APPLICATION NOTE

R01AN2839EJ0100
Rev.1.00
June 16, 2015

RX Family

Sample programs to use SRC FIT module for sampling rate conversion Using Firmware Integration Technology Modules

Introduction

This is document provided with two sets of sample program to show sampling rate conversion using SRC FIT module with Sampling Rate Converter (SRC) peripheral. Those are provided as e2studio C projects consisted of all needed software components. Those project shows how sampling rate conversion operation practically works on Renesas evaluation board.

In this document, firstly overview of sample program and secondly how to use the samples on e2studio are shown.

Target Device

- RX64M Group

Contents

1. Overview

The two sets of sample project provided with this document are made to show typical sampling rate conversion operations using SRC FIT module. The both projects change sampling rate from 8kHz to 44.1kHz. The difference between them is PCM data transfer from/to SRC peripheral. The one sample is using programmed input /output (PIO), the other one is using DMAC.

PCM data format

The input and output PCM data format is shown as follows.

- Input PCM data : 16 bit stereo, sampling frequency: 8kHz
- Output PCM data : 16 bit stereo, sampling frequency: 44.1kHz

Test wave

The test wave is contained in the both projects.

- Fs: 8kHz, Left channel: full scale, 500Hz sine wave, Right channel: full scale, 250Hz sine wave

1.1 Provided projects

This document and two sets of sample project are contained in the r01an2839ej0100_src_rx.zip file.
Following is folder structure after decompress it.

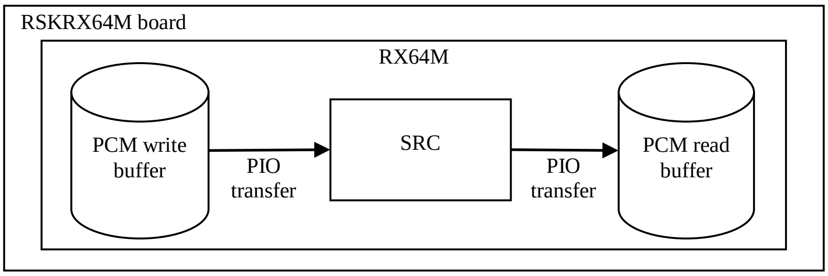
```
r01an2839ej****_src_rx + r01an2839ej****_rx.pdf
                       |
                       \ ---workspace
                              rx64m_example_src.zip
                              rx64m_example_src_dma.zip
```

Those two .zip files are archived C project files of e2studio. Sampling rate conversion operation types and the corresponding C project archive files are shown in Table 1-1. Choose one among them, import and execute it on e2studio. Each projects' operation and audio signal flow by operation types are shown in Table 1-2.

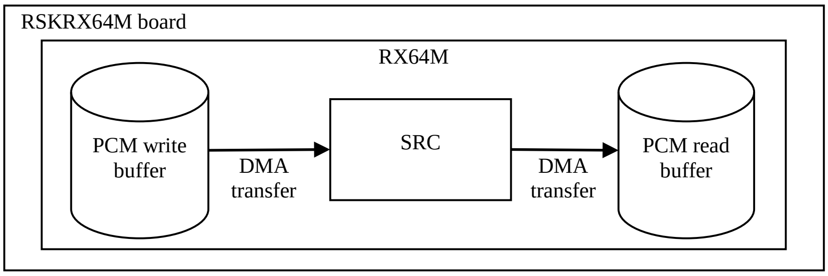
Table 1-1 Operation type and corresponding C project file

Operation type	C project file
Sampling rate conversion with PIO	rx64m_example_src.zip
Sampling rate conversion with DMAC	rx64m_example_src_dma.zip

Table 1-2 Signal flow by operation types

<table>
<tr><td>Operation type</td><td>audio signal flow in RSKRX64M board</td></tr>
<tr><td>Sampling rate conversion with PIO</td><td>Writes PCM data contained in PCM write buffer to SRC's Input FIFO, and after sampling rate conversion read PCM data on Output FIFO and store it in PCM read buffer.


</td></tr>
</table>

Sampling rate conversion with DMAC	Using DMAC, writes PCM data contained in PCM write buffer to SRC's Input FIFO, and after sampling rate conversion read PCM data on Output FIFO and store it in PCM read buffer. 

1.2 Hardware requirement

It shows signal connection between the RSKRX64M CPU board and external components. Refer to the document "RX64M Group Renesas Starter Kit+ User's Manual For e2 studio" and compose the target system.

Components of target system

- Target board : RSKRX64M CPU board
  It is provided with Renesas Starter Kit+ for RX64M (RSK+RX64M).
  Product No: R0K50564MC010BR
- Target MCU : RX64M (R5F564MLCDFC)

1.3 Development tools requirement

- e2studio integrated development environment, version 3.1.2.10, from Renesas Electronics
- RX Family C/C++ compiler package, version 2.02.00, from Renesas Electronics
- E1 emulator from Renesas Electronics

2. Sample program

2.1 Software components

The sample programs consist of several software components as shown in Table 2-1. The main() executes system configuration and sampling rate conversion operation. It controls MCU peripherals indirectly using FIT modules. So look the source file with main() to understand sampling rate conversion operation using FIT modules' APIs. For detailed information about FIT modules, refer to corresponding application notes. (see 4. Reference Documents)

Table 2-1 Software components by operation type

Operation type	Source file with main()	FIT modules
sampling rate conversion with PIO	r_main_src.c (Rev.1.00)	BSP FIT Module (Rev.2.80) SRC FIT Module (Rev.1.11)
Sampling rate conversion with DMAC	r_main_src_dma.c (Rev.1.00)	BSP FIT Module (Rev.2.80) SRC FIT Module (Rev.1.11) DMAC FIT Module (Rev.1.02)

The both samples include PCM data table file "r_test_wave.h".

2.2 Sampling rate conversion using PIO

Figure 2-1 shows main elements for sampling rate conversion operation using PIO. As shown in it, there is just a main routine, no other routine like interrupt service. The main() executes system configuration and sampling rate conversion operation. It is a simple polling system accompanying with SRC peripheral's behavior.

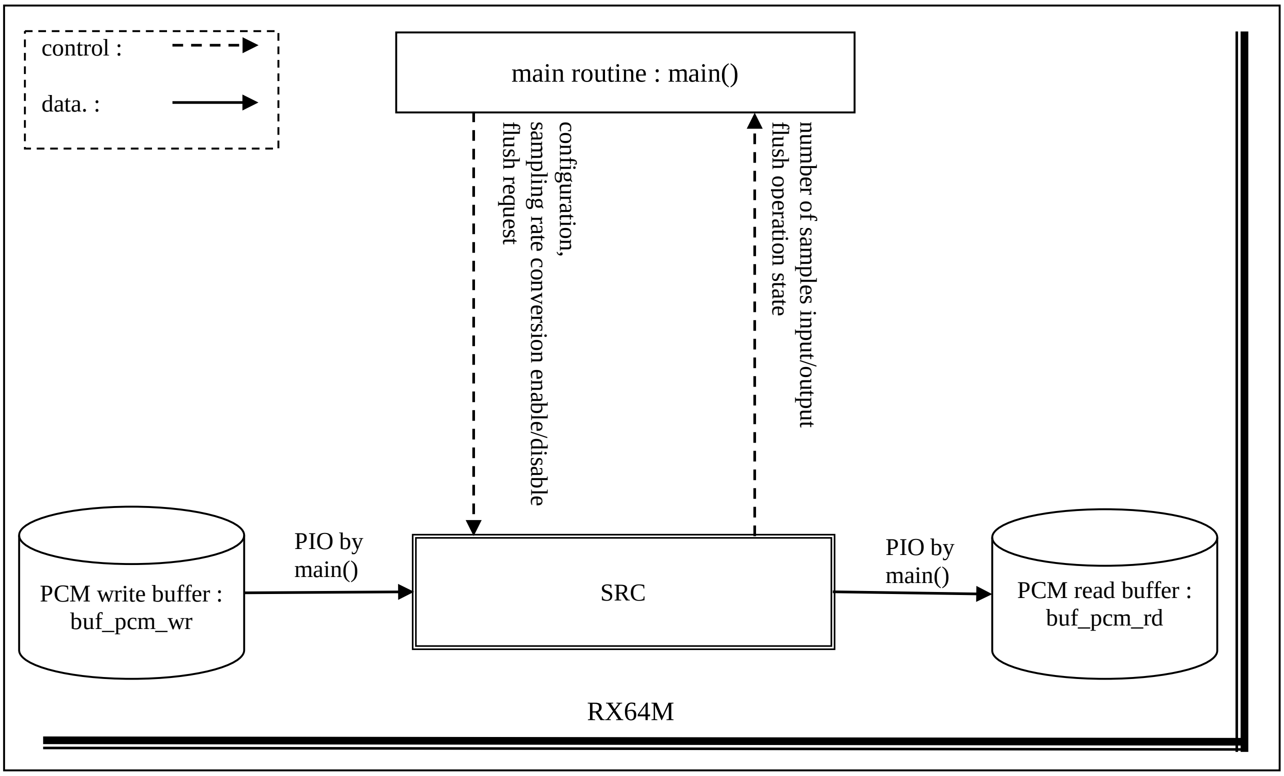


Figure 2-1 Sampling rate conversion operation

Refer to Figure 2-1, operating steps for sampling rate conversion using PIO are described as follows.

1) Peripheral setting
main() configures SRC using SRC FIT module APIs.
2) PCM write and read buffer information setting
main() sets PCM read and write buffer information. The information consists of the start address and the number of samples of PCM buffer. It is a unit of transfer, used for PCM data write and read operation described below.
3) Sample rate conversion start
main() specifies input and output sampling rate and enables SRC to start sampling rate conversion.
4) Sample rate conversion
main() executes following a) and c) repeatedly and asynchronously during sampling rate conversion.
   a) main() writes PCM data before conversion to SRC's Input FIFO.
   b) SRC starts conversion and output PCM data after conversion on its Output FIFO.
   c) main() reads the PCM data after conversion from the Output FIFO.

   PCM data write operation
   The above process a) is data transfer from the PCM write buffer to SRC’s Input FIFO using R_SRC_Write(). The address of buffer and the number of samples are specified at the step 2). main() updates them every time after R_SRC_Write() call using its return value. The return value shows the number of samples successfully transferred.

   PCM data read operation
   The above process c) is data transfer from SRC’s Output FIFO to the PCM read buffer using R_SRC_Read(). The address of buffer and the number of samples are specified at the step 2). main() updates them every time after R_SRC_Read() call. The return value shows the number of samples successfully transferred.
5) Finish conversion
When the specified number of samples (one block of PCM write buffer) is input, main() stops the process a), main() calls R_SRC_Stop() to request SRC to flush processing to finish conversion. Then main() continues the process b) until all PCM data after conversion is read and R_SRC_Read()’s return value shows the end of flush processing. Then all PCM data after sampling rate conversion is stored.

2.3 Sampling rate conversion using DMAC

Figure 2-2 shows main elements for sampling rate conversion operation using DMAC. As shown in it, there is just a main routine, no other routine like interrupt service. The main() executes system configuration and sampling rate conversion operation. It is a simple polling system accompanying with SRC's behavior.

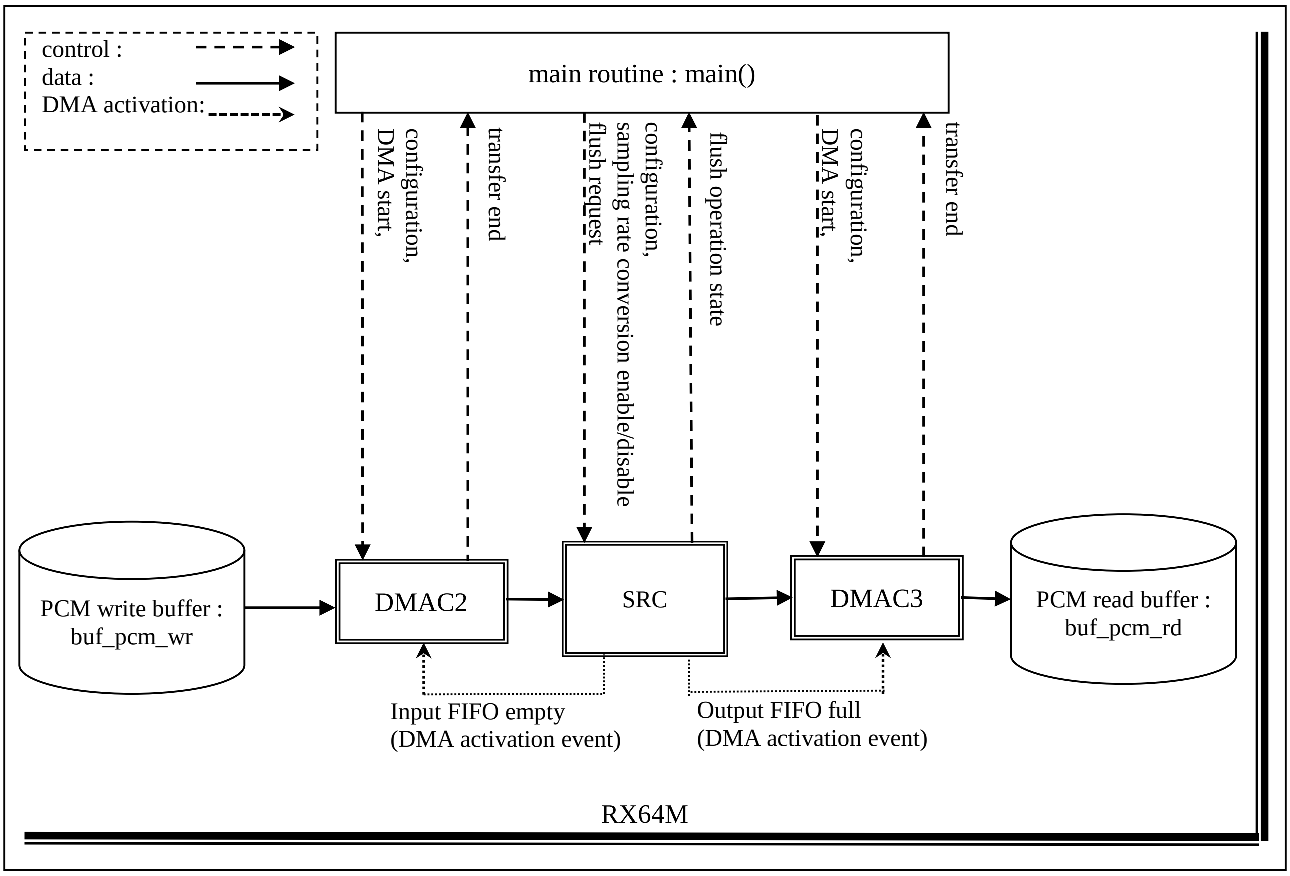


Figure 2-2 Sampling rate conversion operation using DMAC

Refer to Figure 2-2, operating steps for sampling rate conversion using DMAC are described as follows.

1) Peripheral setting
   main() configures DMAC and SRC using DMAC and SSI FIT modules.
2) DMAC transfer information setting
   main() configures DMAC2 and DMAC3 for PCM data transfer using DMAC FIT module's APIs. The unit of transfer is a block of PCM buffer so main() sets the start address and the number of samples of PCM write and read buffer. And main() enables DMAC2 and 3 transfers at this step.
3) Sample rate conversion start
   main() specifies input and output sampling rate and enables SRC to start sampling rate conversion.
4) Sample rate conversion
   DMAC2 and DMAC3 start transfer activated by events occurred from SRC. Input FIFO empty event activates DMAC2 and Output FIFO full event activates DMAC3. The DMA transfers occur repeatedly and asynchronously during sampling rate conversion with occurrence of the events.
   a) DMAC2 transfers PCM data before conversion to SRC's Input FIFO.
   b) SRC starts conversion and output PCM data after conversion on its Output FIFO.
   c) DMAC3 transfers PCM data after conversion from SRC's Output FIFO.

   PCM data write transfer by DMAC2
   The above a) is data transfer from the PCM write buffer to SRC's Input FIFO using DMAC2. The address of buffer and the number of samples are specified at the step 2). When all PCM data is transferred, main() configures DMAC2 for the next transfer and enables it.

PCM data read transfer by DMAC3

The above b) is data transfer from SRC's Output FIFO to the PCM read buffer using DMAC3. The address of buffer and the number of samples are specified at the step 2). When all PCM data is transferred, main() configures DMAC3 for the next transfer and enables it.

5) Finish conversion

When specified times of PCM write transfer by DMAC2 is completed (two units of DMA transfer in this sample program), main() stops the process a) and main() calls R_SRC_Stop() to request SRC to flush processing to finish conversion. Then main() continues the process b) until all PCM data after conversion is read. main() uses R_SRC_CheckFlush() to check the end of sampling rate conversion. When it shows Conversion End, main() stops DMAC3 transfer forcefully. Then all PCM data after sampling rate conversion is stored.

2.4 Required MCU resource

Table 2-2 shows MCU resources used for two sample programs.

Table 2-2 Required MCU resource

	MCU resources by sample programs	
	sampling rate conversion with PIO	Sampling rate conversion with DMAC
SRC	SRC	SRC
DMA activation source	-	SRC's Input FIFO empty (for DMAC2), SRC's Output FIFO full (for DMAC3)
DMAC	-	DMAC2, DMAC3

2.5 PCM data buffer

PCM write buffer and PCM read buffer shown in Figure 2-1 and Figure 2-2 are containers for PCM data before and after sampling rate conversion. The buffers are uint16_t type of two-dimensional array. The array row shows buffer's number of blocks and the array column shows the number of samples per block. Table 2-3 shows the summary of PCM buffer size by conversion operation types.

- The number of blocks: see Table 2-3.
- The number of samples per block: 1024 x2.
- PCM data storing order: Left channel, Right channel, Left channel, Right channel,

Table 2-3 PCM data buffer size

		sampling rate conversion with PIO	Sampling rate conversion with DMAC
PCM write buffer	blocks	1	2
	samples/block	1024 x2	1024 x2
PCM read buffer	blocks	6	12
	samples/block	1024 x2	1024 x2

3. Sample project execution

3.1 Hardware setup

Figure 3-1 shows operating environment. Refer to it, setup the environment keeping connecting order shown as follows.

1. Connect the E1 emulator to the RSKRX64M board of the target system with the user cable.
2. Connect the E1 emulator to the host PC with the USB cable.

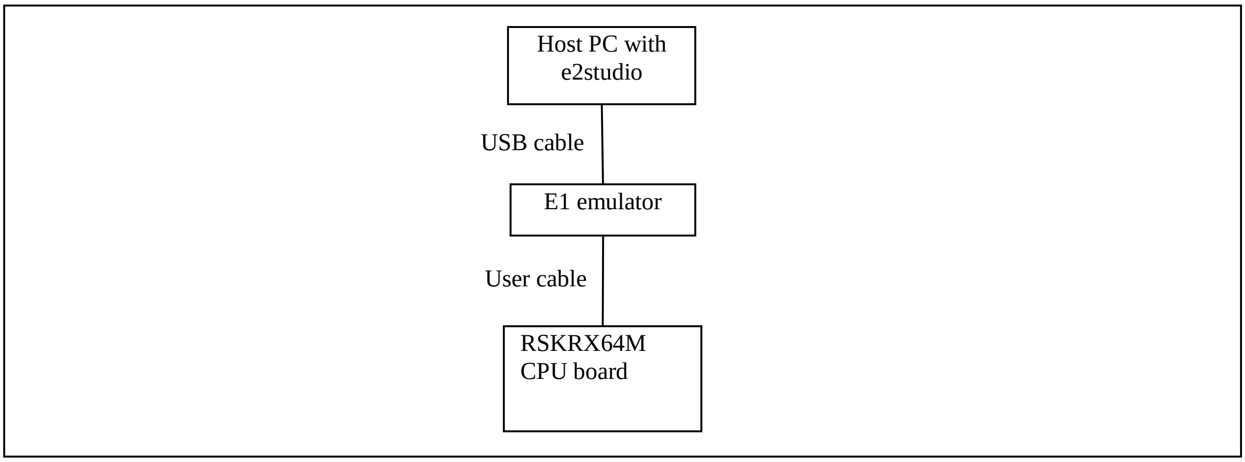


Figure 3-1 Operating environment

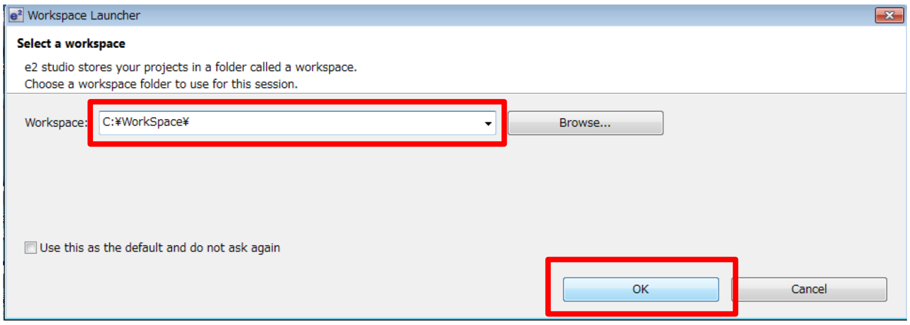
3.2 Sample project execution using e2studio

3.2.1 Import sample project to e2stusio

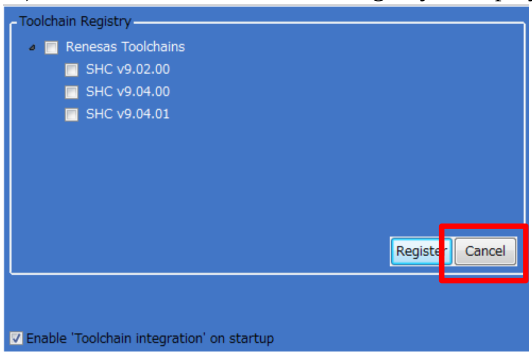
In this part, the instruction to import the project “rx64m_example_src” is shown as an example.

1) Start e2studio

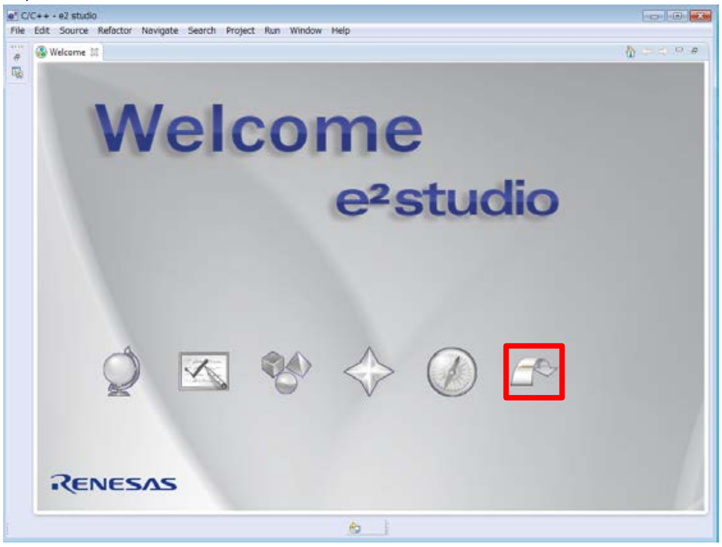
2) Workspace Launcher is displayed, Specify the folder to store the project (e.g. C:\workspace) in this dialog and click OK.



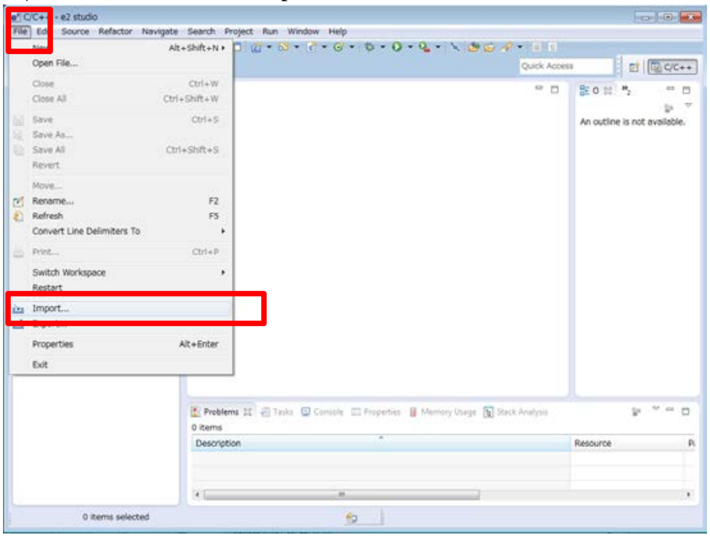
3) Click Cancel if Toolchain Registry is displayed.



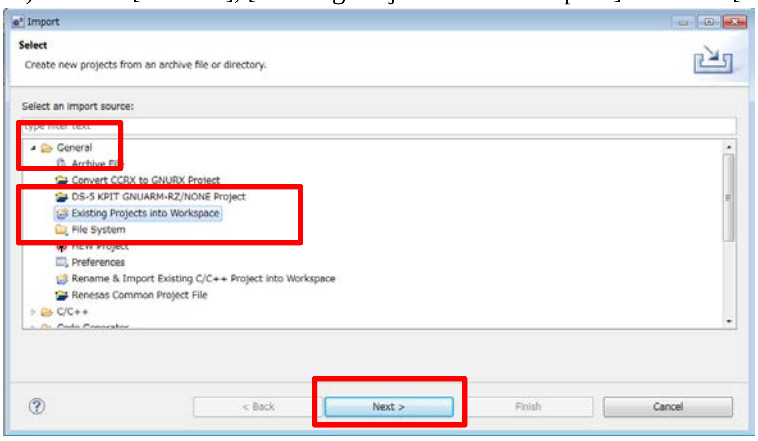
4) Click Workbench.



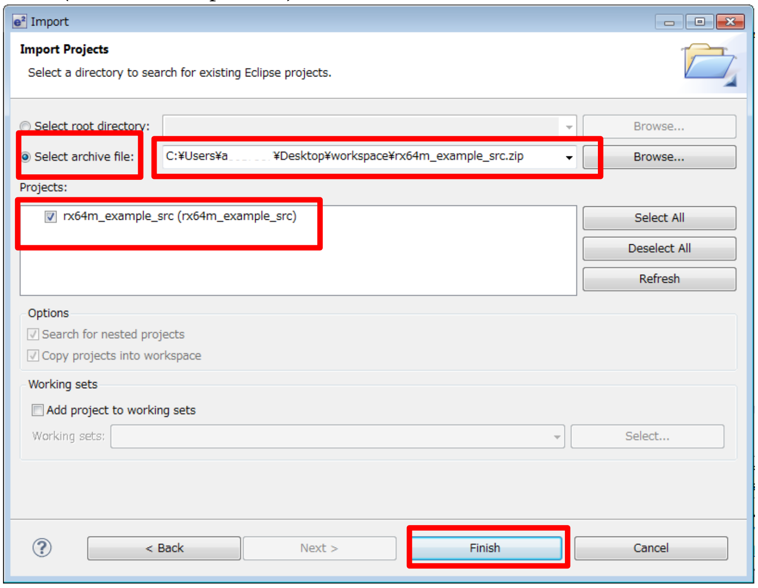
5) Select [File] and [Import…]



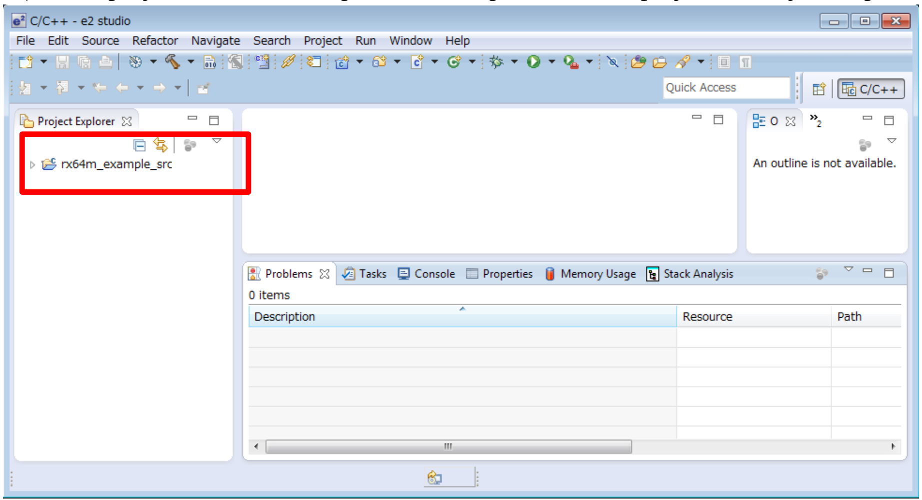
6) Select [General], [Existing Projects into Workspace] and click [Next >]



7) Select [Select archive file], specify the archive file “rx64m_example_src.zip”, check “rx64m_example_src (rx64m_example_src) and click [Finish].



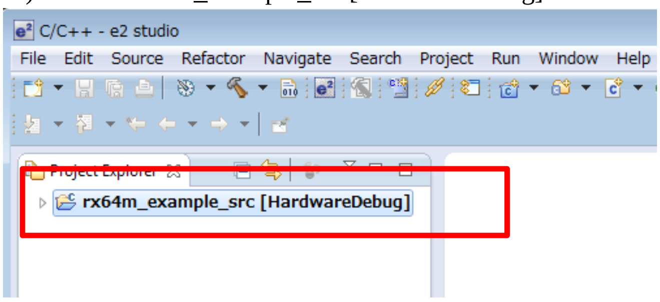
8) The project “rx64m_example_src” is imported and displayed on Project Explorer pane.



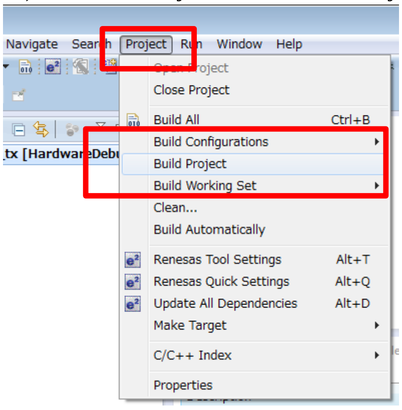
3.2.2 Build the sample project

In this part, the instruction to build the project “rx64m_example_src” is shown as an example.

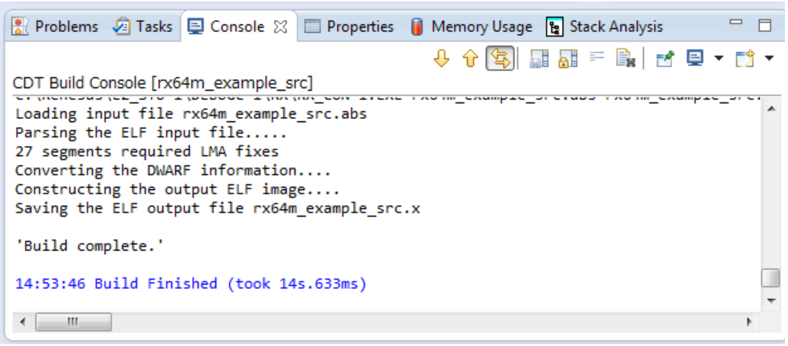
1) Click rx64m_example_src [HardwareDebug].



2) Select [Project] and [Build Project].



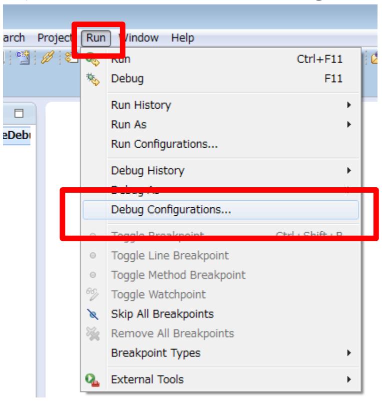
3) After build, 'Build complete' message is shown on the console.



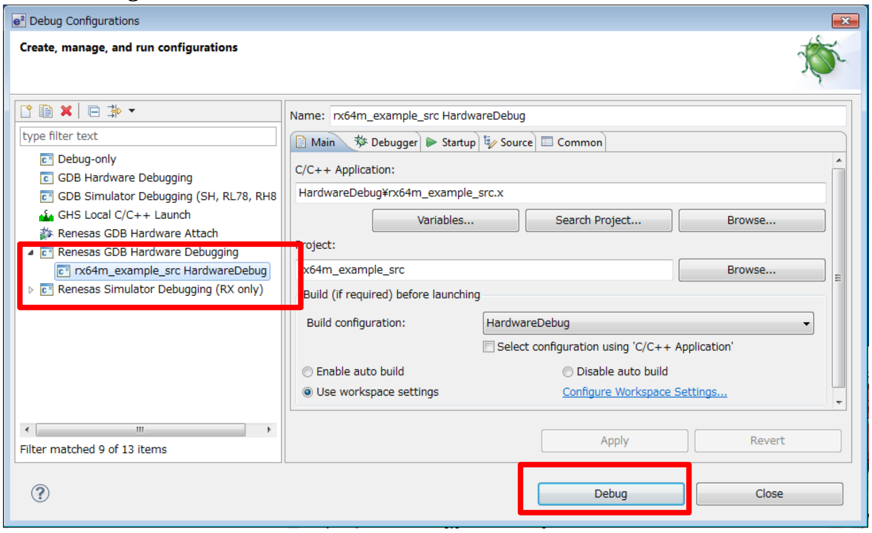
3.2.3 Connect to the target system, download and execute project

In this part, the instruction to connect e2studio with target system, download the road module and execute the "rx64m_example_src" are shown.

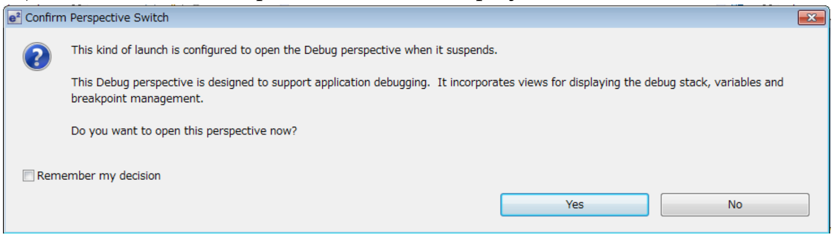
1) Select [Run] and [Debug Configurations…].



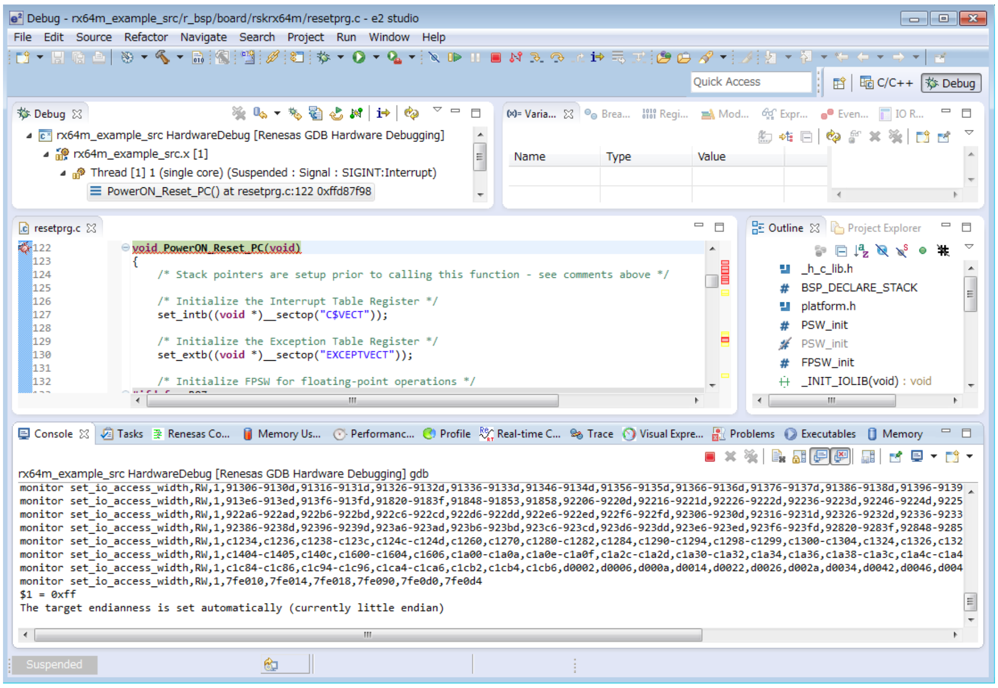
2) Click "Renesas GDB Hardware Debugging", select "rx64m_example_src HardwareDebugging" and click [Debug].



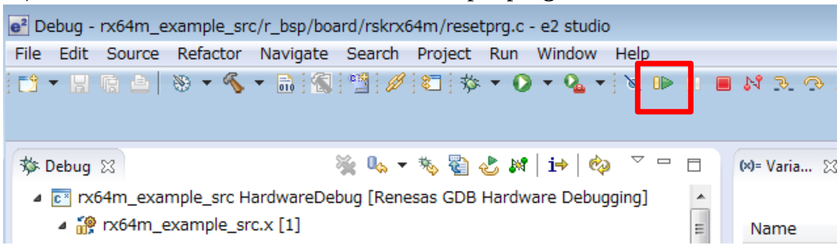
3) Confirmation Perspective Switch is displayed, click [Yes].



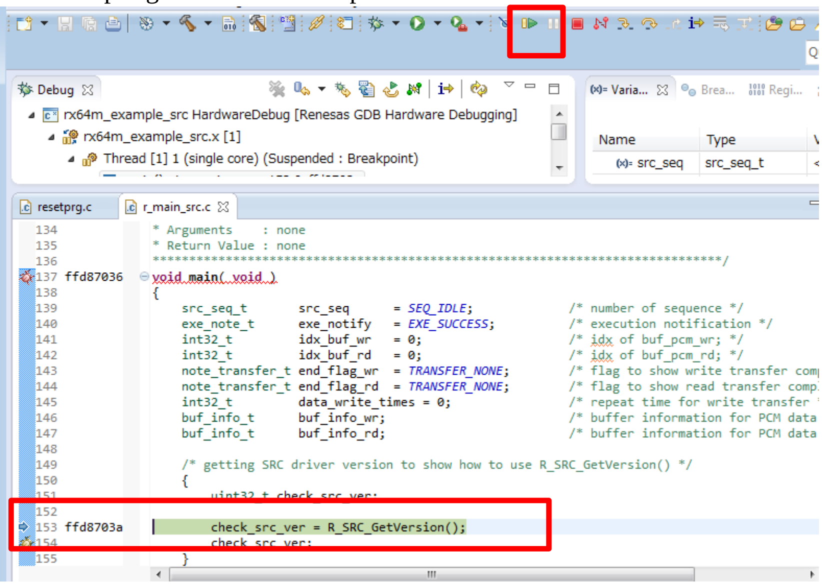
4) Perspective is switched to Debug perspective.



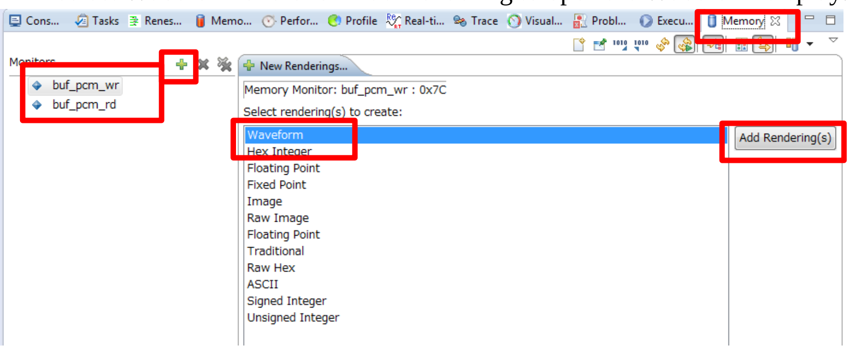
5) Click Resume icon to execute the sample program.



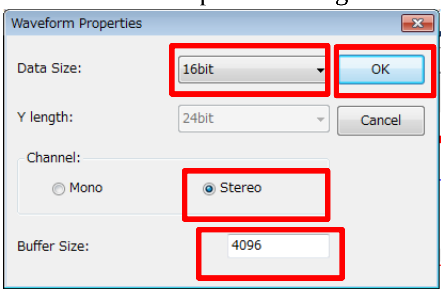
6) As stopping at the first line of main(), click Resume icon again. The sample program execution is restarted. Sampling rate conversion operation is started and PCM data after conversion is stored in PCM read buffer.



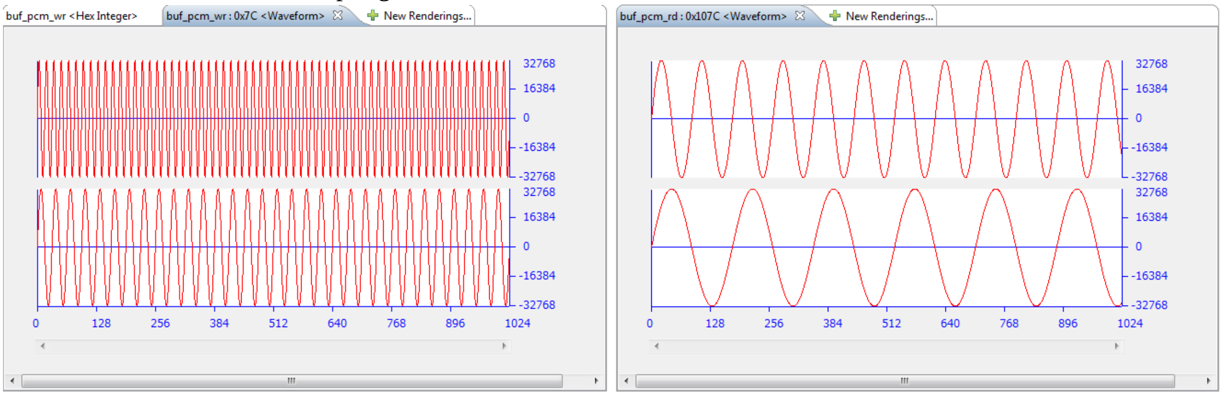
7) PCM data before/after conversion is stored on PCM write/read buffer. Using Waveform display of Memory view, users can see the wave forms of PCM data before/after conversion. Click + to add “buf_pcm_wr”/“buf_pcm_rd”, select Waveform and click Add Rendering to open the Waveform display.



Waveform Properties setting is shown as follow.



Waveform before/after sampling rate conversion is shown as follows.



4. Reference documents

Firmware Integration Technology application note

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

RX Family SRC Module using Firmware Integration Technology (R01AN2090EJ)

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

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 For e2 studio Rev.1.00 (R20UT2593EG)

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)

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.

Website and Support

Renesas Electronics Website

http://www.renesas.com/

Inquiries

http://www.renesas.com/contact/

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	June.16.2015		First edition issued

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.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

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: +44-1628-585-100, Fax: +44-1628-585-900

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, P. R. China 200333
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, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141