RZ/A2M Group

SSIF Sample Application

Introduction

This application note describes a sample application that transmits and receives data using the Serial Sound Interface with FIFO (SSIF-2) with the Direct Memory Access Controller (DMAC).

Target Device

RZ/A2M

Contents

List of Abbreviations and Acronyms

Abbreviation	Full Form
API	Application Programming Interface
ARM	Advanced RISC Machines
COM	COMmunications
CPG	Clock Pulse Generator
CPU	Central Processing Unit
DIP	Dual In-line Package
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
FIFO	First In First Out
GPIO	General Purpose Input Output
INTC	INTerrupt Controller
I/O	Input/Output
IRQ	Interrupt ReQuest
LED	Light Emitting Diode
MCU	MicroController Unit
MHz	MegaHertz
OS	Operating System
PC	Personal Computer
SC	Smart Configurator
SSIF	Serial sound interface with FIFO
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus

1. Specifications

Table 1-1 lists the on-chip peripheral modules and their application, and Figure 1.1 shows the setup required for running the sample application.

Peripheral module	Usage
Serial sound interface (SSIF)	The audio data connection from the CODEC (WM8978) is connected to SSIF interface channel 0.
Direct Memory Access Controller (DMAC)	Transfers data from SSIF registers directly to and from memory
Renesas Serial Peripheral Interface (RSPI)	Configure Audio CODEC (WM8978).
Interrupt controller (INTC)	Configures interrupt settings; the processor will receive interrupts during buffered serial communications, and when transmission or reception has completed after DMA transfers
Clock Pulse Generator (CPG)	Configures the main CPU clock
General Purpose Input Output (GPIO)	Configures I/O lines used by ssif communications

Table 1.1 Peripheral module used

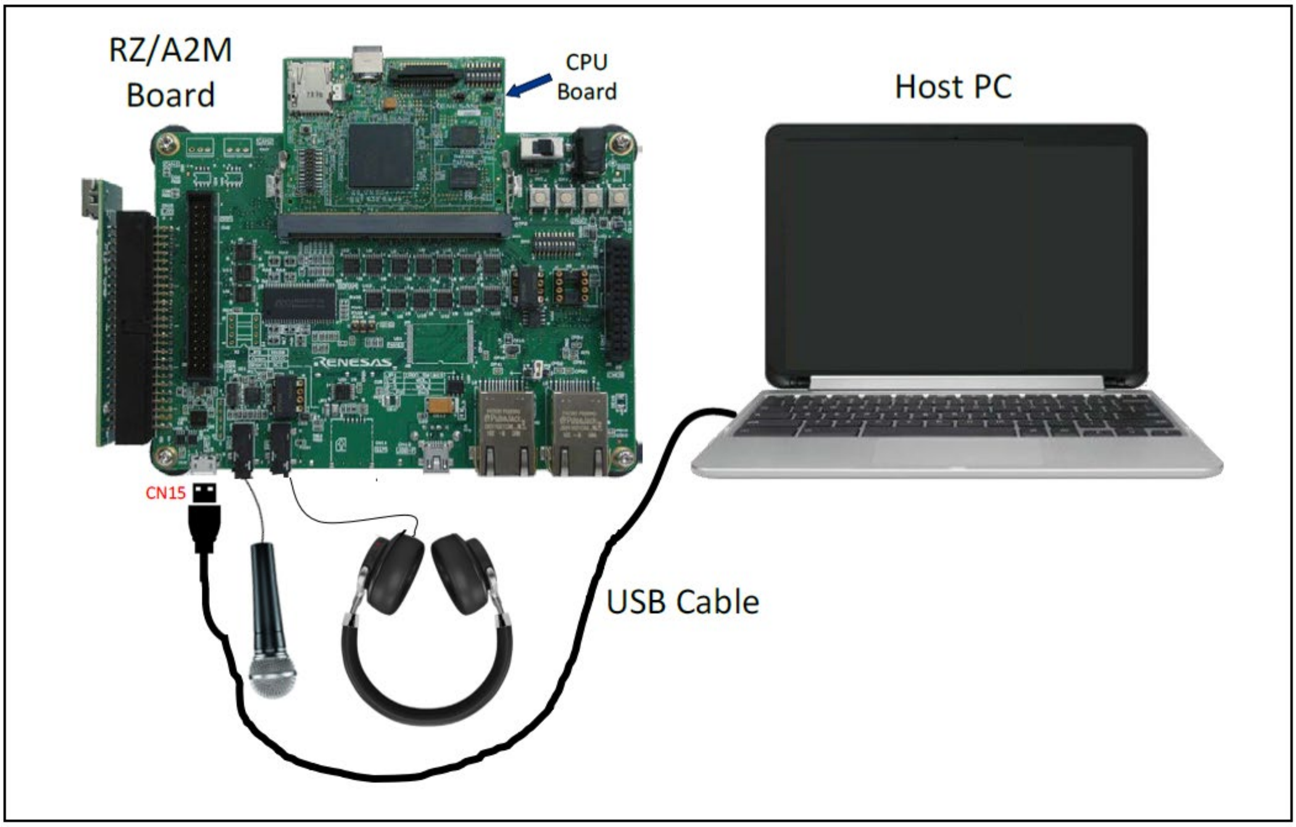

Figure 1.1 Setup for operating the sample program

1.1 Jumper and DIP Switch Settings

Set the DIP switches and jumpers of the sub board as follows.

SW6-1	OFF	Setting to use P9_[7:0], P8_[7:1], P2_2, P2_0, P1_3, P1_[1:0], P0_[6:0], P6_7, P6_5, P7_[1:0] and P7[5:3] as DRP, audio, UART and USB interface pins
SW6-2	OFF	Setting to use P8_4, P8_[7:6], P6_4 and P9_[6:3] as audio interface pins
SW6-3	OFF	Setting to use P9_[1:0], P1_0 and P7_5 as UART and USB interface pins
SW6-4	OFF	Setting to use P6_[3:1] and PE_[6:0] as CEU pins
SW6-5	OFF	Setting to use P3_[5:1], PH_5 and PK_[4:0] as FLCTL pins
SW6-6	ON	Setting to use digital image signal input output connector (CN15)
SW6-7	ON	Setting to use digital image signal input output connector (CN15)
SW6-8	OFF	NC
SW6-9	OFF	P5_3 = "H"
SW6-10	OFF	PC_2 = "H"
JP1	2-JP2	Setting to use PJ_1 as interrupt pin for IRQ0 switch (SW3)

Refer to the CPU board and the sub board user's manual for more details about setting for the DIP switches and jumpers.

2. Verified Operating Conditions

Table 2-1 Verified Operating Conditions

Item	Contents
Microcontroller used	RZ/A2M
Operating frequency (Note 1)	CPU Clock (Iφ): 528MHz Image processing clock (Gφ): 264MHz Internal Bus Clock (Bφ): 132MHz Peripheral Clock 1 (P1φ): 66MHz Peripheral Clock 0 (P0φ): 33MHz QSPI0_SPCLK: 66MHz CKIO: 132MHz
Operating voltage	Sub board power supply voltage: 5.0 V
Integrated Development Environment	e^2 studio 2022-01
C compiler	GNU Arm Embedded Tool chain 9-2020-q2-update Compiler options (except directory path): Release: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Os -ffunction-sections -fdata-sections -Wnull-dereference -Wstack-usage=256 -fabi-version=0 Hardware Debug: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Og -ffunction-sections -fdata-sections -Wnull-dereference -g3 -Wstack-usage=256 -fabi-version=0
Operation mode	Boot mode 3 (Serial Flash boot 3.3V)
Terminal software communication settings	• Communication speed: 115,200bps • Data length: 8 bits • Parity: None • Stop bits: 1 bit • Flow control: None
Board to be used	RZ/A2M CPU board RTK7921053C00000BE RZ/A2M SUB board RTK79210XXB00000BE
Device functionality used on the board	• Serial flash memory allocated to SPI multi-I/O bus space (channel 0) Manufacturer: Macronix Inc. Model Name: MX25L51245GXD • RL78/G1C (converts between USB and serial communications to communicate with the host PC) • WM8978 • LED1

Note 1: The operating frequency used in clock mode 1 (24MHz clock input from EXTAL pin)

3. Running the Sample Code

3.1 Playback Software Application

The playback software application plays the prerecorded audio file through the headphone connection when the "play" command is entered on the console. The audio file is included directly in the source code LR_44_1K16B_S.dat file. This file is encoded in stereo 16-bit, 44.1kHz format.

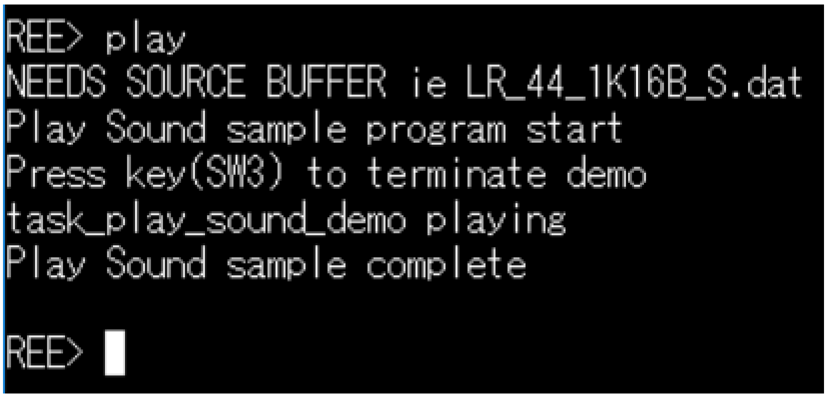

Figure 3.1 Play Sound Application Command Console

3.1.1 Program Operation

The playback application starts in the function R_SOUND_PlaySample, which is run when the command ‘play’ is entered in to the console .
This function initializes a control structure for the sound playback and calls a function play_file_data to manage the audio playback.
The function play_file_data creates a task for the playback, task_play_sound_demo, then waits for completion or a user keypress before finishing. The playback task, task_play_sound_demo, opens the SSIF and sound drivers. It then loops through the audio file, sending blocks of audio data via DMA to the SSIF peripheral to be sent to the CODEC. When playback is complete, the audio is closed and an event is set to signal for the calling function, play_file_data, to close the task and finish.
DMA transfers from the audio file to the SSIF peripheral are managed by the SSIF driver using an array of AIOCB messaging blocks, which define the access control semaphore and callback function for each block of data to be transferred. The appropriate AIOCB messaging block is registered with the SSIF driver and a write initiates the transfer of data to the SSIF peripheral from the relevant point of the audio file. Once the transfer is complete, the access semaphore is released and the next block is set to transfer, until the whole file has been transferred. When complete, the SOUND driver and SSIF drivers are closed and the calling function notified via the event to close the task.

See Figure 3.2 Play Sound Application Flowchart

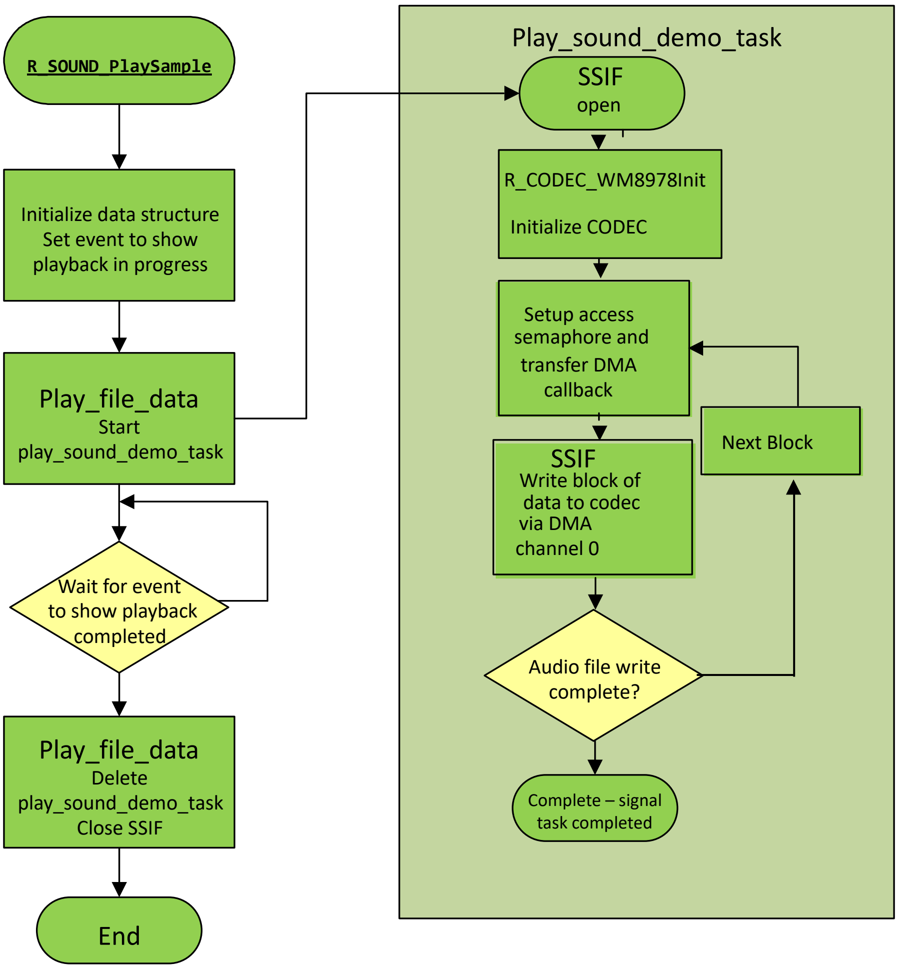

Figure 3.2 Play Sound Application Flowchart

3.2 Record Software Application

When a "record" command is entered at the console, the recording software application reads the input from the line-in connected to the audio jack and outputs it to the headphone connection in software "loopback" mode.

Figure 3.3 Record Software Application Command Console

3.2.1 Program Operation

R_SOUND_RecordSample(), which is run when the command ‘record’ is entered in to the console. This function initializes a control structure for the audio operation and calls the function play_recorded(). The function play_recorded() creates a buffer area, initializes access control semaphores, and configures the SOUND driver, before creating a task. Then task_record_sound_demo() waits for a user keypress before closing the task and finishing.

The task, task_record_sound_demo(), initializes the SSIF messaging structures for the transmit and receive operations, before entering a loop, receiving and transmitting audio information when transfers have completed, when indicated by flagging from end-of-transfer callback functions. This continues until a keypress is detected in the parent function, which then closes the task and completes.

The streaming of the audio data to/from the SSIF peripheral is organized using a number of buffers (set by #define NUM_AUDIO_BUFFER_BLOCKS_PRV_, set to 3). DMA transfers to and from the SSIF peripheral are managed by the SSIF driver using an array of AIOCB messaging blocks, for each receive and transmit channel. These define the access control semaphores and callback functions for each block of data to be transferred. When a SSIF read or write is ready to be setup, the SSIF driver is configured with the relevant AIOCB messaging block and the SSIF driver read or write command initiates the transfer of data. Once the transfer is complete, the access semaphore is released and the next block is set to transfer. As the read and write operations are working on the same data area, they are sharing the same semaphore accessing.

See Figure 3.4 Record Software Application Flowchart

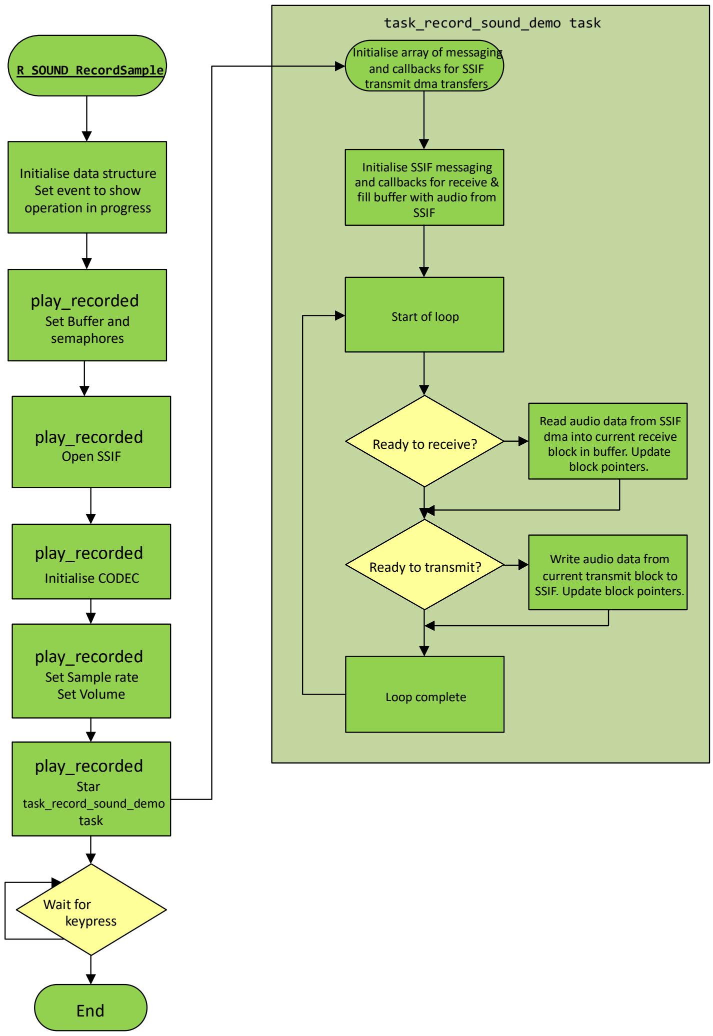

Figure 3.4 Record Software Application Flowchart

3.3 Peripheral Configuration

Smart Configurator is used to configure the two peripherals. This is invoked in e² studio by double-clicking the **ssif_sample_freertos.scfg** file in the root of the project from within Project Explorer.

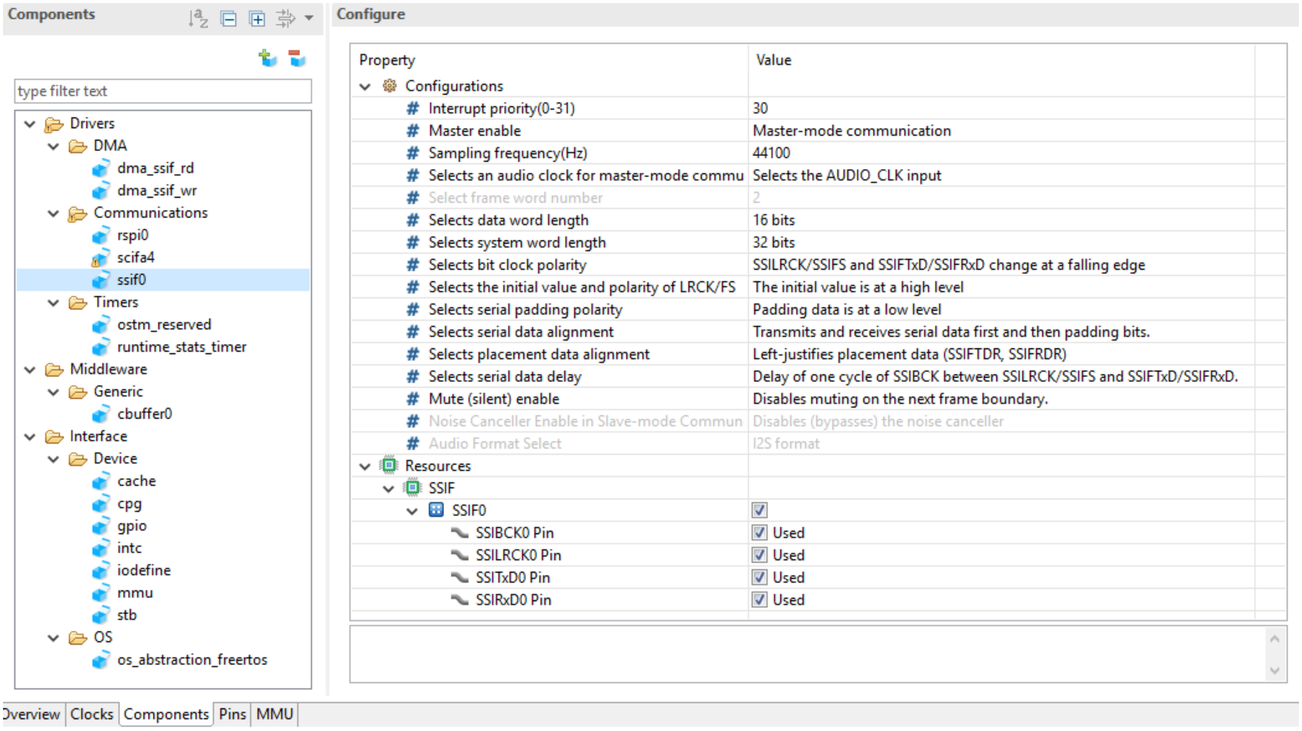

Figure 3.5 Smart Configurator in e2 studio

Peripherals are set up from the Smart Configurator **Components** tab (at the bottom of the window).

In this sample there are three significant peripheral configurations – **ssif0**, **dma_ssif_rd**, and **dma_ssif_wr**. The first of these; **ssif0** is the SSIF configuration for transmit and receive using DMA and is assigned to hardware SSIF channel 0. This configuration is shown in Figure 3.6 and Figure 3.7.

dma_ssif_rd is the DMAC Smart Configuration that is used for the DMA transfer during the receive operation of **ssif0**. DMAC channel 1 is assigned to this configuration. It is shown in Figure 3.6 below.

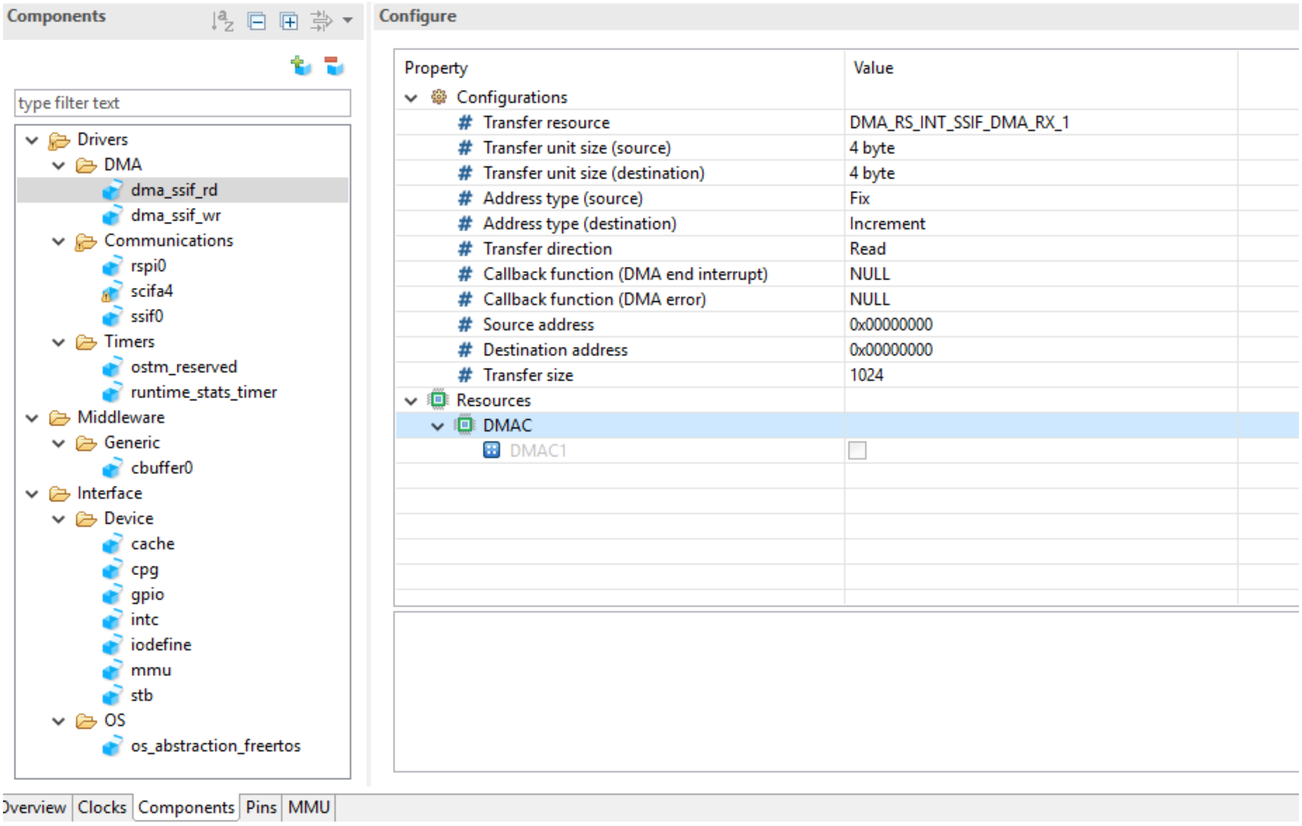

Figure 3.6 Smart Configuration settings for " dma_ssif_rd " SSIF configuration

The settings in Figure 3.6 are described in Table 3-4 below:

Configuration setting	**Value**	**Comment**
Transfer resource	DMA_RS_INT_SSIF_DMA_RX_1	Specifies that the transfer resource is SSIF channel 1 receive
Transfer unit size (source)	4 byte	Data is received by the SSIF in 4-byte chunks
Transfer unit size (destination)	4 byte	Data is written to the memory buffer in 4-byte chunks
Address type (source)	Fix	The source address (the SSIF read register) is fixed
Address type (destination)	Increment	The destination address (the read buffer) increments after each byte is transferred
Transfer direction	Read	This setting is ignored – the DMAC transfer direction register is set automatically by the DMAC driver due to the transfer resource setting of DMA_RS_INT_SSIF_DMA_RX_1
Callback function (DMA end interrupt)	NULL	This setting is not used, but will be set automatically by the SSIF driver to the read complete callback when read() is called

Callback function (DMA error)	NULL	The error callback function is not used in this sample
Source address	0x00000000	This source address setting is not used as it will be set by the SSIF driver to the address of the SSIF peripheral read register when read() is called
Destination address	0x00000000	This destination address setting is not used as it will be set by the SSIF driver to the read buffer address when read() is called
Transfer size	1024	This transfer size setting is not used as it will be set to the transfer size requested when read() is called

Table 3-4 Description of Smart Configuration settings for " dma_ssif_rd"

dma_ssif_wr is the DMAC Smart Configuration that is used for the DMA transfer during the transmit operation of **ssif0**. DMAC channel 0 is assigned to this configuration. It is shown in Figure 3.7 below.

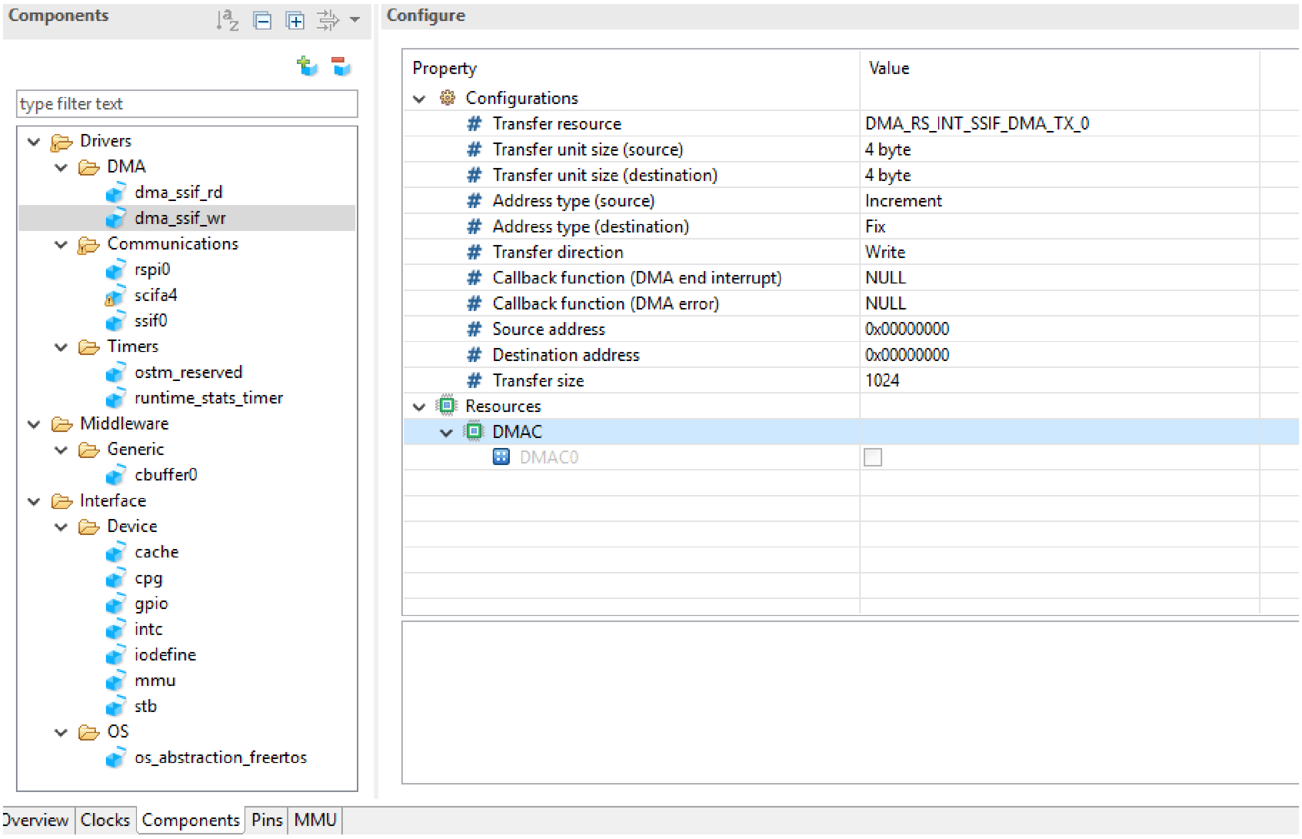

Figure 3.7 Smart Configuration settings for " dma_ssif_wr " DMAC configuration

The settings in Figure 3.7 are described in Table 3-5 below:

Configuration setting	**Value**	**Comment**
Transfer resource	DMA_RS_INT_SSIF_DMA_TX_0	Specifies that the transfer resource is SSIF channel 0 receive
Transfer unit size (source)	4 byte	*Data is received by the SSIF in 4-byte chunks*
Transfer unit size (destination)	4 byte	Data is written to the memory buffer in 4-byte chunks
Address type (source)	Increment	The source address (the transmit buffer) increments after each byte is transferred

Address type (destination)	Fix	The destination address (the SSIF transmit register) is fixed
Transfer direction	Write	This setting is ignored – the DMAC transfer direction register is set automatically by the DMAC driver due to the transfer resource setting of DMA_RS_INT_SSIF_DMA_TX_0
Callback function (DMA end interrupt)	NULL	This setting is not used, but will be set automatically by the SSIF driver to the write complete callback when write() is called
Callback function (DMA error)	NULL	The DMA error callback function is not used in this sample
Source address	0x00000000	This source address setting is not used as it will be set by the SSIF driver to the write buffer address when write() is called
Destination address	0x00000000	This destination address setting is not used as it will be set by the SSIF driver to the address of the SSIF peripheral transmit register when write() is called
Transfer size	1024	This transfer size setting is not used as it will be set to the transfer size requested when write() is called

Table 3-5 Description of Smart Configuration settings for " dma_ssif_wr "

4. Reference Documents

R01AN4467: RZ/A2M DMAC Driver Application Note
The latest version can be downloaded from the Renesas Electronics website.

R11AN5444: RZ/A2M SSIF Driver Application Note
The latest version can be downloaded from the Renesas Electronics website.

R01UH0746: RZ/A2M Group User's Manual: Hardware
The latest version can be downloaded from the Renesas Electronics website.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Sep.30.20	-	First edition issued
1.01	Jan.20.22	4	Update the version of complier. Update the version of e2 studio.

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

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/.