

RZ/N2L Group

Application Note

R01AN6797EJ0110 Rev.1.10 Aug 7, 2023

RZ/N2L Industrial Network SOM Kit Application Note: EtherCAT 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	Build and debug sample code for EWARM	13
5.2	Setting sample code for GCC	16
5.2.1	Erasing the flash memory	16
5.2.2	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	Sync Modes	27
6.3.1	Free Run	27
6.3.2	2 Sync Manager Synchronization	27
6.3.3	B DC Synchronization	28



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

6.4	Testing the I/O Controller	.30
7.	Appendix: FSP Configuration for VSC8531 and SSC port	.33
Rev	ision History	.36



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	12C	Inter-Integrated Circuit
7	MB	Mailbox
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
- I/O function:
 - I/O Input DIP SW
 - I/O Output LED



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



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	• e ² 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. <u>https://www.segger.com/downloads/jlink/</u>



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 and SSC port".



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 🔿
m	EEPROM Programmer
	EtherCAT Development Docume
SS	SSC Tool

2. Select File > New.

ssc	EtherCAT	Slave* - Slav	ve Stack	Code Tool	
File	Proje	ect Tool	Help		
	New	Ctrl+N		Slave Settings	
2	Open	Ctrl+O		SSC Version	5.12
Save Ctrl+S			Config File Vers	sion1.4.0.0	
	Save Ac uir	Save As	vine	File name	
	Jave A	3	_	aceapplic	
	Exit			aoeapplh	
	- Mailbo:	X		applInterfac	eh:
	Compil	er		bootmode.c	

1. Click the [Import] button and select the SSC Tool configuration file at,

"common\ecat_IO\SSCconfig\Renesas_RZN2_config.xml"

Slave Stack C	ode Tool New Project	x
 Default Custom 	EL9800 2Axis CiA402 Sample	~
Default Slav All settings	eStackCode configuration. are available.	
	_	
Import		OK



2. After the configuration file is read, the window changes as follows:

Slave Stack Code Tool New Project	×
◯ Default	
Custom Renesas RZ/N2 < Renesas Electronics Corp >	~
Vendor: Renesas Electronics Corp. (0x766). Version: 0.0.0.2 NOTE: This configuration is not provided by Beckhoff Automation and files or file fragments may be added which are NOT covered by the license from Beckhoff Automation GmbH.	
Shall be set if the Slave code executes on an Renesas development board for the RZ/N2.	
Import	

Once the configuration file is read, it is registered in Custom and is selectable from the drop-down list.

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

5 EtherCAT Slave* - Slave Stack	Code Tool		-		Х
File Project Tool Help					
Slave Project Navigation	Slave Settings				
- EtherCAT Slave - SlaveInformation - Generic	SSC Version 5.12 Config File Version1.4.0.0				
Hardware EtherCAT State Machine	File name	Description		Version	^
- Synchronisation	aceapplic	AoE ADS over EtherCAT		5.11	
Application Process Data	aoeapplh			5.11	
Mailbox	applInterface h	EcatAppl EtherCAT application		5.12	
Compiler	bootmode.c	ESM EtherCAT State Machine		5.12	
	bootmodeh			5.11	
	bootloaderappl.c	Bootloader Bootloader Sample		5.12	
	bootloaderapp1h			5.12	
	cia402appl.c	CiA402appl CiA402 Sample Applica	tion	5.12	
	cia402app1h			5.12	
	coeappl.c	CoE CAN Application Profile over E	EtherCAT	5.12	
	coeapplh			5.12	~
			Reload File Remove	Add File	(s)
	Conflicts				
	👥 Info 👍 Warning 🌘	8 Error			
New project created	2				.:
1					



4. Select Project > Create new Slave Files.

sse EtherCAT	Slave* - Slave Stack	Code Tool	
File Proje	ect Tool Help		
Slave Pr 🌀	Project Update		IES
Ethe -	Find Setting	Ctrl+F	in 5.12
			Version1.4.0.0
	Create new Slave Fil	es F5	ame
	vm i State Machine ronisation	аоеар	ple
📄 - Applica	 Application ProcessData Mailbox 		plh
Pro Mailbox			iterface h
Compil	ler	bootm	odelc

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

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#]	Change				
	Doc Folder	C#RZT2_esc_dual_WS1_RSK_beta3#CPU1#rz#fsp#src#r_ecat#utilities#ssc_config	Change				
Progress							



6. 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_IO\SSCconfig\Src"

		WTTNOK_DECONFOTOTHIZETSPESICET_ECONFORMACESESSC_CONTREFT%_TZ_ECONFOEPT@sp	
	Source Folder	C:¥RZT2_esc_dual_WS1_RSK_beta3¥CPU1¥rz¥fsp¥src¥r_ecat¥utilities¥ssc_config¥	Change
	ESI File	C:¥RZT2_esc_dual_WS1_RSK_beta3¥CPU1¥rz¥fsp¥src¥r_ecat¥utilities¥ssc_config¥	Change
	Doc Folder	C:¥RZT2_esc_dual_WS1_RSK_beta3¥CPU1¥rz¥fsp¥src¥r_ecat¥utilities¥ssc_config	Change
Progress			
"coeappl. "coeappl "ecatappl "ecatappl "ecatoe "ecatoe "ecatslv. "ecatslv. "ecatslv. "acatslv. "acatslv. "acatslv. "sobjdef.c "sdoserv. Generate	<pre>c" : new fil h" : new fil c" : new fil h" : new fil c" : new fil c" : new fil h" : new fil h" : new fil h" : new fil " : new fil " : new fil h" : new fil h" : new fil files finish vice descrip</pre>	e written e written le wri le wri e writ e writ e writ open Folder written written written e written e written	\r_ecat

7. Move the generated EtherCAT slave stack code to the EtherCAT application source folder. The sampleappl.c and sampleappl.h are stored in the destination "\application\ecat" folder. When moving the slave stack code to the application folder, be careful not to delete these files. Remove the original Src code (code generated in the Src folder by SSC) from the folder or exclude the Src code from your build target.

Source folder: (code generated in the Src folder by SSC) "common\ecat_IO\SSCconfig\Src"

Move destination folder:

"project\rzn2l_som\ecat_IO\ewarm (e2studio) \src\ ethercat\beckhoff"



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_IO\ESI\Renesas EtherCAT RZN2.xml"

4.2 Add Driver

Add the Ether driver for TwinCAT. (First time only) From the start menu, select [TwinCAT3] \rightarrow [Show Realtime Ethernet Compatible Devise].

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





5. Running the sample application

5.1 Build and debug 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_IO\ewarm\RZN2L_SOM_ESC_IO.eww"

RZN2L_SOM_ESC_IO.ewd	
RZN2L_SOM_ESC_IO.ewp	
RZN2L_SOM_ESC_IO.eww	

2. Open the "RZ Smart Configurator"

File Edit View Project I-jet 🛛	Too	Is Window Help	
🗄 🗅 🔛 🕋 🔚 🕹 🛍 (🕯	¢	Options	
Workspace 👻 🕂 🗄		Filename Extensions	
Debug		Configure Viewers	
Files 🌣		Configure Custom Argument Variables	
🖻 🌒 RZN2L_esc_RSK 🗸 4	6	Configure Tools	
⊢⊞ ■ Flex Software		IAR Project Converter	
uildinfo.ipcf		RZ Smart Configurator	

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

					0
acks Configuration					Generate Project Conte
nreads 🚯 New Thread 🛍 Remove 📄	HAL/Common Stacks			New Stack >	🗄 Extend Stack > 🔊 Remov
 ✓ ALL/Common <i>g_joport I/O Port Driver on r_jopor</i> <i>g_ethercat_ssc_port0 EtherCAT SSC</i> 	g_ioport I/O Port Driver on r_ioport	 g_ethercat_ssc_port0 Eth g_ether_phy0 Ethernet Driver on r_ether_phy g_ether_selector0 Ethernet Driver on r_ether_selector 	g_ether_phy1 Ethernet Driver on r_ether_phy g g_ether_selector1 Ethernet Driver on r_ether_selector	Add Ethernet PHY Driver ch2	g_timer0 Timer Driver on r_cmt

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



Compile Ctrl+F7 J_DIR\$/src	File Edit View Workspace Debug Files P RZN2L_es P Files Build C P Generi B Compo P Co	Pro □ □ □ □ □ □ □ □ □ □ □ □ □	ject I-jet Tools Window Help Add Files Add Group Import File List Add Project Connection Edit Configurations Edit Configurations Remove Create New Project Add Existing Project Options Alt+F7 Version Control System Make F7	<pre></pre>
<pre>Rebuild All h></pre>			Compile Ctrl+F7)J_DIR\$/src
Clean J_DIR\$/rzn/arm/CMSIS_5/CMSIS/Core_R/Include		9	Rebuild All	h>
Batch build F8 J_DIR\$/rzn/fsp/inc/api J_DIR\$/rzn/fsp/inc/instances J_DIR\$/rzn/fsp/inc/instances		¢ P	Clean Batch build F8	<pre>J_UIR\$/rzn/arm/CMSIS_S/CMSIS/Core_R/Include J_DIR\$/rzn/fsp/inc J_DIR\$/rzn/fsp/inc/api J_DIR\$/rzn/fsp/inc/instances </pre>

- 5. Press the "RESET" switch of the RZ/N2L Industrial Network SOM Kit.
- 6. While the board and I-jet are connected, click on the "Download and debug" button in the "Project" toolbar.

File Edit View Proj	ect I-jet Tools Window Help	
i 🗅 🗅 🔛 🚇 🖡	Add Files	🚽 < Q > 🤹 HI < 🟮 > 🕢 🔋 📗 🌒 🛲
Workspace	Add Group	
Debug 🛃	Import File List	
	Add Project Connection	<pre>" encoding="UTF-8" standalone="yes"?></pre>
Files	Edit Configurations	ion version="1.8" name="Flex Software">
□ ■ RZN2L_es		75024102 (/
├───	Remove	7/0004900
	Create New Project	
Genera 📬		<pre>DJ_DIR\$/rzn/arm/CMSIS_5/CMSIS/Core_R/Include</pre>
	Add Existing Project	JJ_DIR\$/rzn/tsp/inc
	Ontions Alt+E7	J_DIR\$/r2n/TSp/Inc/api
		1] DIR\$/rzn/fsn/src/rm_ethercat_ssc_nort
	Version Control System)] DIR\$/rzn cfg/fsp cfg
Dhuildinfo i	-)J DIR\$/rzn cfg/fsp cfg/bsp
	Make F7)J_DIR\$/rzn_gen
	Compile Ctrl+F7)J_DIR\$/src
9	Rebuild All	th>
	Clean)J_DIR\$/rzn/arm/CMSIS_5/CMSIS/Core_R/Include
	Patek build 50)J_DIR\$/rzn/fsp/inc
· · · · · · · · · · · · · · · · · · ·	Batch Dullu Po	0J_DIR\$/rzn/tsp/inc/api
I I I I	Clean Browse Information	JJ_DIR\$/rzn/tsp/inc/instances
I I I I		JJ DIR\$/rzn cfg/fsp cfg
I	C-STAT Static Analysis)J_DIR\$/rzn_cfg/fsp_cfg/bsp
)J_DIR\$/rzn_gen
	Stop Build Ctrl+Break	0]_DIR\$/src
	Download and Debug Ctrl+D	
	Debug without Downloading	true
		<pre>J_DIR\$/script/fsp_ram_execution.icf</pre>



7. Press the "Resume" button for the project. Program is running.





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.

🔝 J-Link Commander V7.82f	_	×
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 USBO.K. Firmware: J-Link OB-S124 compiled Sep 1 2022 15:38:25 Hardware version: V1.00 J-Link uptime (since boot): Od OOh O6m O5s 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.

	🔜 J-Link Commander V7.82f	_	Х
	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 USBO.K. Firmware: J-Link OB-S124 compiled Sep 1 2022 15:38:25 Hardware version: V1.00 J-Link uptime (since boot): Od OOh O6m O5s S/N: 831650215 VTref=3.300V		
\rightarrow	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) e) swn</default>		
\rightarrow	s) smu T) cJTAG TIF>S Specify target interface speed [kHz]. <default>: 4000 kHz Speed></default>		
			~

Figure 5-2 Connection conditions (1/2)



Figure 5-3 Connection conditions (2/2)



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

- >exec EnableEraseAllFlashBanks

- >Erase 0x6000000, 0x60100000

After that, confirm the message "Erasing done." Is displayed.

Enter "q" to exit J-Link Commander.



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



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.

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_IO\e2studio" folder → [Finish].

Import Projects Select a directory to sea	rch for existing Eclipse projects.	
Select root directory:	C:¥Users¥a5000352¥Desktop¥RZN2L_EtherCAT_RSK_rev0100_rc1 ~	Browse
○ Select archive file:	~	Browse
Projects:	C:¥Users¥a5000352¥Desktop¥RZN2L_EtherCAT_RSK_rev0100_rc1¥project¥rzn2l_rsk_rzn2l¥ecat_CiA402¥e2studio)	Select All
RZN2L_esc_RSK (C	.:#Users#a5000352#Desktop#KZN2L_EtherCAT_KSK_rev0100_rc1#project#rzn2l_rsk_rzn2l#ecat_IO#e2studio)	Deselect All
		Refresh

2. Open "cofiguration.xml" in the "RZN2L_SOM_ESC_IO " project



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

				Generate Proje	ct Cor
 Src configuration.xml RZN2L_esc_RSK Debug_Flat,jlink RZN2L_esc_RSK Debug_Flat,launch 	Thread	HAL/Common Stacks g_ioport I/O Port Driver on r_ioport 3	New Stack g_ethercat_ssc_port0 EtherCAT g_ether_phy0 Ethernet Driver on r_ether_phy g g_ether_selector0 Ethernet Driver on r_ether_selector gether_selector gether_selector	k > ≗ Extend Stack > € T SSC Port g_ether_phy1 Ethernet Driver on r_ether_phy g_ether_selector1 Ethernet Driver on r_ether_selector) Rem



4. Select and build the "RZN2L_SOM_ESC_IO" project.



- 5. Press the "RESET" switch of the RZ/N2L Industrial Network SOM Kit.
- 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 × ✓ Se RZN2L_esc_RSK (i > Se Binaries > Si Includes > Se rzn 		New Go Into Open in New Window Show In	Alt+Shift+W >	rati	ion × HAL/Common Stacks	€ N
> 😂 rzn_gen > 🥵 src > 🗁 Debug > 🗁 rzn_cfg > 🇁 script 👼 configuration.x		Copy Paste Delete Source Move Rename	Ctrl+C Ctrl+V Delete >	er c the	g_ioport I/O Port Driver on r_ioport # (1) (1)	g_ethercat_ssc_port0
III RZN2L_esc_RSł III RZN2L_esc_RSł		Import Export Renesas FSP Export	>	>	0	Driver on r_ether_phy
		Build Project Clean Project Refresh Close Project Close Unrelated Project	F5		•	g_ether_selector0 Ethernet Driver on r_ether_selector
		Build Targets Index Build Configurations	> > >	up īd	Stacks Components - Ŵ スマート・ブラウザー 印 スマート・マニュ	アル 🌣 Debug
		Run As	>	K]		
		Debug As Team Compare With Restore from Local History MISRA-C	>	C* C* C*	1 GDB OpenOCD Hardware Debugg 2 GDB Simulator Debugging (RH850 3 Local C/C++ Application 4 Renesas GDB Hardware Debugging 5 Renesas Simulator Debugging (RX	ying (DSF) 0) ig (, RL78)
		C/C++ Project Settings Renesas C/C++ Project Settings	Ctrl+Alt+P	Lat	Debua Configurations The NETZE_COLLINATION NET NET	<u>vze_ese_nsk.ne</u> x" f"



 $[Renesas DBG Hardware Debugging] \rightarrow [RZN2L_SOM_ESC_IO Debug_Flat] item, then press [Debug]$

Debug Configurations		-	- □ ×		
reate, manage, and run configurations)Ø		
3 🖗 闷 🗶 🖻 🖌 🗕	Name: RZN2L_esc_RSK Debug_Flat				
type filter text	📄 Main 🅸 Debugger 🕨 Startup 🧤	Source Common			
 C/C++ Application C/C++ Remote Application 	Project:		^		
I EASE Script	RZN2L_esc_RSK		Browse		
© GDB Hardware Debugging	C/C++ Application:				
GDB OpenOCD Debugging	Debug/RZN2L esc RSK.elf				
🖻 GDB Simulator Debugging (RH850) 💌 Java Applet		Variables Search Project	Browse		
Java Application	Build (if required) before launching				
Launch Group Remote Java Application	Build Configuration: Use Active		~		
Renesas GDB Hardware Debugging	🔿 Enable auto build	O Disable auto build			
RZN2L_esc_RSK Debug_Flat	Use workspace settings	Configure Workspace Settings	<u> </u>		
Renesas Simulator Debugging (RX, RL78)			Y		
ilter matched 13 of 15 items		Revert	Apply		
(?)		Debug	Close		

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

Cor	firm Perspective Switch X
?	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?
Ren	nember my decision Switch No



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
 M General Registers 	
No. 101	0x1
1000 r1	0x80281a10
888 r2	0xa500
1999 r3	0x80281300
¹⁹¹⁰ otof r4	0xbb92caf
¹⁹¹⁰ r5	0x2aac537e
888 r6	0x0
888 r7	0x0
819 r8	0x0
888 r9	0x0
888 r10	0x0
1000 r11	0x0
## r12	0xe51ff004
lill sp	0x101fe8
1010 lr	0x10006d
bill pc	0x100000
litte 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] \rightarrow [TwinCAT3] \rightarrow [TwinCAT XAE].

After the program is started, by selecting [File] \rightarrow [New] \rightarrow [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



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



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.





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] \rightarrow [EEPROM] \rightarrow [Hex Editor].
- 5. Select [Download from List] → Select ESI File "common\ecat_IO\ESI\esi\Renesas EtherCAT RZN2.xml"
- 6. Select "Renesas EtherCAT RZ/N2 2port"
- 7. OK and Download.

General EtherCAT D	C Process Data Startup CoE - Online Online
Type:	Renesas EtherCAT RZ/T2
Producy newsion:	2048 / 256
Auto Inc Addr:	0 2
EtherCAT Addr:	1001 ÷ Advanced Settings
Identification Value:	0
Previous Port:	Master 🗸
Idvanced Settings	·
⊡ General Behavior	Hex Editor
Timeout Settings Identification	0000 80 0E 44 EE 10 00 00 00 00 00 00 00 00 00 00 CB 00D 0010 66 07 00 00 08 00 00 00 01 00 00 00 00 00 00 00 f
FMMU / SM	0020 00 00 00 00 00 00 00 00 00 00 00 00
Init Commands	0040 00 00 00 00 00 00 00 00 00 00 00 00
Distributed Clock 3	
ESC Access	0070 00 00 00 00 00 00 00 00 00 00 00 00
E-PROM	0090 FF
Enhanced Link De	00A0 FF
Smart View	00C0 FF
Hex Editor	00D0 FF
Memory	OOFO FF
4	
	5
	,
	Download Read from File Download from List
	Upload Write to File
	oprodu milita to milita

Figure 6.1: EEPROM update



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

- 1. SSC Tool \rightarrow [Tool] \rightarrow [EEPROM Programmer].
- 2. [FILE] \rightarrow [OPEN] \rightarrow Browse and select the ESI file.
- 3. [FILE] \rightarrow [Save AS] \rightarrow Select type as binary.
- 4. A binary file will be generated in the specified folder.
- 5. [Read from File] Select the ESI binary file \rightarrow [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 Sync Modes

The Slave Stack Code supports different modes of synchronization which are based on three physical signals: (PDI_) IRQ, Sync0 and Sync1.

After setting the synchronous mode, please reflect the setting in [TwinCAT] \rightarrow [Restart TwinCAT (Config Mode)] \rightarrow [Reload Devices].

6.3.1 Free Run

In this mode there is no slave application synchronization, AL_EVENT_ENABLED and DC_SUPPORTED disabled.

6.3.2 Sync Manager Synchronization

In this mode the slave application is executed as Sync Manager synchronous. AL_EVENT_ENABLED enabled and DC_SUPPORTED disabled.

Solution Explorer 🔹 🖡 🗙	TwinCAT Project98 👳 🗙		
000 0-0 10-0	General EtherCAT DC	Process Data Startup CoE - Online	Online
Search Solution Explorer (Ctrl+:)	Operation Mode:	SM-Synchron Advan	v ced Settings
MOTION PLC SAFETY C++ C++ C++ C++ C++ C++ C++ C	Advanced Settings	Distributed Clock Cyclic Mode Operation Mode: Enable SYNC 0 Cycle Time (µs): © Sync Unit Cycle	SM-Synchron
		User Defined Enable SYNC 0	+ SYNCO Cycle
Mappings		SYNC 1 Sync Unit Cycle SYNC 0 Cycle Enable SYNC 1	 ✓ Cycle Time (µs): ✓ Shift Time (µs):

Figure 6.2: SM- Synchronization



6.3.3 DC Synchronization

SyncManager/Sync0 & SyncManager/Sync0/Sync1 synchronous, both AL_EVENT_ENABLED and DC_SUPPORTED are enabled.

1. Master setting for enabling DC synchronization.

Solution Explorer 👻 👎 🗙	TwinCAT Project98 👳 🗙	✓ Pr
Solution Explorer (Ctrl+c) Search Solution TwinCAT Project98 (1 project) Solution TwinCAT Project98 (1 project) SySTEM MOTION PLC SAFETY SAFETY C++ Devices	General Adapter EtherCAT Online CoE - Online NetId: 10.166.85.159.3.1 Advanced Settings Export Configuration File Advanced Settings	
	B State Machine Image: State Settings Distributed Clocks B Cyclic Frames Distributed Clocks Image: Clock Clocks B Cyclic Frames Distributed Clocks Image: Clock Cloc	tur
Mappings	Continuous Run-Time Measuring Percent of cycle time: 30% Sync Window Monitoring For Outputs: 1.300 + 0 Sync Window (µs): 0 - + 0 + 0 Show DC System Time (64 bit) Dc Sync Task: Highest Priority ~ - - - + 0	~ ОК

Figure 6.3: DC Setting-Master



2. Slave setting for enabling DC synchronization.

Solution Explorer 👻 👎 🗙	TwinCAT Project98 + ×	-
© ② 値 つ ・ 部 ル Search Solution Explorer (Ctrl+:) の・	General EtherCAT DC Process Data Startup CoE - Online Online	Î
Search Solution TwinCAT Project98 (1 project)	Operation Mode: DC-Synchron ~ Advanced Settings	
PLC SAFETY	Advanced Settings	×
Image Image	Distributed Clock Operation Mode: Operation Mode: Sync Unit Cycle (µs): SYNC 0 Cycle Time (µs): Sync Unit Cycle Sync Unit Cycle	
	Use as potential Reference Clock	キャンセル

Figure 6.4: DC Setting Slave

Note: SYNC0 and SYNC1 are level triggered interrupt. It can cause synchronization issues due to multiple interrupts in a single pulse. To avoid this issue, pulse width can be reduced by changing the word 2 of EEPROM configuration value in ESI file. It sets the register 0x982 register of EtherCAT slave during power up. Also can be solved by waiting for the line to be low in the interrupt handler by reading the status register 0x98E.



6.4 Testing the I/O Controller

I/O communication can be confirmed by the LED and the Dip Switch of the SOM Kit..



Figure 6.5: LED and DIP SW

To confirm the I/O output, use TwinCAT3 to select [Output Counter] \rightarrow [Online] \rightarrow [Write] and enter the desired value.

LED1 to LED4 glow up according to the set value.











To confirm the I/O input, use TwinCAT3 to select [Input Counter] \rightarrow [Online], 4-bit input value of Dip Switch (J7-1 to J7-4) is displayed in Value.



Figure 6.7: I/O Input setting



7. Appendix: FSP Configuration for VSC8531 and SSC port

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.

In addition, since interrupts are used for IO control, SSC port driver also needs to change.

(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_IO\e2studio
- When using EWARM, \project\rzn2l_som\ecat_IO\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 7-1.

acks Configuration		G	enerate Project Content	g_ether_	phy0 Ethernet Driver on r_ether_ph	y
New Thread	a athar0 faharaat Daiwaa aa	New Stack >		Settings	Property Common	Value
reads Remove	r_ether Stacks	Remove			> Select PHYs to use	
		-			Parameter Checking	Default (BSP)
MAL/Common	🐣 a ether0 Ethernet Driver o	n r ether			 Module g_ether_phy0 Ethernet Driver on 	r_e
g_ioport I/O Port Driver on i	g_callelo calleliner biller o	in_calci			Name	g_ether_phy0
Main Ihread		/			Channel	A 0
g_ether0 Ethernet Driver on	í	1			PHY-LSI Address	1
 ⊕ g_uart0 UARI Driver on r_sci ⊕ Heap 4 ⊕ g_i2c_master0 I2C Master Di 	· · ·				PHY-LSI Reset Completion Timeout	0x00020000
	g_ether_phy0 Ethernet Driver on r_ether_phy Driver ch1		Add Ethernet PHY Driver ch1	e	Flow Control	🔒 Disable
		Driver ch1			Port Type	Ethernet
		Diliver citi			Select PHY	VSC8541
	6				Select MDIO type	GMAC
>					Auto Negotiation	ON
					Speed	10/100/1000M
New Object >	g_ether_selector0				Duplex	FULL
jects	Ethernet Driver on				Reset Port	13
Remove	T_ether_selector				Reset Pin	4
	U				Reset assert time	15000
	<		>			

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

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 also 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);</pre> reg |= 1U << ETHER_PHY_REG_LED1_FEATURE_DISABLE_OFFSET;</pre> 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_GPI0_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_GPI0_PAGE, 0x00); #endif } /* End of function ether_phy_targets_initialize() */



(4) Add code for nested interrupt.

The following code for nested interrupt should be added to "ethercat_ssc_port_isr_esc_cat" function in rzn/fsp/src/rm_ethercat_ssc_port/rm_ethercat_ssc_port.c.

```
void ethercat_ssc_port_isr_esc_cat (void)
{
    /* Enable nested interrupt. */
    __asm volatile ("cpsie i");
__asm volatile ("isb");
    ethercat_ssc_port_callback_args_t callback_arg;
    IRQn_Type irq = R_FSP_CurrentIrqGet();
    ethercat_ssc_port_instance_ctrl_t * p_instance_ctrl =
    (ethercat_ssc_port_instance_ctrl_t *) R_FSP_IsrContextGet(irq);
    /* SSC ESC CAT Interrupt handler */
    PDI_Isr();
    /* Callback : Interrupt handler */
    if (NULL != p_instance_ctrl->p_cfg->p_callback)
    {
         callback arg.event
                                  = ETHERCAT SSC PORT EVENT ESC CAT INTERRUPT;
         callback_arg.p_context = p_instance_ctrl->p_cfg->p_context;
         (*p_instance_ctrl->p_cfg->p_callback)((void *) &callback_arg);
    }
    /* Disable nested interrupt */
    __asm volatile ("cpsid i");
__asm volatile ("isb");
}
                                             /* End of function ethercat_ssc_port_isr_esc_cat() */
```



Revision History

Description		on	
Rev.	Date	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 is supplied until the power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).

7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

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

Notice

- 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.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
 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/.