RZ/T2N Group

Encoder I/F EnDat Sample Program

Summary

This document describes the package contents of the RZ/T2N Encoder I/F EnDat sample program.

For EnDat 2.2 communication protocol specifications and encoder specifications (EQN1035), please contact DR. JOHANNES HEIDENHAIN GmbH.

Functionality Checked Device

RZ/T2N Evaluation Board (RTK9RZT2Nxxxxxxxxx)

Contents

1. Package Contents 2
1.1 Software 2
1.2 Documents 2

2. File Structure 3

3. About EnDat Sample Program 4
3.1 Software Information 4
3.1.1 Base OS 4
3.1.2 Memory Size 4
3.2 Hardware Information 5
3.2.1 Device 5
3.2.2 Target Board 5
3.3 Procedures on Development Environments: CR52 version 7
3.3.1 Preparation before Executing the Sample Program 7
3.3.2 EWARM from IAR Systems 7
3.3.3 e² studio from RENESAS 9
3.4 Procedures on Development Environments: CA55 version 11
3.4.1 Preparation before Executing the Sample Program 11
3.4.2 EWARM from IAR Systems 11
3.4.3 e² studio from RENESAS 13

Revision History 15

1. Package Contents

This package contains the following contents.

The EnDat encoder interface of the RZ/T2N supports up to 8 axes, but the sample program uses only 1 of those axes. If you use 2 or more axes simultaneously, modify the sample program to support the required number of axes.

1.1 Software

Source Code

No.	Name	Version
1	RZ/T2N EnDat sample program (CR52 ver. *)	4.0
2	RZ/T2N EnDat sample program (CA55 ver. *)	4.0

Note: This sample program has a CR52 version that runs on the CPU core Cortex-R52 and a CA55 version that runs on the CPU core Cortex-A55. CR52 ver. and CA55 ver. are descriptions of the respective version.

1.2 Documents

No.	Document name	Version	File name
1	RZ/T2N Group Encoder I/F EnDat Sample Program Release Notes	4.00	(j) r11an1016jj0400-rzt2n.pdf (e) r11an1016ej0400-rzt2n.pdf (this document)
2	RZ/T2N Group EnDat Sample Program	4.00	(j) r11an1030jj0400-rzt2n-endat.pdf (e) r11an1030ej0400-rzt2n-endat.pdf

2. File Structure

The file structure and contents of this package are shown below.

```
Top
 ├── r11an1016jj0400-rzt2n.pdf
 ├── r11an1016ej0400-rzt2n.pdf
 ├── SBOM
 │    ├── RZT2N_endat22_evb_rev4.00.json        : Sample program SBOM information
 │    └── RZT2N_endat22_evb_rev4.00.spdx        : Sample program SBOM information
 └── workspace
      ├── Software
      │    ├── iccarm
      │    │    ├── RZ_T2N_CR52_endat.zip       : RZ/T2N EnDat sample program set
      │    │    │                                 CR52 version (EWARM)
      │    │    └── RZ_T2N_CA55_endat.zip       : RZ/T2N EnDat sample program set
      │    │                                      CA55 version (EWARM)
      │    └── gcc
      │         ├── RZ_T2N_CR52_endat.zip       : RZ/T2N EnDat sample program set
      │         │                                 CR52 version (e2 studio)
      │         └── RZ_T2N_CA55_endat.zip       : RZ/T2N EnDat sample program set
      │                                           CA55 version (e2 studio)
      └── Documents
           ├── r11an1030jj0400-rzt2n-endat.pdf
           └── r11an1030ej0400-rzt2n-endat.pdf
```

The file structure of the RZ_T2N_CR52_endat.zip and RZ_T2N_CA55_endat.zip is shown below.

```
Top folder
 ├── configuration.xml                  : FSP Configuration data
 ├── ( Environment File Depending on Build Tool )
 └── src
      ├── hal_entry.c                   : EnDat sample program
      ├── endat_main.c                  : EnDat sample program
      ├── siochar.c                     : SCI_UART sample program
      ├── syscalls.c                    : SCI_UART sample program
      ├── sio_char.h                    : SCI_UART sample program
      └── drv
           └── endat
                ├── iodefine_endat.h    : EnDat register definition file
                ├── r_endat_rzt2.c      : EnDat driver file
                ├── r_endat_rzt2_config.h : EnDat driver file
                ├── r_endat_rzt2_dat.h  : EnDat driver file
                └── r_endat_rzt2_if.h   : EnDat driver file
```

3. About EnDat Sample Program

This section contains information necessary to use the complete set of EnDat sample program.

3.1 Software Information

3.1.1 Base OS

This sample program is OS-independent.

3.1.2 Memory Size

Memory size used by this sample program and EnDat driver is shown in the following table. This table does not include memory size used by Flexible Software Package or C language libraries of the compiler.

(1) CR52 version

Items		Memory Size	
		EWARM [kBytes]	e² studio [kBytes]
EnDat driver	Code	4.5	3.4
	Data (with initial value)	0.0	0.0
	Data (without initial value)	0.4	0.4
	Constant Data	0.2	1.3
Sample program	Code	2.8	3.2
	Data (with initial value)	0.0	0.0
	Data (without initial value)	0.4	0.4
	Constant Data	1.2	0.1

(2) CA55 version

Items		Memory Size	
		EWARM [kBytes]	e² studio [kBytes]
EnDat driver	Code	6.6	5.4
	Data (with initial value)	0.0	0.0
	Data (without initial value)	0.5	0.5
	Constant Data	0.3	2.0
Sample program	Code	4.5	5.6
	Data (with initial value)	0.0	0.0
	Data (without initial value)	0.4	0.4
	Constant Data	1.3	0.1

3.2 Hardware Information

3.2.1 Device

RZ/T2N

3.2.2 Target Board

(1) Board Name

RZ/T2N Evaluation Board (RTK9RZT2Nxxxxxxxxx)

(2) Setting of the Target Board

The target board configuration is as follows.

SW7-1: OFF, SW7-3: OFF, SW7-4: OFF, SW7-5: OFF, SW7-6: OFF, SW7-7: OFF, SW7-8: OFF

SW9-3: OFF, SW9-5: OFF, SW9-6: OFF, SW9-7: OFF, SW9-8: OFF

SW12-5: ON, SW13-1: ON, SW13-2: OFF, SW13-6: ON, SW13-7: OFF

SW8-1: ON, SW8-2: OFF, SW8-3: ON (Set xSPI1 boot mode)

SW10-3: OFF, J18: Short between 2-3pin (Set VCC1833_2 to 3.3V)

SW10-6: OFF, J39: Short between 2-3pin (Set VCC1833_5 to 3.3V)

(3) Used Pins of the Target Board

The correspondence between the pin used as the encoder I/F and the pin header of the target board is as follows.

Channel	Pin Name	Pin Header	Input/ Output	Voltage Domain	Description
ENDAT_CH0	ENCIFCK0 (TCLK0)	J49 #41	Output	VCC1833_2	Clock output
	ENCIFOE0 (DE0)	J49 #42	Output	VCC1833_2	Data output enable
	ENCIFDO0 (DATA_DV0)	J49 #43	Output	VCC1833_2	Data output
	ENCIFDI0 (DATA_RC0)	J49 #44	Input	VCC1833_2	Data input
ENDAT_CH1	ENCIFCK1 (TCLK1)	J50 #41	Output	VCC1833_2	Clock output
	ENCIFOE1 (DE1)	J50 #42	Output	VCC1833_2	Data output enable
	ENCIFDO1 (DATA_DV1)	J50 #43	Output	VCC1833_2	Data output
	ENCIFDI1 (DATA_RC1)	J50 #44	Input	VCC1833_2	Data input
ENDAT_CH2	ENCIFCK2 (TCLK2)	J51 #41	Output	VCC33	Clock output
	ENCIFOE2 (DE2)	J51 #42	Output	VCC33	Data output enable
	ENCIFDO2 (DATA_DV2)	J51 #43	Output	VCC33	Data output
	ENCIFDI2 (DATA_RC2)	J51 #44	Input	VCC33	Data input
ENDAT_CH3	ENCIFCK3 (TCLK3)	J52 #41	Output	VCC1833_4	Clock output
	ENCIFOE3 (DE3)	J52 #42	Output	VCC1833_4	Data output enable
	ENCIFDO3 (DATA_DV3)	J52 #43	Output	VCC1833_4	Data output
	ENCIFDI3 (DATA_RC3)	J52 #44	Input	VCC1833_4	Data input
ENDAT_CH4	ENCIFCK4 (TCLK4)	J49 #46	Output	VCC1833_3	Clock output
	ENCIFOE4 (DE4)	J49 #47	Output	VCC1833_3	Data output enable
	ENCIFDO4 (DATA_DV4)	J49 #48	Output	VCC33	Data output
	ENCIFDI4 (DATA_RC4)	J49 #49	Input	VCC33	Data input
ENDAT_CH5	ENCIFCK5 (TCLK5)	J51 #46	Output	VCC1833_5	Clock output
	ENCIFOE5 (DE5)	J51 #47	Output	VCC1833_5	Data output enable
	ENCIFDO5 (DATA_DV5)	J51 #48	Output	VCC1833_5	Data output
	ENCIFDI5 (DATA_RC5)	J51 #49	Input	VCC1833_5	Data input
ENDAT_CH6	ENCIFCK6 (TCLK6)	J50 #46	Output	VCC33	Clock output
	ENCIFOE6 (DE6)	J50 #47	Output	VCC33	Data output enable
	ENCIFDO6 (DATA_DV6)	J50 #48	Output	VCC33	Data output
	ENCIFDI6 (DATA_RC6)	J50 #49	Input	VCC1833_4	Data input
ENDAT_CH7	ENCIFCK7 (TCLK7)	J52 #46	Output	VCC33	Clock output
	ENCIFOE7 (DE7)	J52 #47	Output	VCC33	Data output enable
	ENCIFDO7 (DATA_DV7)	J52 #48	Output	VCC33	Data output
	ENCIFDI7 (DATA_RC7)	J52 #49	Input	VCC1833_3	Data input

3.3 Procedures on Development Environments: CR52 version

3.3.1 Preparation before Executing the Sample Program

This sample program communicates with a PC. The USB connection terminal on the target board is J33. Select the higher-numbered COM port that appears when the board is connected to the host PC.

The terminal software of the host PC is set as shown in the following table.

Function	Setting
Communication method	Asynchronous serial transmission/reception
Transmission/reception order	LSB first
Transfer rate	19200 bps
Character length	8 bits
Stop bit length	1 bit
Parity function	None
Hardware flow control	None

3.3.2 EWARM from IAR Systems

(1) Build Environment

IAR Embedded Workbench for Arm (EWARM)

Version 9.60.3 + patch (EWARM_Patch_for_RZT2N_rev1.0)

RENESAS FSP Smart Configurator (FSP SC) 2026-07

RENESAS Flexible Software Package (FSP) for RZ v4.2.0

(2) Debug Probe

IAR I-jet

Connect the JTAG debug probe to J31 and short between the J29 jumper pins.

(3) Build Procedure of the Sample Program

The build procedure of the sample program is as follows.

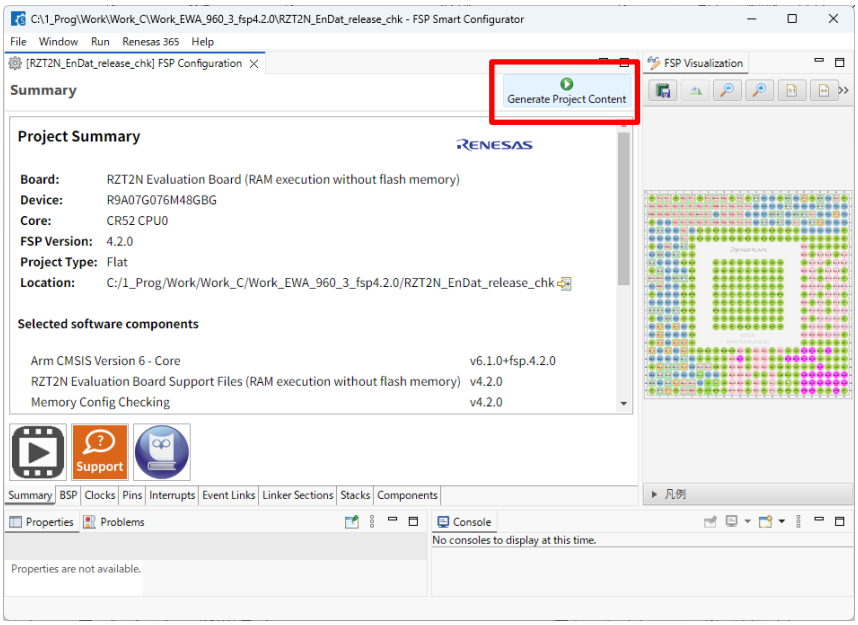
1 Extract RZ_T2N_CR52_endat.zip and copy the extracted source files to the desired location.
2 Activate EWARM.
3 Select [File] menu -> [Open Workspace].
4 Open the extracted source file RZ_T2N_endat.eww.
5 Start the FSP Smart Configurator from the [Tools] menu of the EWARM IDE. *

Note: The following procedure adds the activation of the FSP Smart Configurator to the [Tools] menu of the EWARM IDE. Select [Tools] menu -> [Tool Configuration] in the EWARM IDE. Select the [New] button, specify the strings shown in the table for each field, and press [OK].

Field	String
Menu text	FSP Smart Configurator
Command	$RASC_EXE_PATH$
Argument	--compiler IAR configuration.xml
Initial directory	$PROJ_DIR$

The string in the command field is a variable that holds the path of the FSP Smart Configurator execution file, rasc.exe.
If the path written as RASC_EXE_PATH in the buildinfo.ipcf file does not match your rasc.exe installation path, edit buildinfo.ipcf accordingly.
You can also start the FSP Smart Configurator directly from the command prompt by specifying the folder where it is installed.

6 In the FSP Configuration pane of the FSP Smart Configurator, click Generate Project Content. The rz, rz_cfg, rz_gen, script and .settings folders will be generated.



7 When project generation is complete, close the FSP Smart Configurator.

8 Select [Rebuild All] from the [Project] menu of EWARM.
The file Debug\Exe\RZ_T2N_endat.out will be generated.

(4) Sample Program Execution Procedure

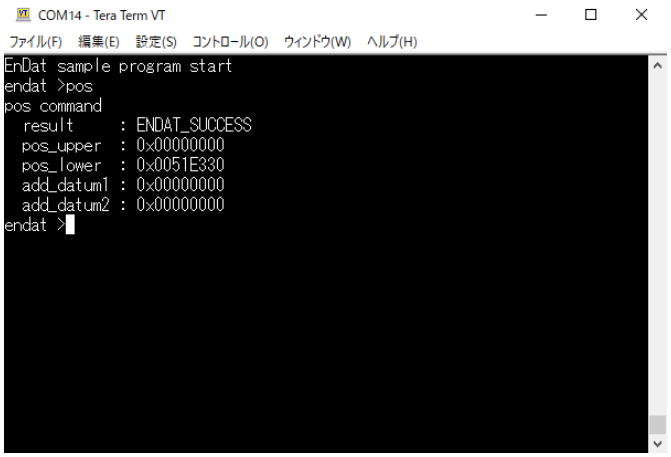
After executing the "build procedure", connect the target board and debugger correctly, and perform the following operations.

1 Select [Project] menu -> [Download and Debug].

2 Select [Debug] menu -> [Execute].

(5) Execution Result of the Sample Program

Run the sample program and enter commands in the terminal software window. For commands, see “4.11.8 console commands” in the “RZ/T2N Group EnDat Sample Program” application note.



3.3.3 e² studio from RENESAS

(1) Build Environment

RENESAS e² studio 2026-07

Toolchain version: GNU Arm Embedded 13.3.1.arm-13-24

RENESAS Flexible Software Package (FSP) for RZ v4.2.0

(2) Debug Probe

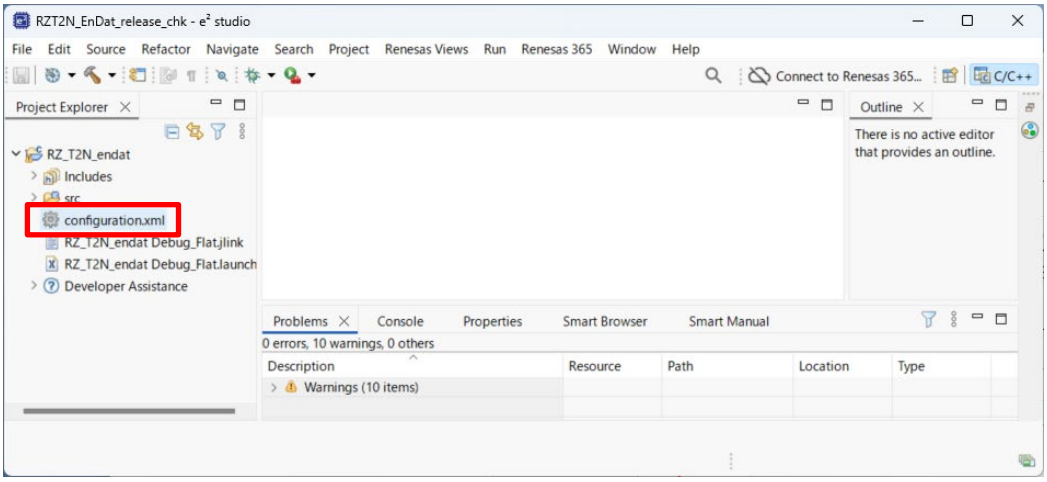
SEGGER J-Link™ v9.44

Connect the JTAG debug probe to J31 and short between the J29 jumper pins. If you use on-board debug probe (J-Link OB) on the target board, open between the J29 jumper pins and connect PC to J32.

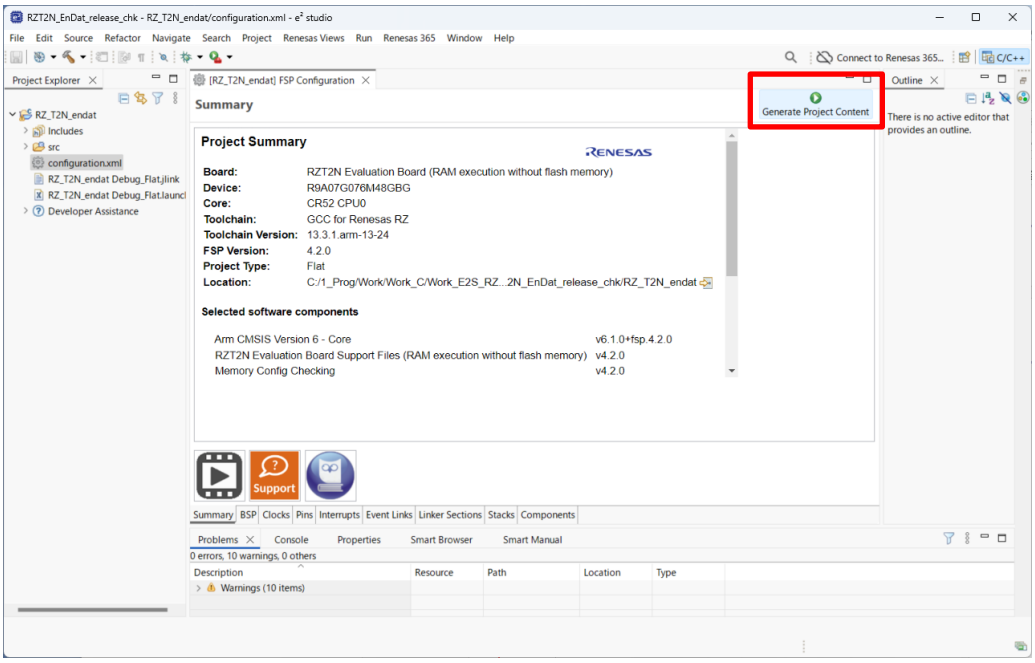
(3) Build Procedure of the Sample Program

The procedure for building the sample program is as follows.

1 Extract RZ_T2N_CR52_endat.zip and copy the extracted source files to the desired location.
2 After launching e² studio and moving to the workspace, click the [File] menu -> [Import] and select General > Existing Projects into Workspace, and click [Next].
3 On the project import screen, select the folder where the sample program was expanded as the root directory.
4 Select a project, check Copy Project to Workspace, and click [Finish].
5 Double-click the configuration.xml in the Project Explorer pane of e² studio to open it.



6 Click Generate Project Content in the FSP Configuration pane of e² studio.
The rz, rz_cfg, rz_gen, script and .settings folders will be generated.



7 Select [Project] menu -> [Build All]
The file Debug\RZ_T2N_endat.elf will be generated.

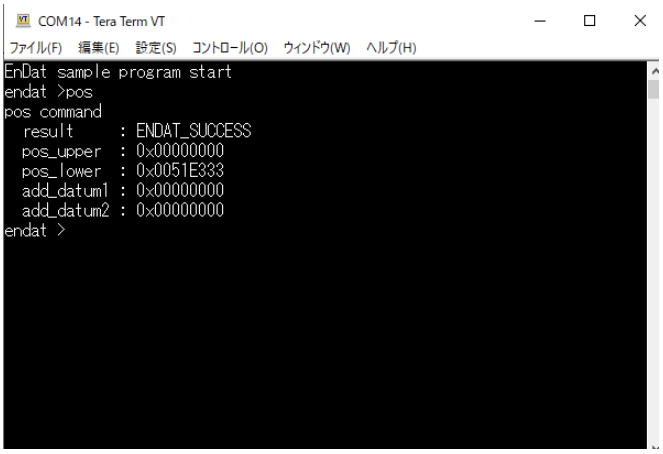
(4) Sample Program Execution Procedure

After executing the "build procedure", connect the target board and debugger correctly, and perform the following operations.

1 Select [Run] menu -> [Debug As] -> [Renesas GDB Hardware Debugging].
2 Click [Debug] to start downloading to internal RAM.
3 Click [Run] menu -> [Resume] to run the sample program.

(5) Execution Result of the Sample Program

Run the sample program and enter commands in the terminal software window. For commands, see “4.11.8 console commands” in the “RZ/T2N Group EnDat Sample Program” application note.



3.4 Procedures on Development Environments: CA55 version

3.4.1 Preparation before Executing the Sample Program

This sample program communicates with a PC. The USB connection terminal on the target board is J33. Select the lower-numbered COM port that appears when the board is connected to the host PC.

The terminal software of the host PC is set as shown in the following table.

Function	Setting
Communication method	Asynchronous serial transmission/reception
Transmission/reception order	LSB first
Transfer rate	19200 bps
Character length	8 bits
Stop bit length	1 bit
Parity function	None
Hardware flow control	None

3.4.2 EWARM from IAR Systems

(1) Build Environment

IAR Embedded Workbench for Arm (EWARM)

Version 9.60.3 + patch (EWARM_Patch_for_RZT2N_rev1.0)

RENESAS FSP Smart Configurator (FSP SC) 2026-07

RENESAS Flexible Software Package (FSP) for RZ v4.2.0

(2) Debug Probe

IAR I-jet

Connect the JTAG debug probe to J31 and short between the J29 jumper pins.

(3) Build Procedure of the Sample Program

The build procedure of the sample program is as follows.

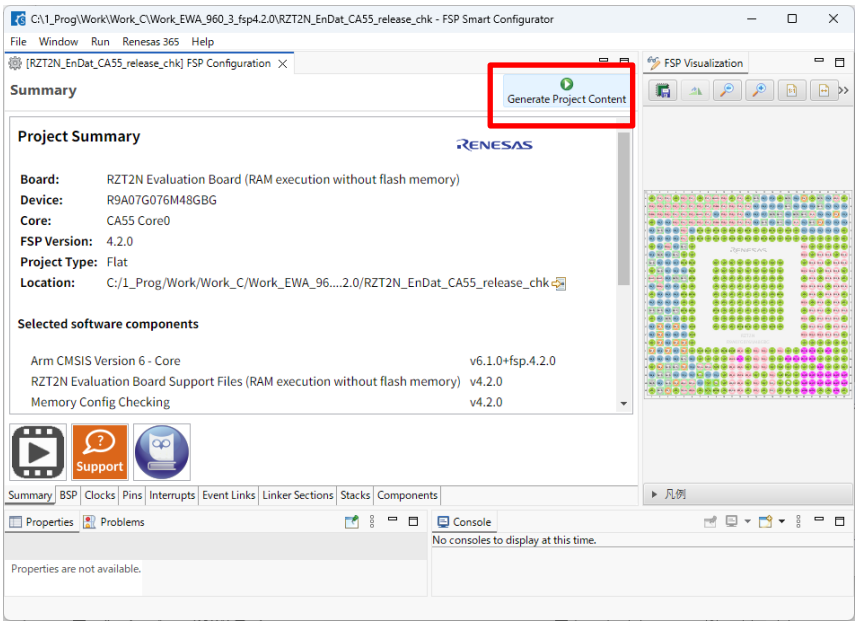
1 Extract RZ_T2N_CA55_endat.zip and copy the extracted source files to the desired location.
2 Activate EWARM.
3 Select [File] menu -> [Open Workspace].
4 Open the extracted source file RZ_T2N_CA55_0_endat.eww.
5 Start the FSP Smart Configurator from the [Tools] menu of the EWARM IDE. *

Note: The following procedure adds the activation of the FSP Smart Configurator to the [Tools] menu of the EWARM IDE. Select [Tools] menu -> [Tool Configuration] in the EWARM IDE. Select the [New] button, specify the strings shown in the table for each field, and press [OK].

Field	String
Menu text	FSP Smart Configurator
Command	$RASC_EXE_PATH$
Argument	--compiler IAR configuration.xml
Initial directory	$PROJ_DIR$

The string in the command field is a variable that holds the path of the FSP Smart Configurator execution file, rasc.exe.
If the path written as RASC_EXE_PATH in the buildinfo.ipcf file does not match your rasc.exe installation path, edit buildinfo.ipcf accordingly.
You can also start the FSP Smart Configurator directly from the command prompt by specifying the folder where it is installed.

6 In the FSP Configuration pane of the FSP Smart Configurator, click Generate Project Content. The rz, rz_cfg, rz_gen, script and .settings folders will be generated.



7 When project generation is complete, close the FSP Smart Configurator.

8 Select [Rebuild All] from the [Project] menu of EWARM.
The file Debug\Exe\RZ_T2N_CA55_0_endat.out will be generated.

(4) Sample Program Execution Procedure

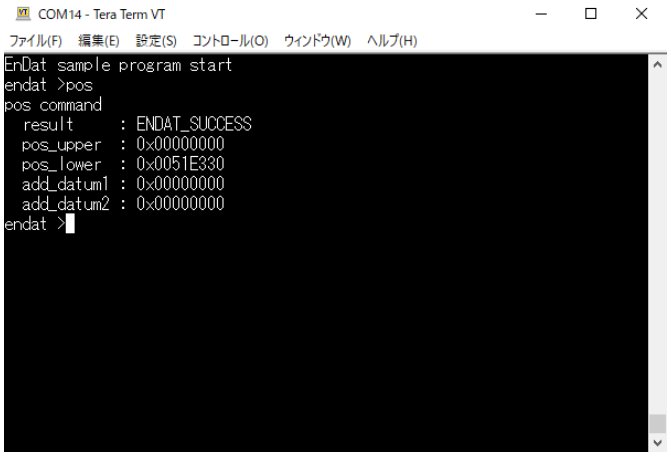
After executing the "build procedure", connect the target board and debugger correctly, and perform the following operations.

1 Select [Project] menu -> [Download and Debug].

2 Select [Debug] menu -> [Execute].

(5) Execution Result of the Sample Program

Run the sample program and enter commands in the terminal software window. For commands, see “4.11.8 console commands” in the “RZ/T2N Group EnDat Sample Program” application note.



3.4.3 e² studio from RENESAS

(1) Build Environment

RENESAS e² studio 2026-07

Toolchain version: GCC Arm A-Profile (AArch64 bare-metal) 13.2.1.20231009

RENESAS Flexible Software Package (FSP) for RZ v4.2.0

(2) Debug Probe

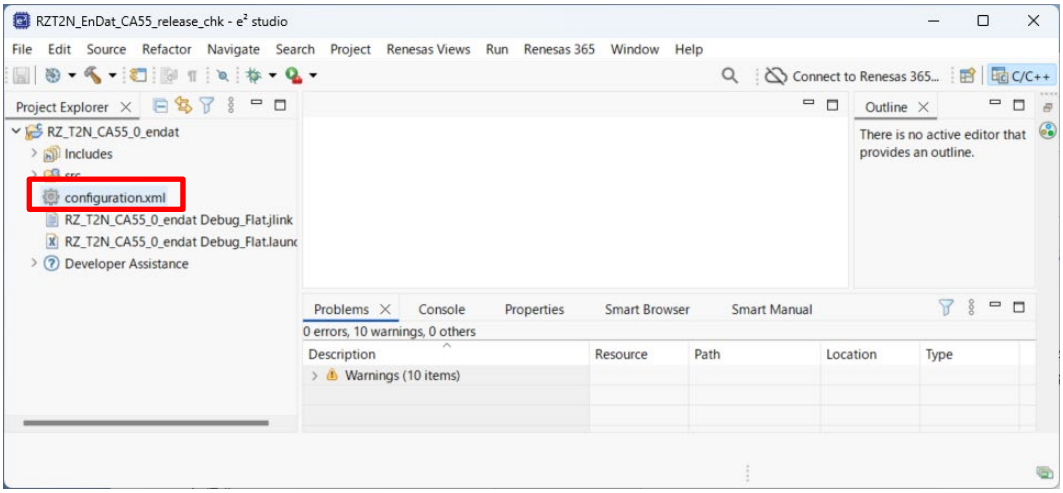
SEGGER J-Link™ v9.44

Connect the JTAG debug probe to J31 and short between the J29 jumper pins. If you use on-board debug probe (J-Link OB) on the target board, open between the J29 jumper pins and connect PC to J32.

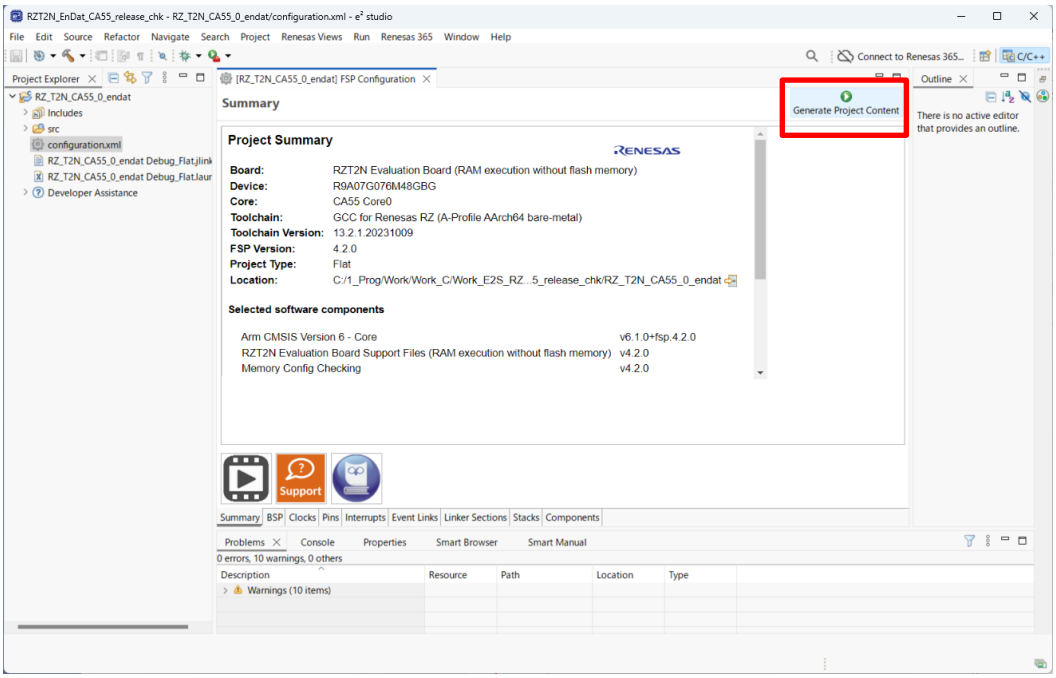
(3) Build Procedure of the Sample Program

The procedure for building the sample program is as follows.

1 Extract RZ_T2N_CA55_endat.zip and copy the extracted source files to the desired location.
2 After launching e² studio and moving to the workspace, click the [File] menu -> [Import] and select General > Existing Projects into Workspace, and click [Next].
3 On the project import screen, select the folder where the sample program was expanded as the root directory.
4 Select a project, check Copy Project to Workspace, and click [Finish].
5 Double-click the configuration.xml in the Project Explorer pane of e² studio to open it.



6 Click Generate Project Content in the FSP Configuration pane of e² studio.
The rz, rz_cfg, rz_gen, script and .settings folders will be generated.



7 Select [Project] menu -> [Build All].
The file Debug\RZ_T2N_CA55_0_endat.elf will be generated.

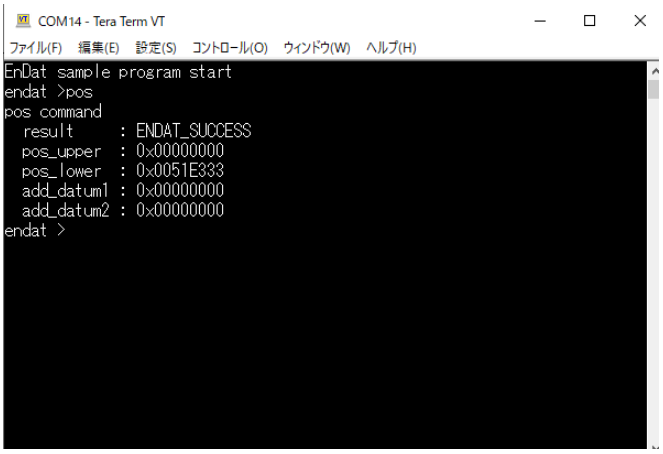
(4) Sample Program Execution Procedure

After executing the "build procedure", connect the target board and debugger correctly, and perform the following operations.

1 Select [Run] menu -> [Debug As] -> [Renesas GDB Hardware Debugging].
2 Click [Debug] to start downloading to internal RAM.
3 Click [Run] menu -> [Resume] to run the sample program.

(5) Execution Result of the Sample Program

Run the sample program and enter commands in the terminal software window. For commands, see “4.11.8 console commands” in the “RZ/T2N Group EnDat Sample Program” application note.



Revision History

Rev.	Date	Description	
		Page	Summary
0.50	Jul 10, 25	-	First Edition issued.
0.80	Mar 19, 26	2, 3	Update revisions of the application note and the release note. Update sample program version to 0.8. (Support FSP v3.1.0-t2n.alpha. Change device to 537-pin version.)
		4	Update memory size information.
		5	Update setting of target board for using 537-pin version.
		6	Correct misprint in input/output descriptions.
		7 to 14	Update build environment for FSP v3.1.0-t2n.alpha. Figures are replaced.
4.00	Nov 4, 26	2, 3	Update revisions of the application note and the release note. Update sample program version to 4.0. (Support FSP v4.2.0.)
		3	Add SBOM information.
		4	Update memory size information.
		6	Correct voltage domain on channel 4.
		7 to 14	Update build environment for FSP v4.2.0. Figures are replaced.

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.

- EnDat is a registered trademark of Dr.Johannes Heidenhain GmbH.
- IAR Embedded Workbench is a registered trademark of IAR Systems.
- J-Link is a trademark of SEGGER Microcontroller GmbH.
- Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.
- Additionally all product names and service names in this document are trademarks or registered trademarks which belong to the respective owners.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

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/.

Trademarks

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

© 2026 Renesas Electronics Corporation. All rights reserved.