

RX Family

Sample programs to use SSI module for PCM data transfer

Using Firmware Integration Technology Modules

Introduction

This application note describes how to use the SSI module that is compliant with Firmware Integration Technology (FIT). It comes with seven sample programs for transferring PCM data, and describes the processing to perform PCM data transfer using a Renesas evaluation board.

The sample programs provided with this document are classified into two types. One is the Basic programs of individual use cases for using a serial sound interface. The other is applications of the Basic programs that use the serial sound interface in RX72N Envision Kit. The Application program can input PCM signals from the on-board MEMS microphone. It can also output sine waves.

There are two serial sound interface types, SSI and SSIE, and the sample programs support both of them. The sample programs are intended for the RX64M or RX72N, but can easily be used with other Renesas MCUs by employing the e² studio integrated development environment or Smart Configurator.

In this document, firstly overview of sample program and secondly how to use the samples on e² studio are shown.

Table 1. Basic programs

Transfer type	Sample programs	Target Device
PCM data transmit	rx64m_example_ssi_tx	RX64M
PCM data receive	rx64m_example_ssi_rx	RX64M
PCM data receive & transmit	rx64m_example_ssi_rx_tx	RX64M
PCM data transmit with DMAC	rx64m_example_ssi_dma_tx	RX64M
PCM data receive with DMAC	rx64m_example_ssi_dma_rx	RX64M
PCM data receive & transmit with DMAC	rx64m_example_ssi_dma_rx_tx	RX64M

Table 2. Application program

Transfer type	Sample programs	Target Device
PCM data receive & transmit with DMAC	rx72n_example_ssi_dma_rx_tx	RX72N

Target Device (SSI)

- RX64M Group

Target Device (SSIE)

- RX72N Group

Contents

1. Overview.....	3
1.1 Provided sample programs	4
1.1.1 Basic programs	4
1.1.2 Application program	6
2. Operation confirmation conditions	7
3. Software explanation	11
3.1 Software components.....	11
3.2 PCM data transfer operation using interrupts	11
3.3 PCM data transfer operation using DMAC.....	13
3.4 PCM data buffer	14
4. Migrating projects with Smart Configurator	15
4.1 Changing the device.....	16
4.2 Setting the clock	17
4.3 Setting the SSI and setting the pins to use with SSI	18
4.4 Changing pin assignments	19
4.5 Making other changes	20
5. Sample project execution	21
5.1 Hardware setup	21
5.1.1 When using Basic program	21
5.1.2 When using Application program	22
5.2 Importing a Project	23
5.2.1 Importing a Project into e ² studio	23
5.2.2 Importing a Project into CS+	24
6. Reference Documents	25
Revision History	26

1. Overview

The all sets of sample program provided with this document are made to show typical PCM data transfer operations (transmit, receive, receive & transmit) using SSI module. The Basic programs use SSI0 interrupts or DMAC for transfer operation. The Application program uses DMAC for transfer operation. All of the sample programs provided with this document transfer PCM data in the following format.

PCM data transfer format

- PCM data transfer format: I²S (SCK: 32bit/ch)
- PCM data: 16-bit stereo (Basic program), 24-bit stereo (Application program)
- Clock master mode

Test wave

The sample programs for transfer operation contain the following test wave, which is transmitted from SSI0 with the Basic programs, and from SSIE1 with the Application program.

- Fs: 48 kHz, Left channel: full scale, 3 kHz sine wave, Right channel: full scale, 1.5 kHz sine wave

For the Basic programs, the sampling frequency is set to be 48 kHz when the master clock is 12.2880 MHz.

For the Application program, the sampling frequency is set to be 48 kHz when the master clock is 24.576 MHz.

1.1 Provided sample programs

This document and the sample programs are contained in the r01an2825xx****_ssi_rx.zip file. Following is folder structure after decompress it. **** indicates the revision of this application note.

```
r01an2825xx****_ssi_rx
|
+ r01an2825ej****_rx.pdf (this document)
+ r01an2825jj****_rx.pdf (this document)
+ workspace
  |
  + rx64m_example_ssi_tx (Basic program)
  + rx64m_example_ssi_rx (Basic program)
  + rx64m_example_ssi_rx_tx (Basic program)
  + rx64m_example_ssi_dma_tx (Basic program)
  + rx64m_example_ssi_dma_rx (Basic program)
  + rx64m_example_ssi_dma_rx_tx (Basic program)
  + rx72n_example_ssi_dma_rx_tx (Application program)
```

The above seven sample programs are projects of e² studio.

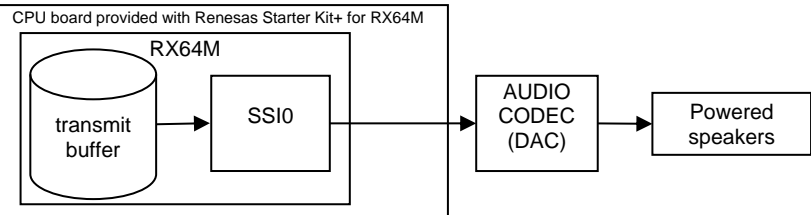
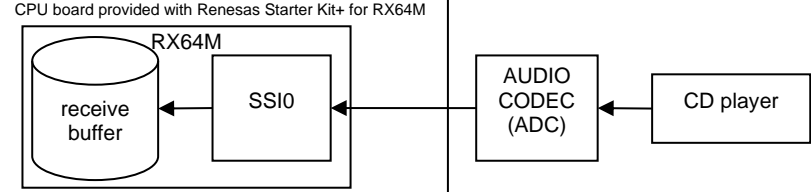
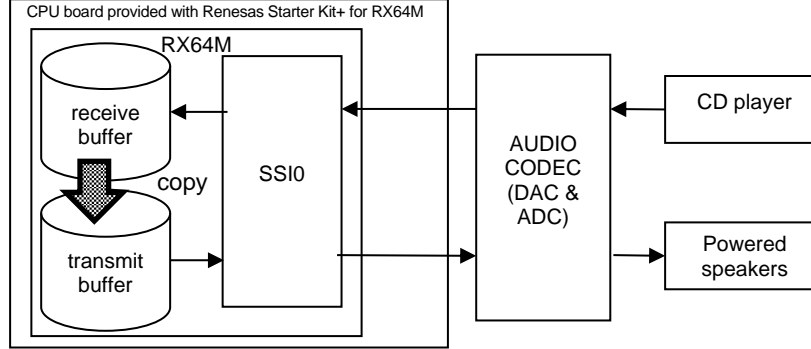
1.1.1 Basic programs

PCM data transfer operation types and the corresponding sample program are shown in Table 1.1. Choose one among them, import and execute it on e² studio. Each sample programs' operation and audio signal flow on assuming target system by operation types are shown in Table 1.2.

Table 1.1 Operation type and corresponding C project file (Basic programs)

Transfer type	Sample programs	Target Device
PCM data transmit	rx64m_example_ssi_tx	RX64M
PCM data receive	rx64m_example_ssi_rx	RX64M
PCM data receive & transmit	rx64m_example_ssi_rx_tx	RX64M
PCM data transmit with DMAC	rx64m_example_ssi_dma_tx	RX64M
PCM data receive with DMAC	rx64m_example_ssi_dma_rx	RX64M
PCM data receive & transmit with DMAC	rx64m_example_ssi_dma_rx_tx	RX64M

Table 1.2 Signal flow by operation types (Basic program)

Transfer type	Operation and audio signal flow with assuming external devices
PCM data transmit, PCM data transmit with DMAC	<p>Transmits sine waves stored on transmit buffer to CODEC.</p>  <p>CPU board provided with Renesas Starter Kit+ for RX64M</p> <p>RX64M</p> <p>transmit buffer → SSI0 → AUDIO CODEC (DAC) → Powered speakers</p>
PCM data receive, PCM data receive with DMAC	<p>Receives signals from CODEC and stores them on receive buffer.</p>  <p>CPU board provided with Renesas Starter Kit+ for RX64M</p> <p>RX64M</p> <p>CD player → AUDIO CODEC (ADC) → SSI0 → receive buffer</p>
PCM data receive & transmit, PCM data receive & transmit with DMAC	<p>Receives signals from CODEC, stores them on receive buffer and copies them on transmit buffer and transmit. (loopback operation)</p>  <p>CPU board provided with Renesas Starter Kit+ for RX64M</p> <p>RX64M</p> <p>CD player → AUDIO CODEC (DAC & ADC) → SSI0</p> <p>SSI0 → receive buffer</p> <p>receive buffer → copy → transmit buffer → SSI0 → AUDIO CODEC (DAC & ADC) → Powered speakers</p>

1.1.2 Application program

The Application program is a project with converted rx64m_example_ssi_dma_rx_tx for RX72N Envision Kit using Smart Configurator. For the conversion method, refer to “4.Migrating projects with Smart Configurator”.

PCM data transfer operation types and the corresponding sample program are shown in Table 1.3. The operation and audio signal flow of the sample program is shown in Figure 1.1. The on-board MEMS microphone or sine waves can be selected as the signal source of SSIE1 as shown in Figure 1.1 and Table 1.4 by using a macro definition in the source code.

Table 1.3 Operation type and corresponding C project file (Application program)

Transfer type	Sample programs	Target Device
PCM data receive & transmit with DMAC	rx72n_example_ssi_dma_rx_tx	RX72N

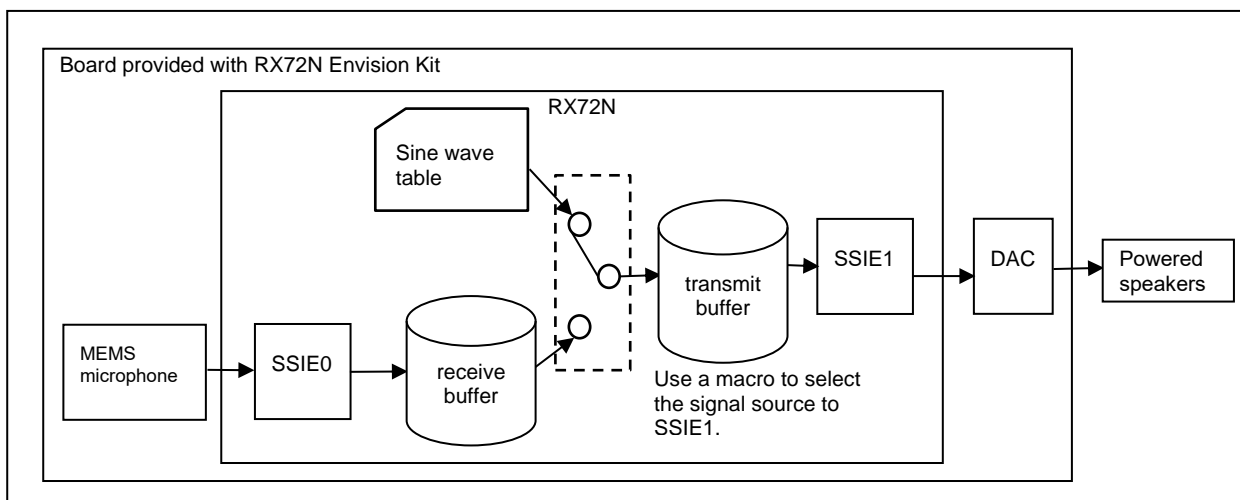


Figure 1.1 Audio signal flow (Application program)

Table 1.4 Selecting signal source of SSIE1 (Application program)

Macro definitions	Signal source of SSIE1
#define OUTPUT_SIGNAL_FROM SINE_WAVE	Sine wave data (default)
#define OUTPUT_SIGNAL_FROM MEMS_MIC	Signal from on-board MEMS microphone

2. Operation confirmation conditions

The operation confirmation conditions are shown in Table 2.1 and Table 2.2. The compositions of the target systems are shown in Figure 2.1 and Figure 2.2. Figure 2.1 shows the external element connections with the CPU board provided with Renesas Starter Kit+ for RX64M. Refer to the document "RX64M Group Renesas Starter Kit+ User's Manual For e² studio" and compose the target system.

Figure 2.2 shows the connections with the RX72N, D/A converter, and other audio system components on the board provided with RX72N Envision Kit. A clock generator IC and other components for PCM data transfer are mounted on the board provided with RX72N Envision Kit. PCM data can thus be transferred from RX72N without adding any components.

Table 2.1 Operation confirmation conditions (Basic programs)

Item	Contents
Microcontroller used	R5F564MLCDFC (RX64M Group)
Operating frequency	<ul style="list-style-type: none"> • Main clock: 24 MHz • PLL circuit: 240 MHz (main clock x 10) • System clock (ICLK): 120 MHz (PLL circuit x 1/2) • Peripheral module clock A (PCLKB): 60 MHz (PLL circuit x 1/4) • Peripheral module clock B (PCLKB): 60 MHz (PLL circuit x 1/4) • Peripheral module clock C (PCLKB): 60 MHz (PLL circuit x 1/4) • Peripheral module clock D (PCLKD): 60 MHz (PLL circuit x 1/4) • External bus clock (BCLK): 60 MHz (PLL circuit x 1/4) • Flash interface clock (FCLK): 60 MHz (PLL circuit x 1/4)
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics e ² studio Version 2020-04
C Compiler ^(Note 1)	Renesas Electronics C/C++ Compiler for RX Family V.3.02.00
	Compile option The default settings of the integrated development environment
iodefine.h version	Version 1.0A
Endianness	Little endian
Operation mode	Single chip mode
Processor mode	Supervisor mode
SSI	Use SSI0
Sample code version	Version 2.00
Board used	CPU board provided with Renesas Starter Kit+ for RX64M (product No.: R0K50564MSxxBE)
Measuring instrument, etc.	Use Audio Precision SYS2722 and PSIA-2722 as a clock generator and AUDIO CODEC.

Note: 1. If the same version of tool chain (C compiler) as that specified in the source project does not exist in the import destination, the tool chain will be in the unselected state and an error will occur. Check the selection state of the tool chain on the project settings screen. For the setting procedure, refer to FAQ 3000404.

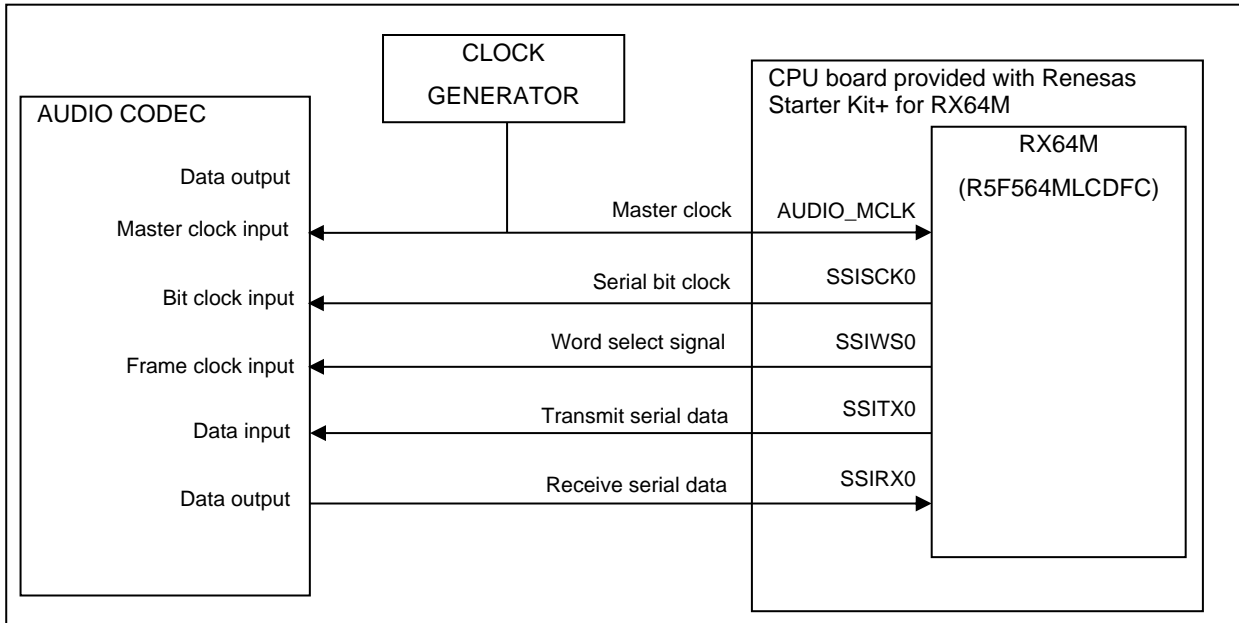


Figure 2.1 Target system composition (Basic programs)

Table 2.2 Operation confirmation conditions (Application program)

Item	Contents
Microcontroller used	R5F572NDHDFB (RX72N Group)
Operating frequency	<ul style="list-style-type: none"> • Main clock: 16 MHz • PLL circuit: 240 MHz (main clock x 15) • System clock (ICLK): 240 MHz (PLL circuit x 1/1) • Peripheral module clock A (PCLKB): 60 MHz (PLL circuit x 1/4) • Peripheral module clock B (PCLKB): 60 MHz (PLL circuit x 1/4) • Peripheral module clock C (PCLKB): 60 MHz (PLL circuit x 1/4) • Peripheral module clock D (PCLKD): 60 MHz (PLL circuit x 1/4) • External bus clock (BCLK): 60 MHz (PLL circuit x 1/4) • Flash interface clock (FCLK): 60 MHz (PLL circuit x 1/4)
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics e ² studio Version 2020-04
C Compiler ^(Note 1)	Renesas Electronics C/C++ Compiler for RX Family V.3.02.00
	Compile option The default settings of the integrated development environment
iodefine.h version	Version 1.00C
Endianness	Little endian
Operation mode	Single chip mode
Processor mode	Supervisor mode
SSI	Use SSIE0 and SSIE1.
Sample code version	Version 2.00
Board used	Board provided with RX72N Envision Kit (product No.: RTK5RX72N0C00000BJ)
Measuring instrument, etc.	Clock generator IC (5X35023) D/A converter (D2-41051-QR) MEMS microphone (ICS-43434)

Note: 1. If the same version of tool chain (C compiler) as that specified in the source project does not exist in the import destination, the tool chain will be in the unselected state and an error will occur. Check the selection state of the tool chain on the project settings screen. For the setting procedure, refer to FAQ 3000404.

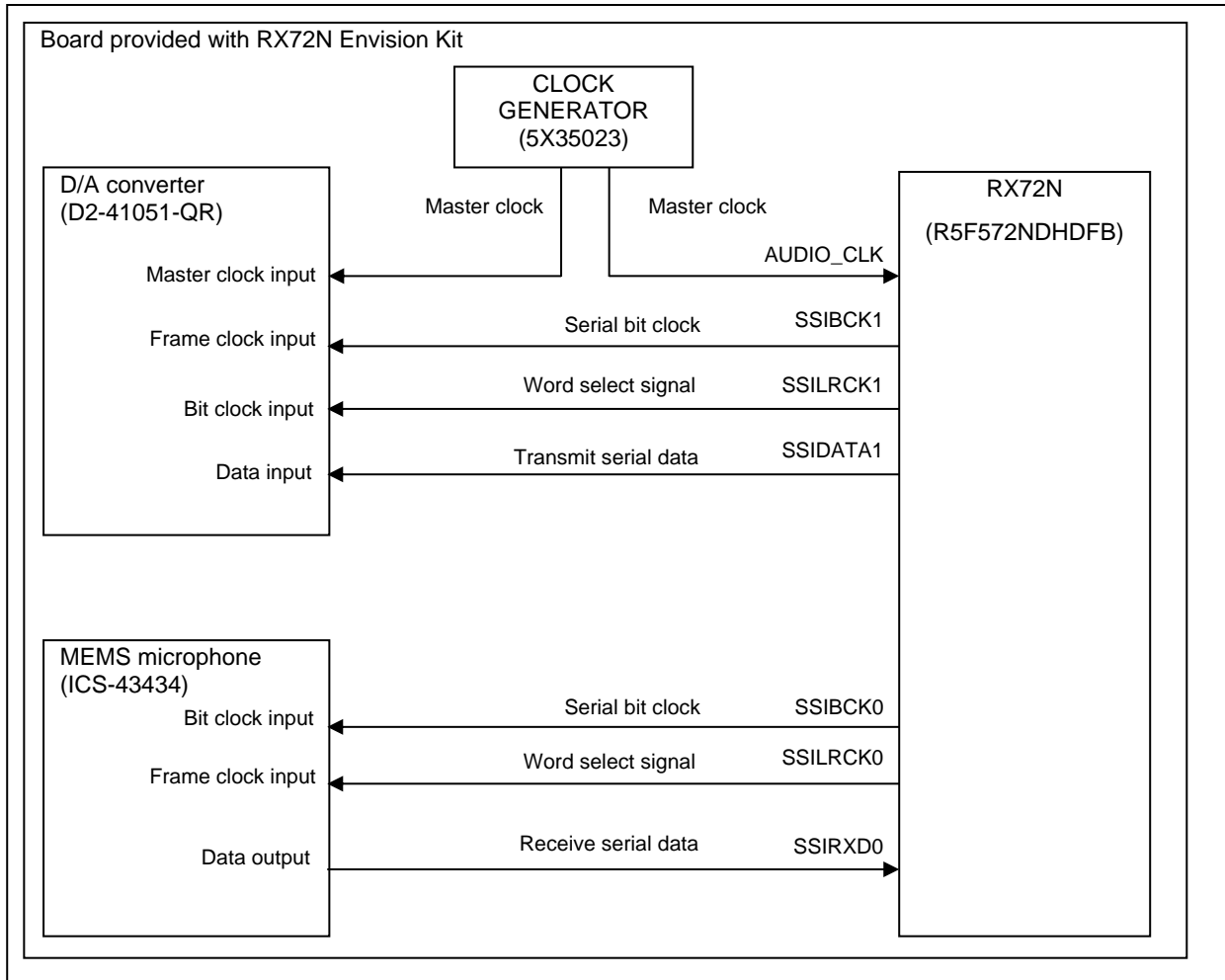


Figure 2.2 Target system composition (Application program)

3. Software explanation

3.1 Software components

Sample programs consist of several software components as shown in Table 3.1 and Table 3.2. System configuration and PCM data transfer operation using SSI module APIs are written on main(). So look sample program file including main() to understand PCM data transfer operation using FIT modules. For detailed information about FIT modules refer to corresponding application notes. (See 6. Reference Documents)

This section describes software operation in the Basic programs.

Table 3.1 Software components (Basic programs)

Transfer type	Sample program including main()	FIT modules
PCM data transmit	r_main_ssi_tx.c (Rev.2.00)	BSP FIT Module (Rev.5.52)
PCM data receive	r_main_ssi_rx.c (Rev.2.00)	SSI Module (Rev.2.02)
PCM data receive & transmit	r_main_ssi_rx_tx.c (Rev.2.00)	MPC Module (Rev.3.40) GPIO Module (Rev.3.40)
PCM data transmit with DMAC	r_main_ssi_dma_tx.c (Rev.2.00)	BSP FIT Module (Rev.5.52)
PCM data receive with DMAC	r_main_ssi_dma_rx.c (Rev.2.00)	SSI Module (Rev.2.02)
PCM data receive & transmit with DMAC	r_main_ssi_dma_rx_tx.c (Rev.2.00)	DMAC Module (Rev.2.30) MPC Module (Rev.3.40) GPIO Module (Rev.3.40)

“PCM data transmit” and “PCM data transmit with DMAC” include PCM data table file “r_test_wave.h”.

Table 3.2 Software components (Application program)

Transfer type	Sample program including main()	FIT modules
PCM data receive & transmit with DMAC	r_main_ssi_dma_rx_tx.c (Rev.2.00)	BSP FIT Module (Rev.5.52) SSI Module (Rev.2.02) DMAC Module (Rev.2.30) MPC Module (Rev.3.40) GPIO Module (Rev.3.40)

Includes PCM data table file “r_test_wave.h”.

3.2 PCM data transfer operation using interrupts

Figure 3.1 shows main elements for PCM data transfer operation using SSI0 interrupts. Elements to use are different from operation types. In case of PCM data transmit & receive operation all elements are used. But in case of PCM data transmit operation just elements for transmit are used. It is same way as in PCM data receive operation case.

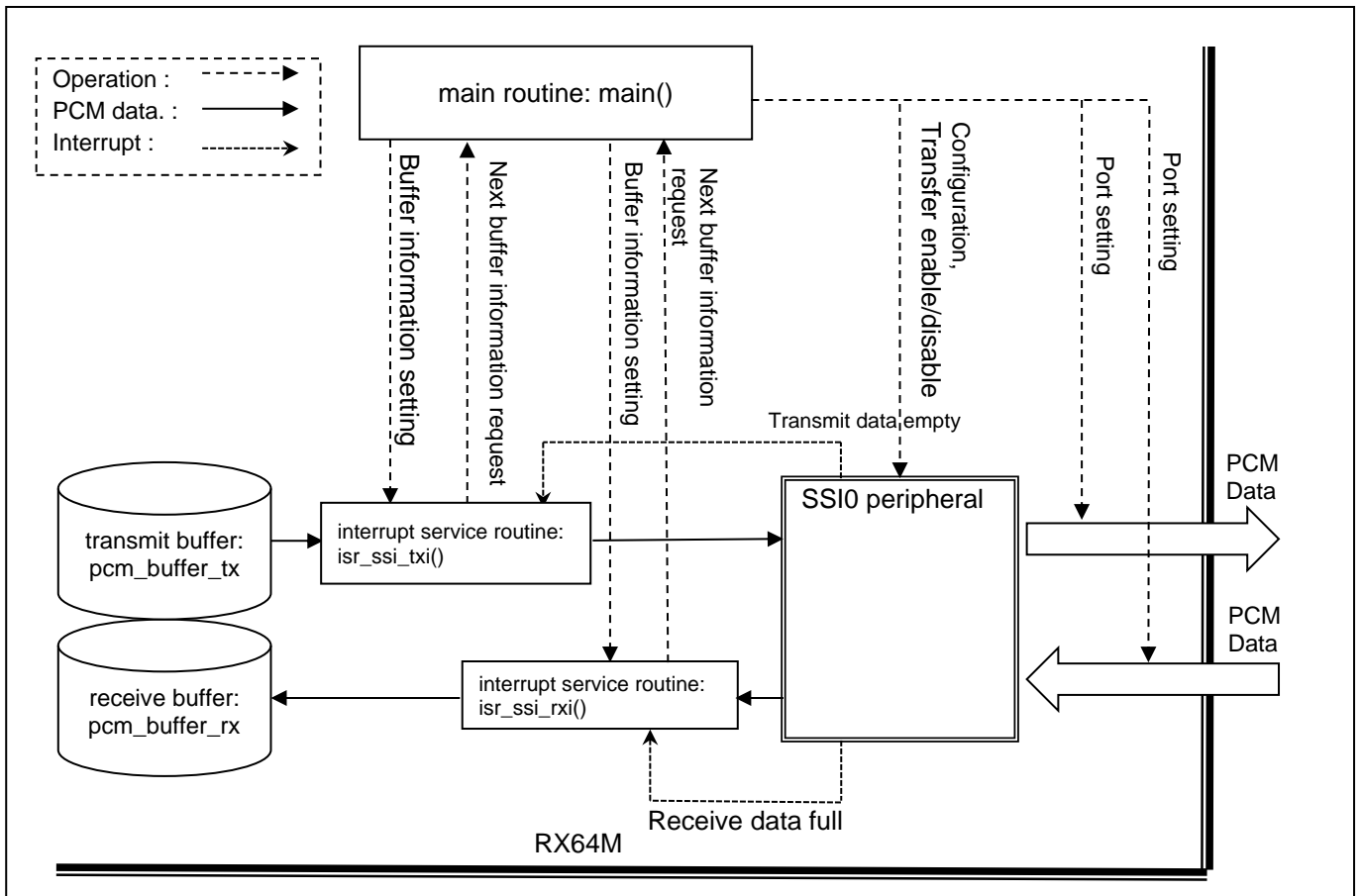


Figure 3.1 PCM data transfer operation (using interrupts)

Refer to Figure 3.1, operating steps for PCM data transfer using SSI0 interrupts are described as follows.

- 1) Peripheral setting
main() configures GPIO, MPC, and SSI peripherals using APIs of GPIO, MPC and SSI modules.
- 2) Transmit and receive buffer information setting
main() sets transmit and receive buffer information. The information consists of the number of samples and transfer start address and it shows a unit of transfer. So the information is used for write and read operation described below.
- 3) Transfer start
main() enables SSI0 to start PCM data transfer.
- 4) Operation during PCM data transfer
After transfer is started, Transmit data empty and Receive data full interrupts occur from SSI0. When Transmit data empty interrupt occurs, corresponding interrupt service routine: isr_ssi_tx() calls R_SSI_Write() to write PCM data in transmit buffer to SSI0's Transmit FIFO. This is periodic operation caused by the interrupt. When all PCM data set at step 2) are transferred, isr_ssi_tx() requests main() to set the next buffer information. Repeating isr_ssi_tx() call and buffer information update, PCM data goes out through SSI0 continuously without break.
When Receive data full interrupt occurs, corresponding interrupt service routine: isr_ssi_rx() calls R_SSI_Read() to read PCM data in SSI0's Receive FIFO and store it on receive buffer. This is periodic operation caused by the interrupt. When all PCM data set at step 2) are transferred, isr_ssi_rx() requests main() to set the next buffer information. Repeating isr_ssi_rx() call and buffer information update, PCM data comes in through SSI0 continuously without break.

Table 3.3 Required MCU resource (using interrupt)

MCU Resources	Transfer type (using interrupt)		
	transmit	receive	receive and transmit
SSI	SSI0	SSI0	SSI0
SSI0 pins	SSISCK0, SSIWS0, SSITXD0, AUDIO_MCLK	SSISCK0, SSIWS0, SSIRXD0, AUDIO_MCLK	SSISCK0, SSIWS0, SSITXD0, SSIRXD0, AUDIO_MCLK
interrupts	transmit data empty	receive data full	transmit data empty, receive data full

3.3 PCM data transfer operation using DMAC

Figure 3.2 shows main elements for PCM data transfer operation using DMAC. Elements to use are different from operation types. In case of PCM data transmit & receive operation all elements are used. But in case of PCM data transmit operation just elements for transmit are used. It is same way as in PCM data receive operation case.

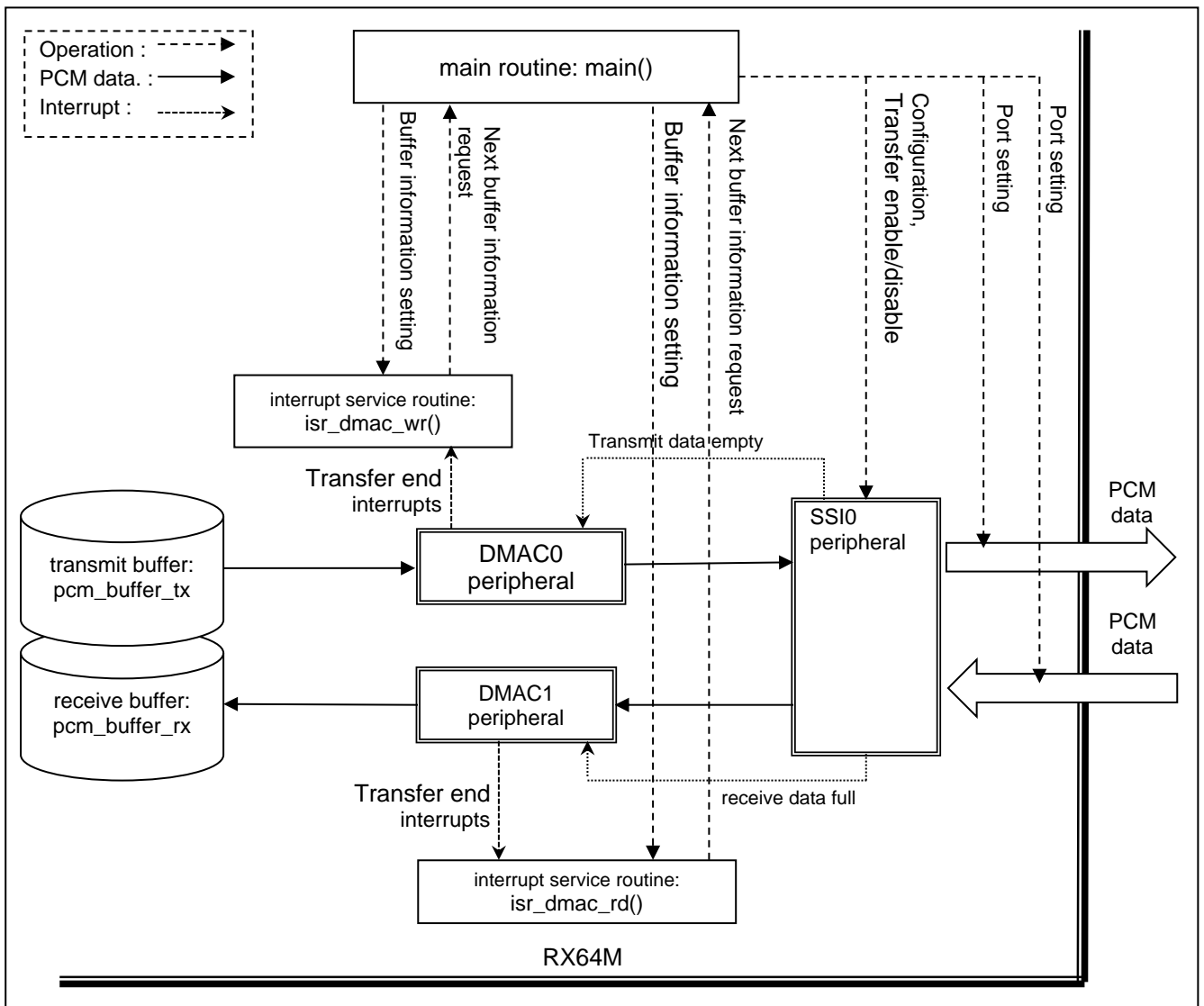


Figure 3.2 PCM data transfer operation using DMAC

Refer to Figure 3.2, operating steps for PCM data transfer using DMAC are described as follows.

- 1) Peripheral setting
main() configures GPIO, MPC, DMAC and SSI peripherals using APIs of GPIO, MPC, DMAC and SSI modules.
- 2) Transmit and receive buffer information setting
main() sets transmit and receive buffer information. The information consists of the number of samples and transfer start address and it shows a unit of transfer. So the information is used for DMAC operation described below.
- 3) Transfer start
main() enables SSI0 to start PCM data transfer after enabling DMA transfer.
- 4) Operation during PCM data transfer
DMAC0 transfers PCM data from transmit buffer to SSI0's Transmit FIFO, DMAC1 transfers PCM data from SSI0's Receive FIFO to receive buffer.
When DMAC0 transfer end interrupt occurs, interrupt service routine: isr_dmac_wr() is called. It means that the specified number of samples at step 2) transfer completion. isr_dmac_wr() requests main() to set the next buffer information. Repeating isr_dmac_wr() call and buffer information update, PCM data goes out through SSI0 continuously without break.
When DMAC1 transfer end interrupt occurs, interrupt service routine: isr_dmac_rd() is called. It means that the specified number of samples at step 2) transfer completion. isr_dmac_rd() requests main() to set the next buffer information. Repeating isr_dmac_rd() call and buffer information update, PCM data comes in through SSI0 continuously without break.

Table 3.4 Required MCU resource (using DMAC)

MCU Resources	Transfer type (using DMAC)		
	transmit	receive	receive and transmit
SSI	SSI0	SSI0	SSI0
SSI0 pins	SSISCK0, SSIWS0, SSITXD0, AUDIO_MCLK	SSISCK0, SSIWS0, SSIRXD0, AUDIO_MCLK	SSISCK0, SSIWS0, SSITXD0, SSIRXD0, AUDIO_MCLK
interrupts	DMAC0 transfer end	DMAC1 transfer end	DMAC0 transfer end, DMAC1 transfer end
DMAC	DMAC0	DMAC1	DMAC0, DMAC1

3.4 PCM data buffer

Transmit and Receive buffer shown in Figure 3.1 and Figure 3.2 are containers for PCM data to transmit or receive PCM data. The buffers are int16_t type of two-dimensional array. The row shows the number of blocks and the column shows the number of samples per block.

- The number of blocks: See Table 3.5
- The number of samples per block: 1024 x 2.
- PCM data storing order: Left channel, Right channel, Left channel, Right channel,

Table 3.5 Number of PCM data buffer blocks

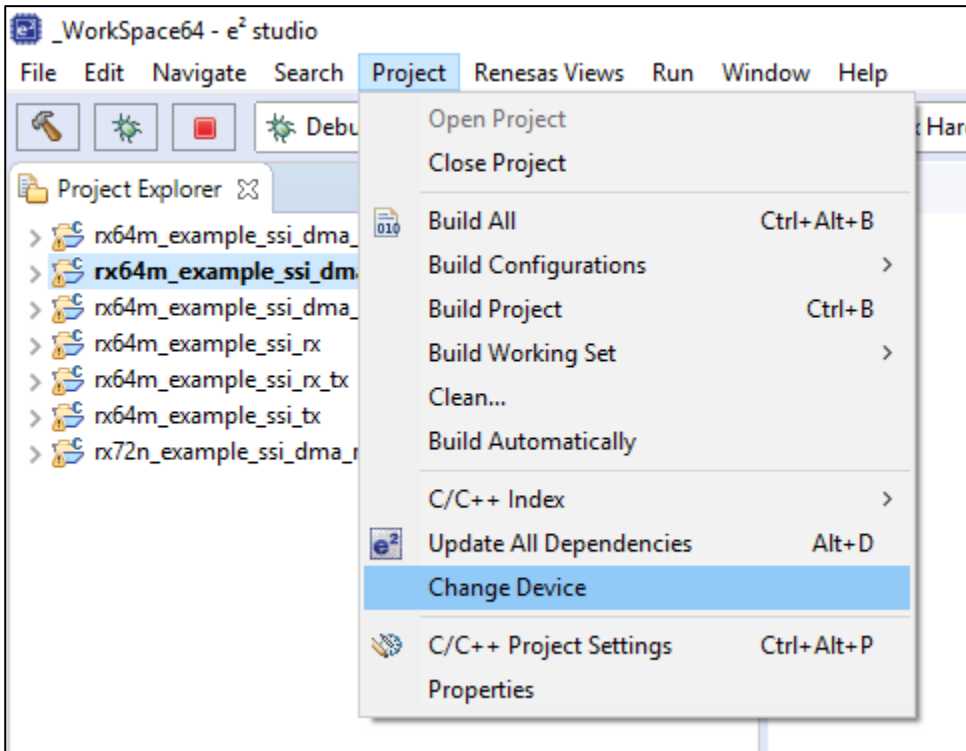
	Transfer type		
	transmit	receive	receive and transmit
blocks	2	2	3

4. Migrating projects with Smart Configurator

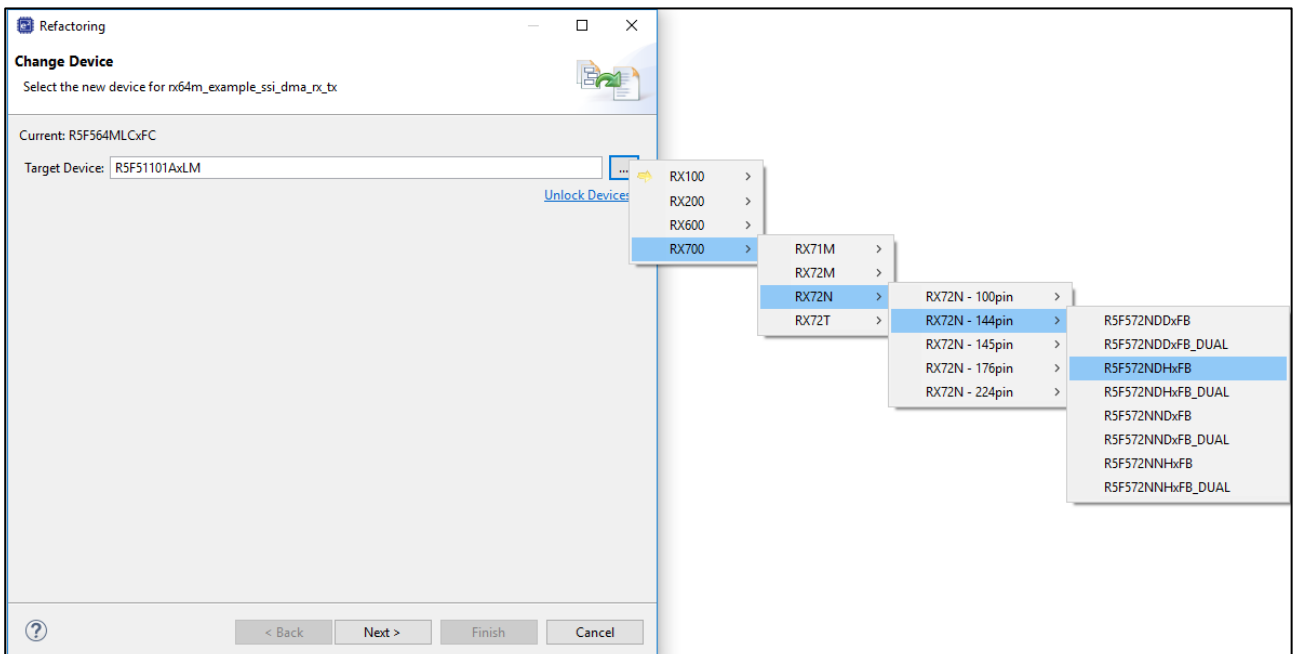
If you use Smart Configurator, you can migrate your project settings between different devices. This section describes the procedure to migrate a project of rx64m_example_ssi_dma_rx_tx to a project for RX72N Envision Kit. Also refer to rx72n_example_ssi_dma_rx_tx, which is provided with this document as an example of a project that has been migrated.

4.1 Changing the device

Select the project and then select [Change Device] from the [Project] menu.



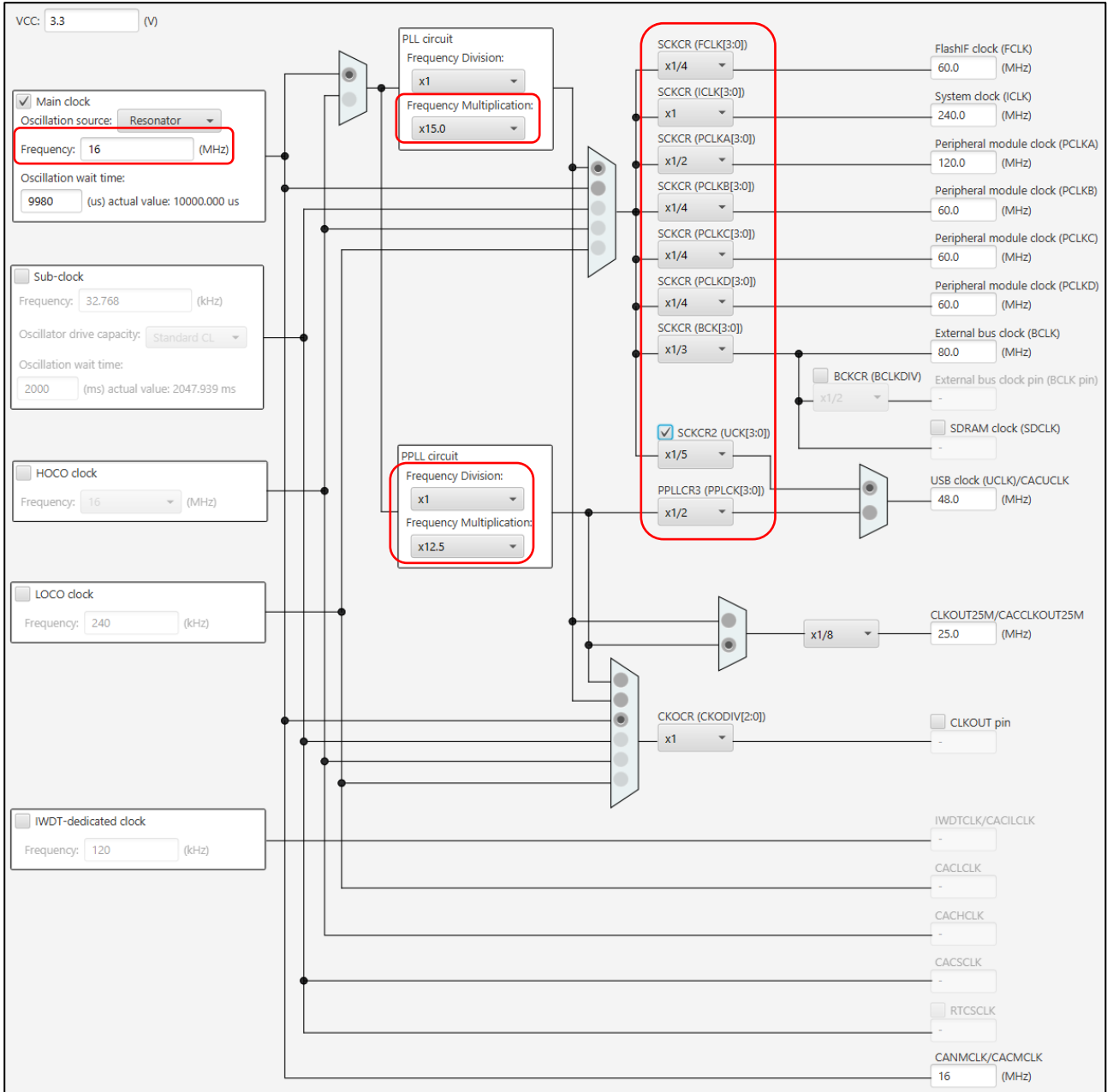
Click the [Select Device] button, select “R5F572NDHxFB”, and click [Next].



4.2 Setting the clock

Set the clock to match RX72N Envision Kit.

Open rx64m_example_ssi_dma_rx_tx.scfg from Project Explorer, display Smart Configurator, select the Clock tab, and then configure the following settings. (The places to change are indicated by red frames.)



4.3 Setting the SSI and setting the pins to use with SSI

Set the SSI and set the pins to use with SSI.

Open rx64m_example_ssi_dma_rx_tx.scfg from Project Explorer, display Smart Configurator, select the Component tab, select r_ssi_api_rx, and then set the SSI and set the pins to use with SSI with the following procedure. The items and settings are described on the settings screen.

Table 4.1 Setting the SSI and setting the pins to use with SSI

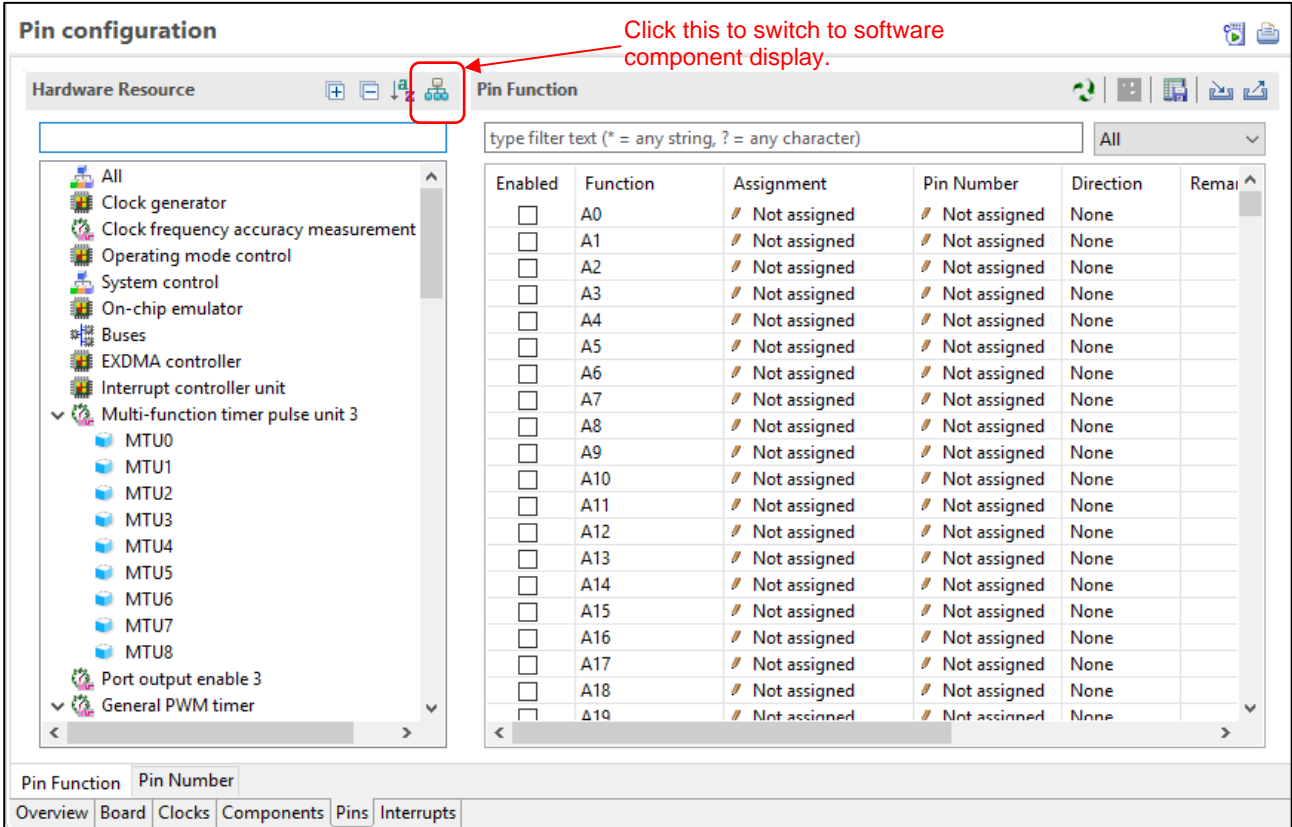
Property	Setting
Use Standard configuration	Standard
SSI or SSIE	SSIE
Ch0 Transmit / Receive mode	SSIE to Receive data
Ch1 Transmit / Receive mode	SSIE to Transmit data
Ch0 Audio serial interface format	I2S
Ch1 Audio serial interface format	I2S
Ch0 PCM data width	24 bits
Ch1 PCM data width	24 bits
Ch0 Bit Clock	64 fs
Ch1 Bit Clock	64 fs
Master Clock	512 u
Ch0 Clock Supply Mode	Master mode
Ch1 Clock Supply Mode	Master mode
Ch0 Tx Empty trigger	8
Ch1 Tx Empty trigger	8
Ch0 Rx Full trigger	8
Ch1 Rx Full trigger	8
Ch0 Byte Swap	Byte swapping is disabled
Ch1 Byte Swap	Byte swapping is disabled
SSIE0	Select check box
SSIBCK0 pins	Use
SSILRCK0 pins	Use
SSITXD0 pins	Do not use
SSIRXD0 pins	Use
AUDIO_CLK pin	Use
SSIE1	Select check box
SSIBCK1 pins	Use
SSILRCK1 pins	Use
SSIDATA1 pins	Use
AUDIO_CLK pin	Use

4.4 Changing pin assignments

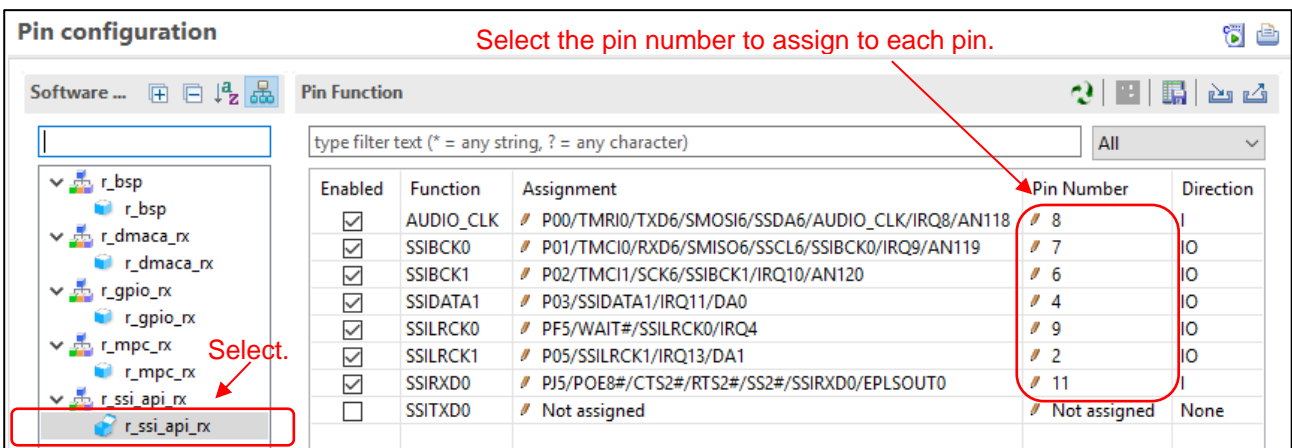
Change the assignments of the pins to use with SSI.

Open rx64m_example_ssi_dma_rx_tx.scfg from Project Explorer, display Smart Configurator, select the Pin tab, and then configure the following settings.

Select the Pin tab and then switch to software component display.



Select r_ssi_api_rx from among the software components and select the pin number to assign to each pin.



4.5 Making other changes

Data is transmitted and received with SSI0 in rx64m_example_ssi_dma_rx_tx, but change "r_main_ssi_dma_rx_tx.c" as shown below so that data is received with SSIE0 and transmitted with SSIE1 since you are migrating the project to a project for RX72N Envision Kit.

Table 4.2 Changes of r_main_ssi_dma_rx_tx.c

Part to change	Change details
PCM data buffer	Change the type of buffer to int32_t.
main()	Change this so that uses SSIE0 and SSIE1. Add processing to select whether to use the sine wave data or use an input signal from the MEMS microphone as the signal source of SSIE1.
isr_ssi_status_error()	Add the error judgment processing of SSIE1.
init_port_ssi0()	Change the pin settings of SSIE0 and SSIE1 to match RX72N Envision Kit.
init_dmac_wr() isr_dmac_wr() init_dmac_rd() isr_dmac_rd()	Change the number of transfers of DMA.

5. Sample project execution

5.1 Hardware setup

5.1.1 When using Basic program

Figure 5.1 shows an example of an operating environment for using a Basic program. Refer to Figure 2.1 and Figure 5.1, setup the environment. Use Renesas Starter Kit+ for RX64M with the Basic program.

Use Renesas Starter Kit+ for RX64M accompanying AC adaptor and setup the developing environment keeping connecting order shown as follows.

1. Connect the powered speakers to the AUDIO CODEC. If the powered speakers have a volume, connect them while the volume is set to the minimum level and set the volume to an appropriate value after executing the program.
2. Connect the E2 Lite emulator to the CPU board provided with the Renesas Starter Kit+ for RX64M of the target system with the user interface cable.
3. Connect the AC adaptor to the CPU board provided with Renesas Starter Kit+ for RX64M, and supply power.
4. Connect the E2 Lite emulator to the host PC with a USB cable.

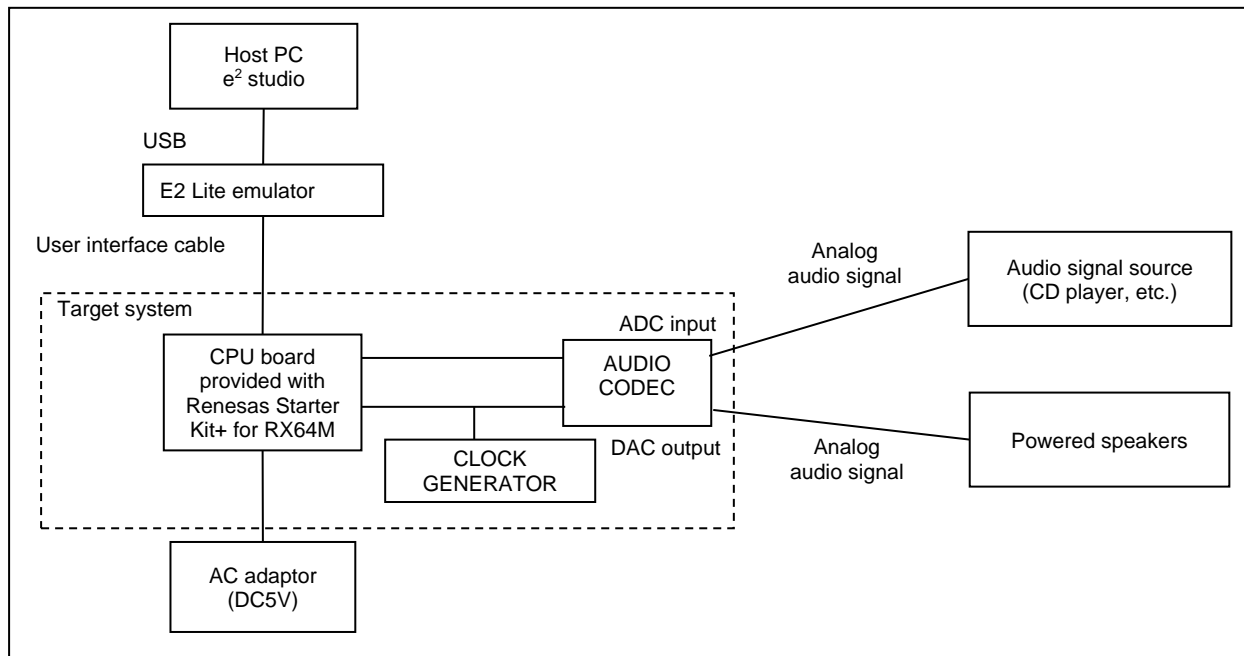


Figure 5.1 Operating environment example (Basic program)

5.1.2 When using Application program

Figure 5.2 shows an example of an operating environment for the Application program. Use RX72N Envision Kit with the Application program. Refer to Figure 2.2 and Figure 5.2, setup the environment.

Set up the hardware as described in the following procedure.

1. Set the volume of the powered speakers to the minimum level and connect the powered speakers to the audio output jack of the board provided with RX72N Envision Kit. If the powered speakers have a volume, connect them while the volume is set to the minimum level and set the volume to an appropriate value after executing the program.
2. Connect the host PC to the board provided with RX72N Envision Kit of the target system with a USB cable. The power of the board is supplied from the USB connector.
3. Whether the input to the D/A converter is a sine wave or an input signal from the on-board MEMS microphone can be selected by using a macro in the sample program. The default is a sine wave.

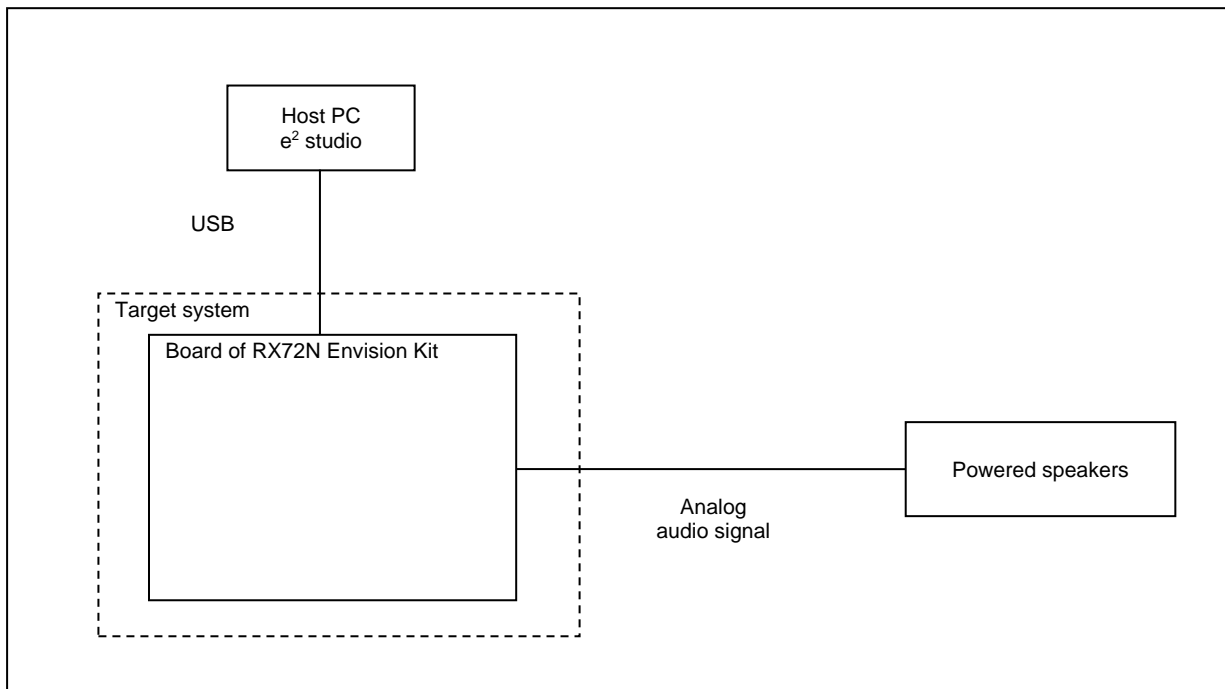


Figure 5.2 Operating environment example (Application program)

5.2 Importing a Project

The sample program is provided in the project format of e² studio. This chapter shows how to import the project to e² studio and CS+. After importing is complete, check the build and debug settings.

5.2.1 Importing a Project into e² studio

Follow the steps below to import your project into e² studio. Pictures may be different depending on the version of e² studio to be used.

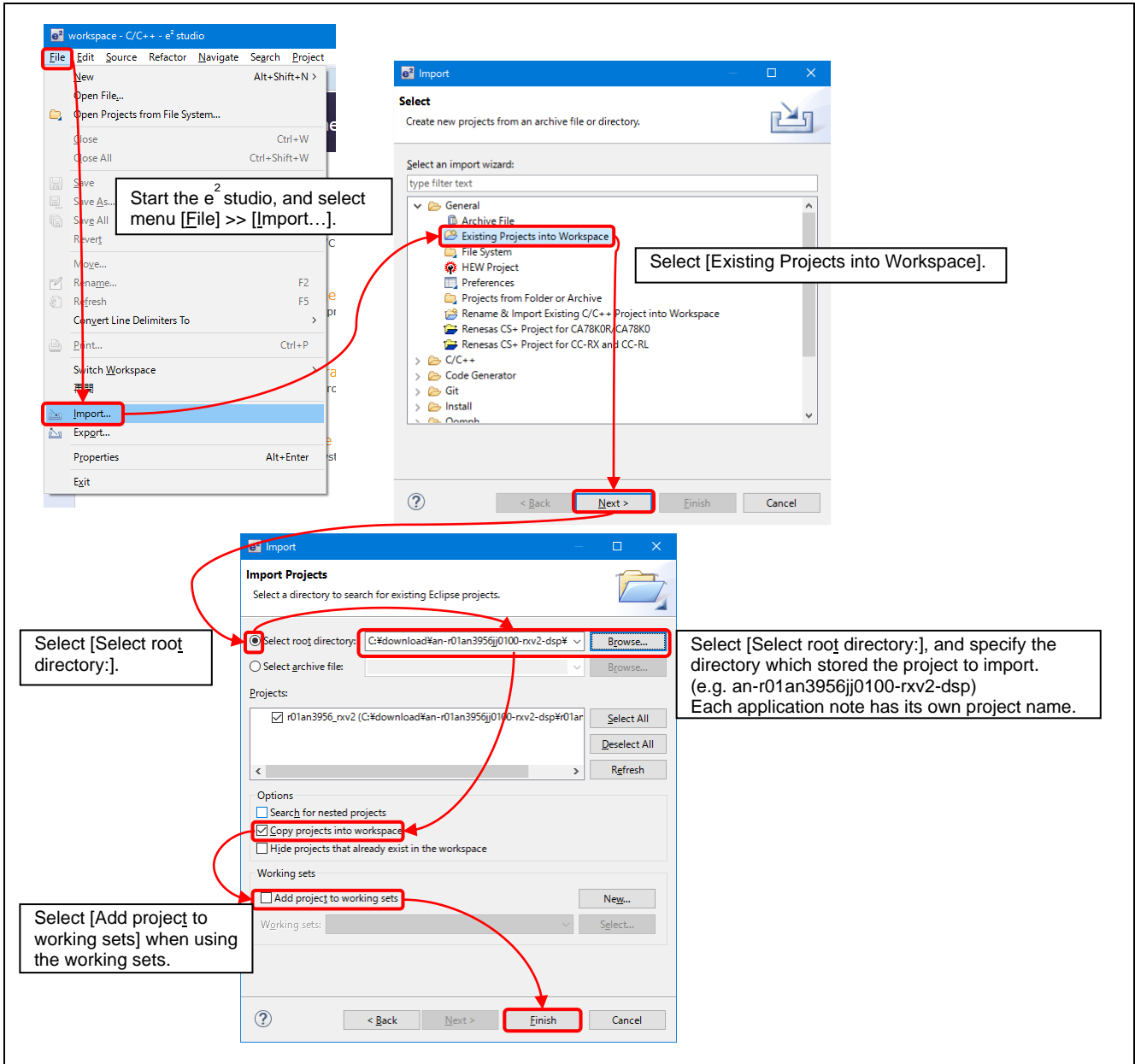


Figure 5.3 Importing a Project into e² studio

5.2.2 Importing a Project into CS+

Follow the steps below to import your project into CS+. Pictures may be different depending on the version of CS+ to be used.

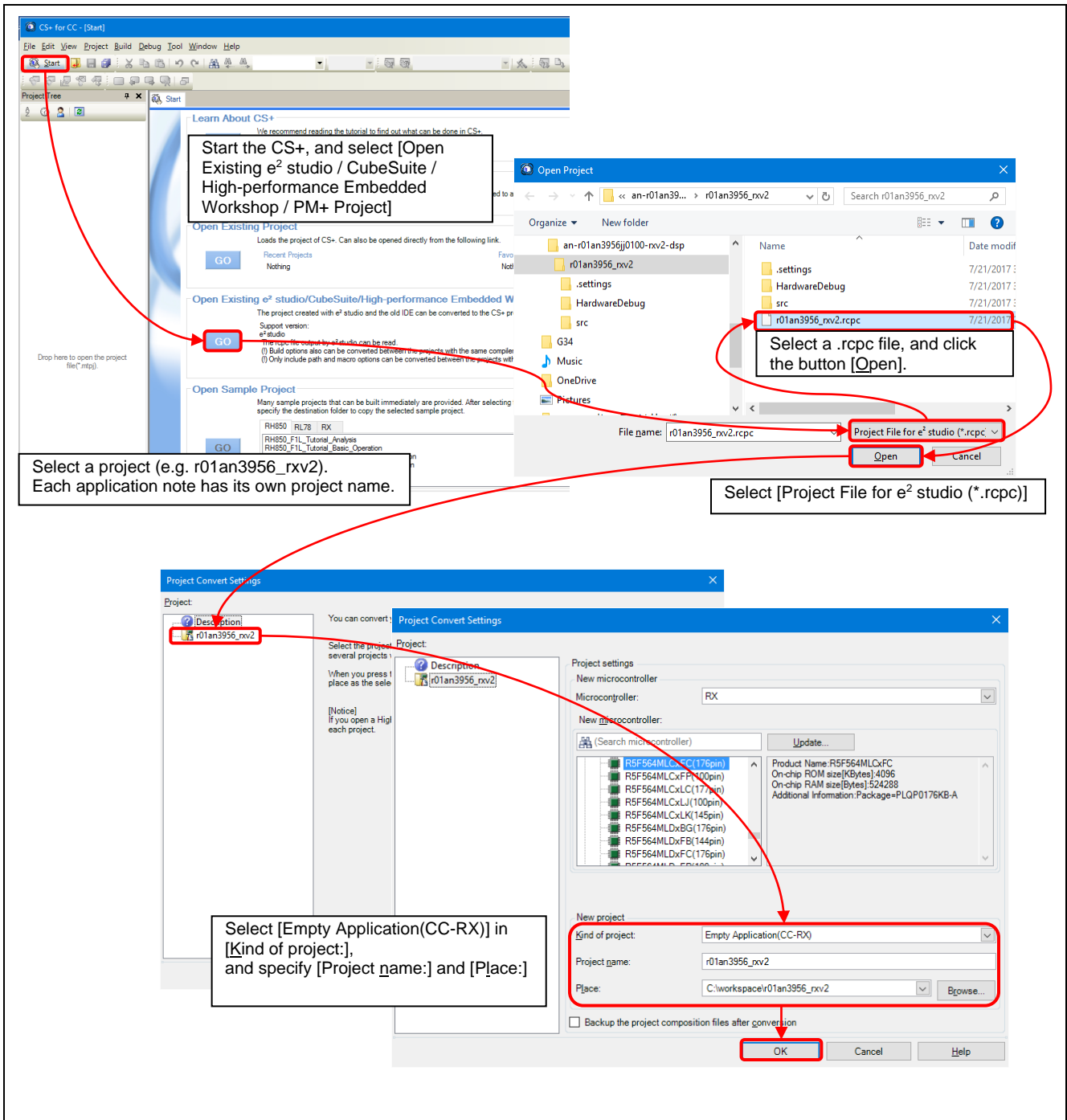


Figure 5.4 Importing a Project into CS+

6. Reference Documents

Firmware Integration Technology application note

RX Family Board Support Package Module Using Firmware Integration Technology (R01AN1685EU)

RX Family SSI Module using Firmware Integration Technology (R01AN2150EJ)

RX Family DMA Controller DMACA Control Module Using Firmware Integration Technology (R01AN2063EJ)

RX Family General Purpose Input/Output Driver Module Using Firmware Integration Technology (R01AN1721EU)

RX Family Multi-Function Pin Controller Module Using Firmware Integration Technology (R01AN1724EU)

The latest version can be downloaded from the Renesas Electronics website.

Renesas Starter Kit+ for RX64M User's Manual

RX64M Group Renesas Starter Kit+ User's Manual (CubeSuite+) (20UT2590)

RX64M Group Renesas Starter Kit+ User's Manual For e² studio (R20UT2593)

The latest version can be downloaded from the Renesas Electronics website.

RX72N Envision Kit User's Manual

RX72N Group Envision Kit User's Manual (R20UT4788)

The latest version can be downloaded from the Renesas Electronics website.

User's Manual: Hardware

RX64M Group User's Manual: Hardware Rev.1.00 (R01UH0377EJ)

RX72N Group User's Manual: Hardware Rev.1.00 (R01UH0824EJ)

The latest version can be downloaded from the Renesas Electronics website.

Technical Update/Technical News

The latest information can be downloaded from the Renesas Electronics website.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun.05.2015	-	First edition issued
2.00	Jun.30.2020	-	Following target device added RX72N

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.

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 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.
 5. 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.
6. 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.
 7. 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.
 8. 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.
 9. 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.
 10. 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.
 11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
 12. 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.4.0-1 November 2017)

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/.