RX Family

Sample Program for Performing FFT on Analog Input Signals

Introduction

This application note provides the sample program to perform FFT processing with RX MCU.

The FFT is an abbreviation for Fast Fourier Transform and a signal processing used for a frequency spectrum analysis on sensor, audio signals, etc. The sample program performs A/D conversion on analog input signals and FFT processing with one RX MCU alone. The A/D conversion is performed with 12 bits resolution and approximately 1 kHz of the sampling frequency. And the FFT processing is performed with 1024 points.

Figure 1 shows an application example of the sample program. It assumes evaluation for systems working periodically or cyclically. Firstly, it performs FFT processing on the signals with a sensor attached a target system and acquires magnitudes of the frequency spectrum. Secondly, it compares the acquired spectrum with the expected spectrum and detects abnormal behaviors of target.

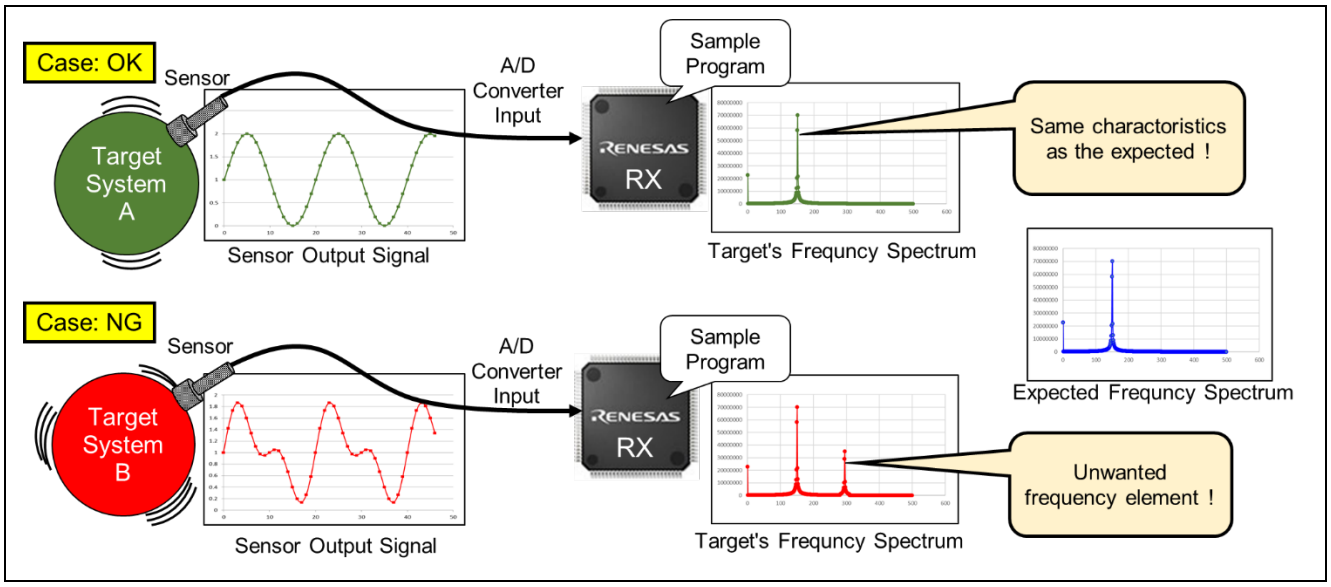


Figure 1 Application Example of the Sample Program

Target Readers

This application note describes the sample program for engineers with experience of MCU system development. The sample program uses RX DSP library APIs for the FFT processing, therefore, any special knowledge on FFT or Digital Signal Processing are not required.

Target Device

- **RX231 Group**

When applying the information in this application note to a microcontroller other than the above, modifications should be made as appropriate to match the specification of the microcontroller and careful evaluation performed.

Contents

1. Specifications

Figure 2 shows the system overview of the sample program.

The system performs the FFT processing on analog signals input to the RX231. Figure 3 shows sine waveforms input to this system. The FFT processing is performed on the input and magnitudes of the acquired frequency spectrum are shown in Figure 4 as an example.

Processing performed in the system are as follows:

- A/D conversion
  Using the 12-Bit A/D converter (S12AD), the compare match timer (CMT), and the event link controller (ELC), A/D conversion is performed with approximately 1 kHz of the sampling frequency. As shown in Figure 5, the CMT generates compare match events of approximately 1 µs period, the ELC notifies the S12AD of the events as triggers to start A/D conversion. The data after A/D conversion is transferred to the input buffer with the DMA controller (DMAC).

- FFT processing
  1024-point FFT is performed using the RX DSP library API version 5.0. The magnitudes of the acquired frequency spectrum are stored into the output buffer.

- CPU work load reducing
  The S12AD, CMT, ELC and DMAC work by themselves alone without control by the CPU after they are configured once. Using to the ability, as shown in Figure 5, the CPU operates in normal mode only when processing is performed by the software. Other than that, the CPU enters sleep mode by the WAIT instruction, and it helps reducing CPU work load. The CPU returns to normal operation mode from sleep mode by the DMA transfer end interrupt.

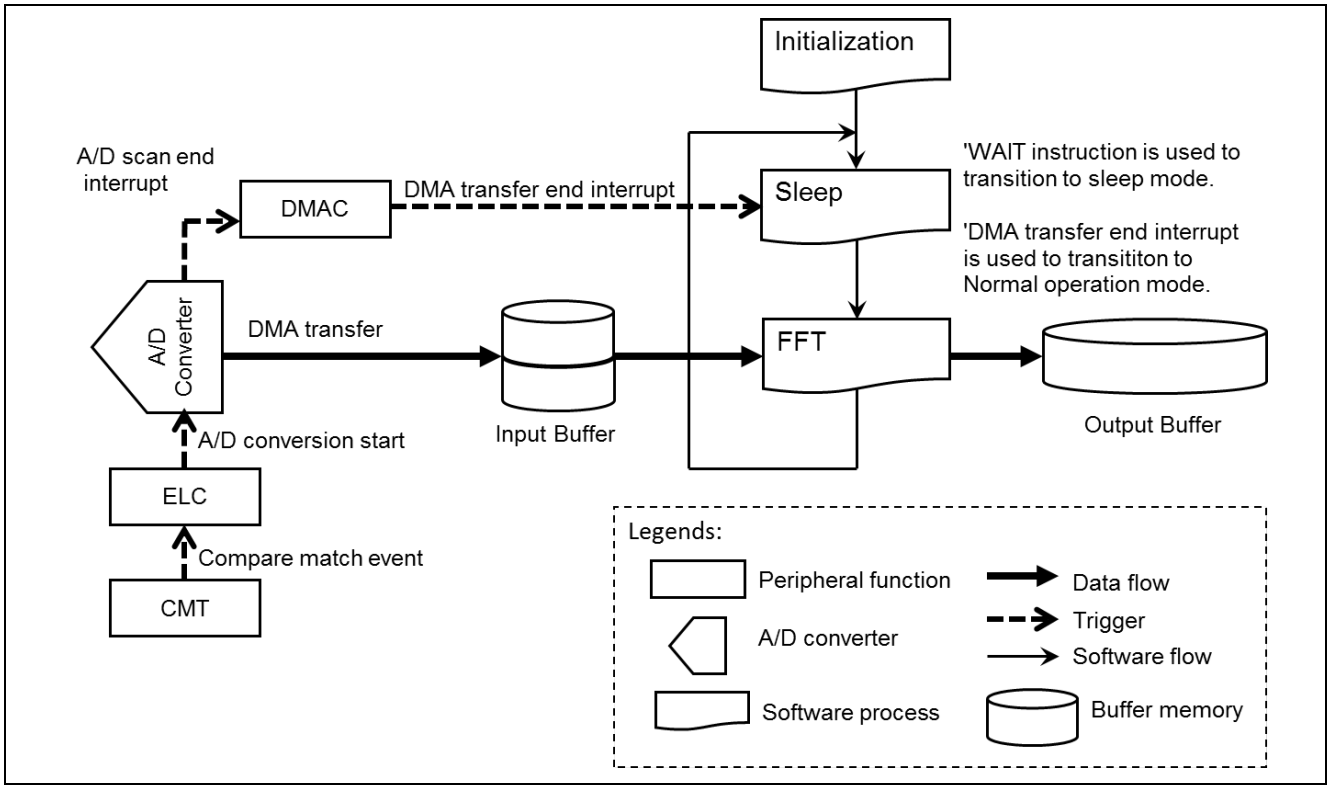


Figure 2 System Overview

From here, this document describes the sample program. For details on the peripheral function drivers or the library, refer to each application note.

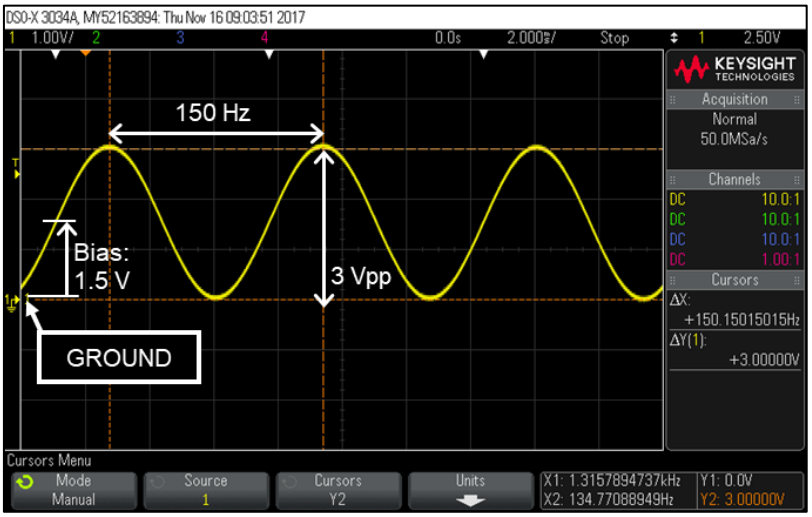


Figure 3 Input Signals (Sine wave, Frequency150 Hz, Amplitude 3Vpp)

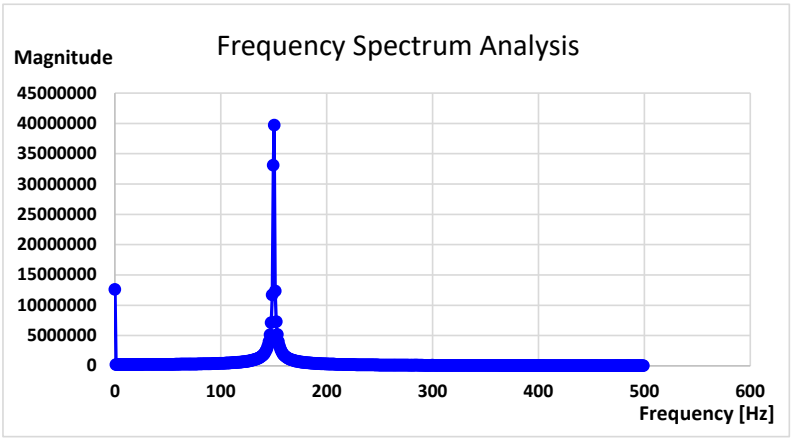


Figure 4 Example of an FFT Result Processing Result

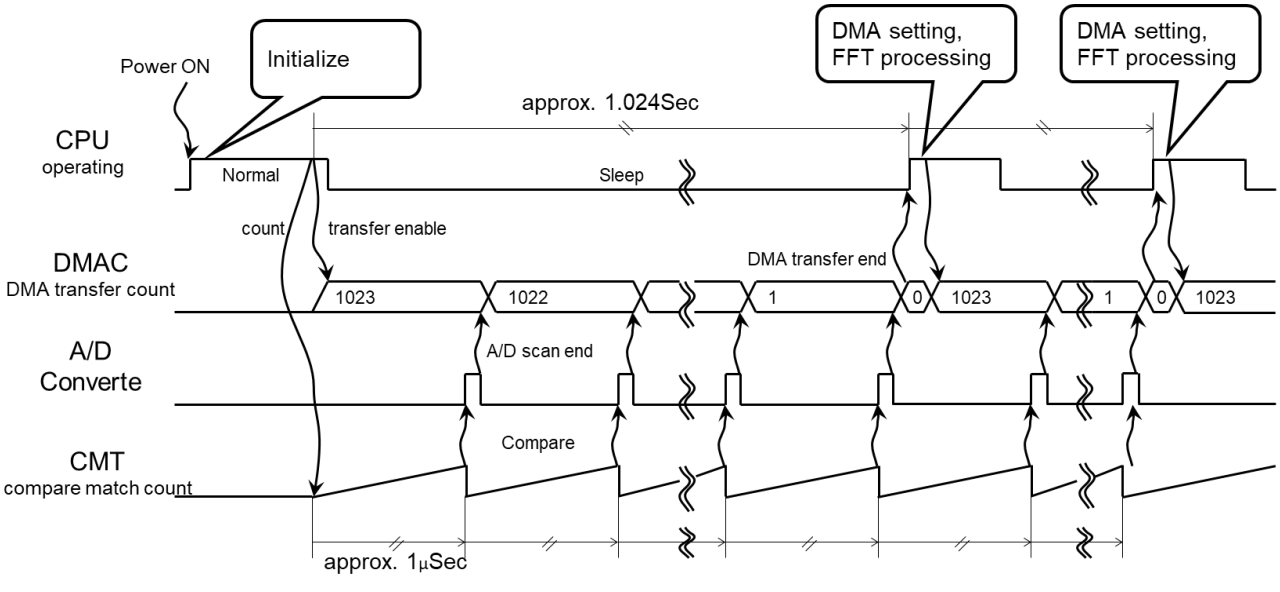


Figure 5 CPU Operation Timing Chart

1.1 Structure of the Application Note

Figure 6 shows the Structure of the Application Note. When the ZIP file provided by this application note is unzipped, the folder is created with the same name as the ZIP. "e² studio project (rx231_fft_sample)" under the unzipped folder is the e² studio project. The project folder contains the source code of the sample program, e² studio configuration file, and this application note as shown in Figure 7. This document explains these c source code shown in the figure.

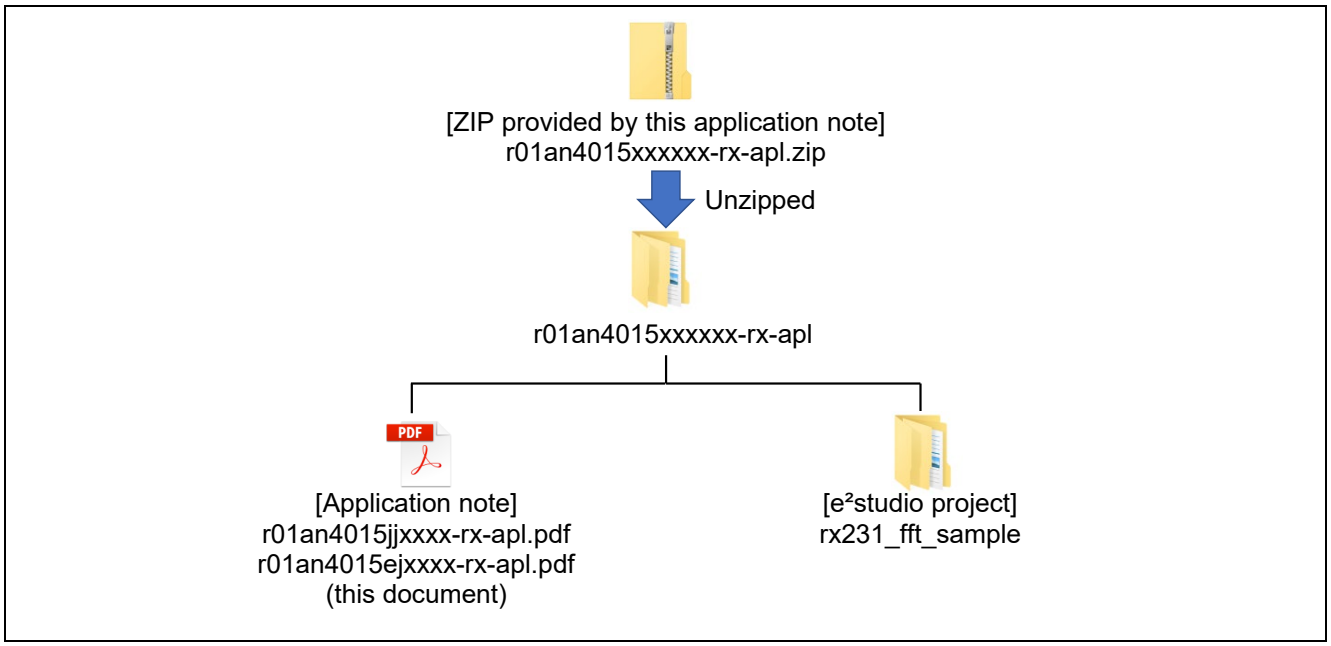


Figure 6 Structure of the Application Note

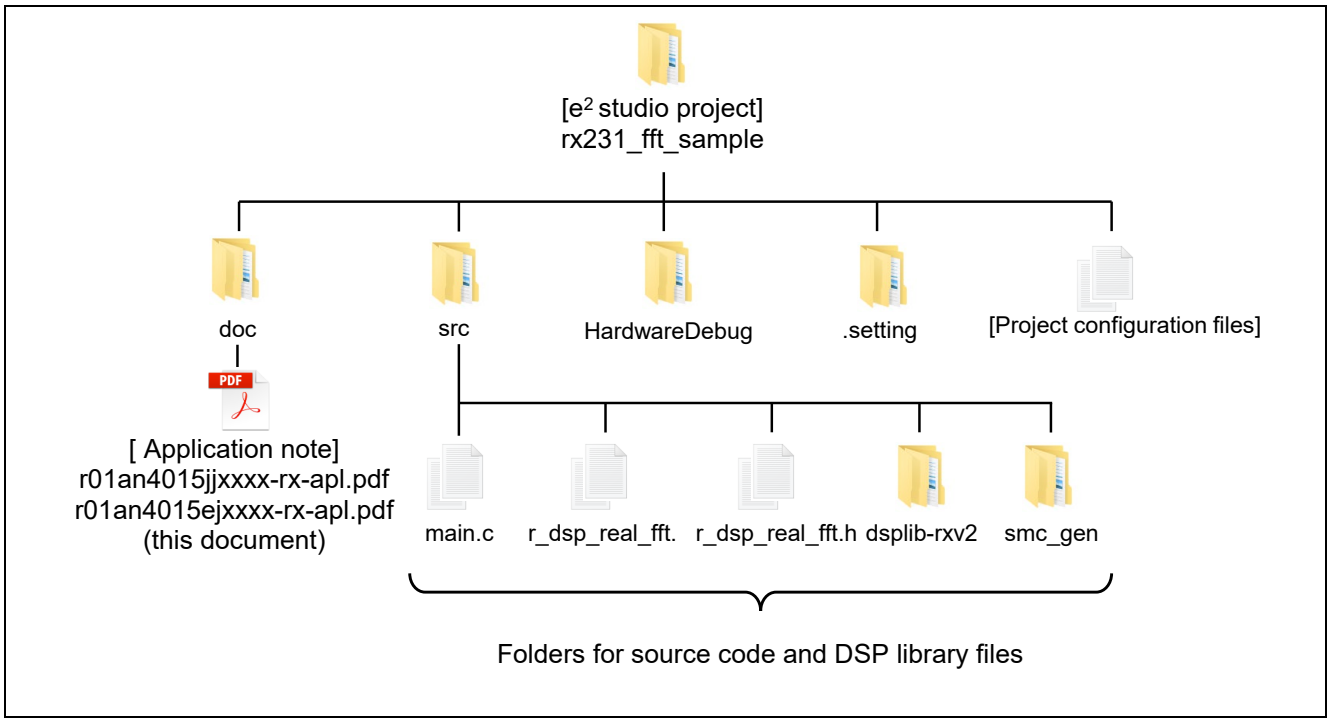


Figure 7 Folder Structure of the Sample Project

1.2 Structure of the Software

The Figure 8 shows software structure of the sample program. The software modules used in the software are listed in Table 1. The FIT modules and the DSP library can be obtained from the Renesas website. Functions for the peripheral function are generated with the Code Generator of the e^2 studio. For details on the software modules, refer to their application notes.

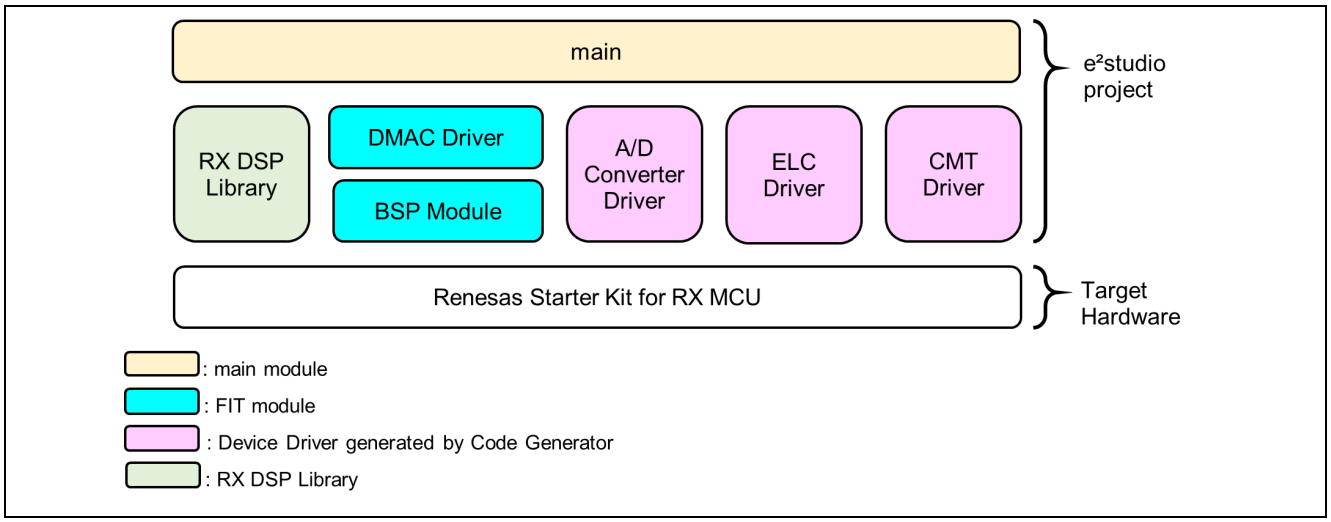


Figure 8 Structure of the Software

Table 1 Software Modules Used in the Sample Program

Module	Document Title	Document No.	Category
main	—	—	Module developed for this application note, contains main function
BSP	RX Family Board Support Package Module Using Firmware Integration Technology	R01AN1685EJ	FIT module
DMAC	RX Family DMA Controller DMACA Control Module Firmware Integration Technology	R01AN2063EJ	FIT module
S12AD	—	—	Driver functions generated by the Code Generator
CMT	—	—	Driver functions generated by the Code Generator
ELC	—	—	Driver functions generated by the Code Generator
DSP library	RX Family RX DSP Library version 5.0 (CCRX)	R01AN4359EJ	DSP library

1.3 Operation Confirmation Environment

The operation of the sample program in this application note has been confirmed under the following conditions.

Table 2 Operation Confirmation Conditions

Item	Description
MCU	R5F52318ADFP (RX231 Group)
Operating frequency	・ Main clock: 8 MHz ・ PLL: 54 MHz (main clock × 1/2 × 13.5) ・ System clock (ICLK): 54 MHz (PLL × 1)
Operating voltage	3.3 V
Operating mode	Single-chip mode
Processor mode	Supervisor mode
Integrated development environment	Renesas Electronics e² studio 2021-01
C compiler	Renesas Electronics RX Compiler CC-RX V3.03.00
	Compiler option ・ -lang = c99 ・ -fpu ・ -save_acc
Endian	・ Data endian: Little endian ・ Setting in the debug tool: Little endian
iodefine.h version	Version 1.0I
Sample program	Version 1.1
Evaluation board	Renesas Electronics CPU board: Renesas Starter Kit for RX231 (product No.: R0K505231S000BE) ・ On-board MCU: See above ・ Board configuration: Default ・ Power: Supplied by the emulator
Emulator	Renesas Electronics E2 emulator Lite
Function generator	Signal generator capable of analog signal output to output sine waveforms. Output signal setting to GND is as follows: Bias: 1.5 V, amplitude: 3.0 Vpp (see Figure 3): ・ Analog signal output (+) is connected to the JA1.10 pin on the evaluation board. A signal applied to the JA1.10 pin becomes an input to AN001 of S12AD. ・ Analog signal output (GND) is connected to the JA1.2 pin on the evaluation board. The JA1.2 pin is connected to the GROUND on the evaluation board.

2. Sample Program

2.1 Overview

The sample program consists of main and DMA transfer end interrupt processing. Table 3 lists roles of each processing. The A/D conversion result is transferred to the input buffer with the DMAC.

Table 3 Roles of Processing

Processing	Role
main processing	· Initialization of peripheral modules and FFT processing · Starting the first DMA transfer · Notifying the DMA transfer end interrupt processing of the next DMA transfer destination · FFT processing · CPU transition to sleep mode
DMA transfer end interrupt processing	· Setting and starting the second and the subsequent DMA transfers

2.2 Structure of the Sample Program

Table 4 shows the file composition of the main modules listed in Table 1. The functions of the source files are listed in Table 5 and Table 6. For details on the other software modules, refer to their application notes. And for the settings of them, refer to 3.4 Software Module Settings.

Table 4 File Composition

File	Description
main.c	· main and DMA transfer end interrupt processing
r_dsp_real_fft.c	· Initialization and performing FFT processing
r_dsp_real_fft.h	· Header file for r_dsp_real_fft.c

Table 5 Functions in main.c

Function	Description
main	· Initializes peripheral functions using software modules · Starts the first DMA transfer · Notifies DMA transfer end interrupt processing the next DMA destination address · Initializes and performs FFT processing using the functions of r_dsp_real_fft.c
set_buf_info	· Sets the next DMA transfer destination address and the number of data to the variables for communication between main and DMA transfer end interrupt processing
init_dmac_s12ad	· Initializes DMAC channel 0 with the S12AD as the transfer source
callback_dmac_s12ad	The callback function registered with the DMAC module by the main processing. · DMA transfer end interrupt processing for DMAC channel 0 · Sets the next transfer destination on DMAC and enables DMA transfer · Requests the information of the next transfer destination

Table 6 Functions in r_dsp_real_fft.c

Function	Description
R_DSP_REAL_FFT_Init	· Initializes FFT processing using the DSP library API.
R_DSP_REAL_FFT_ Operation	· Performs FFT processing using the DSP library API.

2.3 Processing Sequence

Figure 9 shows the processing sequence in the sample program.

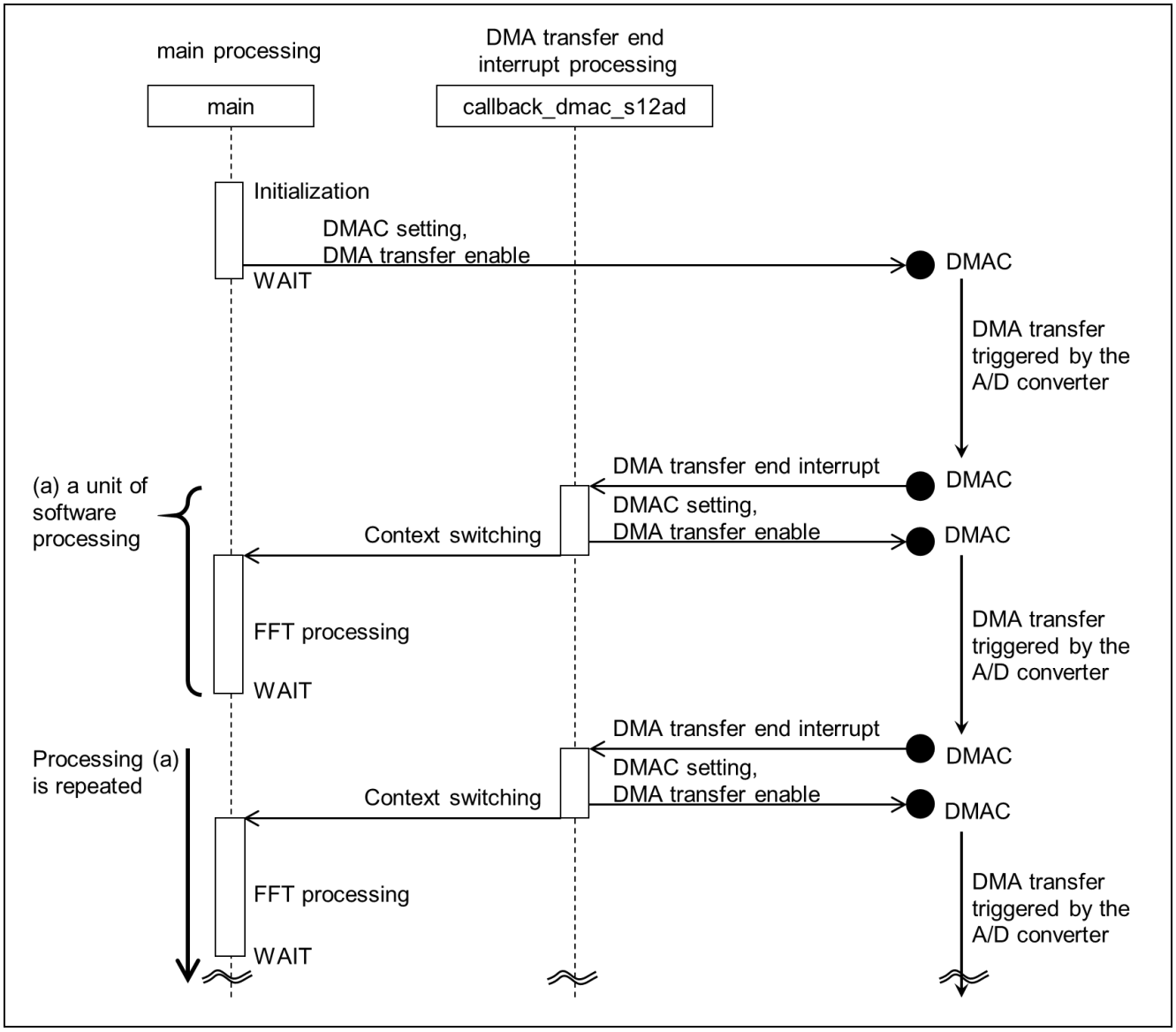


Figure 9 Processing Sequence in the Sample Program

As shown in the figure above, the first DMA transfer is configured and started by main processing. When DMA transfer is performed for the specified number of data, the DMA transfer end interrupt occurs. This becomes a trigger to execute DMA transfer end interrupt processing and main processing in order. A unit of software processing will be repeated.
DMA transfer source is the A/D converter which operates periodically with approximately 1 kHz of the sampling frequency. The DMA transfer is executed while main processing is being executed.
The WAIT instruction is used to cause the CPU to transition to sleep mode and sleep mode is released with the DMA transfer end interrupt.

2.4 Processing Flow

Figure 10 shows the Processing Flow of the Sample Program.

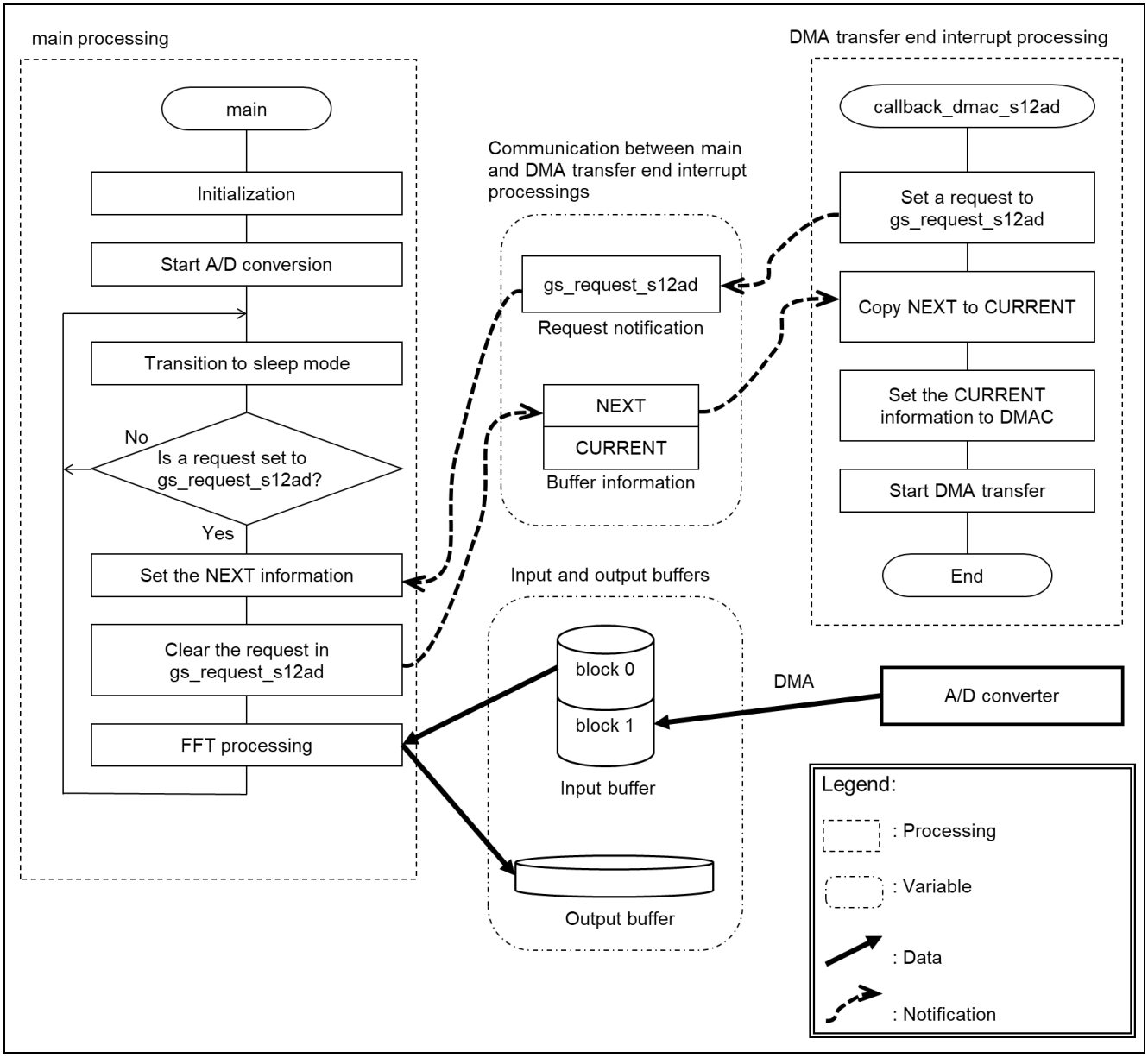


Figure 10 Processing Flow of the Sample Program

Elements shown in Figure 10 are described below.

- main processing
  This processing performs initializations first and transitions to sleep mode. After the A/D conversion result is stored in the buffer by the DMAC, main processing specifies the next DMA transfer destination and performs FFT processing. Then the CPU transitions to sleep mode. This unit of processing will be repeated.
- DMA transfer end interrupt processing
  This processing requests main to update the next DMA transfer destination and starts the current DMA transfer. The DMA transfer end interrupt causes the CPU to transition to normal operation mode.
- Communication between main and DMA transfer end interrupt processing
  When main notifies DMA transfer end interrupt processing of the next DMA transfer destination and when DMA transfer end interrupt processing requests main to update the next DMA transfer

destination, specific variables are used for communication between main and DMA transfer end interrupt processing. Table 7 lists variables used in the communication.

- Input buffer
  The DMAC stores the A/D conversion result into this buffer and FFT processing reads the result as input data.

  The buffer consists of two blocks to avoid access conflict between the CPU (FFT processing) and the DMAC. The blocks the CPU and the DMAC accessing are switched by DMA transfer end interrupt, Figure 11 shows switching. Figure 12 shows data alignment.
- Output buffer
  FFT processing stores the result of the processing into the output buffer. Figure 12 shows data alignment.

Table 7 Variables for Communication Between Main and DMA Transfer End Interrupt Processing

Contents		Description
(a) Request notification		The DMA transfer end interrupt processing uses this variable to request the main to update (b) Buffer information. The request is set by the DMA transfer end interrupt processing. When the variable is set, the main updates “NEXT” of (b) Buffer information and clears the request. The request is first cleared by main processing at the initialization.
(b) Buffer information		The start address and the number of data in the DMA transfer destination buffer are stored in this variable. The buffer consists of two sides; NEXT and CURRENT to avoid access conflict between processings. The initial values for NEXT and CURRENT are set by the main processing.
	NEXT	The main stores the start address and the data count in this side in response to (a) Request notification. The initial values are as follows: ・ Start address: data 0 of block 1 in the input buffer ・ Data count: 1024
	CURRENT	The DMA transfer end interrupt refers to the start address and the data count stored in this side to specify in the DMAC. The DMA transfer end ISR copies the information in NEXT to CURRENT. The initial values are as follows: ・ Start address: data 0 of block 0 in the input buffer ・ Data count: 1024

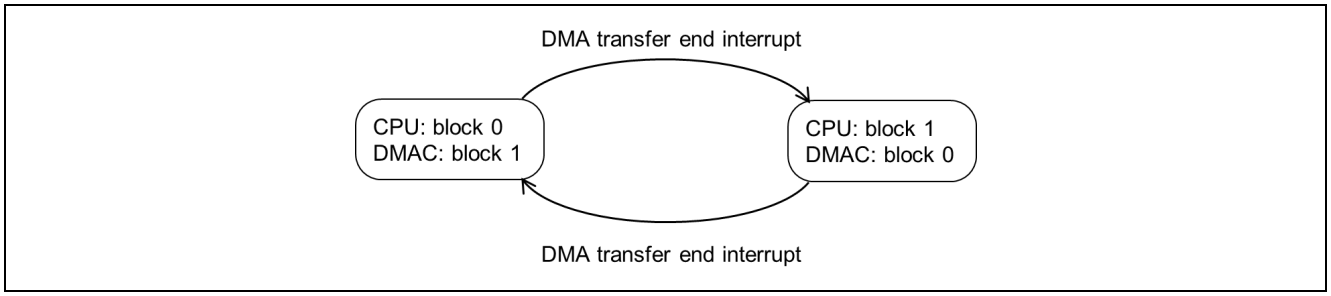
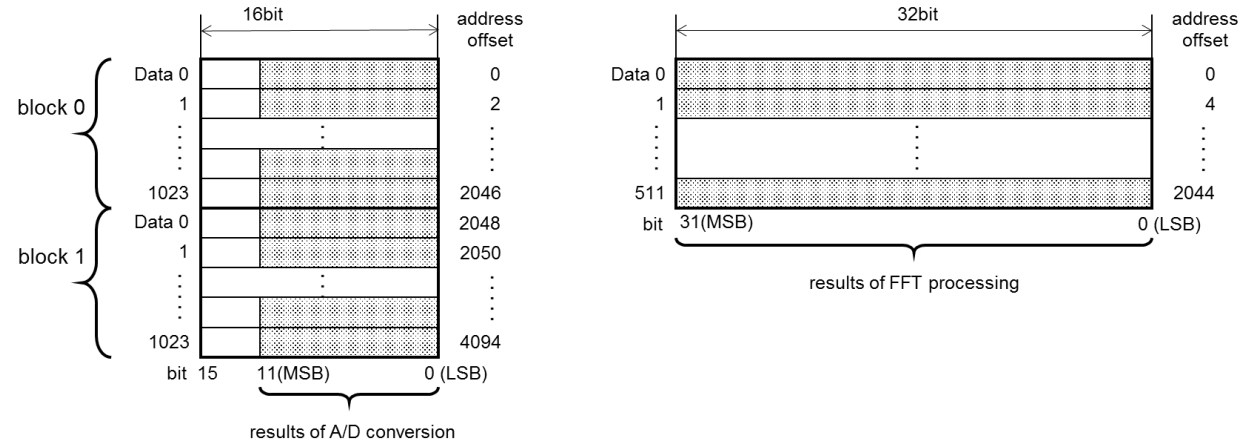


Figure 11 Switching Blocks of Input Buffer where the CPU and the DMAC access



Note. The point count of the FFT result is a half of the input in compliance with the specification of the real FFT API in the DSP library. Therefore, the data count of the output buffer shall be a half of the data count of the input buffer.

Figure 12 Input Buffer and Output Buffer

2.4.1 Initialization

The initialization procedure in the sample program is as follows:

1. Initialize the peripheral functions.

   The CMT, S12AD, and the ELC are initialized.

2. Initialize FFT processing.

   The FFT point count and options are specified.

3. Initialize the DMAC module.

   The DMA channel is specified, the callback function is registered and interrupt priority for DMA transfer end interrupt is specified.

4. Initialize variables for communication between main and DMA transfer end interrupt processing

   Variables used for communication between main and DMA transfer end interrupt processing (see Table 7) are initialized.

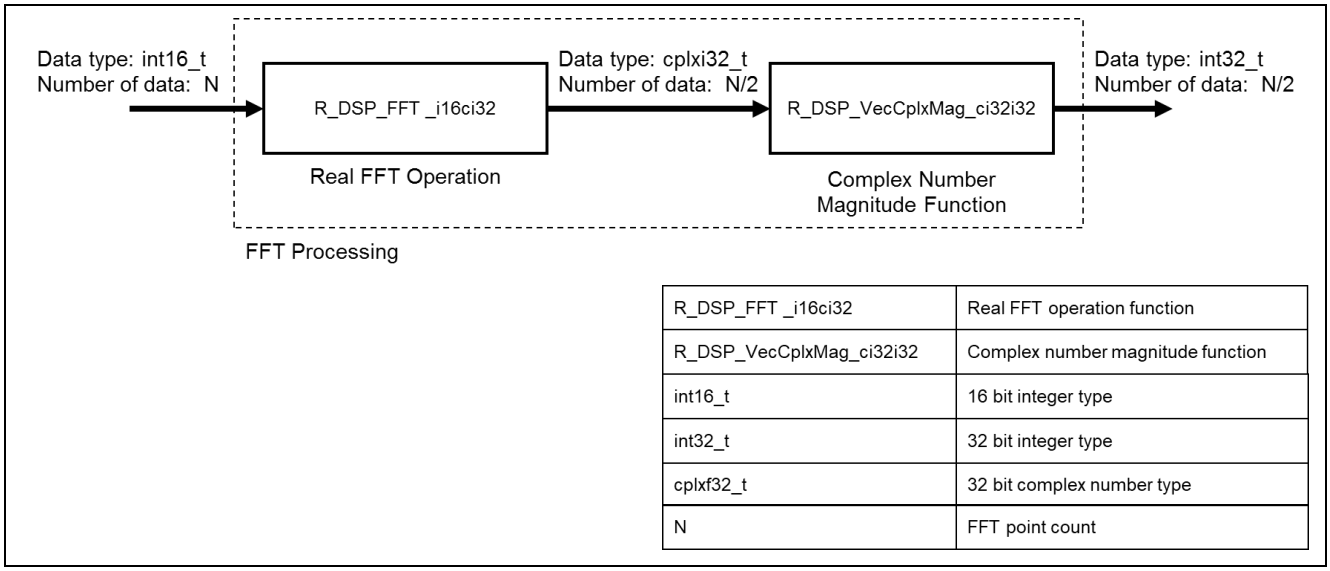
5. Enable DMA transfer.

   Parameters like the activation source, the transfer mode, the start Address of source, the start address of destination, etc. are specified and DMA transfer is enabled.

Perform the above procedure and then start A/D conversion, DMA transfer is started.

2.4.2 FFT Processing

The sample program outputs magnitudes of the frequency spectrum corresponding to input signals using the APIs of the DSP library as shown in Figure 13.



R_DSP_FFT _i16ci32	Real FFT operation function
R_DSP_VecCplxMag_ci32i32	Complex number magnitude function
int16_t	16 bit integer type
int32_t	32 bit integer type
cplxf32_t	32 bit complex number type
N	FFT point count

Figure 13 FFT Processing

- APIs used for FFT processing

  FFT processing consists of the real FFT operation function R_DSP_FFT_i16ci32 and the complex number magnitude function R_DSP_VecCplxMag_ci32i32. R_DSP_FFT_i16ci32 outputs FFT results as complex number. In order to get magnitudes of the frequency spectrum on input signals, the sample program converts the complex number outputs of R_DSP_FFT_i16ci32 into integer number of magnitudes with R_DSP_VecCplxMag_ci32i32.

- Input and output data

  FFT processing outputs N/2 counts of results for N counts of inputs. The N corresponds to the FFT point count and a unit of executing R_DSP_FFT_i16ci32. The point count of results is 1/2 of the input in compliance with the specification of R_DSP_FFT_i16ci32. And as R_DSP_VecCplxMag_ci32i32 is a specification that outputs in Q2.30 format, the output magnitude value is 1/2 of the input.

  For details, refer to the RX DSP Library APIs Version 5.0 User's Manual: Software.

- Output data corresponding to frequencies

  The outputs of the FFT processing shows magnitudes corresponding to frequencies. As shown in Table 8, the magnitudes corresponding to frequencies are stored in the output buffer shown in Figure 12.
  The frequency interval is calculated with the sampling frequency and the FFT point count. In the sample program, it equivalents approximately 0.97 Hz by the following calculation.

  Sampling Frequency / FFT Point Count = 1000.296384/1024 = 0.976851938 Hz

Table 8 Output Data corresponding to Frequencies

Data number	Frequency [Hz]
0	0.97… x 0 = 0
1	0.97… x 1 = 0.97
:	:
510	0.97… x 510 = 498.04
511	0.97… x 511 = 499.02

3. Reference Information

3.1 Memory Usage

Table 9 lists the memory usage of the sample program.

The ROM and the RAM are values calculated in reference to the map file generated with the conditions of Table.2. The User stack and the Interrupt stack are values measured really used stack memory while the sample program is running.

Table 9 Memory Usage of the Sample Program (Reference)

Item	Measurement Value (byte)	Measurement Condition
ROM	20398	Overall ROM consumption of the sample program.
RAM	16426	Overall RAM consumption of the sample program.
User stack	156	Maximum stack memory consumption of the sample program.
Interrupt stack	88	

3.2 Current Consumption

Table 10 lists the current consumption while the sample program is running in case of using sleep mode or not.

While the sample program is running on only the RSKRX231 board, the measurements of current are performed. With the following conditions, the currents from power supply to the board are measured.

- Using a DC Regulated Power Supply, supply RSKRX231 board's PWR pins with 3.3V
- A/D conversion and FFT processing of the sample program are both running

Table 10 Current Consumption of the Sample Program (Reference)

Sleep mode	Measurement Value (mA)	Measurement Condition
Used	25	A measurement of the sample program without any modifications.
Not used	40	A measurement of the sample program with removal of only WAIT instruction.

3.3 About Sampling Frequency

The sampling period of A/D is corresponding to compare match events generated by the CMT1. The following is the sampling frequency generated in the sample program and Table 11 shows settings in the CMT1.

- Sampling Frequency
  CMT1's clock source / Count = (27000kHz / 8) / (3374 + 1) =1.000kHz

Table 11 Settings in the CMT1

Items	Value
Peripheral Clock B (PCLKB)	27MHz
Clock source	PCLK/8
Count	3374

3.4 Software Module Settings

Table 12 to Table 17 describe Settings for FIT modules used in the sample program, the Code Generator in the e^2 studio, and the DSP library. For details on software modules, refer to their application notes.

Table 12 BSP Module Settings

Category	Item	Setting/Description
Smart Configurator >> Components >> r_bsp		Other than the changes listed below, properties are left in the default settings.
	Parameter checking	Disabled
Smart Configurator >> Clock		The following settings are made on the “Clocks” tab and reflected in r_bsp_config.h.
	VCC setting	3.3 (V)
	Main clock settings	Operation: Checked. Oscillation source: Resonator Frequency: 8 (MHz) Wait time: 8,192 cycles, 2,048 (μs)
	PLL circuit settings	Frequency Division: ×1/2 Frequency Multiplication: ×13.5
	Sub-clock settings	Operation: Unchecked.
	HOCO clock settings	Operation: Unchecked.
	LOCO clock settings	Operation: Unchecked.
	System clock settings	Clock source: PLL circuit System clock (ICLK): ×1 54 (MHz) Peripheral module clock (PCLKA): ×1 54 (MHz) Peripheral module clock (PCLKB): ×1/2 27 (MHz) Peripheral module clock (PCLKD): ×1 54 (MHz) External bus clock (BCLK): ×1/64 0.84375 (MHz) FlashIF clock (FCLK): ×1/64 0.84375 (MHz)
	IWDT dedicated low-speed clock settings	Operation: Unchecked.
	USB dedicated clock (UCLK) settings	Operation: Unchecked.
	BCLK pin output settings	Operation: Unchecked.

Table 13 DMAC Module Settings

Category	Item	Setting/Description
r_dmaca_rx_config.h		Other than the changes listed below, default settings are used.
	DMACA_CFG_PARAM_CHECKING_ENABLE	Changed to 0 (Omit parameter checking when compiling code.)

Table 14 Smart Configurator Settings (S12AD)

Category	Item	Setting
Smart Configurator >> Components >> Single Scan Mode S12AD (Config_S12AD0)		Code is generated using the settings below.
	Analog input mode setting	Double trigger mode: Unchecked.
	Analog input channel setting	Only AN001 checked.
	Conversion start trigger setting	Start trigger source: Trigger from ELC
	Interrupt settings	Enable A/D conversion end interrupt (S12ADI0) checked. Priority: Level 0 (disabled)
	Add/Average AD value setting	AN001 unchecked.
	A/D conversion select	High-speed
	High-Potential reference voltage setting	VREFH0
	Low-Potential reference voltage setting	VREFL0
	Self diagnosis setting	Mode: Unused.
	Disconnection detection assist setting	Charge setting: Unused.
	Data registers setting	Data placement: Right-alignment Automatic clearing: Disable automatic clearing Addition/Average mode select: Addition mode Addition count: 1-time
	Data storage buffer setting	Disable
	Window function setting	Disable
	Window A/B operation setting	Enable comparison window A: Unchecked. Enable comparison window B: Unchecked.
	Input sampling time setting	AN001: 3.667 (µs)
	Event link control setting	ELC scan end event generation condition: On completion of all scans

Table 15 Smart Configurator Settings (CMT1)

Category	Item	Setting/Description
Smart Configurator >> Components >> Compare Match Timer (Config_CMT1)		Code is generated using the settings below.
	Count clock settings	PCLK/8
	Compare match setting	Interval value: 1,000 µs (Actual value: 1,000) Register value (CMCOR): 3,374 Compare match interrupt (CMI1): Unchecked.

Table 16 Smart Configurator Settings (ELC)

Category	Item	Setting/Description
Smart Configurator >> Components >> Event Link Controller (Config_ELC)		Code is generated using the settings below.
	SOURCE	Configuration: Config_CMT1 Resource: CMT1 Event: CMT1 compare match 1
	DESTINATION	Configuration: Config_S12AD0 Resource: S12AD0 Operation: Start A/D conversion

Table 17 Setting in the DSP Library

Category	Item	Setting/Description
Files used		The following files included in the RX_DSP_Sample_CCRX_e2studio project of the DSP library are used.
	.lib file	RX_DSP_FPU_LE.lib
	.h file	r_dsp_complex.h r_dsp_transform.h r_dsp_typedefs.h r_dsp_types.h r_dsp_ver_info.h

4. Obtaining the Development Environment

4.1 e² studio

Visit the following URL and download the e² studio.

https://www.renesas.com/en-us/products/software-tools/tools/ide/e2studio.html

This document assumes that 2021-01 or later version of e² studio is used. If a version earlier than 2021-01 is used, some features of e² studio may not be supported. Make sure to download the latest version of e² studio on the website.

4.2 Compiler Package

Visit the following URL and download the RX Family C/C++ Compiler Package.

https://www.renesas.com/en-us/products/software-tools/tools/compiler-assembler/compiler-package-for-rx-family.html

5. Importing a Project

The sample code is provided as the e² studio project. This section describes importing a project into the e² studio. After importing a project, confirm that the build settings and the debug settings are correct.

5.1 Importing a Project into the e² studio

Follow the steps below to import your project into the e² studio. Windows/dialogs may differ depending on the e² studio version used.

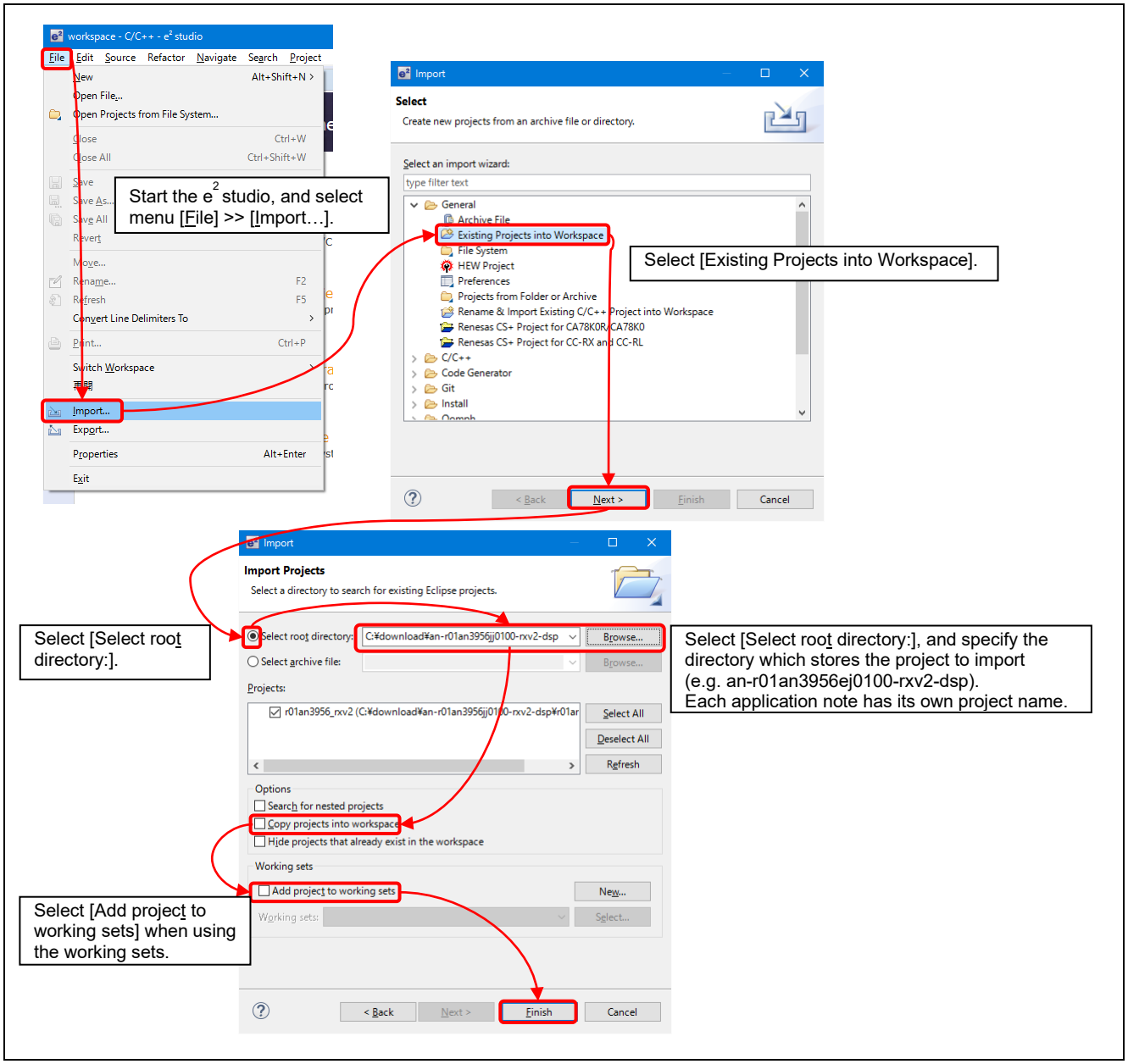


Figure 14 Importing a Project into the e² studio

5.2 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.

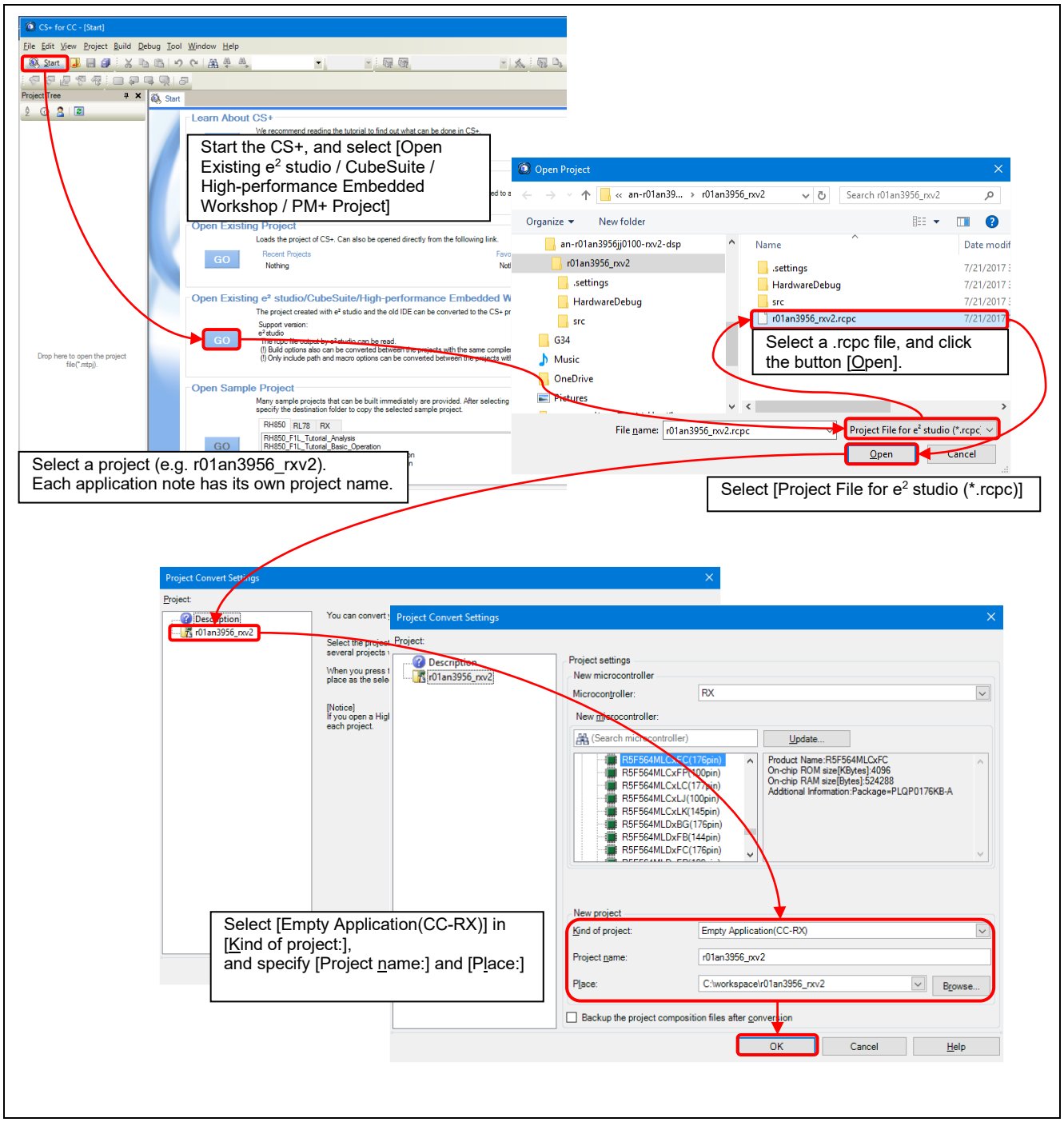


Figure 15 Importing a Project into CS+

6. Others

6.1 Notes on Using the Evaluation Version of C/C++ Compiler Package for RX Family

When using the evaluation version of C/C++ Compiler Package for RX Family, the evaluation period and usage limitations apply. When the evaluation period expires, the size of linkable object is reduced to 128 Kbytes or less and this may cause the incorrect generation of the load module.

For details, refer to the following software tool page for evaluation versions on the Renesas website:

https://www.renesas.com/en-us/products/software-tools/evaluation-software-tools.html

7. Reference Document

- RX Family Board Support Package Module Using Firmware Integration Technology (R01AN1685)
- RX Family DMA Controller DMACA Control Module Firmware Integration Technology (R01AN2063)
- e^2 studio Code Generator User's Manual: RX API Reference (R20UT2864)
- RX Family RX DSP Library version 5.0 (CC-RX) (R01AN4359EJ0100)
- RX230 Group, RX231 Group User’s Manual: Hardware (R01UH0496)
- Renesas Starter Kit for RX231 User's Manual (R20UT3027)
- Renesas Starter Kit for RX231 CPU Board Schematics (R20UT3026)

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec,15,17	-	First edition issued
1.10	Apr,26,21	3	1. Specifications Updated DSP library version Figure 4 Example of an FFT Result Processing Result Updated.
		5	1.1 Structure of the Application Note Figure 6 Structure of the Application Note Figure 7 Folder Structure of the Sample Project Updated
		6	1.2 Structure of the Software Table 1 Software Modules Used in the Sample Program Updated
		7	1.3 Operation Confirmation Environment Table 2 Operation Confirmation Conditions Updated
		13	2.4.2 FFT Processing Input and output data Added description. Table 8 Output Data corresponding to Frequencies Fixed typo
		14	3.1 Memory Usage Updated. 3.2 Current Consumption Updated. 3.3 About Sampling Frequency Fixed typo.
		15-16	3.4 Software Module Settings Updated.
		17	4.1 e^2 studio Updated.
		19	5.2 Importing a Project into CS+ Added.
		20	7. Reference Document Updated.
		program	Updated operation confirmation environment. (e^2 studio, compiler, FIT modules, DSP Library etc.) Changed DSP processing specifications as follows - Changed the output format of the frequency magnitude characteristic of the FFT processing result from Q1.31 to Q2.30 due to the update of the DSP library. - Changed the method of deriving the element [0] of the frequency magnitude characteristic of the FFT processing result so that the absolute value of $R_{N/2}$ is acquired and the format is Q2.30. (r_dsp_real_fft.c)

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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. 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.
6. 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. 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.5.0-1 October 2020)

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/.