RZ/T1 Group

SSI Sample Program

R01AN2602EJ0140
Rev.1.40
Jun. 07, 2018

Introduction

This application note explains a sample program that uses the RZ/T1 SSI function to record and play audio.

The SSI sample program has the following features:

- Performing the following SSI communications with the WM8978 on the RSK RZ/T1 Evaluation Board:
  - Receiving inputs from the MIC pin on the WM8978
  - Returning the received data to the HP pin on the WM8978
  - Starting and stopping communications repeatedly every five seconds
  - SSI communication settings: 16 bits, 44.1 kHz, stereo, I2S format
- Displaying the operating status of the sample program on the SCIFA2 device

Target Devices

RZ/T1

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

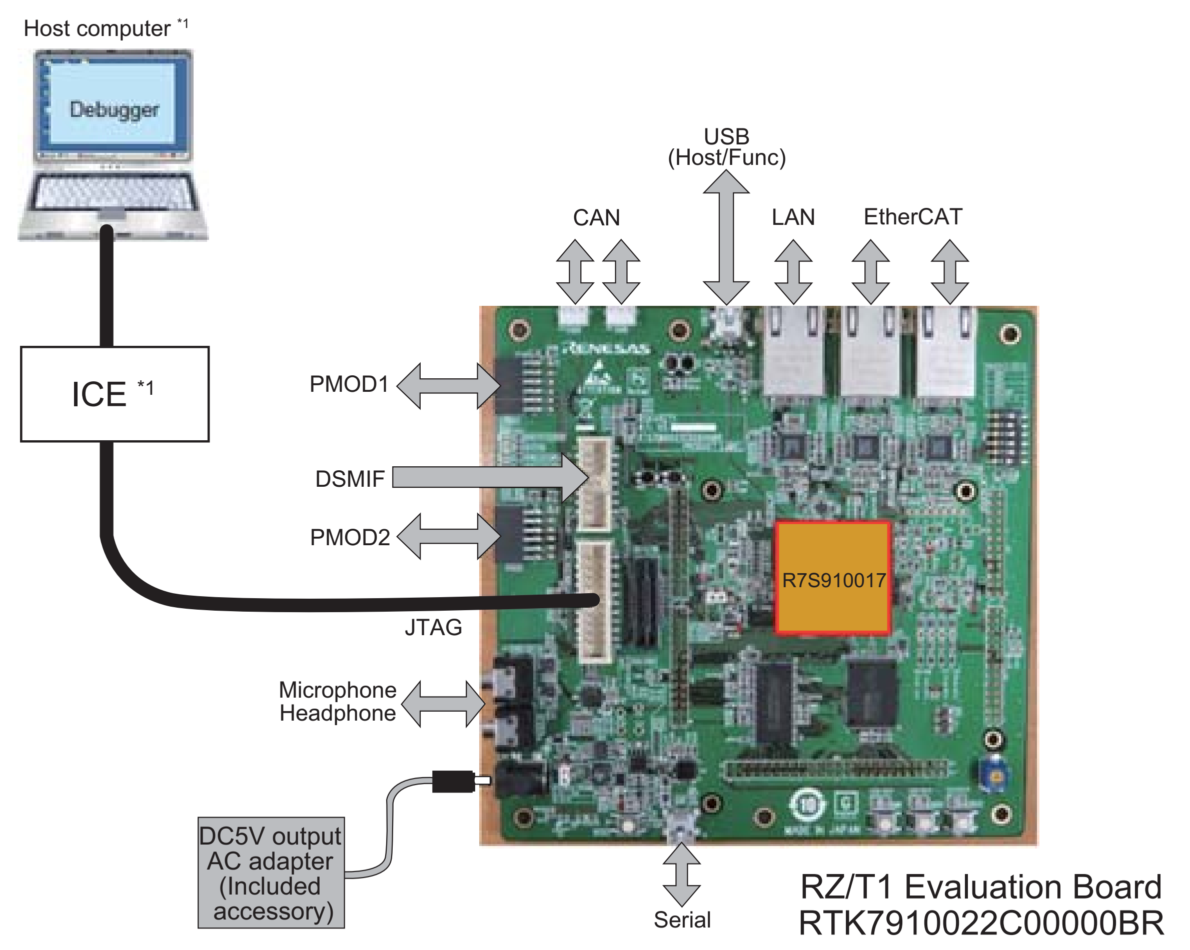
Table of Contents

1. Specifications

Table 1.1 Peripheral Functions and Applications lists the peripheral functions to be used and their applications, and Figure 1.1 shows the Operating Environment where the sample code is executed.

Table 1.1 Peripheral Functions and Applications

Peripheral Function	Application
RZ/T1 built-in serial sound interface (SSI)	Performs SSI communications with the WM8978 on the RSK RZ/T1 Evaluation Board. SSI is used to send and receive PCM data.
RZ/T1 built-in serial peripheral interface (RSPIa)	Performs RSPI communications with the WM8978 on the RSK RZ/T1 Evaluation Board. RSPI1 is used to set the WM8978 operating mode.
RZ/T1 built-in interrupt controller (ICUA)	Uses the following interrupt sources to wait for the SSI driver to complete transfers: SSI receive data full interrupt (ID: 127) SSI transmit data empty interrupt (ID: 128)
RZ/T1 built-in serial communication interface with FIFO (SCIFA)	Uses SCI2 to display debug messages.
RZ/T1 built-in compare match timer (CMT)	Performs timeout waiting polling processing within the sample program. A single channel is used within the RSPI sample program function.
RZ/T1 built-in compare match timer W (CMTW)	Performs timeout waiting polling processing within the sample program. A single channel is used within the SSI sample program function (a channel can be selected).
RZ/T1 built-in power saving function	Controls clock signal supply to the following modules used by the sample program: SSI, RSPIa, SCIFA, CMT, CMTW
RZ/T1 built-in I/O port RZ/T1 built-in multi-function pin controller (MPC)	Sets up the pins related to the following modules used by the sample program: SSI, RSPIa, SCIFA
WM8978 mounted on the RSK RZ/T1 Evaluation Board	Performs A/D and D/A conversion for PCM data.
R78/G1C-S mounted on the RSK RZ/T1 Evaluation Board	Converts SCIF communications to USB serial communications.



Note 1. Indicates the device that the user needs to prepare.

Figure 1.1 Operating Environment

2. Operating Environment

The sample code covered in this application note is for the environment below.

Table 2.1 Operating Environment

Item	Description
Microcomputer	RZ/T1 Group
Operating frequency	CPUCLK = 450 MHz
Operating voltage	3.3 V
Integrated Development Environment	Manufactured by IAR Systems Embedded Workbench® for Arm Version 8.20.2 Manufactured by Arm DS-5™ 5.26.2 Manufactured by RENESAS e2studio 6.1.0
Operating mode	SPI boot mode 16-bit bus boot mode
Board	RZ/T1 Evaluation Board (RTK7910022C00000BR)
Device (functions to be used on the board)	• NOR flash memory (connected to CS0 and CS1 spaces) Manufacturer: Macronix International Co., Ltd., Model: MX29GL512FLT2I-10Q • SDRAM (connected to CS2 and CS3 spaces) Manufacturer: Integrated Silicon Solution Inc, Model: IS42S16320D-7TL • Serial flash memory Manufacturer: Macronix International Co., Ltd., Model: MX25L51245G • Audio CODEC Manufacturer: Wolfson, Model: WM8978GEFL/V • USB-Serial controller Manufacturer: Renesas Electronics Co., Ltd., Model: R78/G1C-S

3. Related Application Note

The application notes related to this application note are listed below for reference.

- RZ/T1 Group: Application Note Initial Settings (R01AN2554EJ)
- RZ/T1 Group: Application Note RSPI Sample Program (R01AN2601EJ)

Note: For any registers not covered by this application note, use the values specified in the RZ/T1 Group: Application Note Initial Settings.

4. Peripheral Functions

The basics of the operating modes, serial sound interface (SSI), serial peripheral interface (RSPIa), interrupt controller (ICUA), serial communication interface with FIFO (SCIFA), compare match timer (CMT), compare match timer W (CMTW), power saving function, I/O port, and multi-function pin controller (MPC) are described in the RZ/T1 Group User's Manual: Hardware.

5. Hardware

5.1 Hardware Configuration

Figure 5.1 shows the Hardware Configuration.

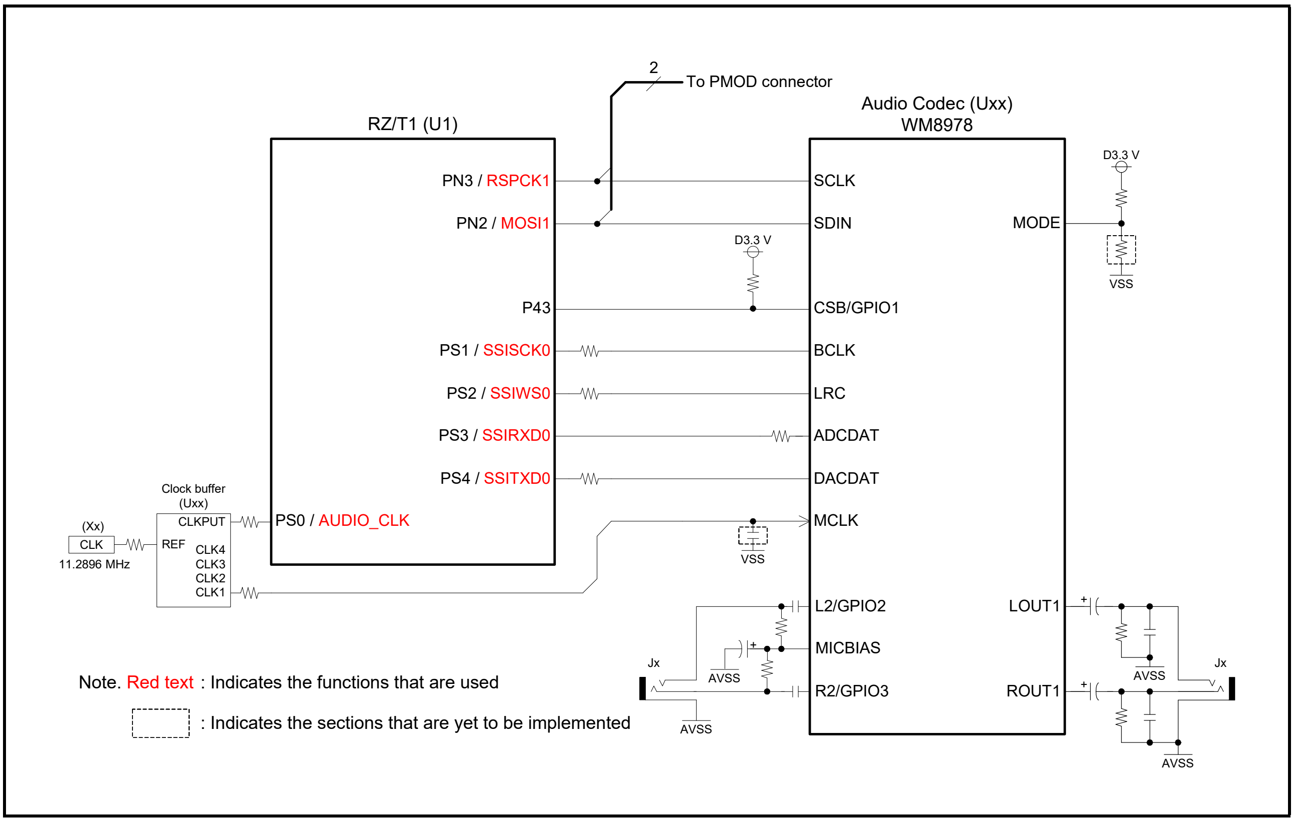


Figure 5.1 Hardware Configuration

5.2 Pins

Table 5.1 shows the Pins Used and Their Functions.

Table 5.1 Pins Used and Their Functions

Pin Name	Input/Output	Application
PS0/AUDIO_CLK	Input	Audio master clock input (11.2896 MHz) from the oscillator on the RSK RZ/T1 Evaluation Board to the RZ/T1 built-in SSI
PS1/SSISCK0	Input/Output	Serial bit clock input/output from the RZ/T1 built-in SSI to the WM8978 on the RSK RZ/T1 Evaluation Board (this pin is used as an output pin by the sample program)
PS2/SSIWS0	Input/Output	Word select (LR clock) input/output from the RZ/T1 built-in SSI to the WM8978 on the RSK RZ/T1 Evaluation Board (this pin is used as an output pin by the sample program)
PS3/SSIRXD0	Input	Serial audio data input from the WM8978 on the RSK RZ/T1 Evaluation Board to the RZ/T1 built-in SSI
PS4/SSITXD0	Output	Serial audio data output from the RZ/T1 built-in SSI to the WM8978 on the RSK RZ/T1 Evaluation Board
PN3/RSPCK1	Input/Output	RSPI clock input/output from the RZ/T1 built-in RSPIa to the WM8978 on the RSK RZ/T1 Evaluation Board (this pin is used as an output pin by the sample program)
PN2/MOSI1	Input/Output	Master transmission data input/output from the RZ/T1 built-in RSPIa to the WM8978 on the RSK RZ/T1 Evaluation Board (this pin is used as an output pin for the WM8978 control commands by the sample program)
P43	Output	RSPI slave select signal output from the RZ/T1 built-in I/O port to the WM8978 on the RSK RZ/T1 Evaluation Board (Because the SSL1x pins of RSPI1 are connected to the PMOD interface and not connected to the WM8978, the I/O port is used to control SSL signals for RSPI communications.)
P91/TXD2	Output	SCIF data output from the RZ/T1 built-in SCIFA to the R78/G1C-S on the RSK RZ/T1 Evaluation Board
P92/RXD2	Input	SCIF data input from the RZ/T1 built-in SCIFA to the R78/G1C-S on the RSK RZ/T1 Evaluation Board

Note: In addition to these pins, there are also pins that are used to initialize RZ/T1 products. For details, see the RZ/T1 Group: Application Note Initial Settings.

6. Software

6.1 Operation Outline

Table 6.1 Operation Outline presents a functional overview of the SSI sample program. Figure 6.1 shows the System Block Diagram for this program.

Table 6.1 Operation Outline

Function	Outline
SSI initialization	Executes the SSI driver initialization function.
WM8978 initialization	Initializes the WM8978 using the following settings: Audio interface Data Format: I2S Word length: 16 bits Sample rate: 44.1 kHz input PGA volume: 0 dB headphone output volume: 0 dB
SCIFA initialization	Initializes the SCIFA2 using the following settings: Synchronization mode: Asynchronous Baud rate: 115200 bps Character length: 8 bits Parity bit: None Stop bit: 1 bit Flow control: None This function also sets up the following SCIFA2 pins: • TXD2 (P91: Output) • RXD2 (P92: Input)
Compare match timer (CMTW) initialization	Initializes the compare match timer to measure the waiting time within the sample program.
SCIFA character output	Implements a single-character output function into the SCIFA2.
SSI sample program operation	Runs the sample program using the following parameter settings that causes the SSISCK pin to generate output signals with the same frequency as input clock signals (11.2896 MHz) to the AUDIO_CLK pin. • Sampling frequency: 44.1 kHz • System word length: 128 bits • Data word length: 32 bits

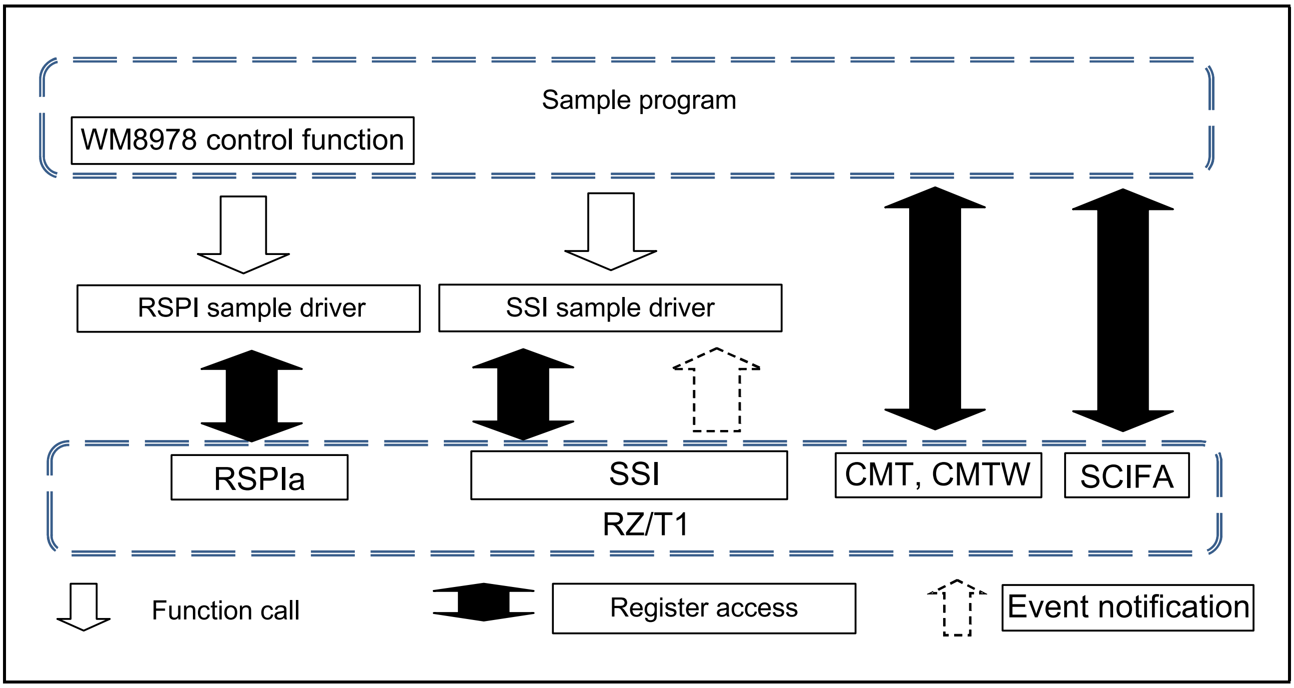


Figure 6.1 System Block Diagram

6.1.1 Project Setup

How to set up projects used in the EWARM development environment is described in the RZ/T1 Group Application Note Initial Settings.

6.1.2 Preparation

This section explains how to prepare for executing this sample program.

(1) Start the terminal emulator on the host PC, and specify settings as shown in the following window. (In this example, Tera Term is used as the terminal emulator.)

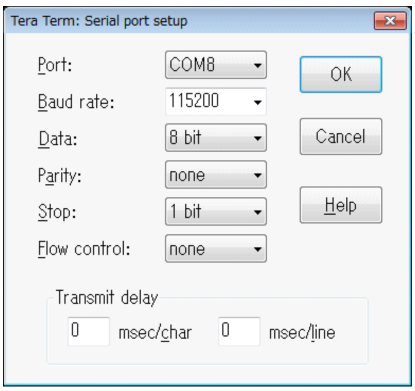


Figure 6.2 Serial Port Settings

(2) For the new-line code, select CR for both transmission and reception. (For Tera Term, select Terminal from the Setup menu, and select CR for both Transmit and Receive in the New-line section in the Terminal setup dialog box.)
(3) Connect the J8 connector on the RSK RZ/T1 Evaluation Board to the USB port on the host PC.
(4) Connect the J3 connector on the RSK RZ/T1 Evaluation Board to any sound generator.
(5) Connect the J5 connector on the RSK RZ/T1 Evaluation Board to a speaker.
(6) Executing the sample program outputs the following message to the terminal.

```
SSI driver version 00010000
WM8978 power on
HP Volume 0dB
Input PGA volume 0dB
Initialize SSI driver
Record and Play: start
Record and Play: pause
Record and Play: start
Record and Play: pause
Record and Play: start
Record and Play: pause
Record and Play: start
Record and Play: pause
Record and Play: start
Record and Play: pause
Record and Play: start
Record and Play: pause
Record and Play: start
Record and Play: pause
Record and Play: start
Record and Play: pause
Record and Play: start
Record and Play: pause
Record and Play: start
Record and Play: pause
Finish SSI driver
WM8978 power off
```

6.2 Memory Mapping

Memory mapping for the address spaces in the RZ/T1 Group MCU and the memory in the RZ/T1 Evaluation Board is described in the RZ/T1 Group Application Note Initial Settings.

6.2.1 Section Allocation for the Sample program

The sections used by the sample program, the section allocation for the sample program in the initial state (load view), and the section allocation for the sample program after the scatter loading function is used (execution view) are described in the RZ/T1 Group Application Note Initial Settings.

6.2.2 MPU Setup

MPU setup is described in the RZ/T1 Group Application Note Initial Settings.

6.2.3 Exception Processing Vector Table

Exception processing vector tables are described in the RZ/T1 Group Application Note Initial Settings.

6.3 Interrupts

Table 6.2 shows the Interrupts for the Sample Code.

Table 6.2 Interrupts for the Sample Code

Interrupt (Source ID)	Priority	Process Outline
RSPI Unit1 transmit buffer empty interrupt (85)	7	Waiting for the completion of WM8978 command transmission
SSI receive data full interrupt (127)	7	Transferring the PCM data to be recorded
SSI transmit data empty interrupt (128)	7	Transferring the PCM data to be played
SSI error interrupt (266)	7	Detecting SSI error occurrences

6.4 Fixed-Width Integer Types

Table 6.3 shows the Fixed-Width Integer Types for the Sample Code.

Table 6.3 Fixed-Width Integer Types for the Sample Code

Symbol	Description
int8_t	8-bit signed integer (defined in the standard library)
int16_t	16-bit signed integer (defined in the standard library)
int32_t	32-bit signed integer (defined in the standard library)
int64_t	64-bit signed integer (defined in the standard library)
uint8_t	8-bit unsigned integer (defined in the standard library)
uint16_t	16-bit unsigned integer (defined in the standard library)
uint32_t	32-bit unsigned integer (defined in the standard library)
uint64_t	64-bit unsigned integer (defined in the standard library)

6.5 Constants/Error Codes

Table 6.4 shows the Constants for the Sample Code, and Table 6.5 shows the Error Codes for the Sample Code.

Table 6.4 Constants for the Sample Code

Constant Name	Setting Value	Description
SSI_STS_CLOSE	(0x00000000u)	Driver status: Not open
SSI_STS_IDLE	(0x00000001u)	Driver status: Transfer stopped
SSI_STS_TRANS	(0x00000002u)	Driver status: Transfer in progress
SSI_BUF_PAGE_NUM	(2u)	Number of pages in the SSI data buffer
SSI_BUF_PAGE_DWSZ	(3528u)	Page size of the SSI data buffer (unit: 4 bytes)
SSI_SSICR_INIVAL	(0x000BC000u)	Initial values of the SSICR register • I2S format • Data word length: 16 bits • System word length: 32 bits The CKDV field is overwritten with the value calculated by ssi_calc_ckdv() after the initial values are set. For details on the SSICR register, refer to the RZ/T1 Group User's Manual: Hardware.
SSI_SSIFCR_INIVAL	(0x80000000u)	Initial value of the SSIFCR register • Master mode For details on the SSIFCR register, refer to the RZ/T1 Group User's Manual: Hardware.
SSI_SSITDMR_INIVAL	(0x00000000u)	Initial value of the SSITDMR register • WS continue mode disabled For details on the SSITDMR register, refer to the RZ/T1 Group User's Manual: Hardware.
SSI_AUDIO_CLK	(11289600u)	Audio input clock (Hz)
SSI_SAMPLE_FREQ	(44100u)	Sampling frequency (Hz)
SSI_ERR_MSGFN	uart_puts	Pointer to the error message output function
CMTW_CH	(0)	Timer for time-wait processing for the sample program The CMTW channel with the specified number is used.
SERICLK	(1)	The value to be set in the SERICK bit of the SCKCR register that determines the input clock to the message output SCIF for the sample program 0: 150 MHz, 1: 120 MHz

Table 6.5 Error Codes for the Sample Code

Constant Name	Setting Value	Description
SSI_OK	0	Normal termination
SSI_ER_CLOSED	-1	An attempt was made to call a function when the SSI driver is closed.
SSI_ER_OPENED	-2	An attempt was made to call a function that cannot be executed when the SSI driver is open.
SSI_ER_RUNNING	-3	An attempt was made to call a function that cannot be executed when transfer is in progress.
SSI_ER_STOPPED	-4	An attempt was made to call a function that cannot be executed when transfer is stopped.

6.6 Structures, Unions, and Enumerated Types

There are no structures, unions, or enumerated types to be defined in the sample code.

6.7 Global Variables

Table 6.6 shows the Global Variables for the sample code.

Table 6.6 Global Variables

Type	Variable Name	Description	Function
static uint32_t	gb_ssi_sts	SSI driver status	R_SSI_Open() R_SSI_Close() R_SSI_Start() R_SSI_Stop()
static uint32_t	gb_ssi_buf[]	The SSI ring buffer size is as follows: SSI_BUF_PAGE_NUM * SSI_BUF_PAGE_DWSZ	R_SSI_Start() ssi_rxf_isr() ssi_txe_isr()
static ssi_ringbuf_t	gb_ssi_buf_ctl	Ring buffer control	R_SSI_Start() ssi_rxf_isr() ssi_txe_isr()

6.8 Functions

Table 6.7 lists the functions defined in this document.

Table 6.7 Functions

Function Name	Page Number
R_SSI_Open	18
R_SSI_Close	19
R_SSI_Start	19
R_SSI_Stop	20
R_SSI_GetVersion	20
ssi_rxf_isr	21
ssi_txe_isr	21
ssi_err_isr	22
ssi_set_pin	23
ssi_calc_ckdv	24
main	24
uart_init	25
uart_putc	25
uart_getc	26
uart_puts	26
cmtw_delay	27

Table 6.8 lists the functions defined in the related application notes that are referred to in this sample program.

Table 6.8 Externally Referenced Functions

Function Name	Description
wm8978_init	Used to turn on and initialize the WM8978. This function sets the following items: • Audio interface Data Format: I2S • Word length: 16 bits • Sample rate: 44.1 kHz For details, refer to the Application Note: RZ/T1 RSPI Sample Driver.
wm8978_output_volume_set	Used to set the headphone output volume level for the WM8978. This function sets the following item: • headphone output volume: 0 dB For details, refer to the Application Note: RZ/T1 RSPI Sample Driver.
wm8978_input_volume_set	Used to set the microphone input volume level for the WM8978. This function sets the following item: • input PGA volume: 0 dB For details, refer to the Application Note: RZ/T1 RSPI Sample Driver.
wm8978_shutdown	Used to turn off the WM8978. For details, refer to the Application Note: RZ/T1 RSPI Sample Driver.

6.9 Specifications of Functions

This section presents the specifications of the functions for the sample code.

6.9.1 R_SSI_Open

R_SSI_Open

Synopsis	Initializing the SSI driver
Header	r_ssi_rzt1_if.h
Declaration	int32_t R_SSI_Open(void);
Description	This function performs the following processing as the SSI driver initialization processing: • Setting up the power saving function - Starting clock supply to the SSI module • Setting up the I/O port and MPC - Setting up the SSI pins (ssi_set_pin) • Setting up the SSI - Setting up the audio format - Calculating clocks (ssi_calc_ckdv) • Setting up the ICUA - Setting up the interrupt enable register to enable SSI receive data full interrupts - Setting up the interrupt enable register to enable SSI transmit data empty interrupts - Setting up the interrupt enable register to enable SSI error interrupts - Registering the interrupt handlers (ssi_rxf_isr, ssi_txe_isr) with the respective address storage registers for SSI receive data full interrupts and SSI transmit data empty interrupts - Registering the interrupt handler (ssi_err_isr) with the address storage register for SSI error interrupts - Setting each interrupt priority level in the interrupt priority registers
Arguments	None
Return values	SSI_OK : Normal termination SSI_ER_OPENED : Function called when the SSI driver is open
Remarks	After the driver is initialized, the SSI operating mode is set up as follows: • I2S format • System word length: 32 bits • Data word length: 16 bits • SSI master mode • WS continue mode disabled For details on the register settings for the SSI, see Section 6.5, Constants/Error Codes.

6.9.2 R_SSI_Close

R_SSI_Close	
Synopsis	Closing the SSI driver
Header	r_ssi_rzt1_if.h
Declaration	int32_t R_SSI_Close(void);
Description	This function performs the following processing as the SSI driver closing processing: • Setting up the ICUA - Setting up the interrupt enable register to disable SSI receive data full interrupts - Setting up the interrupt enable register to disable SSI transmit data empty interrupts • Setting up the power saving function - Stopping clock supply to the SSI module
Arguments	None
Return values	SSI_OK : Normal termination SSI_ER_CLOSED : Function called when the SSI driver is closed SSI_ER_RUNNING : Function called when transfer is in progress
Remarks	SSI driver closing processing does not perform setup processing for the I/O port, MPC, or SSI. This function can be executed only when the transfer is stopped.

6.9.3 R_SSI_Start

R_SSI_Start	
Synopsis	Starting transfers by the SSI driver
Header	r_ssi_rzt1_if.h
Declaration	int32_t R_SSI_Start(void);
Description	This function performs the following processing as the transfer start processing for the SSI driver: • Resetting the transmit FIFO and receive FIFO for the SSI • Initializing the SSI data buffer - Setting the pointer to transmitted data to the top of page 0 - Setting the pointer to received data to the top of page 1 - Clearing page 0 to zero • Writing the amount of data equivalent to the FIFO size from the pointer to transmitted data to the transmit FIFO • Enabling various SSI interrupts • Enabling both transmission and reception simultaneously for the SSI (transfer then starts)
Arguments	None
Return values	SSI_OK : Normal termination SSI_ER_CLOSED : Function called when the SSI driver is closed SSI_ER_RUNNING : Function called when transfer is in progress

6.9.4 R_SSI_Stop

R_SSI_Stop

Synopsis	Stopping the SSI driver from transferring data
Header	r_ssi_rzt1_if.h
Declaration	int32_t R_SSI_Stop(void);
Description	This function performs the following processing as the transfer stop processing for the SSI driver: • Disabling various SSI interrupts (stopping the handler from adding data to be transmitted) • Performing transfer stop processing for the SSI (waiting for the transmit FIFO to become empty, and changing the transmission size to a multiple of 64 bits) • Disabling both transmission and reception simultaneously for the SSI (transfer then stops) • Waiting for the SSI to enter an idle state
Arguments	None
Return values	SSI_OK : Normal termination SSI_ER_CLOSED : Function called when the SSI driver is closed SSI_ER_STOPPED : Function called while transfer is stopped

6.9.5 R_SSI_GetVersion

R_SSI_GetVersion

Synopsis	Obtaining the SSI driver version
Header	r_ssi_rzt1_if.h
Declaration	uint32_t R_SSI_GetVersion(void);
Description	This function returns the version of the SSI driver.
Arguments	None
Return values	SSI driver version : Bits 31 to 16 - Major version : Bits 15 to 0 - Minor version
Remarks	For the first version, "1.0" returns.

6.9.6 ssi_rxf_isr

ssi_rxf_isr	
Synopsis	SSI receive data full interrupt handler
Header	–
Declaration	void ssi_rxf_isr(void);
Description	This function is a handler used for the following interrupt: • SSI receive data full interrupt This interrupt performs the following reception processing: • Obtaining the data count for the receive FIFO • Reading the amount of data (equivalent to the data count for the receive FIFO) from the receive data register • Updating the pointer to the ring buffer
Arguments	None
Return values	None

6.9.7 ssi_txe_isr

ssi_txe_isr	
Synopsis	SSI transmit data empty interrupt handler
Header	–
Declaration	void ssi_txe_isr(void);
Description	This function is a handler used for the following interrupt: • SSI transmit data empty interrupt This interrupt performs the following transmission continuation processing: • Obtaining the empty interrupt count for the transmit FIFO • Adding the amount of data (equivalent to the empty interrupt count for the transmit FIFO) from the ring buffer • Updating the pointer to the ring buffer
Arguments	None
Return values	None

6.9.8 ssi_err_isr

ssi_err_isr	
Synopsis	SSI driver error interrupt handler
Header	–
Declaration	void ssi_err_isr(void);
Description	This function is an interrupt handler that detects the following error interrupts: • SSI transmission underflow interrupt • SSI transmission overflow interrupt • SSI reception underflow interrupt • SSI reception overflow interrupt This function displays error occurrence messages (the values of the SSISR and SSIFSR registers).
Arguments	None
Return values	None
Remarks	Error messages are displayed by the function specified in the SSI_ERR_MSGFN macro. If an error occurs, SSI transfers will stop. To restart the transfer after an error has occurred, execute R_SSI_Stop() once to clear the internal state of the driver, and then execute R_SSI_Start().

6.9.9 ssi_set_pin

ssi_set_pin	
Synopsis	Setting up the pins for the SSI driver
Header	–
Declaration	void ssi_set_pin(void);
Description	This function sets up the following pins used by the SSI sample driver: • AUDIO_CLK • SSISCK0 • SSIWS0 • SSIRXD0 • SSITXD0 This function is called within the R_SSI_Open() function by the SSI sample driver.
Arguments	None
Return values	None
Remarks	The RZ/T1 Group provides two sets of five pins to which each SSI function can be assigned. Because these pins depend on the board where they are mounted, separate functions are provided to facilitate customization. Implementing the SSI sample driver allocates each SSI function to the following pins: • AUDIO_CLK : PS0, input • SSISCK0 : PS1, input/output • SSIWS0 : PS2, input/output • SSIRXD0 : PS3, input • SSITXD0 : PS4, output For details on the I/O port and multi-function pin controller (MPC) that are used to set up pins, refer to the RZ/T1 Group User's Manual: Hardware.

6.9.10 ssi_calc_ckdv

ssi_calc_ckdv	
Synopsis	Calculating the division ratio of serial bit clocks for the SSI driver
Header	–
Declaration	uint32_t ssi_calc_ckdv(void);
Description	This function calculates the division ratio of serial bit clocks to master clocks, and returns the value to be set in the CKDV field of the SSICR register. This function is called within the R_SSI_Open() function by the SSI sample driver.
Arguments	None
Return values	0x00 to 0xC : CKDV field value of the SSICR register
Remarks	Because the division ratio of serial bit clocks depends on audio master clocks (MCLK) input to the RZ/T1 Group, audio transfer format settings, and other factors, separate functions are provided to facilitate customization. Implementing the SSI sample driver calculates clock division ratios based on the following conditions and returns 2 as a return value (MCLK/4). • Master clock: 11.2896 MHz • Audio transfer format: I2S • Sampling frequency: 44.1 kHz • System word length: 32 bits For details on the CKDV field of the SSICR register, refer to the RZ/T1 Group User's Manual: Hardware.

6.9.11 main

main	
Synopsis	The main unit of the sample program
Header	–
Declaration	int32_t main(void);
Description	This function is the main unit of the sample program. For details, see the flowchart in Section 6.10.1, Main Processing.
Arguments	None
Return values	0 : The sample program completed operations

6.9.12 uart_init

uart_init	
Synopsis	Initializing the serial port
Header	ssi_sample.h
Declaration	void uart_init(void);
Description	This function initializes the serial port for debug messages.
Arguments	None
Return values	None
Remarks	Implementing the SSI sample driver performs the following initialization processing: Initializing the SCIFA • Channel: SCI2 • Synchronization mode: Asynchronous • Baud rate: 115200 bps • Character length: 8 bits • Parity bit: None • Stop bit: 1 bit • Flow control: None Setting up the SCIFA pins • TXD2 (P91: Output) • RXD2 (P92: Input)

6.9.13 uart_putc

uart_putc		
Synopsis	Outputting a single character to the serial port	
Header	ssi_sample.h	
Declaration	void uart_putc(uint8_t cha);	
Description	This function outputs a single character to the serial port. If the output buffer is not empty, a busy loop is repeated within the function until the output buffer becomes empty.	
Arguments	uint8_t cha	The character to be output
Return values	None	

6.9.14 uart_getc

uart_getc		
Synopsis	Inputting a single character from the serial port	
Header	ssi_sample.h	
Declaration	int32_t uart_getc(void);	
Description	This function inputs a single character from the serial port. If there is no character to be input, a busy loop is repeated within the function until input occurs.	
Arguments	None	
Return values	0 to 255	: Input character
	-1	: Read error

6.9.15 uart_puts

uart_puts		
Synopsis	Outputting a string to the serial port	
Header	ssi_sample.h	
Declaration	int32_t uart_puts(const uint8_t *p_str);	
Description	This function outputs a string to the serial port. This function repeats transmitting a single character with the uart_putc() function until "\0" is found from the beginning of the string.	
Arguments	const uint8_t *p_str	The string to be output (ending with \0)
Return values	An integer not less than 0	: Number of output bytes

6.9.16 cmtw_delay

cmtw_delay	
Synopsis	Waiting for specified time
Header	ssi_sample.h
Declaration	void cmtw_delay(uint32_t msec);
Description	This function repeats a busy loop for the time specified by the argument. This function measures time with the compare match timer (CMTW) and performs the following processing: • Starting clock supply to the CMTW • Clearing the counter register to zero • Initializing the constant register to 0xFFFFFFFF (to enter a free-run state) • Calculating the elapsed time corresponding to the value specified by the msec argument • Starting the timer • Polling the counter register by repeating a busy loop until the elapsed time is exceeded • Stopping the timer • Stopping clock supply to the CMTW
Arguments	uint32_t msec Waiting time (in milliseconds)
Return values	None

6.10 Flowcharts

6.10.1 Main Processing

Figure 6.3 shows the flowchart for Main Processing.

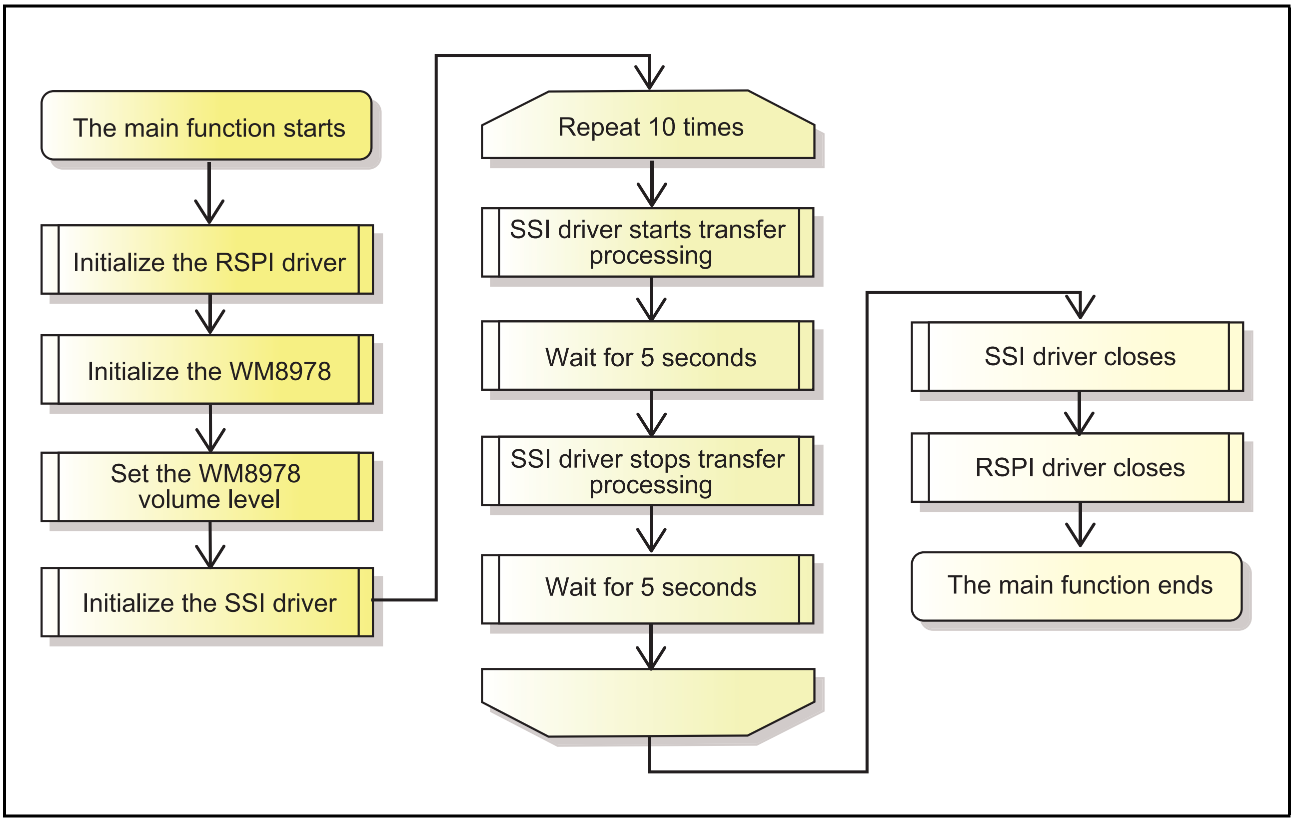


Figure 6.3 Main Processing

7. Sample Code

Download the sample code from the Renesas Electronics website.

8. Related Documents

- User's Manuals: Hardware
  RZ/T1 Group User's Manual: Hardware
  (Download the latest edition from the Renesas Electronics website.)

  RZ/T1 Evaluation Board RTK7910022C00000BR User's Manual
  (Download the latest edition from the Renesas Electronics website.)

- Technical Update and Technical News
  (Download the latest information from the Renesas Electronics website.)

- User's Manuals: Development Environment
  For the IAR Embedded Workbench® for Arm, download the user's manual from the IAR website.
  (Download the latest edition from the IAR website.)

Website and Support

Renesas Electronics website

http://www.renesas.com/

Inquiries

http://www.renesas.com/inquiry

Revision History	Application Note: SSI Sample Program

Rev.	Date	Description	
		Page	Summary
0.10	Apr. 02, 2015	—	First Edition issued
1.00	Apr. 10, 2015	—	Only the revision number was changed to be posted on a website.
1.10	Jul. 16, 2015	2. Operating Environment	
		6	Table 2.1 Operating Environment: Description added to Integrated Development Environment
		6. Software	
		14	6.2.4 Required Memory Size: Description and reference added
		14	Table 6.2: Table title and size description were partially amended
		14	Table 6.2 Memory Requirements: Description on the Note, changed
		15	Table 6.3 added
		15	Table 6.4 added
1.20	Dec. 04, 2015	2. Operating Environment	
		6	Table 2.1 Operating Environment: Integrated Development Environment, information partially amended
1.30	Apr. 05, 2017	2. Operating Environment	
		6	Table 2.1 Operating Environment: Integrated Development Environment, modified
		6. Software	
		—	6.2.4 Required Memory Size, deleted
1.40	Jun. 07, 2018	2. Operating Environment	
		6	Table 2.1 Operating Environment: The description on the integrated development environment, modified
		8. Related Documents	
		30	The name of IAR Embedded Workbench, modified

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. 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.
5. Differences between Products
   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.
   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the 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.

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.0