Bio Sensing Software Platform

Software Library for Recording Heart Sound

R11AN0326EJ0101
Rev.1.01
Sep 14, 2018

Abstract

This document describes the sample program to record the heart sound using microphone and A/D converter.

Target Device

RX231

Target Board

- Renesas Starter Kit for RX231 (R0K505231S000BE) (Renesas Electronics)
  Hereafter, it is abbreviated as RX231 RSK.

Contents

1. Introduction

This application note describes how to record the heart sound design using RX231and microphone.

1.1 Terminology

Table 1-1 Terminology

Term	Meaning
ADC	Analog-Digital Converter
API	Application Programming Interface
DTC	Data Transfer Controller
HPF	High Pass Filter
HSM	Heart Sound Monitoring

1.2 Overview

This application note answers the following topics:

- Overview of Heart Sound recording

1.3 Devices

In this application note, the system is constructed following major devices:

- MCU: RX231 series 32bit microcontroller by Renesas Electronics.

  RX231 series CPU leverages a 32bit RXv2 CPU core with DSP/FPU and low power consumption technology to realize extreme power efficiency.

- ADC: Internal 12-bit ADC of RX231 by Renesas Electronics.
- Microphone

2. Functional Purpose

Heart sounds are the noises generated by the beating heart and the resultant flow of blood through it.

The HSM realizes below functionalities:

- Recording the heart sound

3. Measurement Principle

3.1 Recorded Sound

The recorded sound will be stored in a global buffer. The recorded sound will be filtered before storing it in the buffer. The samples would be filtered for a frequency range of 20Hz to 200Hz, to get the Heart Sound.

3.2 Low Pass Filter

The sampled data may contain noise signals. So the data need to be filtered. Direct form 1 Biquad filter would be used for filtering.

LPF details:

Order: 2nd order Biquad filter

Type: Butterworth filter

Cut-off: 200Hz

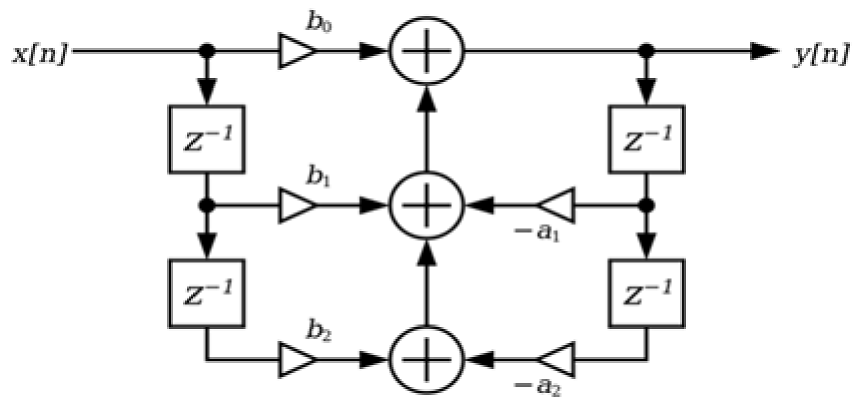

Figure 3-1 Flow Graph of Direct Form 1Biquad Filter

3.2.1 Equations

he normalized direct form 1 equation is given below,

y[n] = (b0 * x[n]) + (b1 * x[n-1]) + (b2 * x[n-2]) – (a1 * y[n-1]) – (a2 * y[n-2])

Where,

b0, b1 and b2 are the coefficients determining zero
a1 and a2 are the coefficients determining poles

3.2.2 Filter Coefficient Calculation

Butterworth filter has been chosen due to it low ripple in the pass band and stop band region than other filter types.

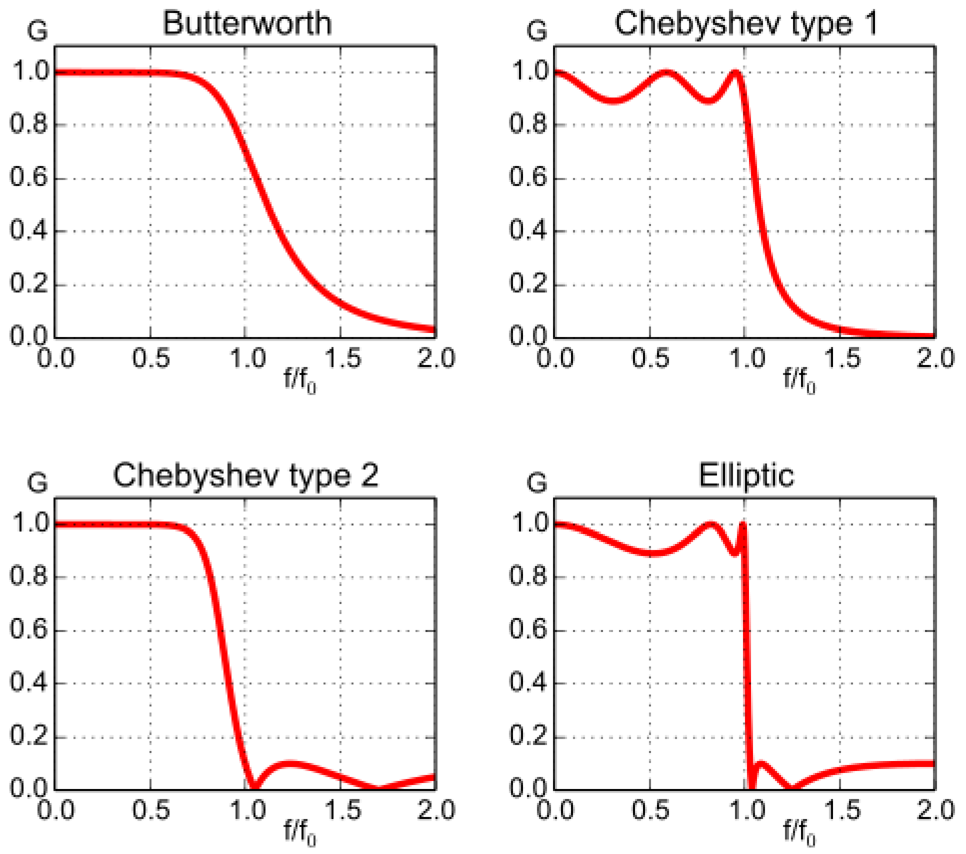

Figure 3-2 Flow Graph of Direct Form 1Biquad Filter

Butterworth filter has a frequency response as flat as possible in the passband. It also has low ripples in the stop band. So Butterworth has been chosen as filter. The Butterworth filter coefficients can be calculated by using below formula:

w0 = (2 * pi * (f0 / fs))
b0 = (1 - cos(w0)) / 2
b1 =   1 - cos(w0)
b2 = (1 - cos(w0)) / 2
a0 =   1 + (sin(w0) / (2 * Q))
a1 = (-2 * cos(w0))
a2 = 1 – (sin(w0) / (2 * Q))

Where,

f0 = cut-off frequency
fs = sampling frequency
Q = Quality factor (0.7071)

3.2.3 Filtering the Recorded Heart Sound

The low pass filter algorithm used to filter the recorded heart sound is given below:

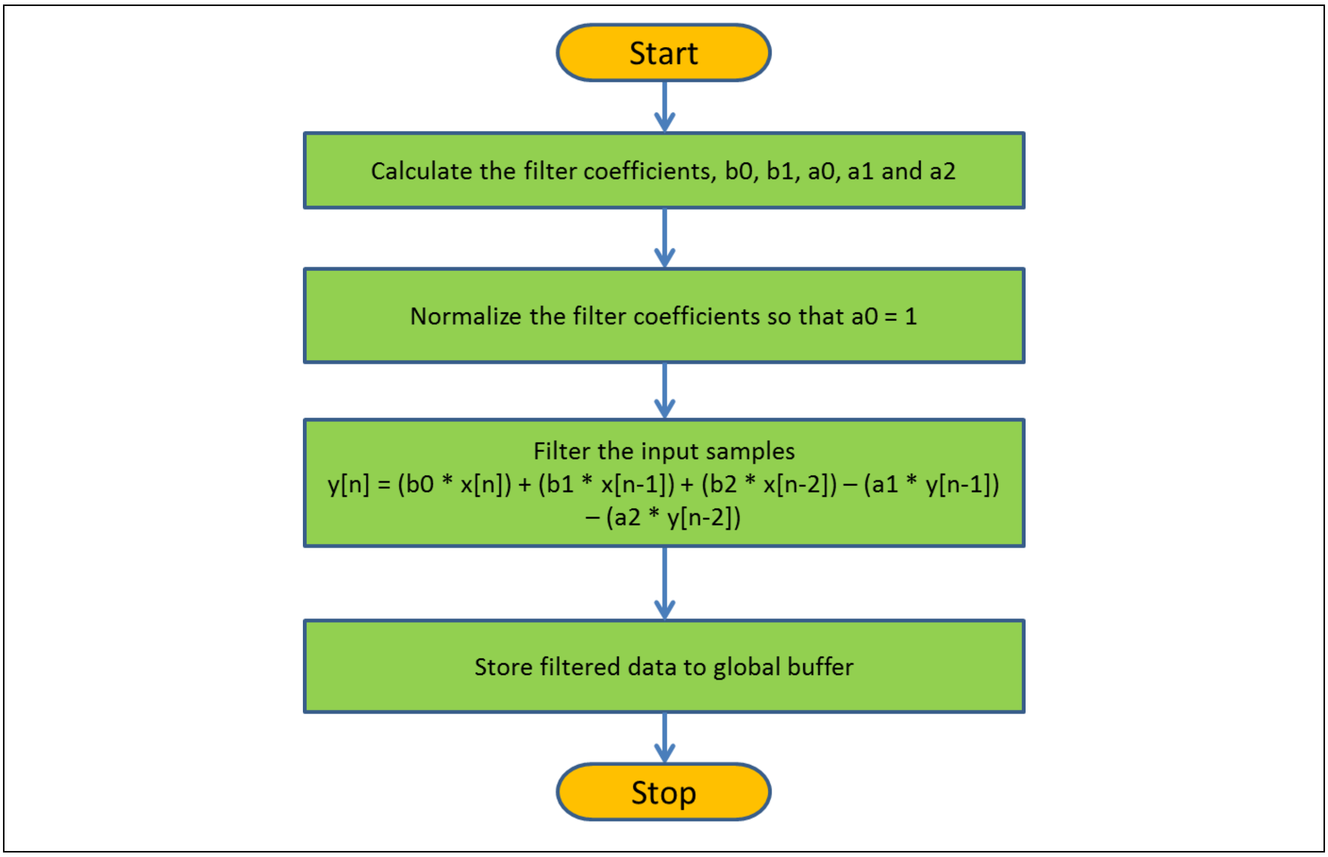

Figure 3-3 Algorithm for the LPF used to Filter the Recorded Heart Sound

3.3 Algorithm

The algorithm used to record Heart Sound is described in the following flow chart:

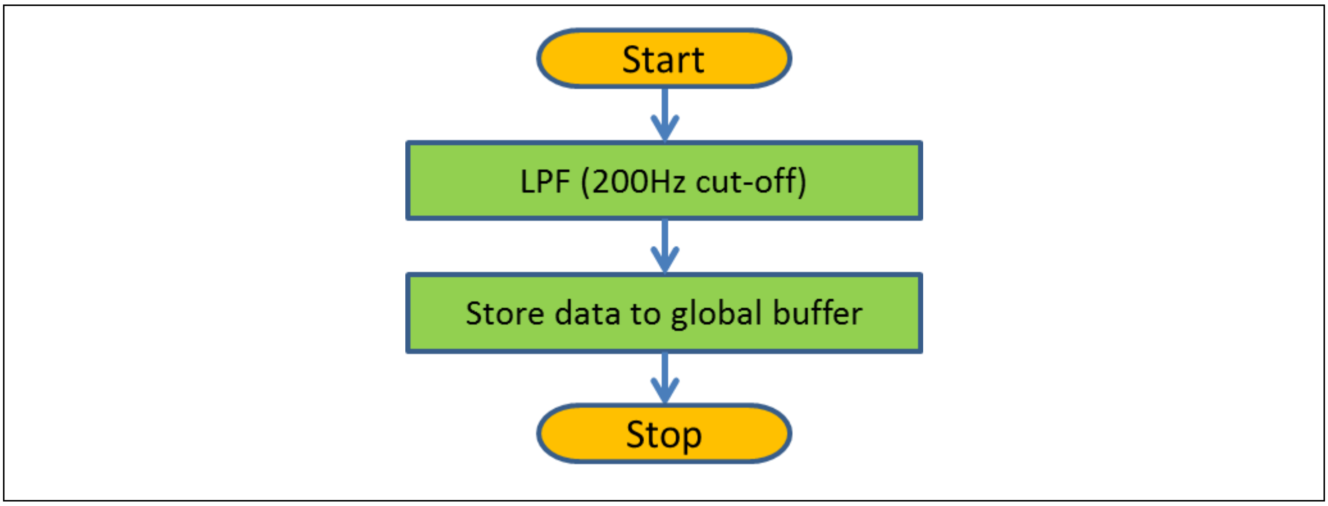

Figure 3-4 Heart Sound Recording Algorithm

4. Libraries

4.1 API List

Each API functions' role is shown in the below table:

Table 4-1 API List

Functional Name	Function
R_HSM_StartSampling(void)	Resets NewDataReady flag, Page variable and global buffer. Starts MTU count.
R_HSM_StopSampling(void)	Stops MTU count. This function is not used in this application. This is for extension by customer.
R_HSM_SetPage(uint8_t page_ready)	When timer interrupt is CMIA (compare match A 128), sets Page=0 and new data ready flag. When timer interrupt is OVI (Overflow), sets Page=1 and new data ready flag.
R_HSM_StoreSound(void)	Reads the ADC samples and stores the values to global buffer after filtering. Resets NewDataReady flag.
R_HSM_IsSampleDataReady(void)	Returns g_hsm_new_data_ready flag status. The flag will indicate whether samples of heart sound data are ready or not.

4.2 Global Variables

The recorded Heart Sound will be stored in the below buffer:

Table 4-2 List of Global Variables

Global Variables	Function
g_hsm_ready_page_number	Global variable used to indicate which page contains 128 samples of heart sound data.
g_hsm_new_data_ready	Flag used to indicate heart sound data of 128 samples are ready.
g_hsm_buffer	global buffer for storing 256 samples, each of size 2 bytes. The buffer would be cleared during initialization.
g_hsm_lpf_handle	Biquad filter's parameters

4.3 Memory Size

Table 4-3 Memory Size

Memory	Size
ROM	1,515 bytes for HSM library [Note1]
RAM	1,574 bytes for HSM Library [Note1]
User Stack	64 bytes
Interrupt Stack	40 bytes

Note1: Refer to "5.2.2 Software Architecture Overview". It does not include device drivers.

5. Applications

5.1 Hardware

5.1.1 Hardware Design Policy

Hardware design policy of the HSM is

- The Hardware components used in the system shall be commercially available in the market. This helps the user to reproduce the development environment easily.

5.1.2 Hardware Block Diagram

The HSM will be developed to run on RX231 RSK.

The Block Diagram for the overall system is shown below:

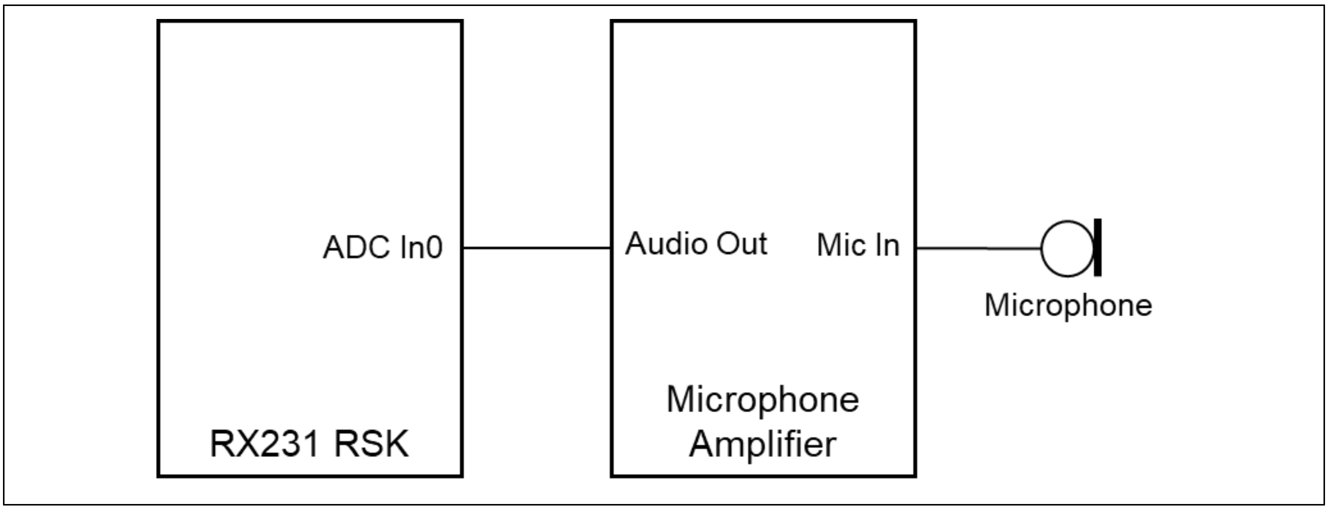

Figure 5-1 Hardware Block Diagram

5.1.3 RX231 MCU Digital Interconnect

The functional block diagram for the HSM library with the RX231 peripheral blocks is shown below:

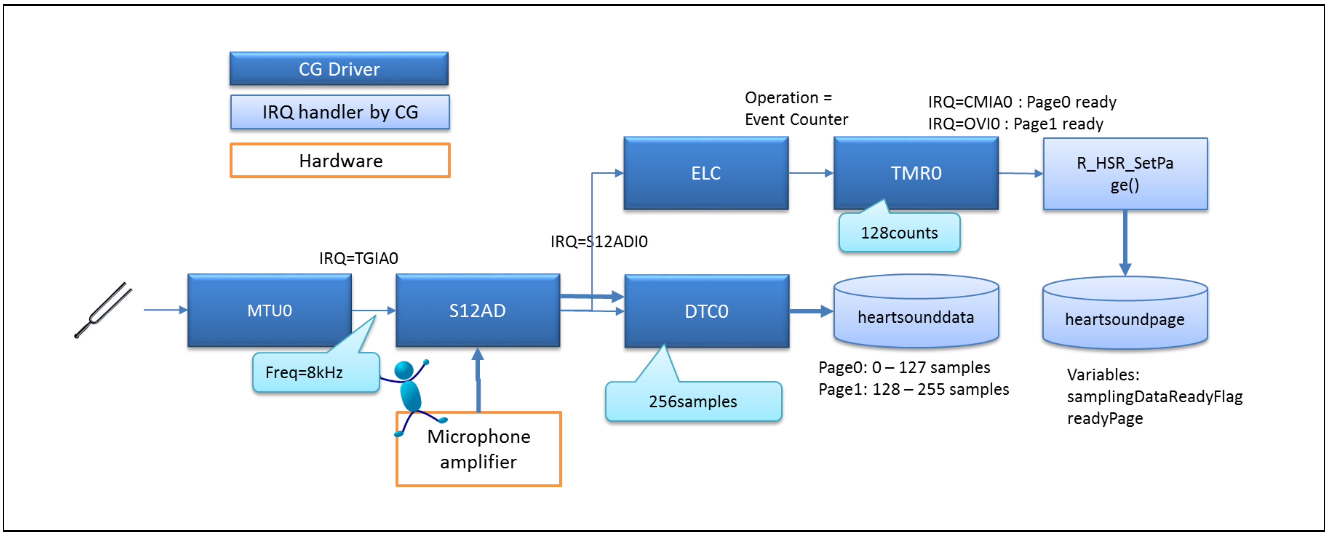

Figure 5-2 HSM Digital Interconnect

- The Heart Sound would be sampled by 12-bit ADC, S12AD channel 0.
- MTU0 is configured to oscillate at a frequency of 8000Hz for ADC sampling and on MTU0 interrupt AD conversion is triggered.
- DTC0 is configured to transfer AD converted sample on S12AD conversion complete interrupt. The transfer would be repeated for each 256-word.
- The ADC conversion complete interrupt is also connected to ELC. On every interrupt ELC generates an event which is used as clock for TMR0.
- TMR0 is configured to generate both compare match interrupt CMIA0 and overflow interrupt OVI0. TMR0 compare match value is configured to 128.
- On CMIA0 interrupt page0 ready flag is set.
- On OVI0 interrupt, page1 ready flag is set.

5.2 Software

5.2.1 Software Design Policy

The Software Design Policy is:

- HSM Software is designed for Non-OS environment.
- Code Generator Utility is used to create device drivers and hence the software framework is driven by the Code Generator Utility.
- The Heart Sound Recording/Data Sampling is automatically started on software initialization, immediately after all the peripherals are initialized and configured.
- The Heart Sound is recorded continuously unless the system is powered off. The sampling can be stopped temporarily during the calculation, if needed.
- Recording and storing of the heart sound are executed continuously using an infinite loop (while (1U) {}) in the main () function.
- The recorded sounds are not stored in any file and no playback or display options are provided.
- The software does not include power management.
- The HSM Software Architecture is designed to be a simple Layered Architecture where each layer exposes a set of APIs to the layer above it. The Software Layered Architecture is described in the next section.

5.2.2 Software Architecture Overview

The Software architecture is a layered architecture, where each layer provides a set of APIs for the above layers to access it.

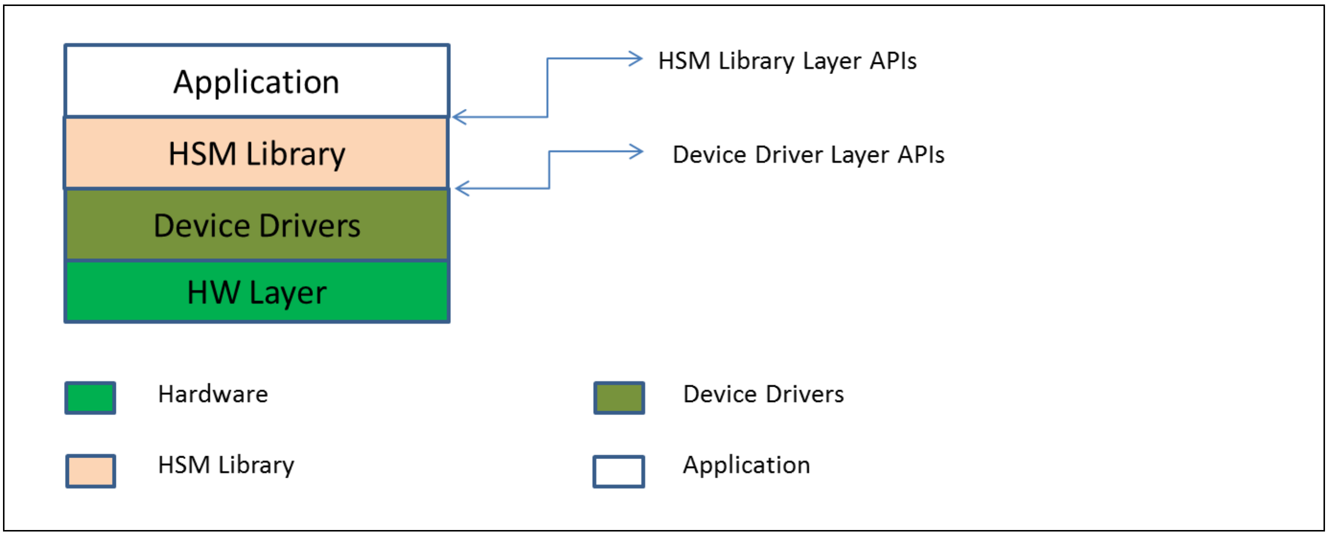

Figure 5-3 HSM Software Architecture

5.2.3 Measurement Signal Flow

The heart sound would be captured using microphone (simulated). The sound would be amplified before transmitting to RX231 RSK. The heart sound recording signal flow is shown in the diagram below:

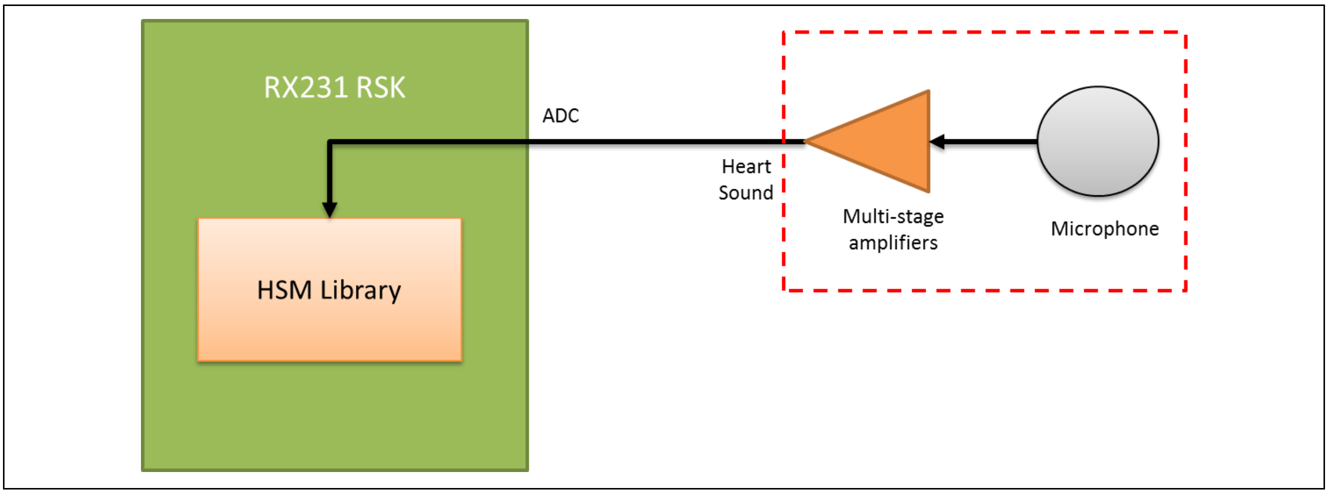

Figure 5-4 Interface of the Microphone with the MCU

5.2.4 Measurement Control Flow

HSM Software implements a simple application to use the HSM Library. The application is automatically started on reset. The heart sound is recorded continuously by the hardware until the device is powered off. The recorded data is transferred from DTC to RAM area by the software whenever 128 samples are collected.

The Control Flow of the main application is shown in the flow chart below:

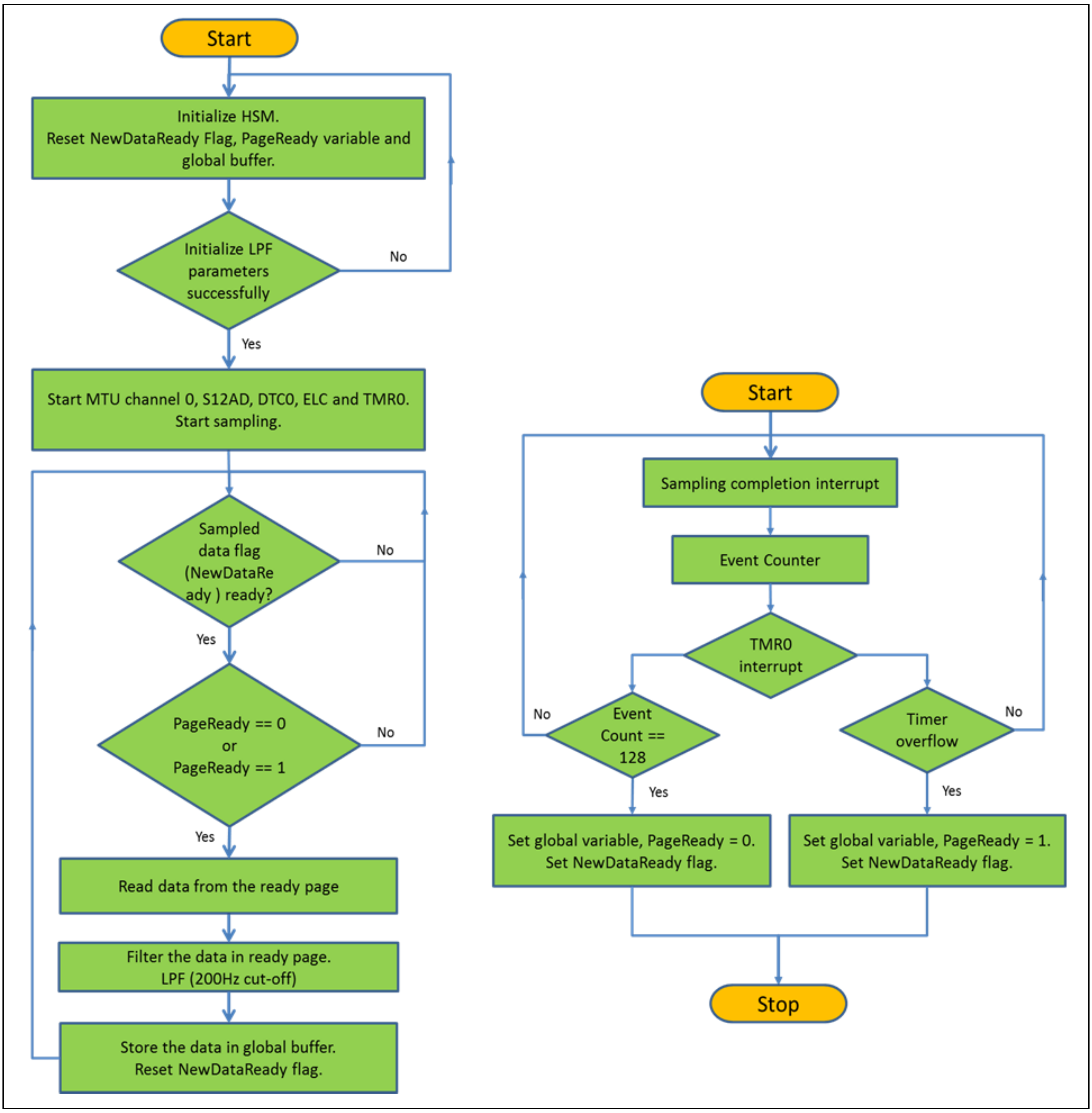

Figure 5-5 HSM Software Control Flow

5.3 Device Drivers

5.3.1 Code Generator

Table 5-1 shows the used peripheral function.

Table 5-2 shows the functions generated by Code Generator.

Table 5-1 Used Peripheral function List

Peripheral function	Macro	Sub Macro	Setting	Status
Clock Generator	CGC		VCC setting	2.7 (V) = VCC = 5.5 (V)
			Main clock oscillation source	Resonator
			Main clock oscillation source Frequency	8(MHz)
			Oscillator wait time	8192cycles2048 (µs)
			Oscillation stop detection function	Disabled
			PLL circuit setting	
			Input frequency division ratio	x 1/2
			Frequency multiplication factor	x 8
			PLL Frequency	32 (MHz)
			Sub-clock oscillator drive capacity	Drive capacity for low CL
			Sub-clock oscillator and RTC (RTCSCLK) setting	32.768 (kHz)
			Low speed clock oscillator (LOCO) setting	4 (MHz)
			Clock source	Main clock oscillator
			System clock (ICLK)	x 18 (MHz)
			Peripheral module clock (PCLKA)	x 18 (MHz)
			Peripheral module clock (PCLKB)	x 18 (MHz)
			Peripheral module clock (PCLKD)	x 18 (MHz)
			External bus clock (BCLK)	x 18 (MHz)
			Flash IF clock (FCLK)	x 18 (MHz)
Data Transfer Controller	DTC	BaseAddress	Transfer data read skip	Disable
			Address mode	Full-address mode (32 bits)
			DTC vector base address	0x0000FC00
		DtcChannel0	Transfer data0	Used
			Chain transfer	Unused
			Activation source	S12AD (S12ADI0 vect=102)
			Transfer mode setting	Repeat mode
			Transfer data size setting	16 bits
			Interrupt setting	An interrupt request to the CPU is generated each time DTC data transfer is performed
			Block / Repeat area setting	Transfer destination
			Source address	0x00089020(Address fixed)
			Destination address	0x00004000(Address incremented)
			Count	256
Event Link Controller	ELC	ELC_TMR0	TMR0	Used
			Event signal	S12AD A/D conversion end
			Operation on event	Event counter

Multi-Function Timer Pulse Unit 2	MTU2_U0	MTU0	MTU0	Normal mode
			Include this channel in the synchronous operation	Unused
			Counter clock selection	PCLK/64
			Clock edge setting	Falling edge
			Counter clear source	TGRB0 compare match/input capture (Use TGRB0 as a cycle register)
			TGRA0 (Output compare register)	125µs, (Actual value: 128)
			TGRB0 (Output compare register)	125µs, (Actual value: 128)
			TGRC0 (Output compare register)	100µs, (Actual value: 104)
			TGRD0 (Output compare register)	100µs, (Actual value: 104)
			TGRE0 (Output compare register)	100µs, (Actual value: 104)
			TGRF0 (Output compare register)	100µs, (Actual value: 104)
			MTIOC0A pin (P34)	MTIOC0A pin output disabled
			MTIOC0B pin (P15)	MTIOC0B pin output disabled
			MTIOC0C pin (P32)	MTIOC0C pin output disabled
			MTIOC0D pin (PA3)	MTIOC0D pin output disabled
			Enable A/D conversion start request on TGRA input capture/compare match (trigger signal of MTU0 TRGA0N)	Used
			Enable TGRA0 input capture/compare match interrupt (TGIA0)	Used
			Enable TGRB0 input capture/compare match interrupt (TGIB0)	Unused
			Enable TGRC0 input capture/compare match interrupt (TGIC0)	Unused
			Enable TGRD0 input capture/compare match interrupt (TGID0)	Unused
			(TGIA/TGIB/TGIC/TGID) Priority	Level 15 (highest)
			Enable TGRE0 compare match interrupt (TGIE0)	Unused
			Enable TGRF0 compare match interrupt (TGIF0)	Unused
			Enable overflow interrupt (TCIV0)	Unused
8-Bit Timer	Tmr0	TmrChannel0	TMR0	8-bit count mode
			Clock source	External clock both edges
			External clock pin TMCI0	P21
			Counter clear	Disabled
			Compare match A value (TCORA)	128 count (Actual value: 128)
			Compare match B value (TCORB)	10 count (Actual value: 10)
			Enable TMO0 output	Unused
			Enable TCORA compare match interrupt (CMIA0)	Used
			Enable TCORB compare match interrupt (CMIB0)	Unused
			Enable TCNT overflow interrupt (OVI0)	Used
			Priority	Level 10

12-Bit A/D Converter	S12AD	AnalogInputChannelMode	S12AD operation setting	Used
			Operation mode setting	Single scan mode
			Double trigger mode setting	Disable
			A/D conversion select	High-speed
			Self diagnosis setting	Unused
			Disconnection detection assist setting	Unused
			A/D converted value count setting	Addition mode
			High-Potential reference voltage select	AVCC0
			Low-Potential reference voltage select	AVSS0
			Window function setting	Unused
			Window A operation setting	Unused
			Window B operation setting	Unused
			Data storage buffer setting	Disable
			Analog input channel setting	-
			AN000 Convert (Group A)	Used
			Conversion start trigger (Group A)	Compare match with or input capture from MTU0.TGRA
			Data placement	Right-alignment
			Automatic clearing	Disable automatic clearing
			AN000 Input sampling time	3.667(us), (Actual value: 3.625)
			Total conversion time (Group A)	8.875(us)
			ELC scan end event generation condition	On completion of all scans

Table 5-2 Functions Generated by Code Generator

Peripheral function	File	Macro	Function
Common	r_cg_main.c		void main(void)
			void R_MAIN_UserInit(void)
	r_cg_dbsct.c		-
	r_cg_intprg.c		void r_privileged_exception(void)
			void r_floatingpoint_exception(void)
			void r_access_exception(void)
			void r_undefined_exception(void)
			void r_reserved_exception(void)
			void r_nmi_exception(void)
			void r_brk_exception(void)
	r_cg_resetprg.c		void PowerON_Reset_PC(void)
	r_cg_sbrk.c		-
	r_cg_vecttbl.c		-
	r_cg_sbrk.h		-
	r_cg_stacksct.h		-
	r_cg_vect.h		-
	r_cg_hardware_setup.c		void R_Systeminit(void)
			void HardwareSetup(void)
	r_cg_macrodriver.h		-
	r_cg_userdefine.h		-
Clock Generator	r_cg_cgc.c		void R_CGC_Create(void)
	r_cg_cgc_user.c		-
	r_cg_cgc.h		-
Data Transfer Controller	r_cg_dtc.c		void R_DTC_Create(void)
		DTC0	void R_DTC0_Start(void)
			void R_DTC0_Stop(void)
	r_cg_dtc_user.c		-
	r_cg_dtc.h		-
Event Link Controller	r_cg_elc.c		void R_ELC_Create(void)
			void R_ELC_Start(void)
			void R_ELC_Stop(void)
			void R_ELC_GenerateSoftwareEvent(void)
	r_cg_elc_user.c		-
	r_cg_elc.h		-
Multifunction timer pulse unit 2	r_cg_mtu2.c		void R_MTU2_Create(void)
		MTU0	void R_MTU2_C0_Start(void)
			void R_MTU2_C0_Stop(void)
	r_cg_mtu2_user.c	MTU0	static void r_mtu2_tgia0_interrupt(void)
	r_cg_mtu2.h		-
8-Bit Timer	r_cg_tmr.c		void R_TMR_Create(void)
		TMR0	void R_TMR0_Start(void)
			void R_TMR0_Stop(void)
	r_cg_tmr_user.c	TMR0	static void r_tmr_cmia0_interrupt(void)
			static void r_tmr_ovi0_interrupt(void)
	r_cg_tmr.h		-

12-Bit A/D Converter	r_cg_s12ad.c		void R_S12AD_Create(void)
			void R_S12AD_Start(void)
			void R_S12AD_Stop(void)
			void R_S12AD_Get_ValueResult(ad_channel_t channel, uint16_t * const buffer)
	r_cg_s12ad_user.c		static void r_s12ad_interrupt(void)
	r_cg_s12ad.h		-

For details, refer to the following files.

- Function.html

  It is stored in the “an-r11an0326ej0100-bsspf-apl/workspace/HSM/doc” folder
- Macro.html

  It is stored in the “an-r11an0326ej0100-bsspf-apl/workspace/HSM/doc” folder

5.4 Application Framework

The application framework generated by Code Generator is shown in the flow chart below:

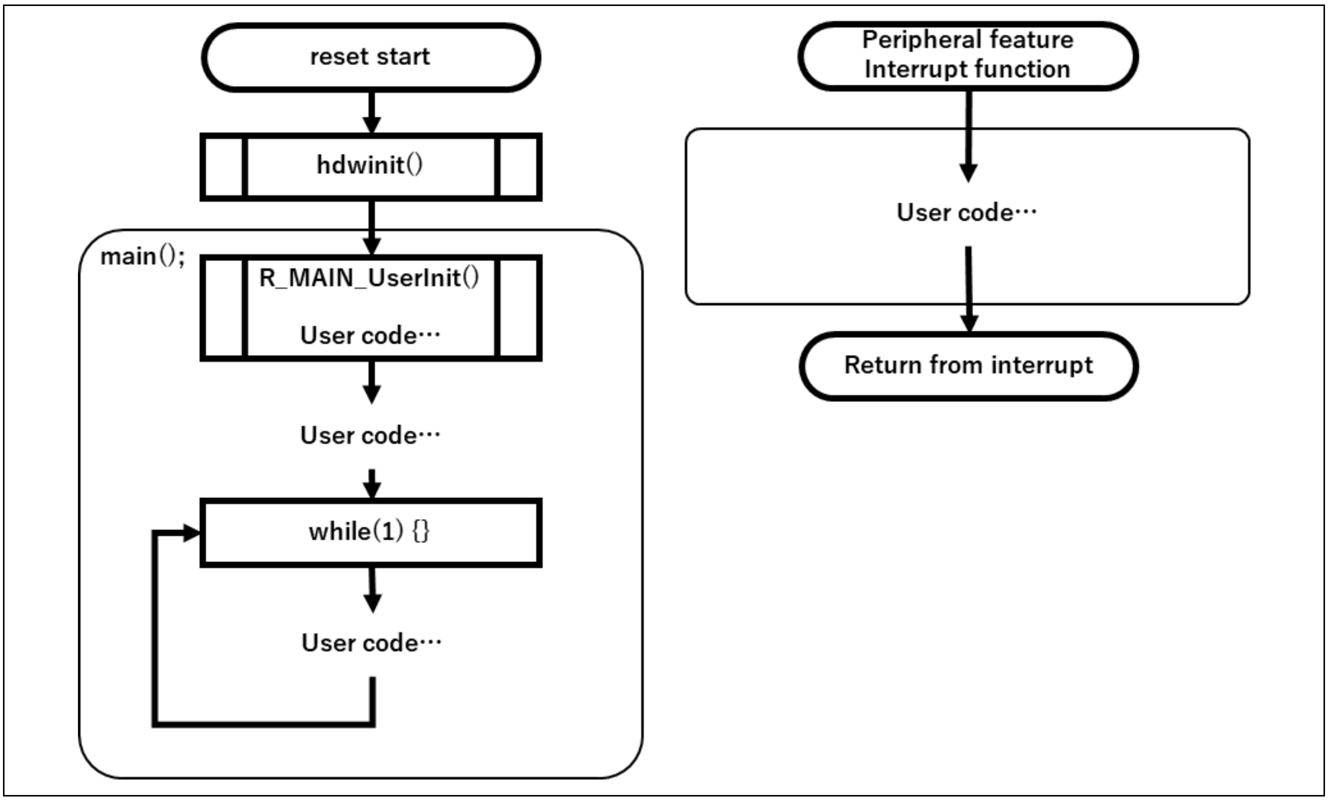

Figure 5-6 Application Framework by the Code Generator

The application program is constructed on this framework generated by the Code Generator.

In the main routine, all the peripherals of the MCU will be initialized before the main () function is executed. The hdwinit () is automatically invoked from the compiler on reset and all the peripherals are initialized according to the Code Generator Configuration.

The main () function will have three sections:

1. R_MAIN_UserInit () – This function is invoked in the beginning of main () function. User code to initialize all the devices outside the MCU is implemented in R_MAIN_UseInit () function.

2. User Code Section - User code to start each of the device operation is implemented here.

3. While loop – main () function in the application framework will have an infinite while loop. User code can be implemented in this loop.

The Code Generator generates Peripheral Interrupt Framework for each of the Interrupt configured in the Code Generator. The user code to handle each of the enabled interrupts shall be implemented in the respective interrupt template generated by the code generator.

5.5 Application Flow

The application flow is shown in the flow chart below:

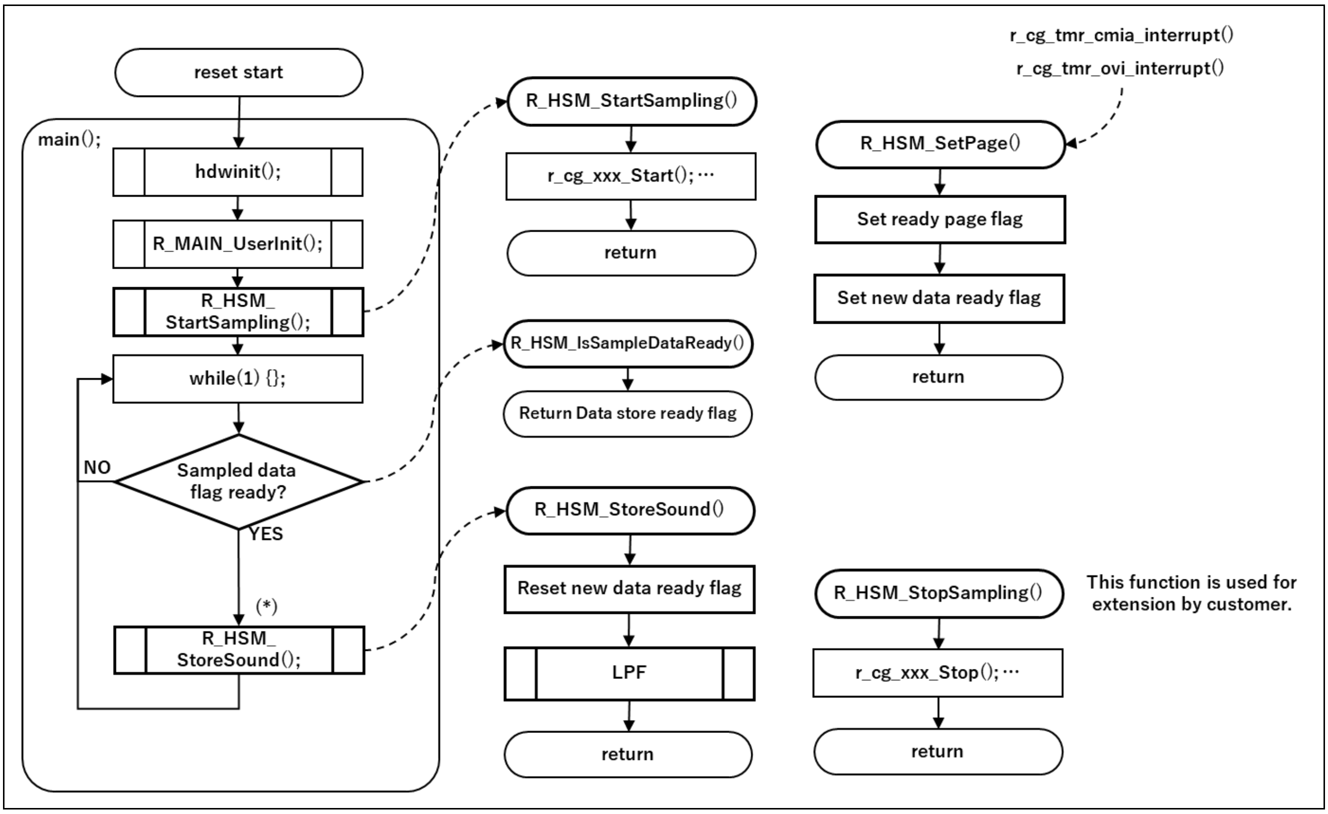

Figure 5-7 Software Flow Chart

The application does the below operations:

- Initialize the MCU and Peripherals on reset.
- Start sampling the microphone data.
- Do below operations repeatedly

    Wait for 128 samples

Store the first 128 samples in Page 0 and next 128 samples in Page 1 of the buffer.

5.6 File Configurations

Figure 5-8 and Figure 5-9 show the file structure.

```
an-r11an0326ej0101-bsspf-apl
|     r11an0326ej0101-bsspf.pdf                   : This application note
|
+--- workspace                                    : Workspace folder
      |
      +--- HSM                                    : Project folder
            |     .cproject                       : ProjectDescription
            |     .project                        : ProjectDescription
            |     SM HardwareDebug.launch         : Launch Configuration
            |
            +--- .settings                        : Configuration folder of e2studio (Omit details)
            |
            +--- demo
            |         r11an0326ej0101-bsspf-hsm.zip   : Archived file of this project
            |
            +--- doc
            |     |     Function.html             : Function Table file for CG
            |     |     Macro.html                : Macro Table file for CG
            |     |
            |     +--- Macro.files                : Macro.files folder (Omit details)
            |
            +--- generate                         : generate folder
            |         iodefine.h                  : IO definition file
            |
            +--- src                              : Source folder
                  |
                  +--- cg_src                     : CG Source folder
                  |         r_cg_cgc.c
                  |         r_cg_cgc.h
                  |         r_cg_cgc_user.c
                  |         r_cg_dbsct.c
                  |         r_cg_dtc.c
                  |         r_cg_dtc.c
                  |         r_cg_dtc_user.c
                  |         r_cg_elc.c
                  |         r_cg_elc.h
                  |         r_cg_elc_user.c
                  |         r_cg_hardware_setup.c
                  |         r_cg_intprg.c
                  |         r_cg_macrodriver.h
                  |         r_cg_main.c
                  |         r_cg_mtu2.c
                  |         r_cg_mtu2.h
                  |         r_cg_mtu2_user.c
                  |         r_cg_resetprg.c
```

Figure 5-8 File Structure (1/2)

```
|           r_cg_s12ad.c
|           r_cg_s12ad.h
|           r_cg_s12ad_user.c
|           r_cg_sbrk.c
|           r_cg_sbrk.h
|           r_cg_stacksct.c
|           r_cg_tmr.sc
|           r_cg_tmr.h
|           r_cg_tmr_user.c
|           r_cg_userdefine.h
|           r_cg_vect.h
|           r_cg_vecttbl.h
|
+--- r_hsm_src                                       : HSM Source folder
     |      r_hsm_if.h
     |
     +--- src
                 r_hsm.c
                 r_hsm_cgw.h
                 r_hsm_driver.c
                 r_hsm_driver.h
                 r_hsm_private.h
```

Figure 5-9 File Structure (2/2)

5.7 System Requirement

The following are required for the execution of the sample project:

- e2studio version: 6.2.0 or above
- RX Family C/C++Compiler Package version: CC-RX 2.07.00 or later
- Language Configuration: C(C99) (-lang=c99)

5.8 Procedure to Execute the Sample Application

The following describes the steps to build and execute the sample application.

(1) **Import the sample project into e2studio workspace by clicking "Import" from File Menu.**

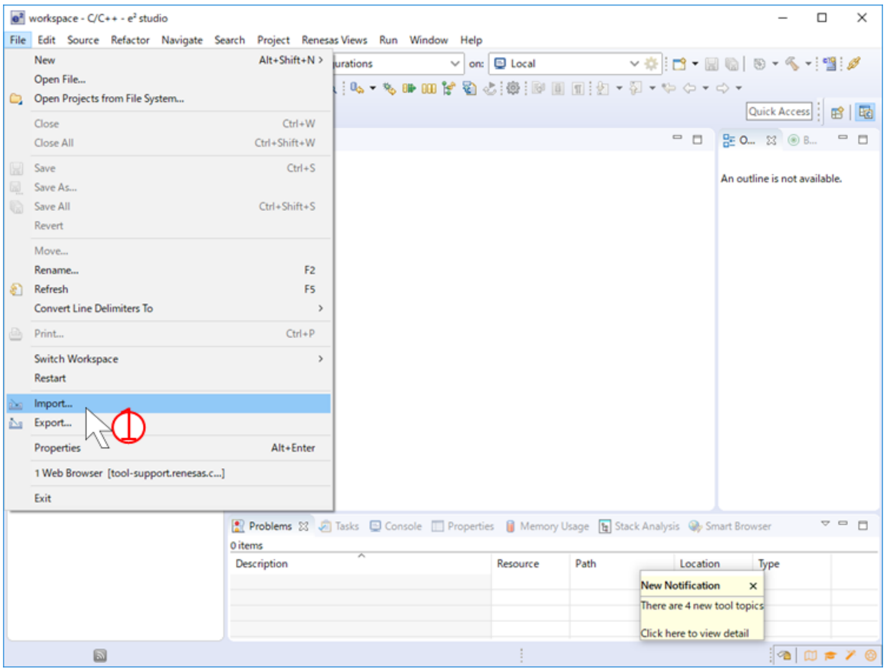

Figure 5-10 Select "Import" Menu

(2) **Select "Existing Projects into Workspace".**

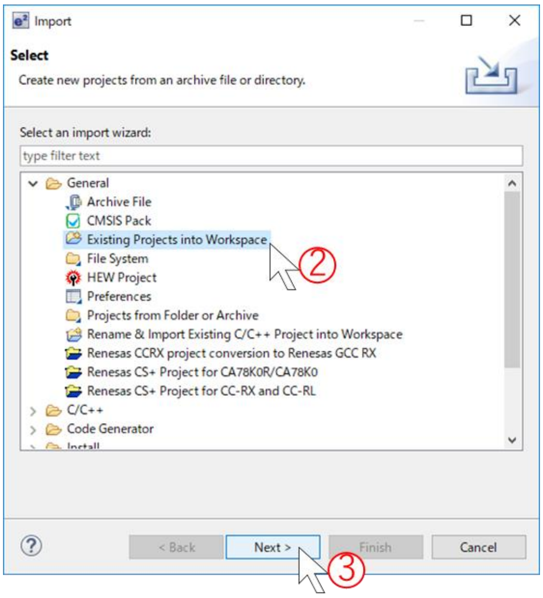

Figure 5-11 Select "Existing Projects into Workspace"

(3) **In the next window, choose "Select archive file:" and browse to the directory of "r11an0326ej0101-bsspf-hsm.zip".**

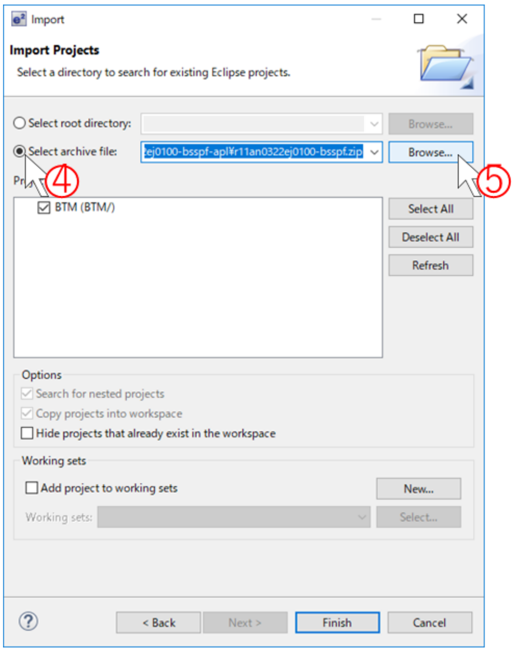

Figure 5-12 Choose "Select archive file:" and Select the Archived File

(4) **After selecting the archive file, the projects it contains will be listed down as shown. Click "Finish" to finish the importing.**

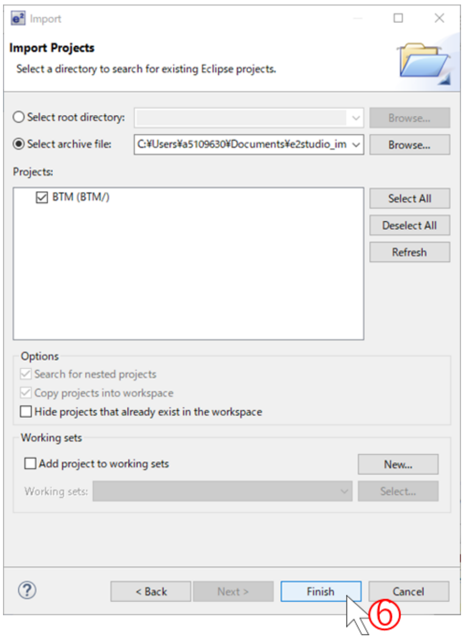

Figure 5-13 Click "Finish"

6. References

- User's Manual for RX231:

  The latest version can be downloaded from the Renesas Electronics website.
- User's Manual for Renesas Starter Kit for RX231 (R0K505231S020BE):

  The latest version 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	Jul 31, 2018	-	1st Released
1.01	Sep 14, 2018	21 to 22	Changed “5.6 File Configurations”.
		22	Added “5.7 System Requirement” and “5.8 Import procedure”.
		23 to 24	Updated SALES OFFICE page.

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 Corporation
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

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

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 7.2