Bio Sensing Software Platform

Software Library for Measuring Body Surface Temperature

R11AN0322EJ0101
Rev.1.01
Sep 14, 2018

Abstract

This document describes the sample program to obtain the human body surface temperature using thermistor.

Target Device

RX231

SAIC101

Target Board

- Renesas Starter Kit for RX231 (R0K505231S000BE) (Renesas Electronics)

  Hereafter, it is abbreviated as RX231 RSK.
- RSK Option Board TSA-OP-IC101 (TESSERA TECHNOLOGY INC.)

  *Included Renesas Electronics SAIC101 (RAA730101)

Contents

1. Introduction

This application note describes how to measure human body surface temperature with RX231, SAIC101 and thermistor.

1.1 Terminology

Table 1-1 Terminology

Term	Meaning
ADC	Analog-to-Digital Converter
API	Application Programming Interface
BTM	Body surface Temperature Measuring
CPU	Central Processing Unit
DSP	Digital Signal Processor
FPU	Floating-Point number processing Unit
IR	Infra-Red Ray
MCU	Micro Control Unit
NTC	Negative Temperature Coefficient (thermistor)
Platform	Means a hardware and fundamental software (i.e. OS) which application software works on.
PTC	Positive Temperature Coefficient (thermistor)
RTD	Resistance Temperature Detector RTDs are sensors for temperature measurement. Usually they are constructed of platinum.

1.2 Overview

This application note answers the following topics:

- Back ground of the body surface temperature
- Principle of temperature measurement using electronic sensor and computer system.
- Sensor type for body surface temperature.
- Algorithm of converting from sensor output to the temperature value.

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: SAIC101 16-bit delta-sigma A/D converter with 4ch analog multiplexer by Renesas Electronics.

  SAIC101 is a flexibly change analog front-end settings in response to environmental changes.
- Sensor. NTC thermistor GA100K6A1I series by TE Connectivity Ltd. family of companies.

  GA100K6A1I series is an NTC (Negative Temperature Coefficient) thermistor with epoxy coating small conductive.
  - Interchangeable
  - Proven stability and reliability
  - Rapid time response
  - Temperature range: from -40 to 257 °F (-40 to +125 °C)
  - Normal resistance at 77 °F (25 °C): 100k Ω

- Beta value and tolerance 77 °F (25 °C) / 185 °F (85 °C): 4261 K ±0.5 %
- PTFE (polytetrafluoroethylene) Insulated lead wires
- Diameter of conductive head: 2.4 mm Max

2. Functional Purpose

Monitoring of body surface temperature is one of the most important clinical diagnostic measures. Body surface temperature varies by the person's age, activities performed, and time of the day. The normal body surface temperature is generally 98.6°F (37°C) except the elderly who tend to have a lower body surface temperature.

The measurement range depends on the range of body surface temperature. So, the requirement is from 25°C to 45 °C.

2.1 Introduction to Body Surface Temperature

Core body surface temperature is a constant 98.24 °F (36.8 °C). Temperature of the inner organs like the liver, kidney, and heart is called the body's core temperature. The core temperature is measured internally (e.g. in the rectum or esophagus) and it usually remains within a narrow range, between 9.77 °F (36.5 °C) and 99.5 °F (37.5 °C).

At room temperature, the temperature of human body surface varies in different regions of the body. It is lower at extremities and markedly curved parts, such as the nose, ears, fingers and toes. The body surface temperature is higher at muscles than at bone or tendons.

2.2 Response and Measurement Interval Time

The change of body surface temperature is so slowly. Then 1minute response time is enough specification. And measurement interval is also 1 minute.

If more quick response and short interval time are needed, the system should implement some prediction mechanism.

3. Measurement Principle

3.1 Thermistor Analog Block

The body surface temperature measurement circuit design is based on the thermistor principle. It uses the fundamental properties of a thermistor to measure resistance which varies proportional to body surface temperature.

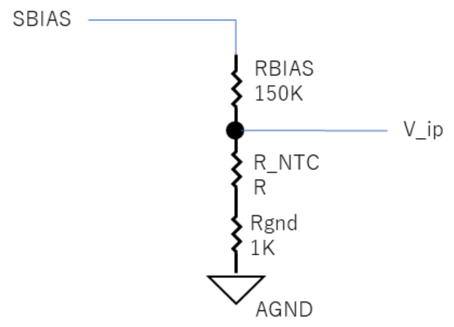

Figure 3-1 Body Surface Temperature Measurement Circuit

The circuit has the following components:

- R_NTC - NTC100K thermistor
- Bias Resistors – bias resistor values are chosen according to the SAIC101 input voltage limits

3.2 Body Surface Temperature Sensor - GA100K6A1IA

GA100K6A1IA is a NTC100K thermistor which is used to measure the body surface temperature. The properties of the thermistor are given below:

- R (25 °C): 100k Ω
- B (25/85 °C): 4261 K

The resistance values for different temperatures are calculated using the following equation:

$$R(T) = R_{25} \cdot e^{\left(B_{25/100}\left(\frac{1}{T} - \frac{1}{T_{25}}\right)\right)} \quad [\Omega]$$

Where,

T = Temperature of Interest in K
T25 = 298.15 K

The resistance values for the thermistor GA100K6A1IA are provided by the component datasheet. The voltage V_ip values are calculated by using the potentiometer shown in Figure 3-1 and tabulated in the below table.

Table 3-1 Body Surface Temperature Measurement Techniques

Body Surface Temperature Sensor - GA100K6A1IA **B(25/85 [°C]) = 4261 [K], R(25 [°C]) = 100.000 [kΩ]** SBAIS 2.200 [V], RBIAS 150k [Ω], Rgnd 1.00 [kΩ]		
T [°C]	**R_NTC [kΩ]**	**V_ip [V]**
45	40.686	0.478
40	50.397	0.561
35	62.861	0.657
30	78.982	0.765
25	100.000	0.885
20	127.635	1.016
15	164.293	1.153

The graph for the above tabulated data is shown below:

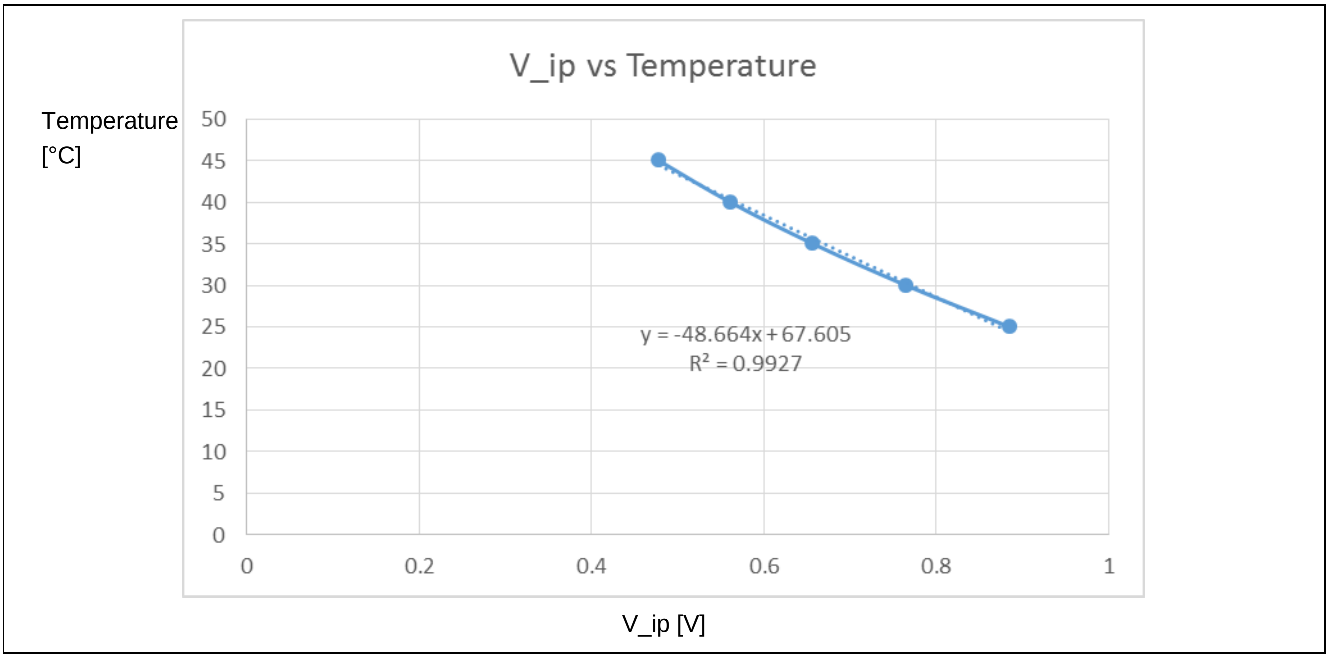

Figure 3-2 Body Surface Temperature Measurement Circuit

The relationship between the temperature T and the input voltage to the SAIC101 V_ip for the temperature range of interest can be approximated to the following linear equation from the above graph:

T (°C) = (-48.664) * V_ip + 67.605

This equation has a R2 (regression) value of 0.9927, which has a 0.3% non-linearity error.

3.3 A/D Converter (ADC)

SAIC101 has two analog input-modes. One is “single-end mode” and the other is “Differential mode”. In this system SAIC101 is set to single-end mode.

This ADC has also a programable analog amplifiers. But Renesas recommends that SAIC101 should be used as gain = x1 in the single-end mode.

Figure 3-3 shows the relation of digital output (AD conversion result) and analog input (voltage input to the ADC).

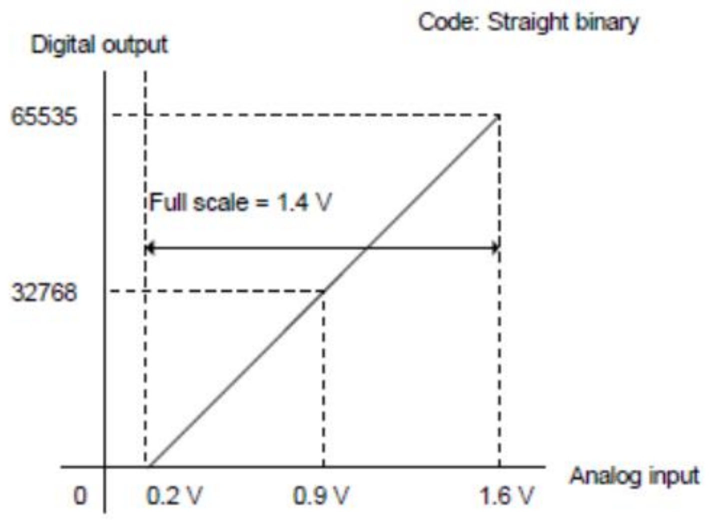

Figure 3-3 Thermistor Circuit

The value of RBIAS and RGND is influenced by this minimum and maximum input voltage.

From this figure, the equation of conversion is:

Vin = (1.6 – 0.2) / 65536 x Dout + 0.2

Note: “65536” shows the maximum resolution of 16-bit ADC.

So,

Vin = 1.4 / 65536 x Dout + 0.2

Measurement Value

The Body Surface Temperature Module measures body surface temperature and calculates the temperature in Celsius and Fahrenheit.

The body surface temperature values will be stored in double float variables.

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_BTM_StartSampling(void)	Starts the operation of MCU peripheral devices and initializes the body surface temperature values (both °C and °F) to 0.
R_BTM_StopSampling(void)	Stops the MCU peripheral devices' operation.
R_BTM_Calculate(void)	Reads the ADC sample values from the buffer, resets the data ready flag, calculates average value of all the samples, calculates the body surface temperature, and stores the values in the respective global variables.
R_BTM_IsSampleDataReady(void)	Returns the SampleDataReady flag. If the flag is set, the function returns true, otherwise returns false.
R_BTM_ContinueSampling(void)	Continues data sampling.

4.2 Global Variables

The body surface temperature values are stored in the below global variables:

Table 4-2 List of Global Variables

Global Variables	Function
g_body_temp_celsius	floating point value indicating the body surface temperature in degree Celsius [°C]
g_body_temp_fahrenheit	floating point value indicating the body surface temperature in degree Fahrenheit [°F]
g_samples_ready	Flag which indicates sampled data for calculation was ready or not

4.3 Memory Size

Memory size is shown in the below table:

Table 4-3 Memory Size

Memory	Size
ROM	1,024 bytes for BTM library [Note1]
RAM	12 bytes for BTM Library [Note1]
User Stack	264 bytes
Interrupt Stack	48 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 BTM 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.
- SAIC101 Starter Kit is used as Analog Front End which has high performance 16-bit delta-sigma ADCs with programmable gain instrumentation amplifier ideal for differential input sensors. The RAA730101 used in the SAIC101 starter kit uses a 36-pin FBGA package, which enables a more compact set design. It has 256-byte flash memory for storing system configuration data. Also, using the serial communication (SPI or UART communication, which is selectable) each of the function blocks can be controlled from an external device and the measured data can be output to the external device. All these features provide the user flexibility and control to configure sensors for measurement.

5.1.2 Hardware Block Diagram

The BTM will be developed to run on RX231 RSK and TSA-OP-IC101.

Thermistor GA100K6A1IA is assumed as measurement device.

The Block Diagram for the overall system is shown below:

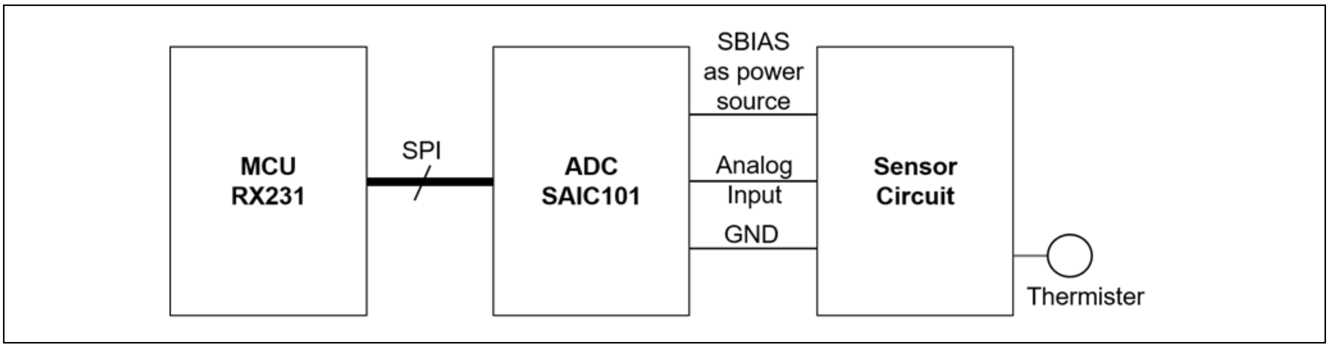

Figure 5-1 Hardware Block Diagram

The signal interconnects between RX231 RSK and TSA-OP-IC101 is shown in the figure below:

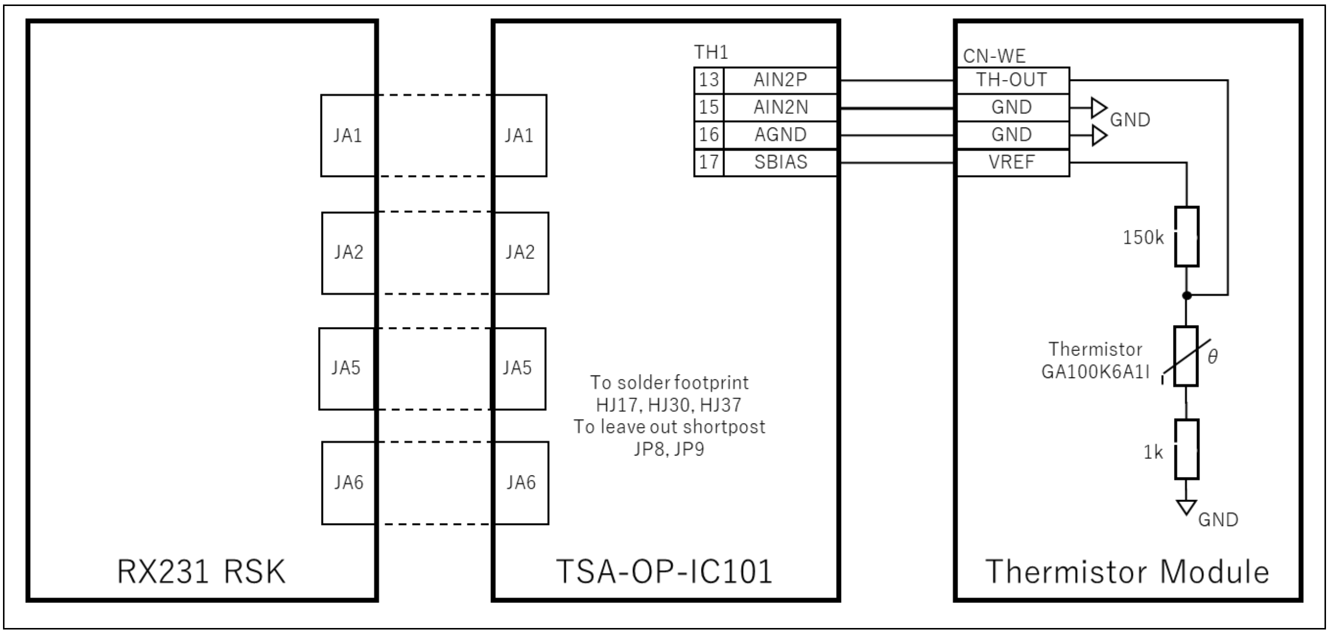

Figure 5-2 Hardware Wiring Chart

5.1.3 RX231 MCU Digital Interconnect

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

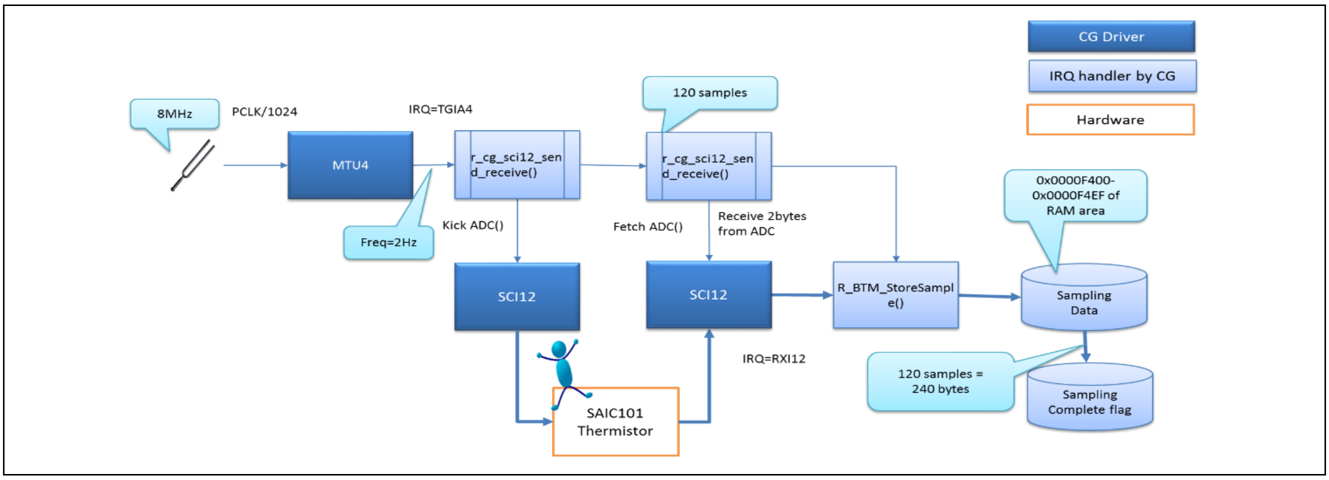

Figure 5-3 BTM Digital Interconnect

- The MTU will be configured to trigger an interrupt every 0.5 second.
- The MTU interrupt triggers SAIC101 AD Conversion.
- On ADC completion, read the 16-bit ADC value over SCI12 and store the read value to RAM area.
- This data sampling is continued for 120 cycles, on acquiring 120 samples the sampling data ready flag is set and body surface temperature calculated out of these 120 values.
- With MTU configured to trigger AD Conversion every 0.5 second and calculate body surface temperature on every 120 samples, body surface temperature will be calculated once every minute

5.2 Software

5.2.1 Software Design Policy

The Software Design Policy is:

- BTM 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 Body Surface Temperature Data Sampling is automatically started on software initialization, immediately after all the peripherals are initialized and configured
- The sensor data is sampled continuously unless the system is powered off. The sampling can be stopped temporarily during the calculation, if needed to get better accuracy
- The data sampling and calculation of the body surface temperature is executed continuously using an infinite loop (while(1U) {}) in the main() function
- The calculate() function calculates body surface temperature, both in degree Celsius and degree Fahrenheit and stores the calculated values in global variables
- The measured values are not displayed on any display or stored in any file
- The software does not include power management
- The BTM 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.

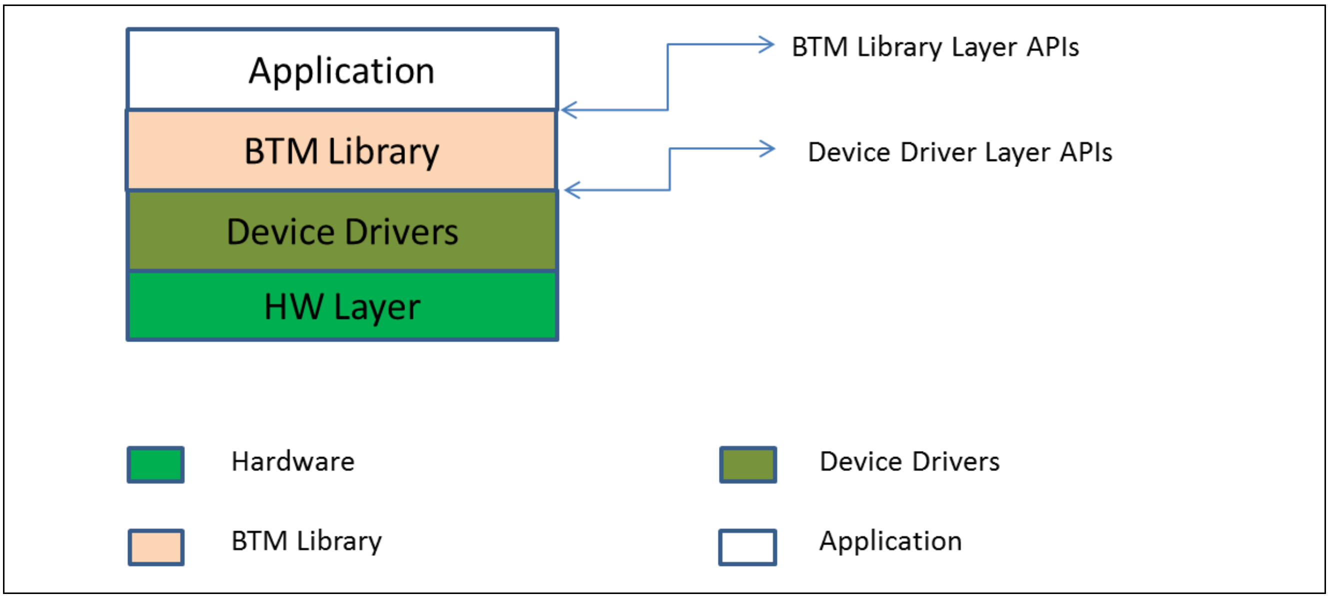

Figure 5-4 BTM Software Architecture

5.2.3 Measurement Signal Flow

This diagram shows the flow of SAIC101 ADC samples from channel 2 to RX231 application. The temperature value will be calculated from these ADC samples by the RX231 application.

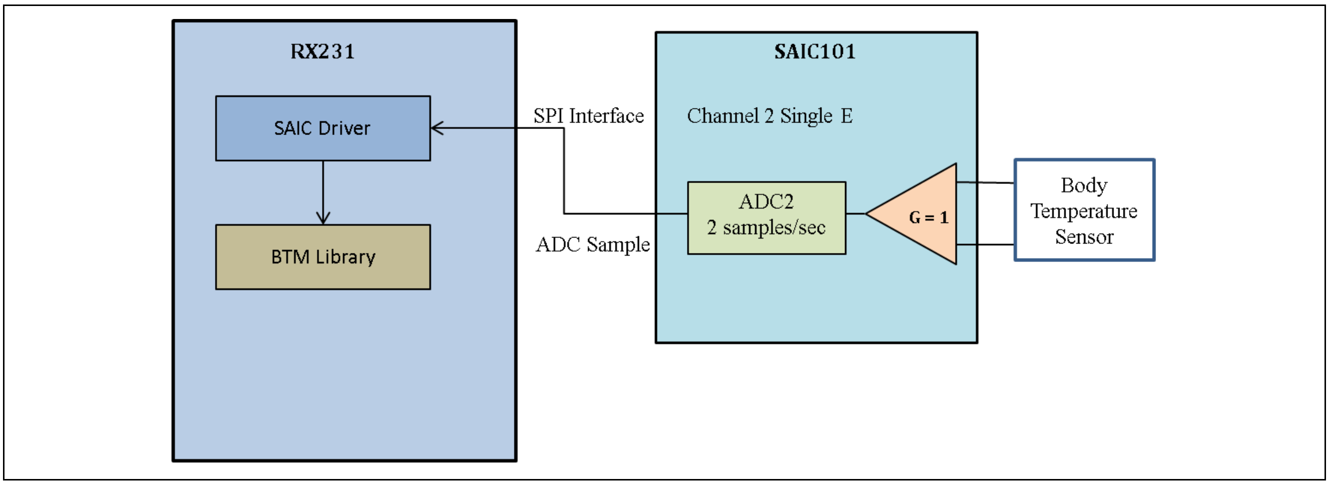

Figure 5-5 Interfacing of the Body Surface Temperature Sensor with the MCU via SAIC101

5.2.4 Measurement Control Flow

The BTM Software implements a simple application to use the BTM Library. The application is started by pressing start measurement. The Body Surface Temperature data is calculated, and values are stored. The Control Flow of the main application is shown in the flow chart below:

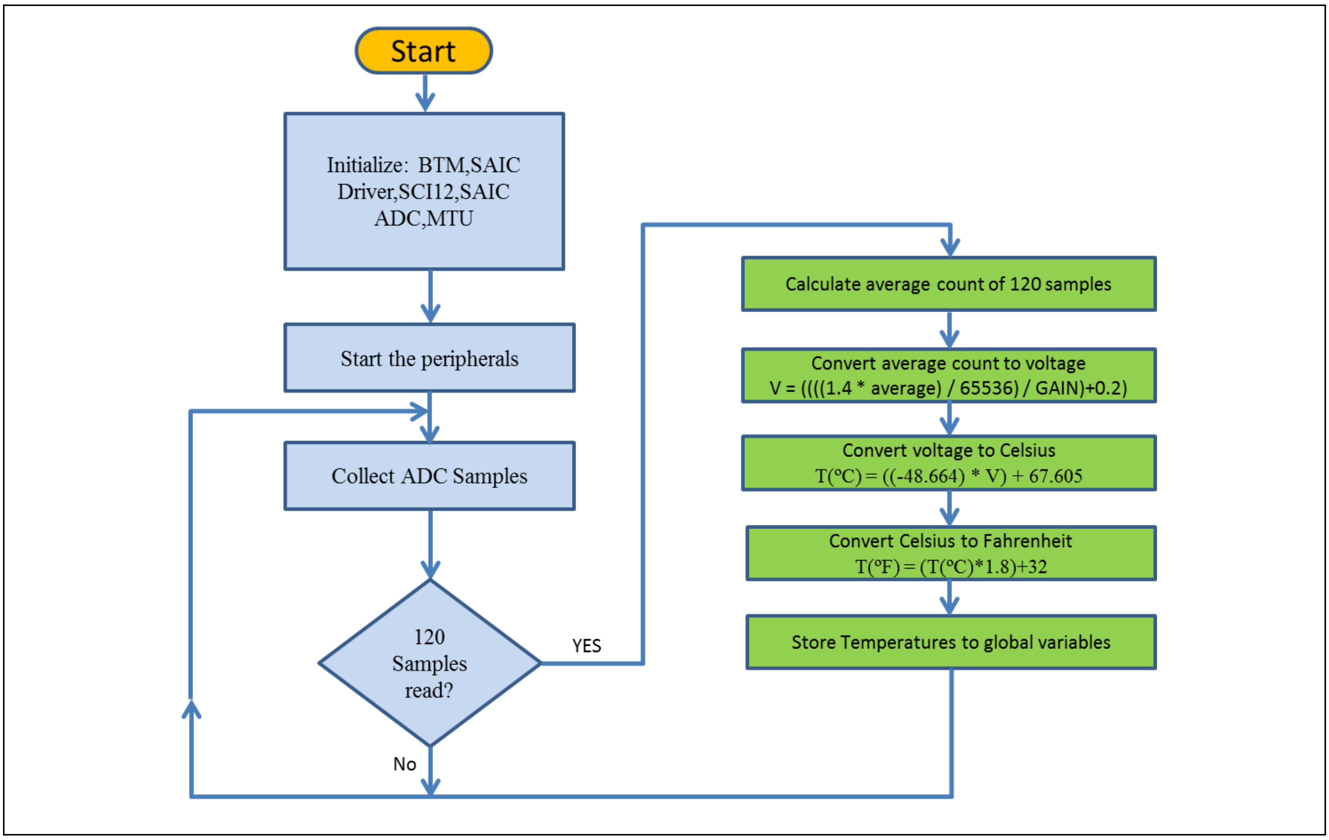

Figure 5-6 BTM Software Control Flow

5.3 Device Drivers

5.3.1 SAIC driver

(1) **Driver Function List**

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

Table 5-1 Driver Function List

Functional Name	Function
R_SAIC101_Initialize(void)	Initializes and sets the input mode, gain, offset, conversion rate, and interrupt setting of the SAIC101.
R_SAIC101_Done(void)	Sets the sampling data ready flag.
R_SAIC101_Kick(void)	Triggers SAIC101 A/D conversion.
R_SAIC101_Fetch(void)	Reads SAIC ADC result. The ADC result is 2 bytes (the lower and upper 8 bits of the 16-bit data) in length.
R_SAIC101_EndRTx(void)	This function disables the SAIC101.

(2) **Conditions**

The temperature reading is obtained by the samples taken from SAIC101 ADC channel via SPI communication between RX231 and SAIC101.

SAIC ADC configuration used is:

ADC ON_OFF = ON
ADC Input mode = Single ended input mode
ADC offset = 0 mV (DC offset)
ADC Over sampling rate = 64 (15625.00 [samples per sec])
ADC Gain = 1 (GSET1 = x1, GSET2 = x1, Total = x 1)
ADC count = 2 (Number of A/D conversions: 2)
ADC Sbias_voltage = 2.2 V

Using below formulas we get the samples collected from SAIC converted to degree Celsius:

- Conversion of average into voltage_value for a single ended SAIC Mode by using below formula

  voltage_value = (((1.4*(average_value))/65536) + 0.2)

  Where,

  SAIC ADC full scale voltage = 1.4
  SAIC ADC offset voltage = 0.2
  average_value = average count of 120 samples collected in one minute
  SAIC ADC 16-bit resolution value = 65536

- Calculation of body surface temperature in degree Celsius using below formula

  body_temp_celsius = ((-48.664) * voltage_value) + 67.605

  Where,

  Compensation constant = 67.605
  Default slope value = -48.664

- Convert the body surface temperature in degree Celsius to degree Fahrenheit using the below formula

  body_temp_fahrenheit = (body_temp_celsius * 1.8) + 32

5.3.2 Code Generator

Table 5-2 shows the used peripheral function.

Table 5-3 shows the functions generated by Code Generator.

Table 5-2 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)
I/O Ports	Port4	P46	Mode	Out
			output value	1
Multi-Function Timer Pulse Unit 2	MTU2_U0	MTU4	Counter clock selection	PCLK/1024
			Clock edge setting	Rising edge
			Counter clear source	TGRA4 compare match/input capture (Use TGRA4 as a cycle register)
			TGRA4 (Output compare register)	500ms, (Actual value: 499.968)
			TGRB4 (Output compare register)	100ms, (Actual value: 99.968)
			TGRC4 (Output compare register)	100ms, (Actual value: 99.968)
			TGRD4 (Output compare register)	100ms, (Actual value: 99.968)
			MTIOC4A pin (PA0)	Initial output of MTIOC4A pin is 0. Toggle output at compare match.
			MTIOC4B pin (P30)	MTIOC4B pin output disabled
			MTIOC4C pin (PB1)	MTIOC4C pin output disabled
			MTIOC4D pin (P55)	MTIOC4D pin output disabled
			Enable TGRA4 input capture/compare match interrupt (TGIA4)	Used
			(TGIA/TGIB/TGIC/TGID) Priority	Level 15 (highest)

Serial Communications Interface	SCI12		Function setting	Simple SPI bus (Master transmit/receive)
			SMOSI12	PE1
			SMISO12	PE2
			Transfer direction setting	MSB-first
			Data inversion setting	Normal
			Transfer clock	Internal clock
			Bit rate	100000 (bps)
			SCK12 pin function selection	Clock output
			SCK12	PE0
			Clock delay	Clock is not delayed
			Transmit data handling	Data handled in interrupt service routine
			Receive data handling	Data handled in interrupt service routine
			TXI12, RXI12, TEI12, ERI12 priority	Level 15 (highest)
			Enable error interrupt (ERI12)	Used
			Transmission end	Used
			Reception end	Used
			Reception error	Used

Table 5-3 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		-
I/O Ports	r_cg_port.c		void R_PORT_Create(void)
	r_cg_port_user.c		-
	r_cg_port.h		-
Multifunction timer pulse unit 2	r_cg_mtu2.c		void R_MTU2_Create(void)
		MTU4	void R_MTU2_C4_Start(void)
			void R_MTU2_C4_Stop(void)
	r_cg_mtu2_user.c	MTU4	static void r_mtu2_tgia4_interrupt(void)
	r_cg_mtu2.h		-
Serial Communication Interface	r_cg_sci.c	SCI12	void R_SCI12_Create(void)
			void R_SCI12_Start(void)
			void R_SCI12_Stop(void)
			MD_STATUS R_SCI12_SPI_Master_Send_Receive(uint8_t * const tx_buf, uint16_t tx_num, uint8_t * const rx_buf, uint16_t rx_num)
	r_cg_sci_user.c	SCI12	static void r_sci12_transmit_interrupt(void)
			static void r_sci12_transmitend_interrupt(void)
			static void r_sci12_receive_interrupt(void)
			static void r_sci12_receiveerror_interrupt(void)
			void r_sci12_callback_transmitend(void)
			void r_sci12_callback_receiveend(void)
			void r_sci12_callback_receiveerror(void)
	r_cg_sci.h		-

For details, refer to the following files.

- Function.html

  It is stored in the "an-r11an0322ej0100-bsspf-apl/workspace/BTM/doc" folder

- Macro.html

  It is stored in the “an-r11an0322ej0100-bsspf-apl/workspace/BTM/doc” folder

5.4 Application Framework

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

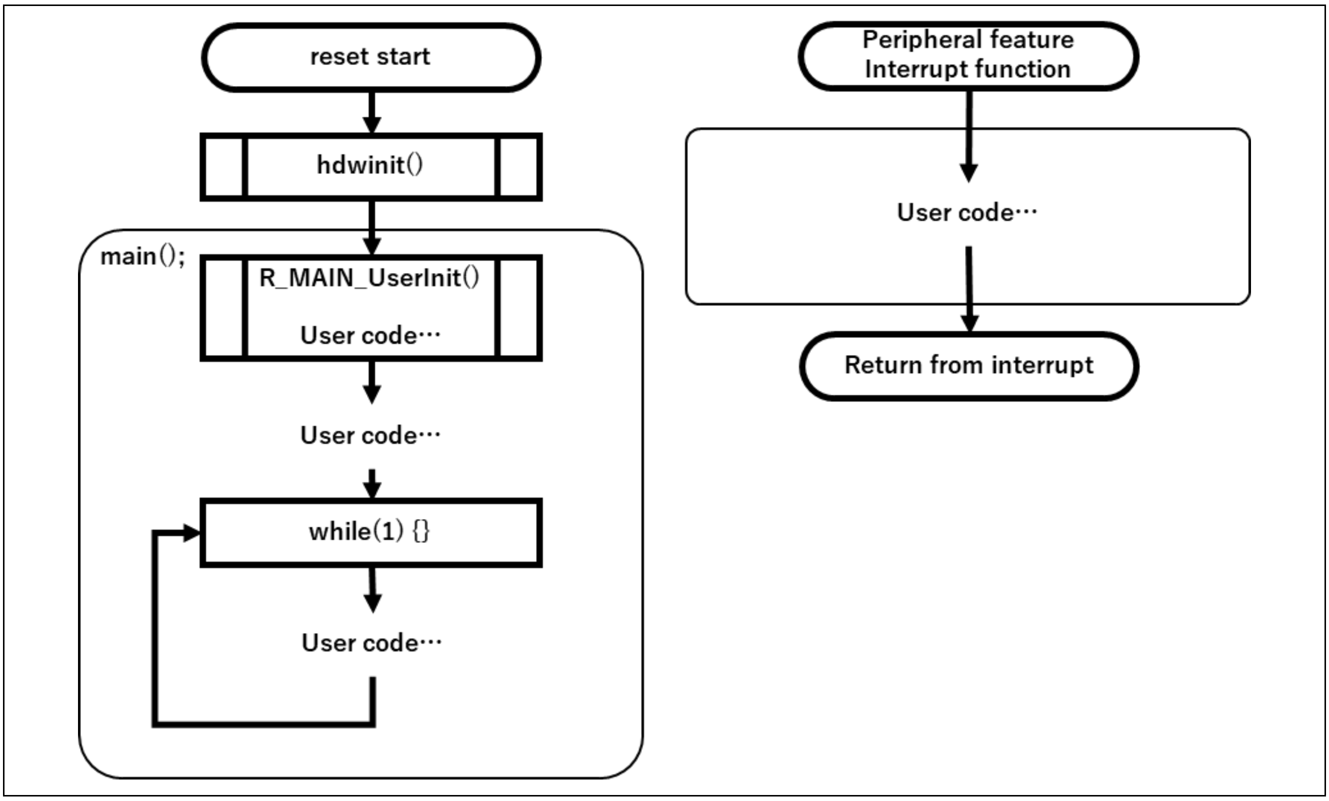

Figure 5-7 Application Framework by the Code Generator

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, such as SAIC will be implemented in R_MAIN_UseInit () function.
2. User Code Section - User code to start each of the device operation will be 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:

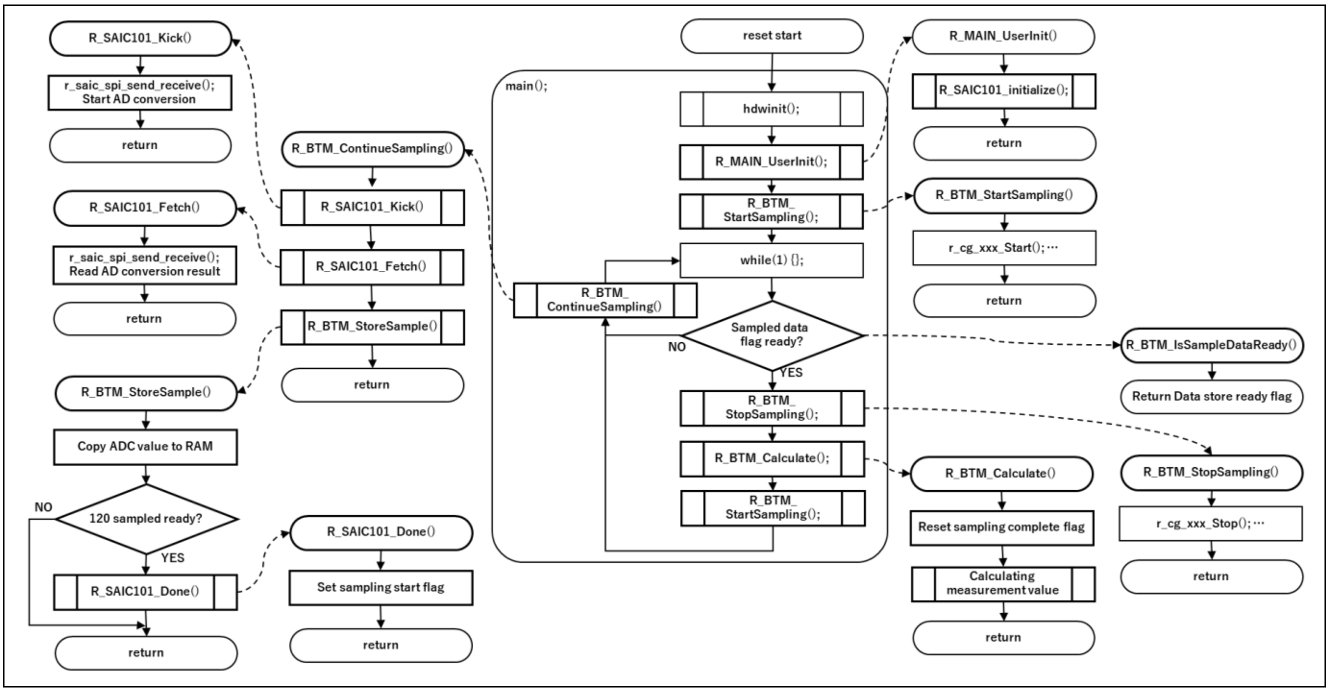

Figure 5-8 Software Flow Chart

The application does the below operations:

- Initialize the MCU, Peripherals and the SAIC on reset.
- Do below operations repeatedly

  Start Data Sampling

  Wait for required number of samples

  Stop Sampling once the required number of samples are collected, if required

  Calculate Body Surface Temperature from the collected samples

  Start Data Sampling again

5.6 File Configurations

Figure 5-9 and Figure 5-10 show the file structure.

```
an-r11an0322ej0101-bsspf-apl
|     r11an0322ej0101-bsspf.pdf                 : This application note
|
+--- workspace                                  : Workspace folder
      |
      +--- BTM                                  : Project folder
            |     .cproject                     : ProjectDescription
            |     .project                      : ProjectDescription
            |     SM HardwareDebug.launch       : Launch Configuration
            |
            +--- .settings                      : Configuration folder of e2studio (Omit details)
            |
            +--- demo
            |          r11an0322ej0101-bsspf-btm.zip   : Archived file of this project
            |
            +--- doc                            : Project folder
            |     |     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_hardware_setup.c
                  |          r_cg_icu.c
                  |          r_cg_icu.h
                  |          r_cg_icu_user.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_port.c
                  |          r_cg_port.h
                  |          r_cg_port_user.c
                  |          r_cg_resetprg.c
```

Figure 5-9 File Structure (1/2)

```
  |       r_cg_sbrk.c
  |       r_cg_sbrk.h
  |       r_cg_sci.c
  |       r_cg_sci.h
  |       r_cg_sci_user.c
  |       r_cg_stacksct.c
  |       r_cg_userdefine.h
  |       r_cg_vect.h
  |       r_cg_vecttbl.h
  |
  +--- r_btm_src                           : BTM Source folder
      |   r_btm_if.h
      |
      +--- src
              r_btm.c
              r_btm_cgw.h
              r_btm_driver.c
              r_btm_driver.h
              r_btm_private.h
```

Figure 5-10 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.**

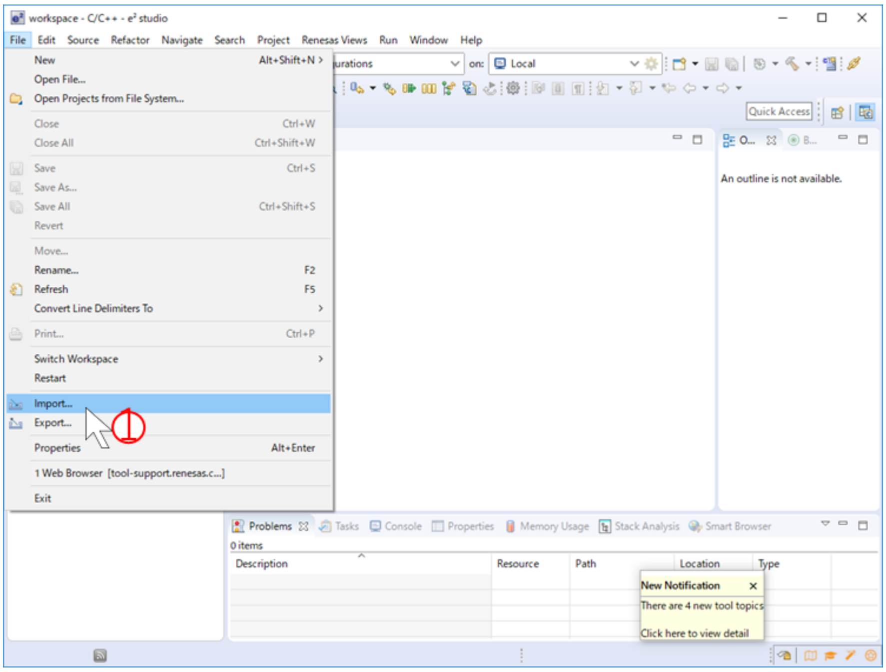

Figure 5-11 Select "Import" Menu

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

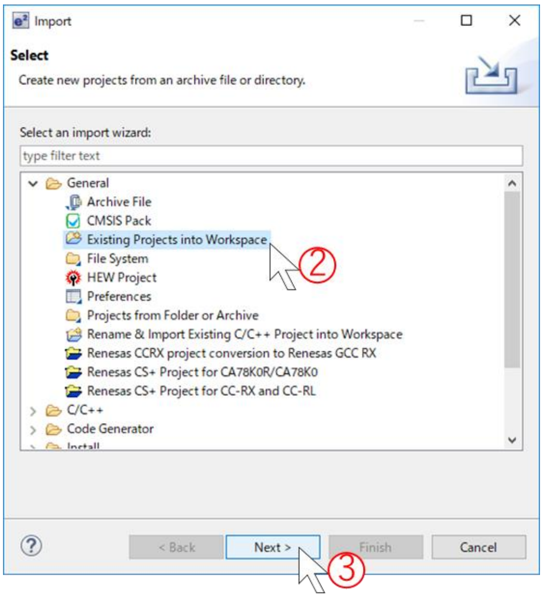

Figure 5-12 Select "Existing Projects into Workspace"

(3) **In the next window, choose "Select archive file:" and browse to the directory of "r11an0322ej0101-bsspf-btm.zip".**

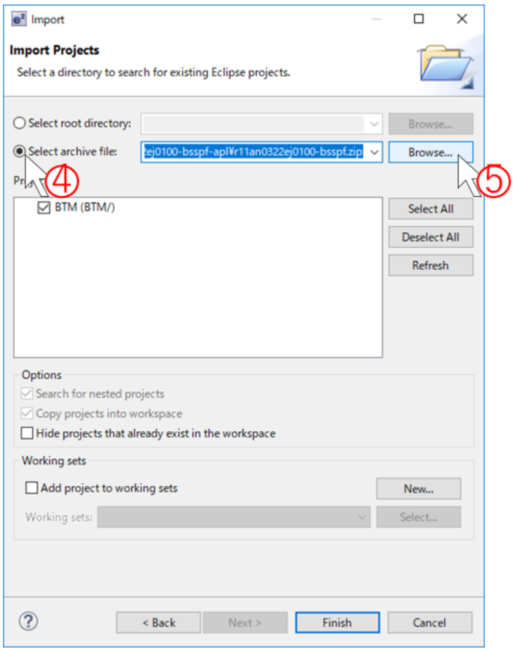

Figure 5-13 Choose "Select archive file:" and Select the Archived File

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

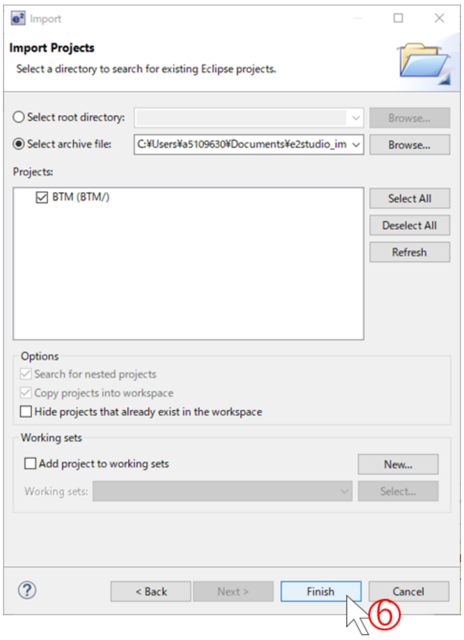

Figure 5-14 Click "Finish"

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.
- Datasheet for SAIC101 (RAA730101):

  The latest version can be downloaded from the Renesas Electronics website.
- User's Manual Datasheet for SAIC101 (RAA730101):

  https://www.tessera.co.jp/Download/TSA_OP_IC101_UM_E_V1_00.pdf
- Datasheet for GA100K6A1I:

  http://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FGA100K6A1i%7FA%7Fpdf%7FEnglish%7FENG_DS_GA100K6A1i_A.pdf%7FGA100K6A1IB

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