RZ/T2L Group

Encoder I/F SSI sample program

Summary

This document describes the RZ/T2L Encoder I/F SSI sample program package.

For Synchronized Serial Interface (SSI) communication protocol specifications and encoder specifications, contact manufacturer of each encoder to obtain it.

Functionality Checked Device

RSK+RZT2L Board (RTK9RZT2Lxxxxxxxxx)

Contents

1. Package Contents

This package contains the following contents.

The Configuration Data included in this package supports up to 2 axes, but the sample program supports only 1 axis; to use it with 2 axes, modify the sample program to support 2 axes.

1.1 Software

- Source code

No.	Name	Version
1	RZ/T2L SSI sample program	4.0

1.2 Document

No.	Document name	Version	File name
1	RZ/T2L Group Encoder I/F SSI sample program Release Note	4.00	(j) r11an0744jj0400-rzt2l.pdf (e) r11an0744ej0400-rzt2l.pdf
2	RZ/T2L Group SSI sample program Application Note	4.00	(j) r11an0743jj0400-rzt2l-ssi.pdf (e) r11an0743ej0400-rzt2l-ssi.pdf

2. File Structure

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

```
Top
├── r11an0744jj0400-rzt2l.pdf
├── r11an0744ej0400-rzt2l.pdf
└── workspace
        ├── Software
        |       ├── iccarm
        |       |       └── RZ_T2L_ssi.zip : RZ/T2L SSI sample program set (IAR)
        |       └── gcc
        |               └── RZ_T2L_ssi.zip : RZ/T2L SSI sample program set (e2 studio)
        └── Documents
                ├── r11an0743jj0400-rzt2l-ssi.pdf
                └── r11an0743ej0400-rzt2l-ssi.pdf
```

The file structure of the RZ_T2L_ssi.zip is shown below.

```
Top folder
├── configuration.xml                    : FSP Configuration data
├── ( Environment File Depending on Build Tool )
└── src
        ├── hal_entry.c                  : SSI sample program
        ├── ssi_main.c                   : SSI sample program
        ├── siochar.c                    : SCI_UART sample program
        ├── siorw.c                      : SCI_UART sample program
        ├── sio_char.h                   : SCI_UART sample program
        └── drv
                └── ssi
                        ├── iodefine_ssi.h       : SSI register definition file
                        ├── r_ssi_rzt2.c         : SSI driver file
                        ├── r_ssi_rzt2_config.h  : SSI driver file
                        ├── r_ssi_rzt2_dat.h     : SSI driver file
                        └── r_ssi_rzt2_if.h      : SSI driver file
```

3. About SSI sample program

This section contains information necessary to use the complete set of SSI 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 SSI driver is shown in following table. This table does not include memory size used by Flexible Software Package or C language libraries of the compiler, but for the SPI driver and DMAC driver portion which is used for the SSI encoder interface.

Items		Memory Size	
		EWARM [kBytes]	e^2 studio [kBytes]
SSI driver	Code	2.8	2.3
	Data (with initial value)	0.0	0.0
	Data (without initial value)	0.3	0.3
	Constant Data	0.0	0.0
SPI driver and DMAC driver for encoder interface *	Code	5.0	3.9
	Data (with initial value)	0.0	0.0
	Data (without initial value)	0.1	0.1
	Constant Data	0.0	0.0
Sample program	Code	3.7	3.6
	Data (with initial value)	0.0	0.0
	Data (without initial value)	0.5	0.5
	Constant Data	1.3	1.4

Note: This is the SPI driver and the DMAC driver which are automatically generated by the Flexible Software Package Smart Configurator. The SPI and the DMAC drivers are used for the SSI encoder interface. It is called from inside of the SSI driver.

3.2 Hardware Information

3.2.1 Device

RZ/T2L

3.2.2 Target Board

(1) Board Name

RSK+RZT2L (RTK9RZT2Lxxxxxxxxx)

(2) Setting of the Target Board

The target board configuration is as follows.

SW4-1: ON, SW4-2: ON, SW4-3: OFF

SW4-4: OFF

SW4-5: ON

SW4-6: ON

SW4-7: OFF

CN32: Short between 1-2 pins.

(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	Description
SSI0	SPI_MISO1	JA2-A #10	input	Data input pin
	SPI_RSPCLK1	JA2-A #25	output	Clock output pin
SSI1	SPI_MISO2	CN27 #3	input	Data input pin
	SPI_RSPCLK2	CN27 #11	output	Clock output pin

3.3 Procedures on Development Environments

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 CN16.
The terminal software of the host PC is set as shown in the following table.

Function	Setting
Communication method	Asynchronous serial transmit/receive
Order of transmission / reception	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: IAR Systems

(1) Build Environment

IAR Embedded Workbench for ARM (EWARM)

Version 9.60.3 + patch (EWARM_Patch_for_RZT2H_N2H_rev1.0)

RENESAS FSP Smart Configurator (FSP SC) 2025-12

RENESAS Flexible Software Package (FSP) for RZ v4.0.0

(2) ICE

IAR I-jet

(3) Build Procedure for the Sample Programs

The build procedure for the sample program is as follows.

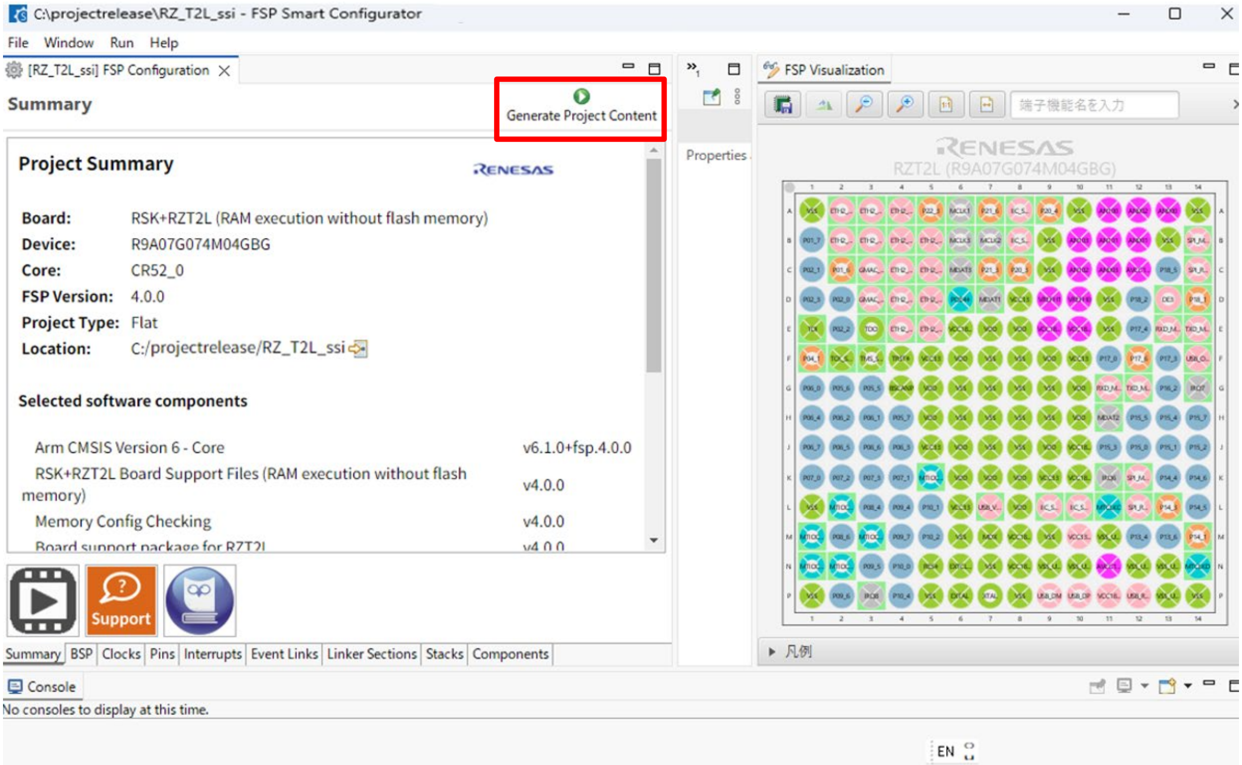
1 Copy the extracted source files to the desired location.
2 Activate EWARM
3 Select [File] menu -> [Open Workspace].
4 Open the extracted source file RZ_T2L_ssi.eww
5 Start the FSP Smart Configurator from the [Tools] menu of the EWARM IDE. *

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 a table string in each field, and press [OK].

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

String for the command is variable holding the path of the Smart Configurator execution file, rasc.exe.
If the path written as RASC_EXE_PATH in the buildinfo.ipcf file does not match with your rasc.exe installation path, please edit the buildinfo.ipcf to fit with your installation path.
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 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 Smart Configurator.

8 Select [Rebuild ALL] from the [Project] menu of EWARM.
The file Debug¥Exe¥RZ_T2L_ssi.out is generated.

(4) Sample Program Execution Procedure

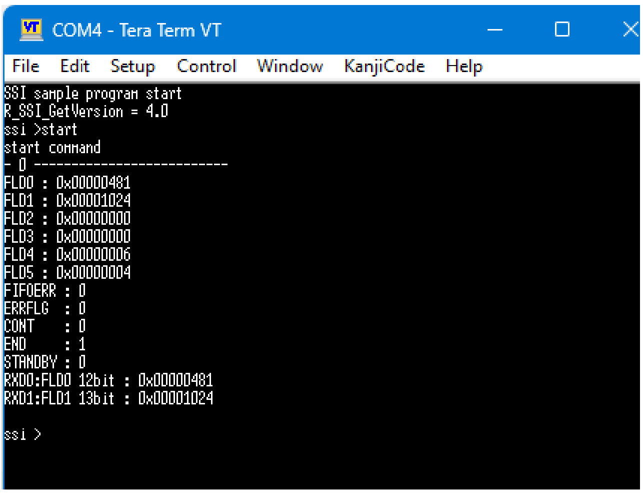
After executing the "build procedure", connect the target board and debugger correctly, and perform the following operations.

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/T2L Group SSI Sample Program Application Note.



3.3.3 e² studio: RENESAS

(1) Build Environment

RENESAS e² studio 2025-12

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

RENESAS Flexible Software Package (FSP) for RZ v4.0.0

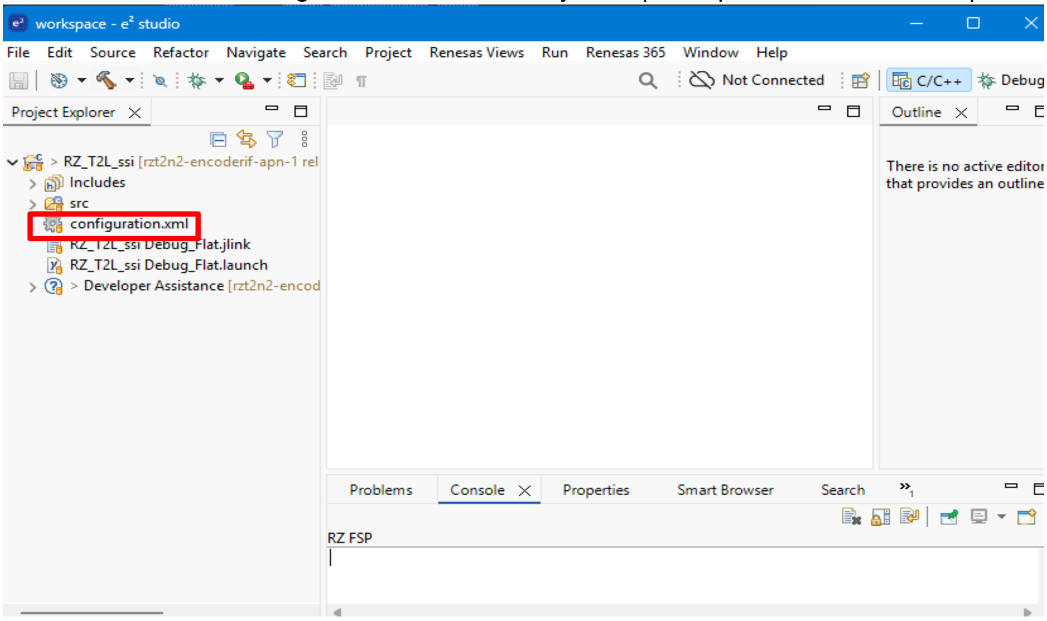
(2) ICE

SEGGER J-Link™ v8.60

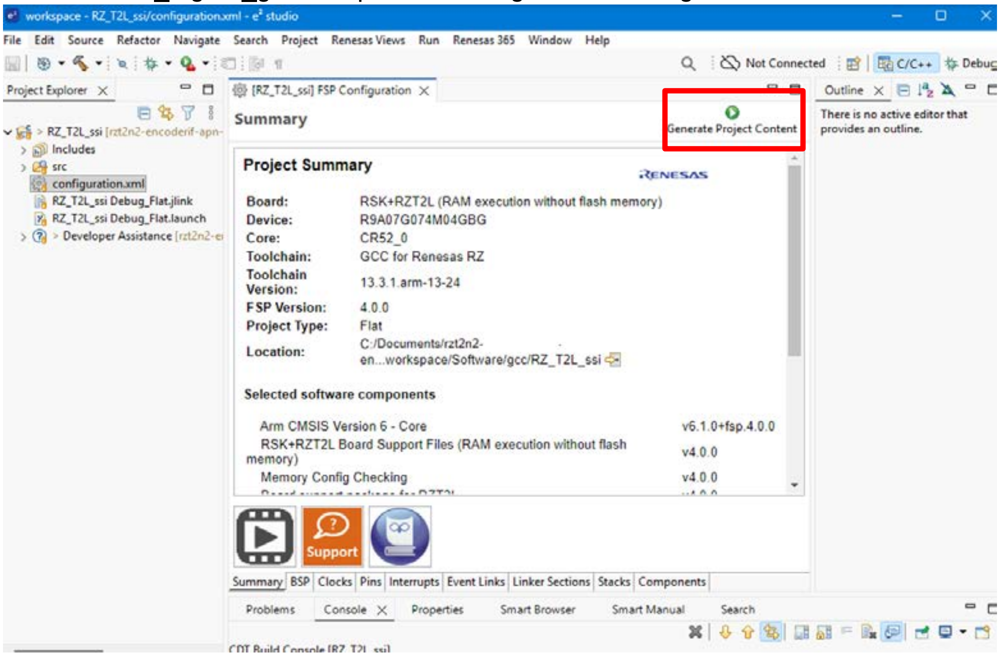
(3) Build Procedure for the Sample Program

The procedure for building the sample program is as follows.

1. Copy the extracted source files to any location.
2. After launching e² studio and moving to the workspace, click the [File] menu -> [Import] and select Existing project to 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 e^2 studio.
The rz, rz_cfg, rz_gen, script and .settings folders are generated.



7 Select [Project] menu -> [Build All]
The Debug¥RZ_T2L_ssi.elf file is generated.

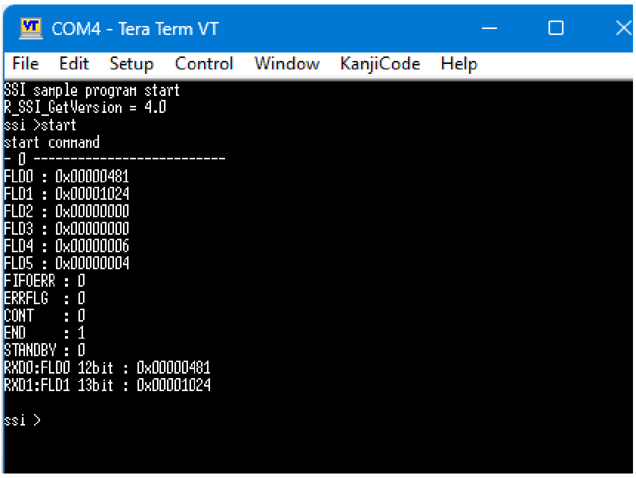
(4) Execution Procedure of the Sample Program

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/T2L Group SSI Sample Program Application Note.



Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar 31.23	-	First Edition issued
1.10	May 31.23	1, 4 2, 3	Appended section 3.1.2 for memory size information. Updated the release note version number.
1.20	Oct 20.23	2, 3 3 4 6 to 9	Updated the source code and the release note version number. Source code is corrected for supporting encoders with various data length. Updated file structure. (Removed RZ/T2L Pin Configuration data from zip file.) Updated memory size information. Updated build environment for FSP v1.3.0. Figures are replaced.
1.30	Mar 15.24	2, 3 4 1, 5 5	Update the application note and release note version number. Update sample program version to 1.2. (It supports elc_start command emulating ELC by using CPU and DMA. Changes SPI channels used for SSI interface.) Update memory size information. Update board name description. Update board setting and pins information by changing used SPI channels.
2.00	Jun 07.24	2, 3 4 6 to 9	Update the release note version number. Update the source code version. (Support FSP v2.0.0.) Update memory size information. Update build environment for FSP v2.0.0. Figures are replaced.
3.00	Oct 3.25	1 2, 3 4 6 to 9	Update description of the trademarks. Update revisions of the application note and the release note. Update sample program version to 3.0. (Support FSP v3.0.0.) Update memory size information. Update build environment for FSP v3.0.0. Figures are replaced.
4.00	Jun 26.26	2, 3 4 6 to 9	Update revisions of the application note and the release note. Update sample program version to 4.0. (Support FSP v4.0.0.) Update memory size information. Update build environment for FSP v4.0.0. 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.

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/.