product/Revision: 2048 / 256

Auto Inc Addr: 0

EtherCAT Addr: □ 1001

Identification Value: 0

Previous Port: Master

Hex Editor

```
0000 80 0E 44 EE 10 00 00 00 00 00 00 00 00 00 CB 00 ..D.............
0010 66 07 00 00 00 08 00 00 00 01 00 00 00 00 00 00 f...............
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0030 00 10 80 00 80 10 80 00 04 00 00 00 00 00 00 00 ................
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0070 00 00 00 00 00 00 00 00 00 00 00 00 01 00 01 00 ................
0080 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
0090 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
00A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
00B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
00C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
00D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
00E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
00F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ................
```

Figure 6.1: EEPROM update

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

1. SSC Tool → [Tool] → [EEPROM Programmer].

2. [FILE] → [OPEN] →Browse and select the ESI file.

3. [FILE] → [Save AS] → Select type as binary.

4. A binary file will be generated in the specified folder.

5. [Read from File] Select the ESI binary file → [Download].

6. Confirm the write status using [Upload] option.

After the data is replaced, restart the RZ/N2L (by turning it off and on, or resetting it) so that the new data is applied to the microcomputer. Execute [Restart TwinCAT System].

6.3 CiA402 Drive Profile check

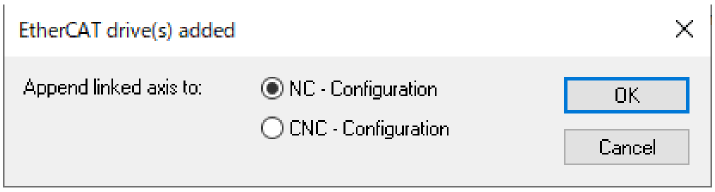
In this chapter, check the operation of the CiA402 drive profile.

6.3.1 Rescanning the Device

1. Press the [Restart TwinCAT (Config Mode)] button.
2. In the [Restart TwinCAT System in Config Mode] dialog box, click on [OK].
3. In the [Load I/O Devices] dialog box, click on [Yes].
4. In the [Active Free Run] dialog box, click on [Yes].

This will be OK if "Box 1" in the System Manager tree has turned to "Box 1 (RZ/N2 EtherCAT CiA402)".

Note). When scanning the device, the CiA402's ESI is used, so the axis configuration settings are displayed.
Select [NC-Configuration]



EtherCAT drive(s) added

Append linked axis to: ● NC - Configuration ○ CNC - Configuration

6.3.2 Checking the Operation Mode

(1) If double-click on [Box 1] in the System Manager tree, panel will be displayed on the right side of the screen.

(2) Select the [Online] tab and check that "Current Status" has turned to "OP".

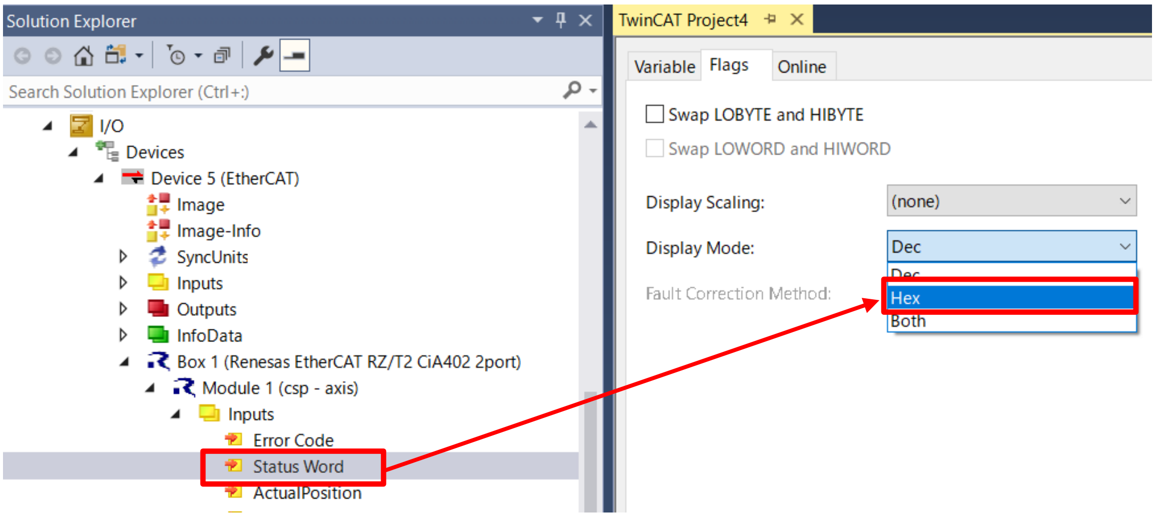
(3) In the System Manager tree, expand + on the left side of "Box 1".

6.3.3 CiA402 State Transition

To check the operation in csp and csv modes, the state must be changed to "Operation Enabled" in both modes.

Change the state by setting a value in the "Control Word" object and check the state by confirming the value of the "Status Word" object.

(1) Select "Inputs" → "Status Word" in the Solution Explorer tree and then select the [Flags] tab on the right-side panel. value will then be displayed.
Change the display mode from [Dec] to [Hex]



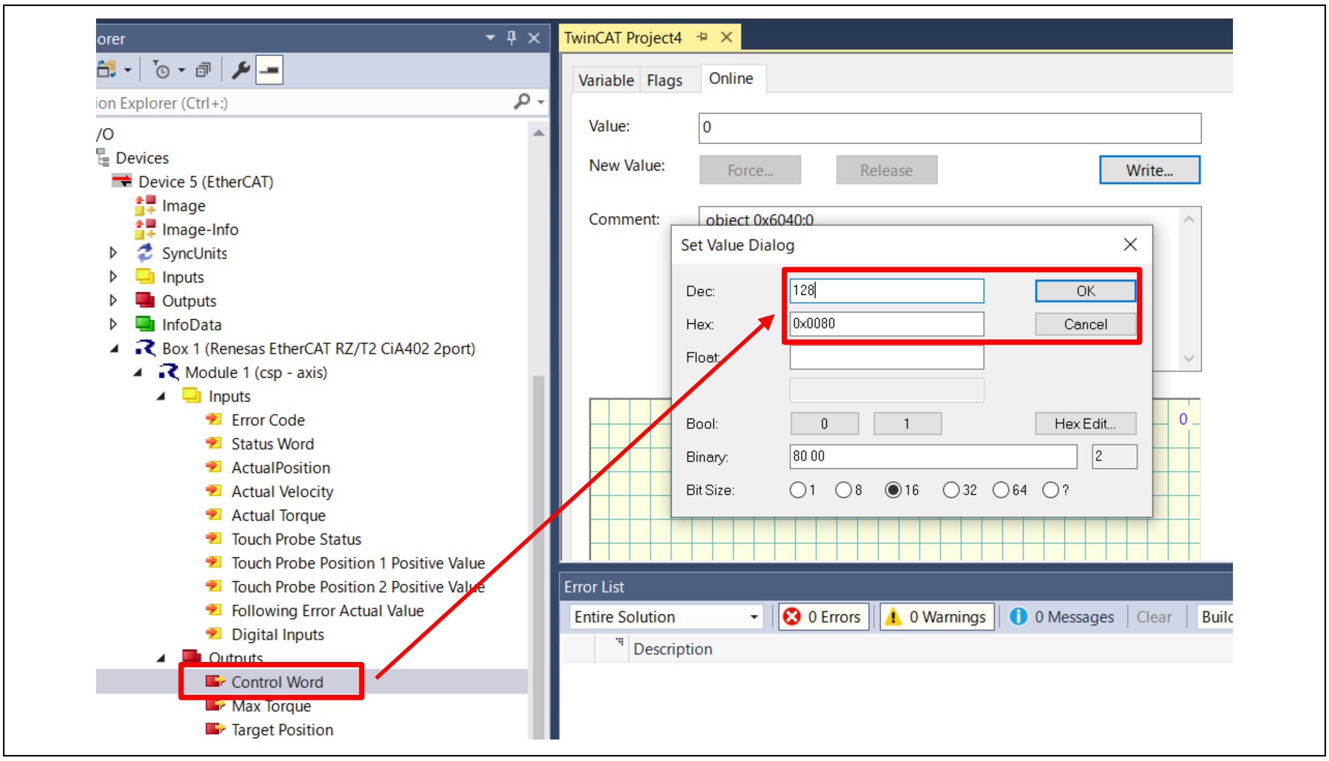
□ Swap LOBYTE and HIBYTE
□ Swap LOWORD and HIWORD

Display Scaling: (none)

Display Mode: Dec

Fault Correction Method:

(2) Check the operation of the profile. Initialize the control word by setting it to 128 (Dec).



Value: 0

Comment: object 0x6040:0

Dec: 128

Hex: 0x0080

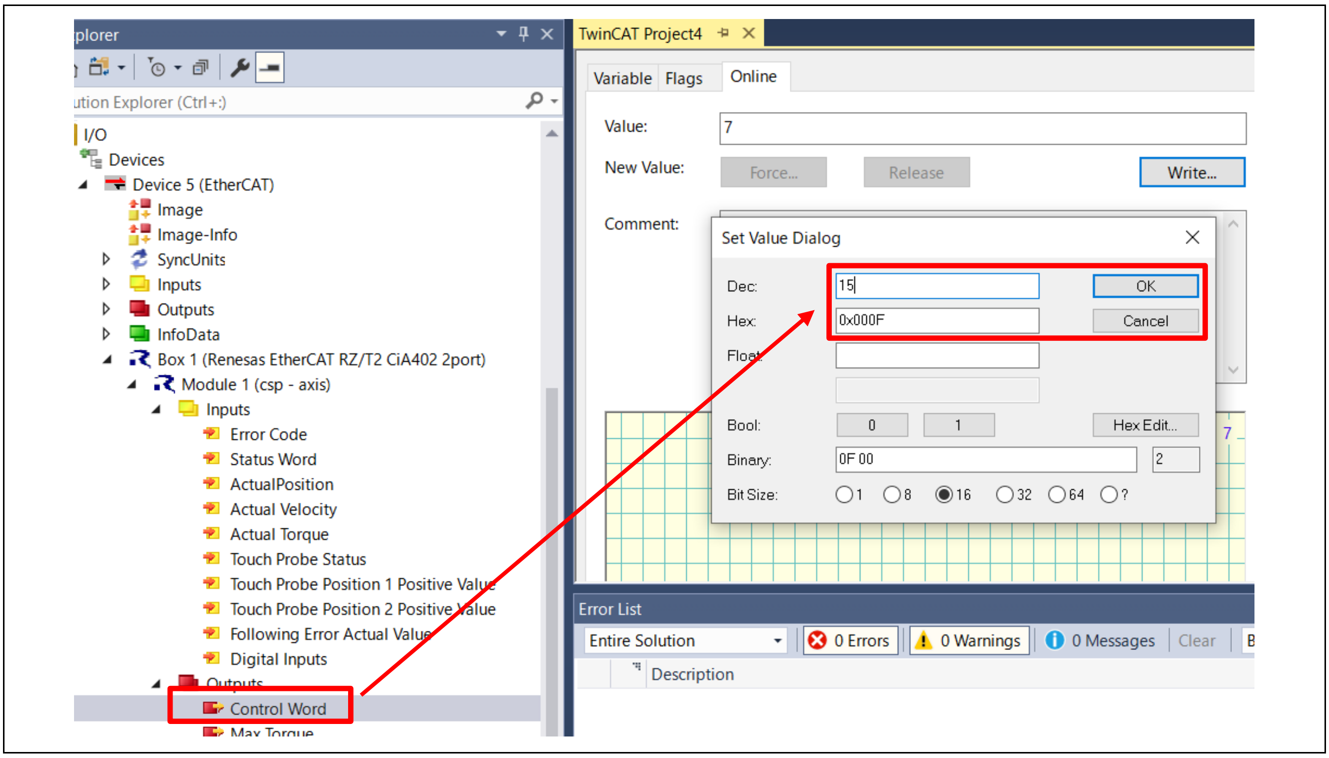
Float:

Bool: 0 1

Binary: 80 00 2

Bit Size: ○1 ○8 ●16 ○32 ○64 ○?

(3) Select "Outputs" → "Control Word" in the System Manager tree and then select the [Online] tab on the right-side panel. value will then be displayed.
Click on [Write] and set values from [7] to [15] in that order.



Value: 7

Comment:

Dec: 15

Hex: 0x000F

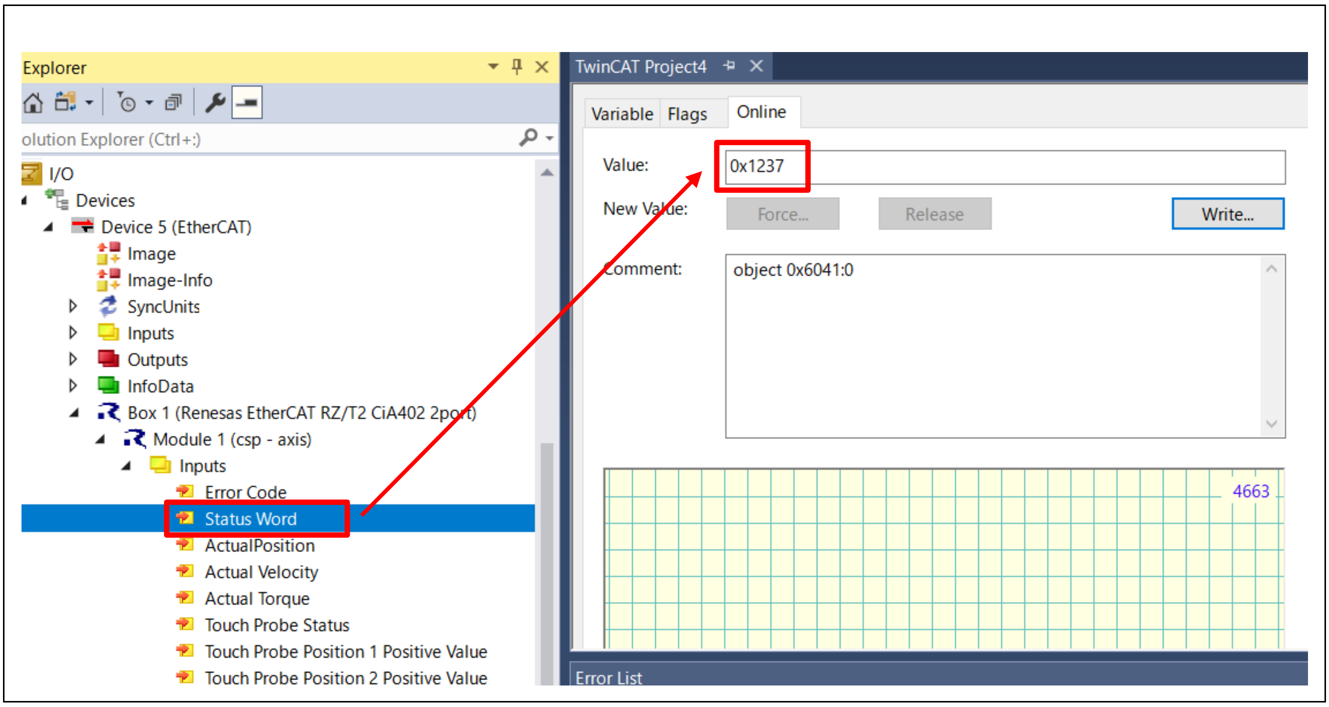
Float:

Bool: 0 1

Binary: 0F 00 2

Bit Size: ○1 ○8 ●16 ○32 ○64 ○?

(4) Select "Inputs" → "Status Word" in the System Manager tree and then select the [Online] tab on the right-side panel. value will then be displayed.
If the value is [0x1237], the state is "Operation Enabled". Proceed to the next step.
If the value is [0x1208], the state is "Fault" for some reason. Set "Control Word" to [128] once and return to step (1).



Value: 0x1237

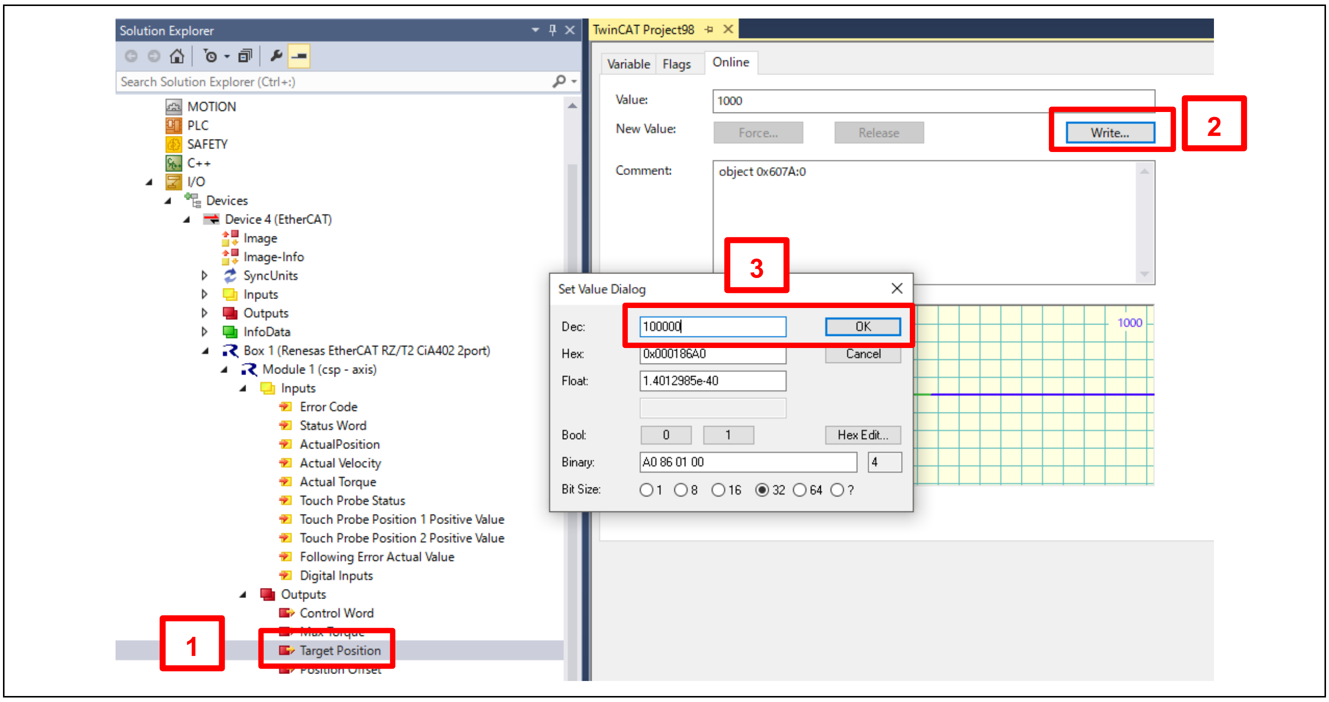
New Value:

Comment: object 0x6041:0

6.3.4 csp Mode

Check that "Module 1" is displayed as "Module 1 (csp-axis)".

(1) Select "Outputs" → "Target Position" in the System Manager tree and then select the [Online] tab on the right-side panel. A value will then be displayed.
Click on [Write] and set a desired value. As an example, set [100000] here.



Value: 1000

Comment: object 0x607A:0

Dec: 100000

Hex: 0x000186A0

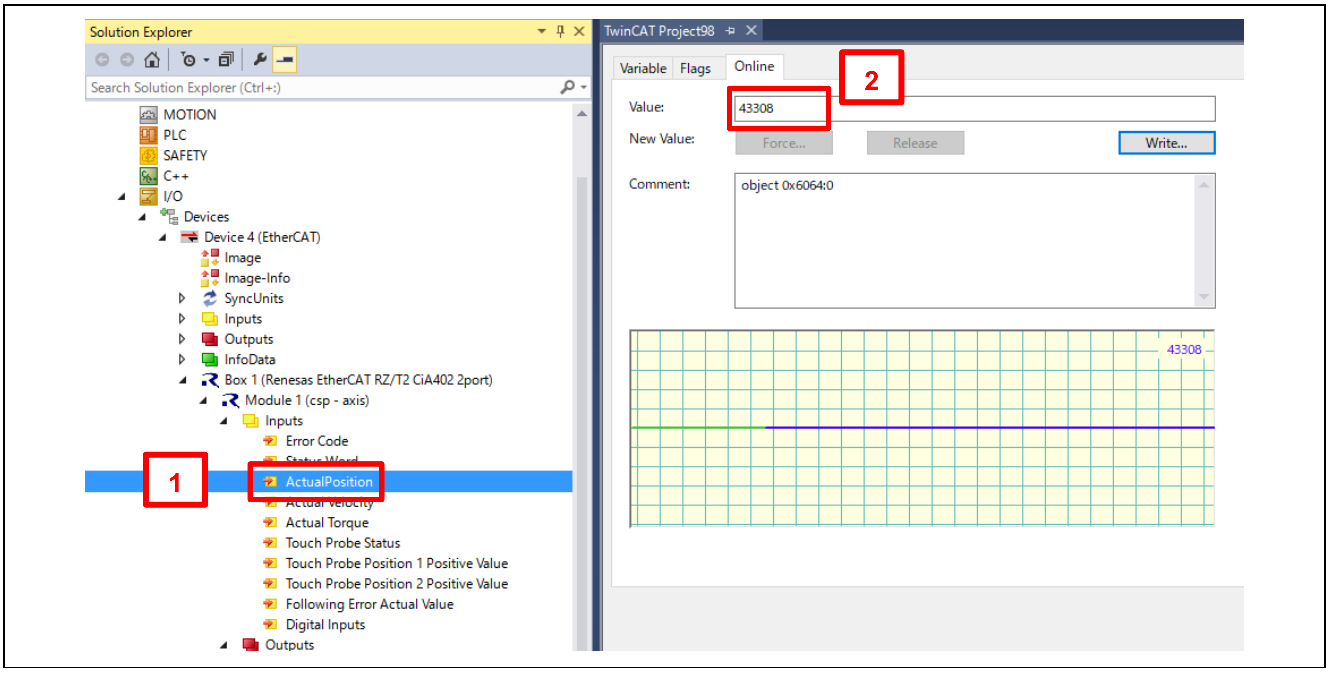
Float: 1.4012985e-40

Bool: 0 1

Binary: A0 86 01 00 4

Bit Size: ○ 1 ○ 8 ○ 16 ● 32 ○ 64 ○ ?

(2) Select "Inputs" → "Actual Position" in the System Manager tree and then select the [Online] tab on the right-side panel. A value will then be displayed.



Value: 43308

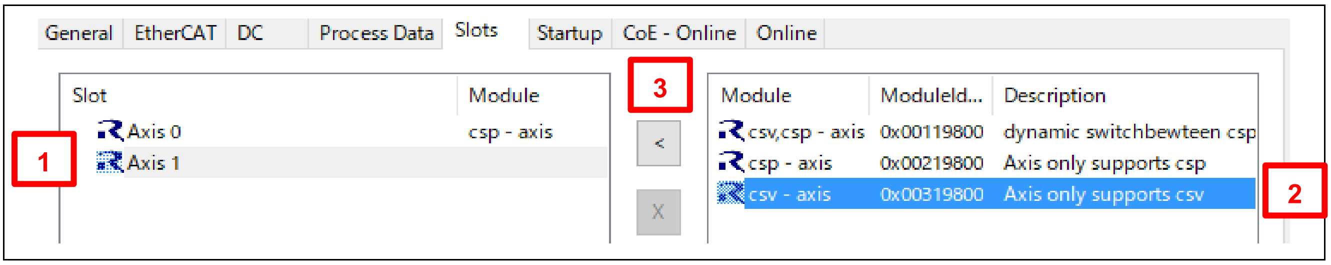
Comment: object 0x6064:0

6.3.5 csv Mode

Change the operation mode

(1) In the System Manager tree, select "Box 1 (RZ/N2 EtherCAT CiA402)" and then select the [Slots] tab on the right-side panel.

(2) If select "Axis 1" under "Slot" in the left frame of the tabbed page, modules which can be added will be displayed in the right frame of the page, so add "csv-axis".



Slot	Module
Axis 0	csp - axis
Axis 1	

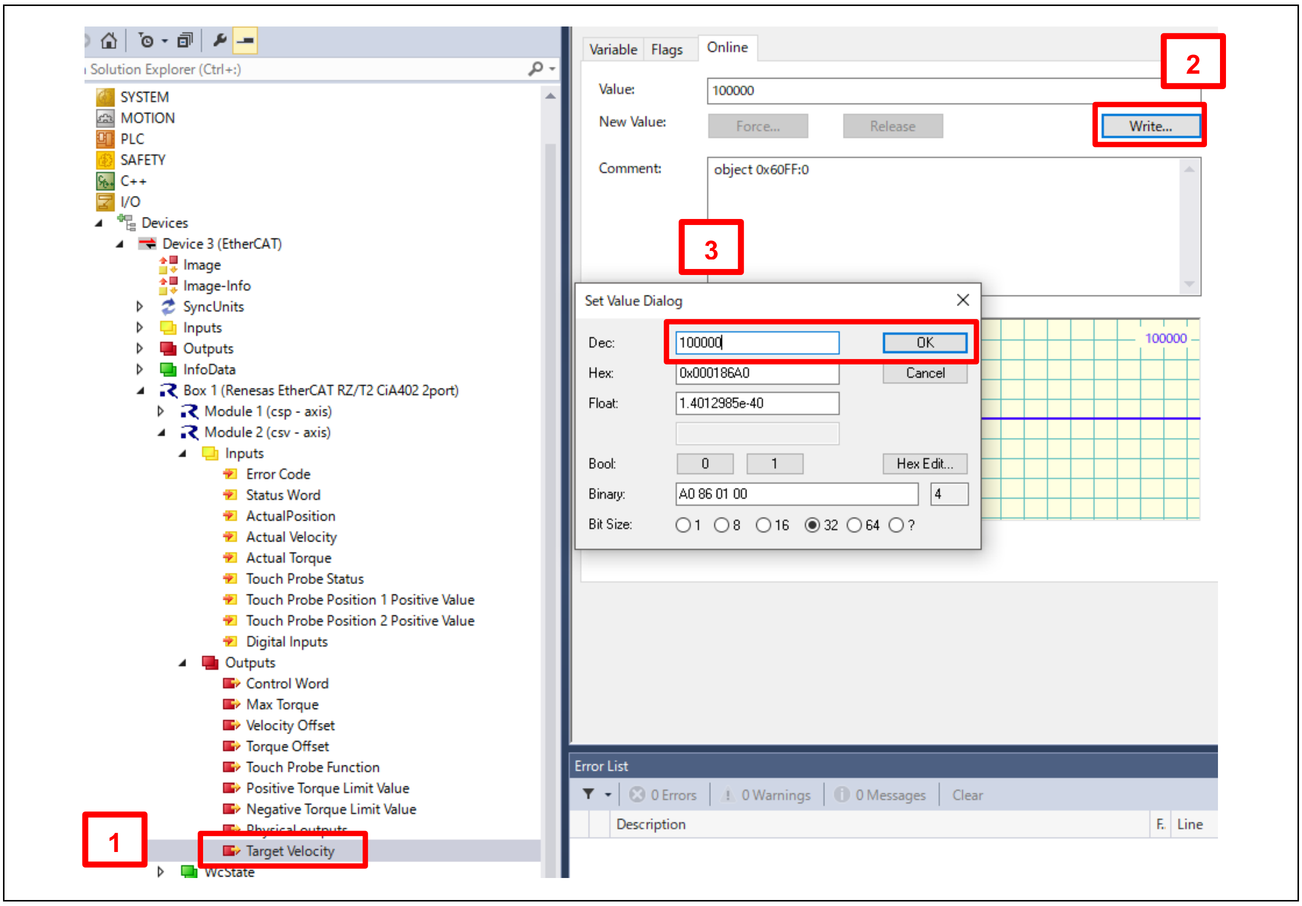
Module	ModuleId...	Description
csv,csp - axis	0x00119800	dynamic switchbewteen csp
csp - axis	0x00219800	Axis only supports csp
csv - axis	0x00319800	Axis only supports csv

(3) Check that "Module 2 (csv-axis)" has been added in the System Manager tree.

(4) Execute “Reload Devices” to reflect the configuration.

(5) Follow the procedure described in section 4.7 to change the state to "Operation Enabled".

(6) Select "Outputs" → "Target Velocity" in the System Manager tree and then select the [Online] tab on the right-side panel. A value will then be displayed.
Click on [Write] and set a desired value. As an example, set [100000] here.



Value: 100000

Comment: object 0x60FF:0

Dec: 100000

Hex: 0x000186A0

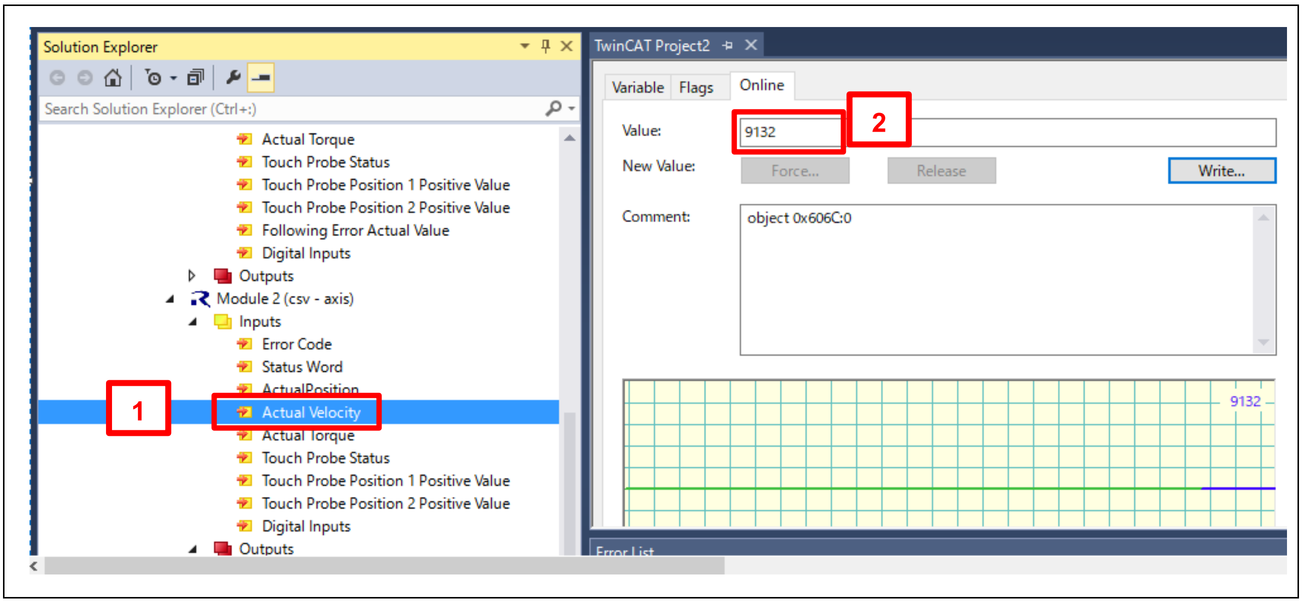
Float: 1.4012985e-40

Bool: 0 1

Binary: A0 86 01 00 4

Bit Size: ○1 ○8 ○16 ●32 ○64 ○?

(7) Select "Inputs" → "Actual Velocity" in the System Manager tree and then select the [Online] tab on the right-side panel. A value will then be displayed.
Check that the value set in "Target Velocity" allows incrementation up to [100000].



Value: 9132

New Value:

Comment: object 0x606C:0

7. CiA402 Drive Profile

The CiA402 drive profile is a device profile for driving motors and motion control and mainly defines functional operations for servo drives, sine-wave inverters and stepping motor controllers. In this profile, the multiple operation modes and corresponding parameters are defined as an object dictionary. Also, Finite State Automaton (FSA) to define the internal and external behavior in every state is included. When changing the state, the result after transition is reflected in the status word object that shows the current state by specifying the state through the control word object. The control word and various command values (such as speed) are assigned to RxPDO, and the status word and various real values (such as position) are assigned to TxPDO. Please see the contents of the CiA402 standard for more details.

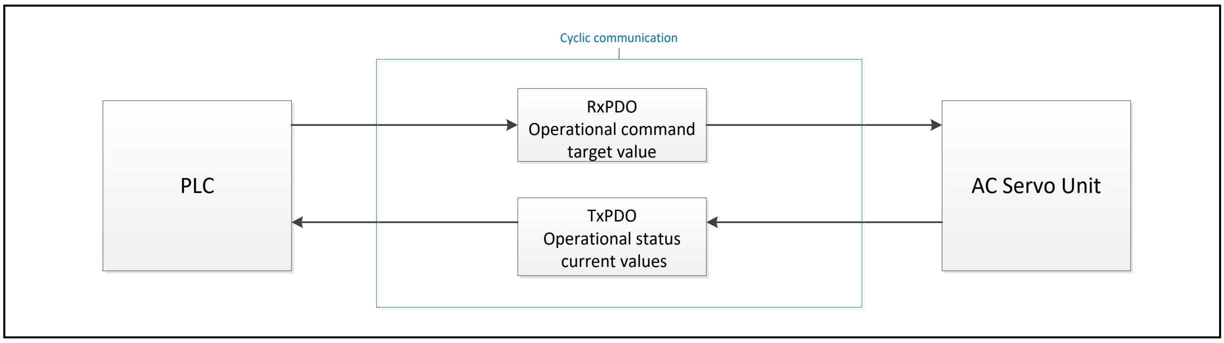


Figure 7-1 CiA402 Communication Flow

7.1 Operation Modes

In the application note, the following modes are supported from among the operation modes defined in the CiA402 standard.

Table 7-1 List of Supported Operation Modes

Operation Mode	Support
Profile position mode	No
Velocity mode (frequency converter)	No
Profile velocity mode	No
Profile torque mode	No
Homing mode	No
Interpolated position mode	No
Cyclic synchronous position mode	Yes
Cyclic synchronous velocity mode	Yes
Cyclic synchronous torque mode	No
Cyclic synchronous torque mode with commutation angle	No
Manufacturer specific mode	No

7.2 State Transition

In this application note, the following is supported as FSA defined in the CiA402 standard.

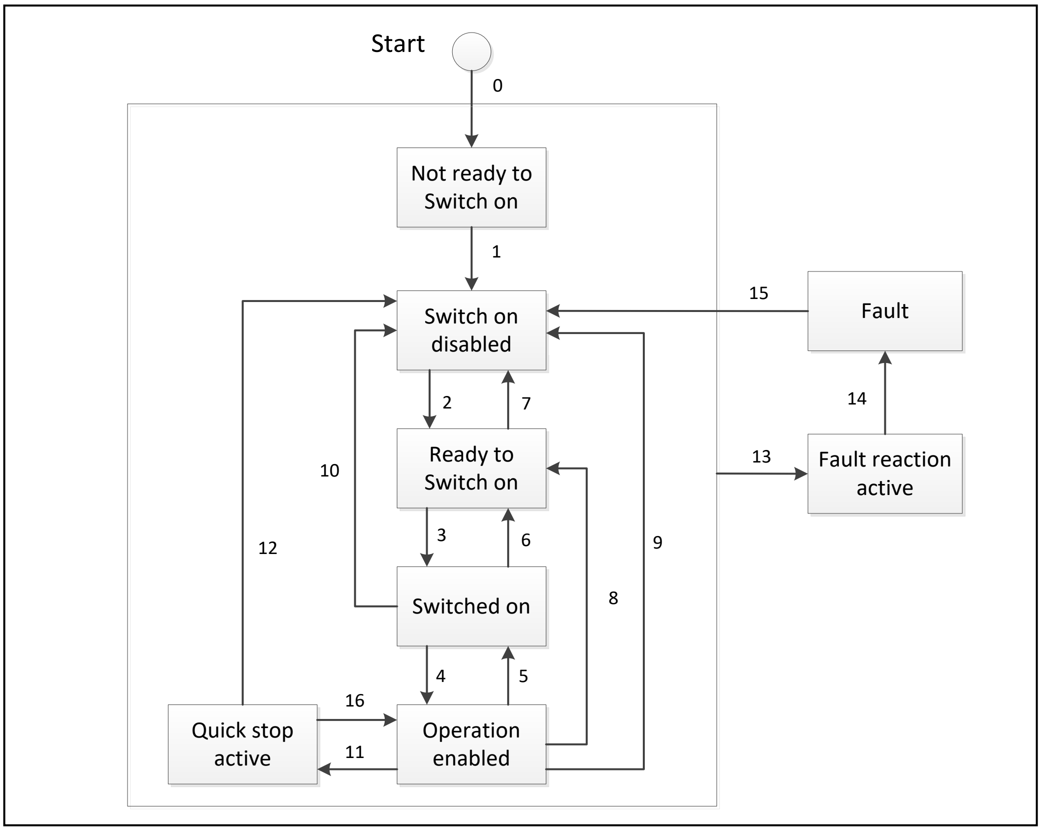


Figure 7-2 CiA402 State Transition Diagram

7.3 Object Dictionary

The following is the list of the object dictionaries supported in this application note.

Table 7-2 List of Supported Object Dictionaries

Operation Mode	OBJECT Name	INDEX	Category	Access	Data Type	PDO Mapping
Cyclic synchronous position mode + Cyclic synchronous velocity mode	Position actual value	0x6064	Mandatory	ro	INT32	Yes
	Following error window	0x6065	Optional	rw	UINT32	No
	Velocity actual value	0x606C	Conditional	ro	INT32	Yes
	Max torque	0x6072	Optional	rw	UINT16	Yes
	Torque actual value	0x6077	Conditional	ro	INT16	Yes
	Target position	0x607A	Optional	rw	INT32	Yes
	Software position limit	0x607D	Optional	c,rw	INT32	No
	Position offset	0x60B0	Optional	rw	INT32	Yes
	Velocity offset	0x60B1	Optional	rw	INT32	Yes
	Torque offset	0x60B2	Optional	rw	INT16	Yes
	Following error actual value	0x60F4	Optional	ro	INT32	Yes
	Target velocity	0x60FF	Conditional	rw	INT32	Yes

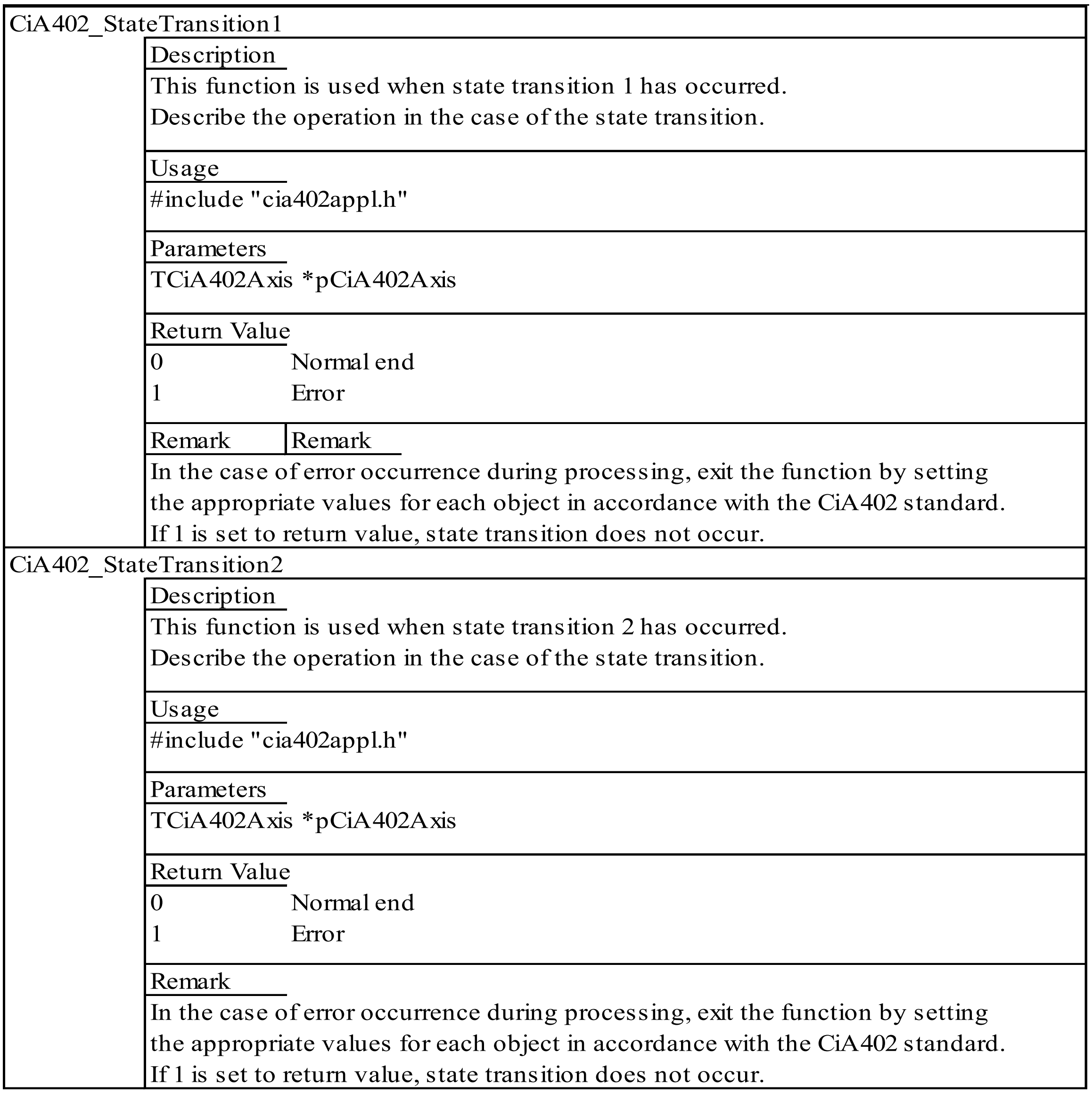
Function Group	OBJECT Name	INDEX	Category	Access	Data Type	PDO Mapping
Torque Limiting	Positive torque limit value	0x60E0	Conditional	rw	UINT16	Yes
	Negative torque limit value	0x60E1	Conditional	rw	UINT16	Yes
Homing	Home Offset	0x607C	Optional	rw	INT32	No
	Homing speeds	0x6099	Conditional	c,rw	UINT32	No
Touch Probe	Touch probe function	0x60B8	Optional	rw	UINT16	Yes
	Touch probe status	0x60B9	Optional	ro	UINT16	Yes
	Touch probe pos 1 pos value	0x60BA	Optional	ro	INT32	Yes
	Touch probe pos 2 pos value	0x60BC	Optional	ro	INT32	Yes
Gear ratio	Gear ratio	0x6091	Optional	c,rw	UINT32	No

Other object	OBJECT Name	INDEX	Category	Access	Data Type	PDO Mapping
Controlling the power drive system	Error code	0x603F	Optional	ro	UINT16	Yes
	Controlword	0x6040	Mandatory	rw	UINT16	Yes
	Statusword	0x6041	Mandatory	ro	UINT16	Yes
	Quick stop option code	0x605A	Optional	rw	INT16	No
	Shutdown option code	0x605B	Optional	rw	INT16	No
	Disable operation option code	0x605C	Optional	rw	INT16	No
	Halt option code	0x605D	Optional	rw	INT16	No
	Fault reaction option code	0x605E	Optional	rw	INT16	No
	Modes of operation	0x6060	Optional	rw	INT8	Yes
	Modes of operation disp	0x6061	Optional	ro	INT8	Yes
	Supported drive modes	0x6502	Mandatory	ro	INT32	No
General object	Motor type	0x6402	Optional	rw	INT16	No
Position control function	Position demand value	0x6062	Optional	ro	INT32	No
	Position actual internal value	0x6063	Optional	ro	INT32	No
	Position window	0x6067	Optional	rw	UINT32	No
Optional application FE	Digital inputs	0x60FD	Optional	ro	UINT32	Yes
	Digital outputs	0x60FE	Optional	c,rw	UINT32	No,Yes

7.4 Implementing the Motor Control Program

According to the CiA402 standard from the list of CiA402 protocol stack I/F functions in Table 7-3, implement the motor control application. Each function links the number of each state transition of CiA402 FSA shown in Figure 7-2 and the corresponding function is called in case of state transition. In each function, describe the processing that calls the motor control program or the relevant processing of the main CPU.

Table 7-3 List of CiA402 Protocol Stack I/F Functions



CiA402_StateTransition1		
	Description	This function is used when state transition 1 has occurred. Describe the operation in the case of the state transition.
	Usage	#include "cia402appl.h"
	Parameters	TCiA402Axis *pCiA402Axis
	Return Value	0 Normal end 1 Error
	Remark	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.
CiA402_StateTransition2		
	Description	This function is used when state transition 2 has occurred. Describe the operation in the case of the state transition.
	Usage	#include "cia402appl.h"
	Parameters	TCiA402Axis *pCiA402Axis
	Return Value	0 Normal end 1 Error
	Remark	In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.

CiA402_StateTransition3	
	Description This function is used when state transition 3 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.
CiA402_StateTransition4	
	Description This function is used when state transition 4 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.
CiA402_StateTransition5	
	Description This function is used when state transition 5 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.

CiA402_StateTransition6	
	Description This function is used when state transition 6 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.
CiA402_StateTransition7	
	Description This function is used when state transition 7 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.
CiA402_StateTransition8	
	Description This function is used when state transition 8 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.

CiA402_StateTransition9	
	Description This function is used when state transition 9 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.
CiA402_StateTransition10	
	Description This function is used when state transition 10 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.
CiA402_StateTransition11	
	Description This function is used when state transition 11 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.

CiA402_StateTransition12	
	Description This function is used when state transition 12 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.
CiA402_LocalError	
	Description This function is used when state transition 13 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters UINT16 ErrorCode
	Return Value None
	Remark If the error corresponding to state transition 13 occurs, call this function after processing required and saving data at error location.
CiA402_StateTransition14	
	Description This function is used when state transition 14 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.

CiA402_StateTransition15	
	Description This function is used when state transition 15 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.
CiA402_StateTransition16	
	Description This function is used when state transition 16 has occurred. Describe the operation in the case of the state transition.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.
APPL_MOTOR_MotionControl_Main	
	Description Implement the motion control code when the state of CiA402 FSA is "Operation enabled". Describe the process for each mode of operation.
	Usage #include "cia402appl.h"
	Parameters TCiA402Axis *pCiA402Axis
	Return Value 0 Normal end 1 Error
	Remark At the initial state, this function is described in "main.c" and calls "CiA402_DummyMotionControl" function for reference.

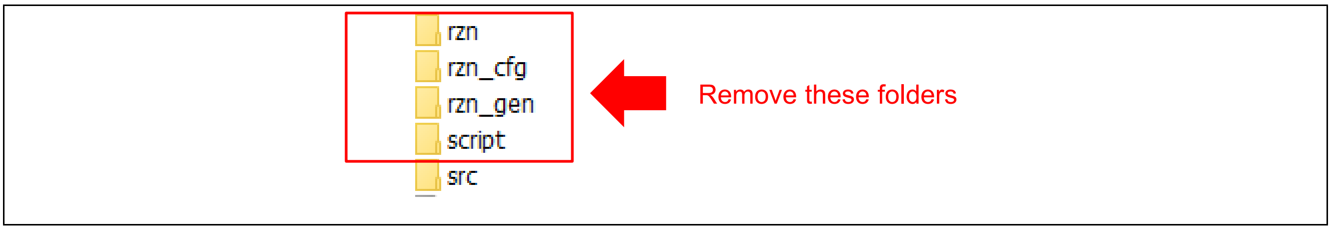
8. Appendix: FSP Configuration for VSC8531

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\ecat_CiA402\e2studio
- When using EWARM, \project\rzn2l_som\ecat_CiA402\ewarm



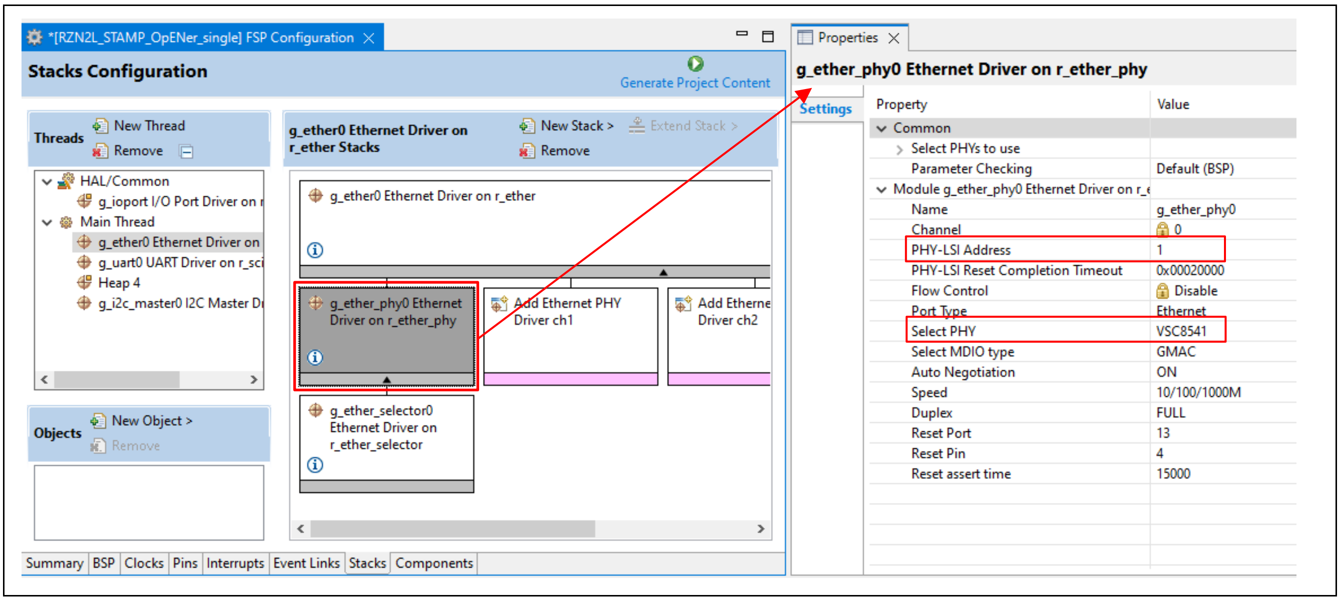
rzn
rzn_cfg
rzn_gen
script
src

Remove these folders

Figure 8-1 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 8-2.
Configuration value for VSC8531 shows in Table 8-1.



*[RZN2L_STAMP_OpENer_single] FSP Configuration
Stacks Configuration
Generate Project Content
Threads
New Thread
Remove
HAL/Common
g_ioport I/O Port Driver on r
Main Thread
g_ether0 Ethernet Driver on
g_uart0 UART Driver on r_sci
Heap 4
g_i2c_master0 I2C Master Dr
Objects
New Object >
Remove
g_ether0 Ethernet Driver on r_ether Stacks
New Stack >
Extend Stack >
Remove
g_ether0 Ethernet Driver on r_ether
g_ether_phy0 Ethernet Driver on r_ether_phy
Add Ethernet PHY Driver ch1
Add Etherne Driver ch2
g_ether_selector0 Ethernet Driver on r_ether_selector
Summary
BSP
Clocks
Pins
Interrupts
Event Links
Stacks
Components
Properties
Settings

g_ether_phy0 Ethernet Driver on r_ether_phy

Property	Value
Common	
Select PHYs to use	
Parameter Checking	Default (BSP)
Module g_ether_phy0 Ethernet Driver on r_e	
Name	g_ether_phy0
Channel	0
PHY-LSI Address	1
PHY-LSI Reset Completion Timeout	0x00020000
Flow Control	Disable
Port Type	Ethernet
Select PHY	VSC8541
Select MDIO type	GMAC
Auto Negotiation	ON
Speed	10/100/1000M
Duplex	FULL
Reset Port	13
Reset Pin	4
Reset assert time	15000

Figure 8-2 Ethernet Driver Configuration for VSC8531 (e.g. ETH0)

Table 8-1 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() */
```

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Feb 6, 2023	-	First edition 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/.