RX Family

Demonstration of Digital Signal Analysis and Judgement Using FFT

Introduction

This application note describes a demo system showing the use of an RX MCU in the sensor field. It shows how a single RX MCU can be used for analog signal input, digital signal analysis, and displaying the analysis results.

The sample program described in this application note is configured with the sensor system shown in the Figure below as the basic model. The elements of the model correspond to the following processing performed by the sample program:

- A/D conversion (data collection)
  Analog signal input is processed using the A/D conversion functionality of the RX MCU.
- Digital signal processing (analysis)
  Frequency analysis is performed using IIR filter processing and fast Fourier transform (FFT) processing.
- Check & judge (judgement)
  A pass/fail judgement is made based on the frequency analysis results.
- Action (control)
  The judgement result, etc., is displayed on an LCD.

This application note describes the demo environment and procedure, and the sample program, in the pages that follow.

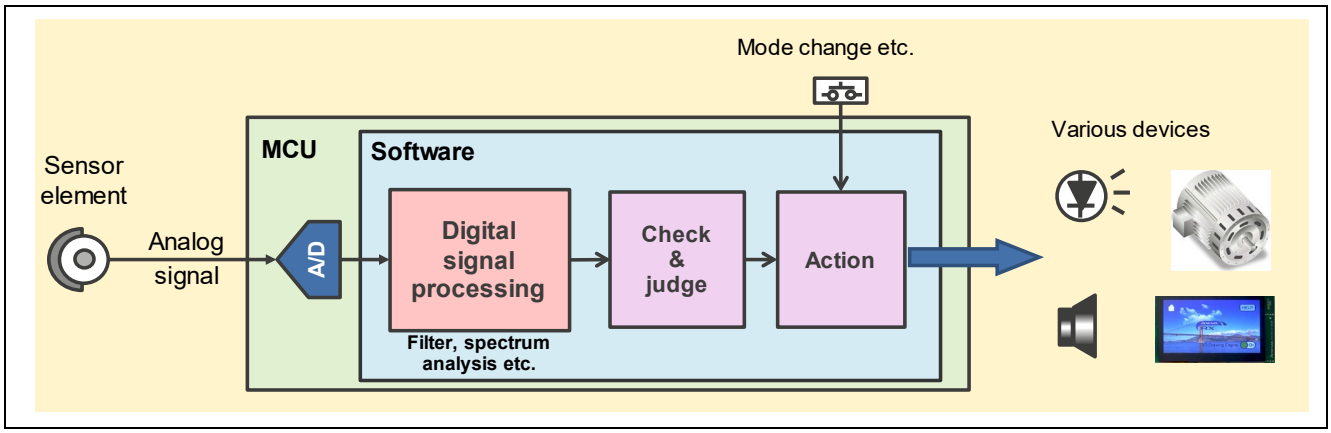


Figure Sensor System Model

Target Device

RX231 Group

Operation Confirmation Board

Renesas Starter Kit for RX231

Target Board for RX231

A sample program that runs on the Renesas Starter Kit for RX231 (RSK) or the Target Board for RX231 (Target Board) is distributed with this application note. Note that the version of the sample program for the Target Board does not include the LCD display functionality.

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. System Overview

Figure 1.1 shows an overview of the system described in this application note.

This system uses a single RX231 MCU for all processing from sampling of the input signal to judgement result output control.

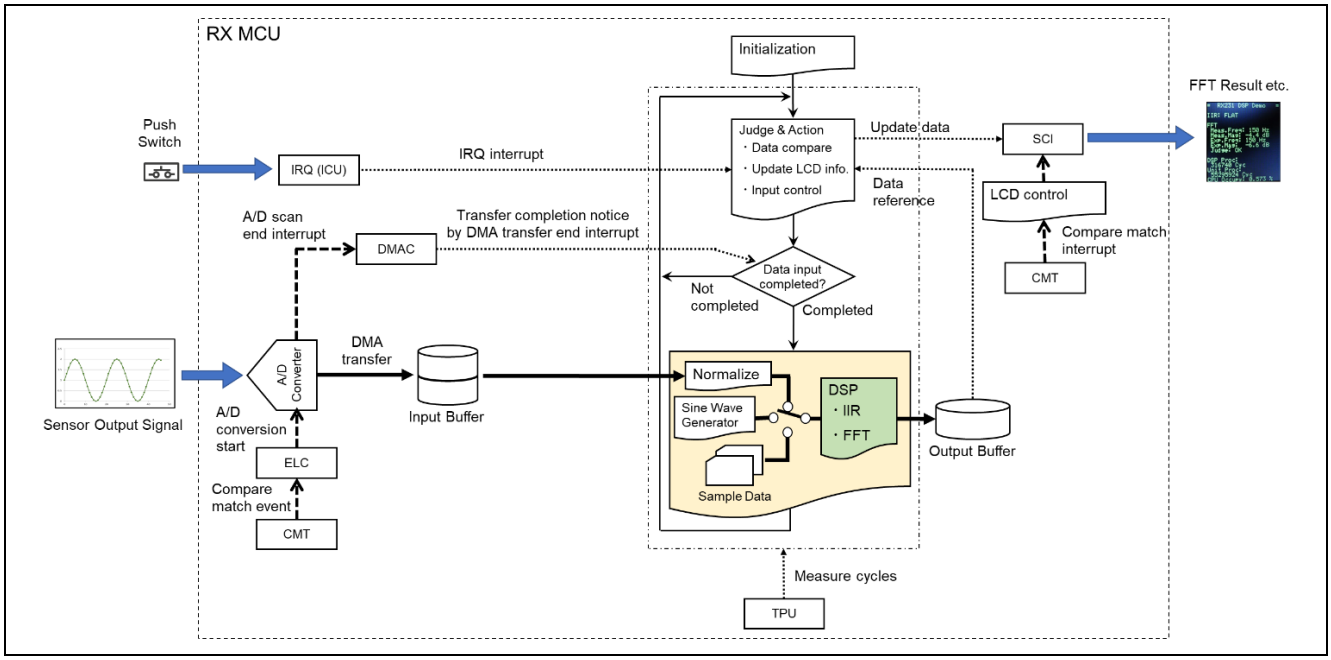


Figure 1.1 System Overview

The system performs the following processing:

- Input signal sampling
  The 12-bit A/D converter (S12AD), compare match timer (CMT), and event link controller (ELC) are used to perform A/D conversion of an input signal sampled at a frequency of approximately 1 kHz. First, the CMT generates compare match events with a period of approximately 1 µs, and these events are passed by the ELC to the S12AD as A/D conversion start triggers. The converted data is transferred to the input buffer by the DMA controller (DMAC).
- Input data normalization
  The input signal A/D converted by the S12AD is stored in the input buffer as 12-bit (unsigned) data. The 12-bit data stored in the input buffer is normalized to 31-bit (signed) format (by bias processing and scaling).
- IIR filter processing and FFT processing
  The RX DSP Library API, version 5.0, is used to perform IIR filter processing. The filter characteristics can be changed among pass-through (FLAT), low-pass filter (LPF), and high-pass filter (HPF) by means of switch input. After IIR filter processing, the data undergoes 1,024-point FFT processing, and the result is stored in the output buffer.
- Judgement of processing results and processing based on judgement result
  The frequency spectrum information obtained by FFT processing is compared with the expected values. A pass/fail judgement is made based on the comparison results, and an indication of the judgement is displayed on the LCD module. The compare match timer (CMT) and serial communications interface (SCI) are used to display information on the LCD module.
- Processing cycle count measurement
  The cycle counts required for the following processing are measured. The 16-bit timer pulse unit (TPU) is used for measurement.
  — DSP processing (normalization processing, IIR filter processing, and FFT processing) cycle count
  — Cycle count for inputting 1,024 samples of input signal

- Sine wave generation processing
  Processing that generates sine waves. The frequency can be adjusted in the range from 0 to 490 Hz. The amplitude can be set to negative infinity or adjusted in the range from -96 to 0 dB. You can use the output of this processing as a test signal to evaluate the IIR filter processing or FFT processing.

Figure 1.2 shows an example of frequency spectrum magnitude characteristics obtained when inputting a sine wave to the system.

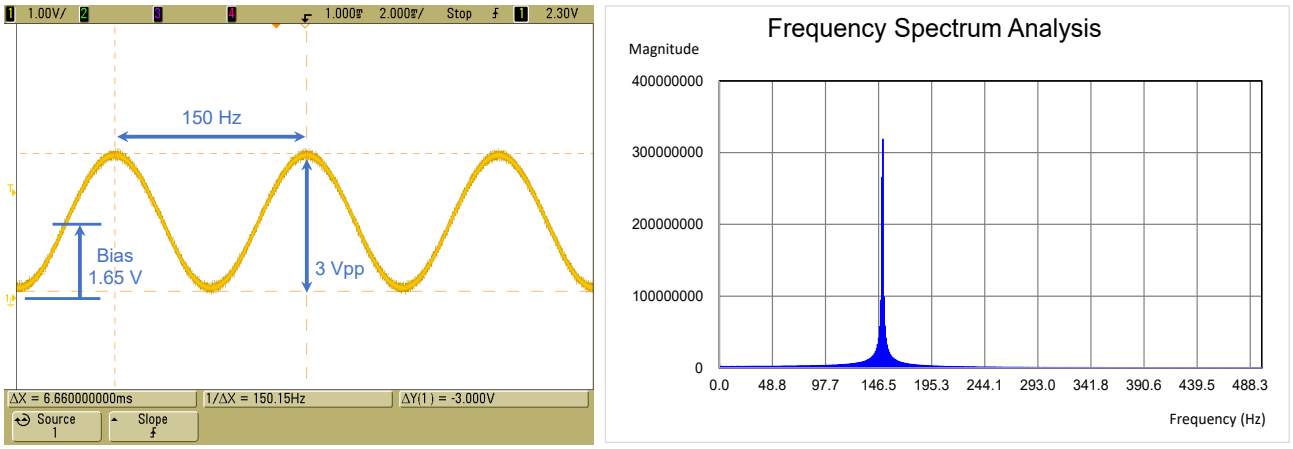


Figure 1.2 Input Signal (Sine Wave of Frequency: 150 Hz, Amplitude: 3 Vpp) (Left) and FFT Processing Results (Right)

Figure 1.3 LCD Module Display Screen

1.1 File Structure Associated with This Application Note

Figure 1.4 shows the file structure associated with this application note. When the contents of the ZIP file in which this application note is distributed is unzipped, a folder is created with the same name as the ZIP file. The “workspace_dsp_example” folder within this folder contains an e^2 studio workspace that includes two projects in e^2 studio format: “dsp_demo_rx231_rsk” (for the RSK) and “dsp_demo_rx231_tb” (for the Target Board). As shown in Figure 1.5, the project folder contains the sample program source code files as well as e^2 studio configuration files and this application note.

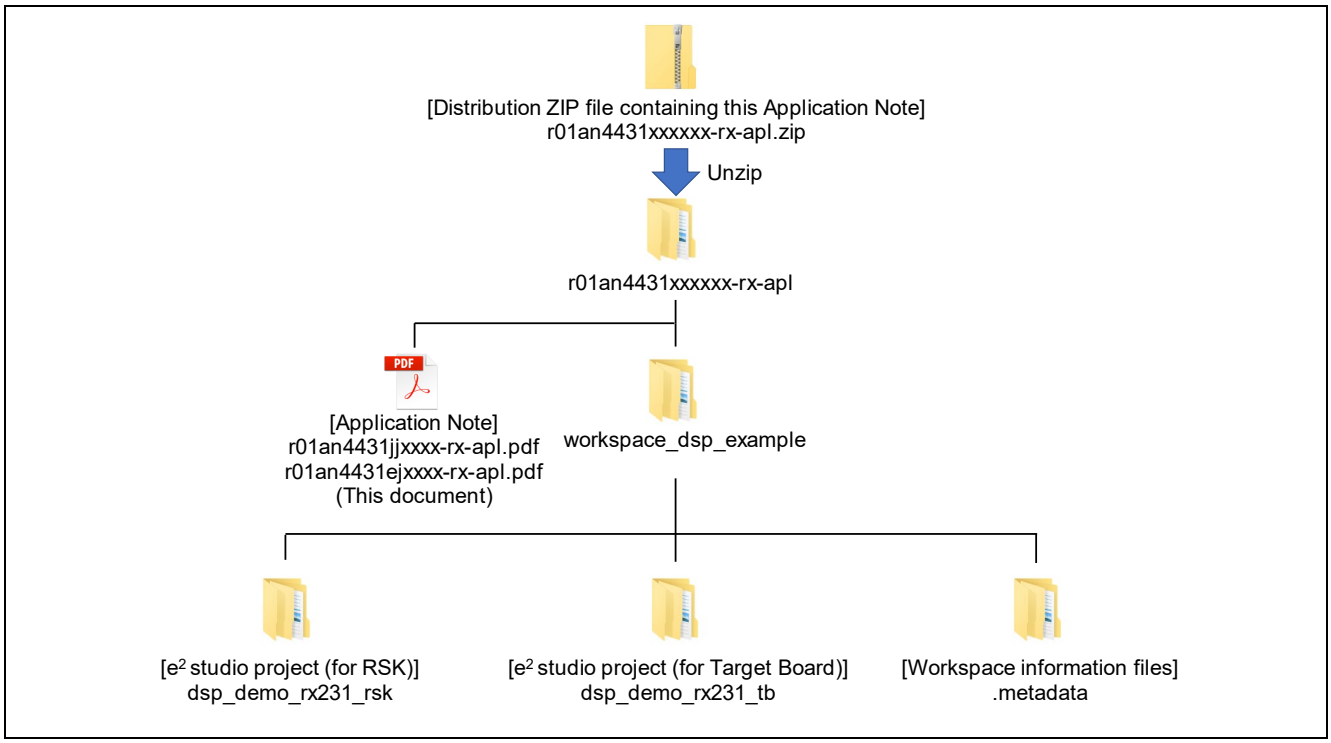
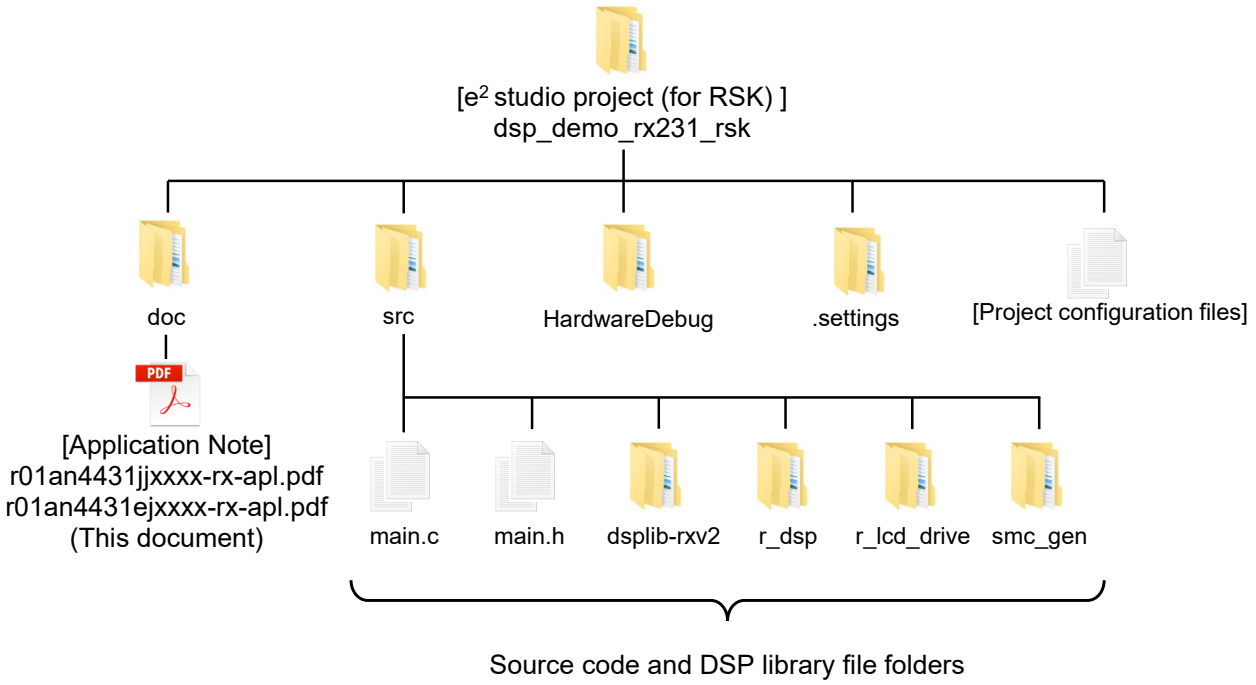


Figure 1.4 File Structure Associated with This Application Note



Note: The sample project for the Target Board does not include the r_lcd_driver folder.

Figure 1.5 Folder Structure of Sample Project

1.2 Structure of Sample Program

Figure 1.6 shows the structure of the sample program and Table 1.1 lists the software modules used. The sample program for the Target Board does not include the LCD display functionality, so the structures of the software modules used for the RSK and for the Target Board differ to some extent. The FIT modules and DSP library can be obtained from the Renesas website. Driver software for the other peripheral functions is generated by using the Code Generator function of e^2 studio. For details of each software module, refer to the associated application note or the e^2 studio help system.

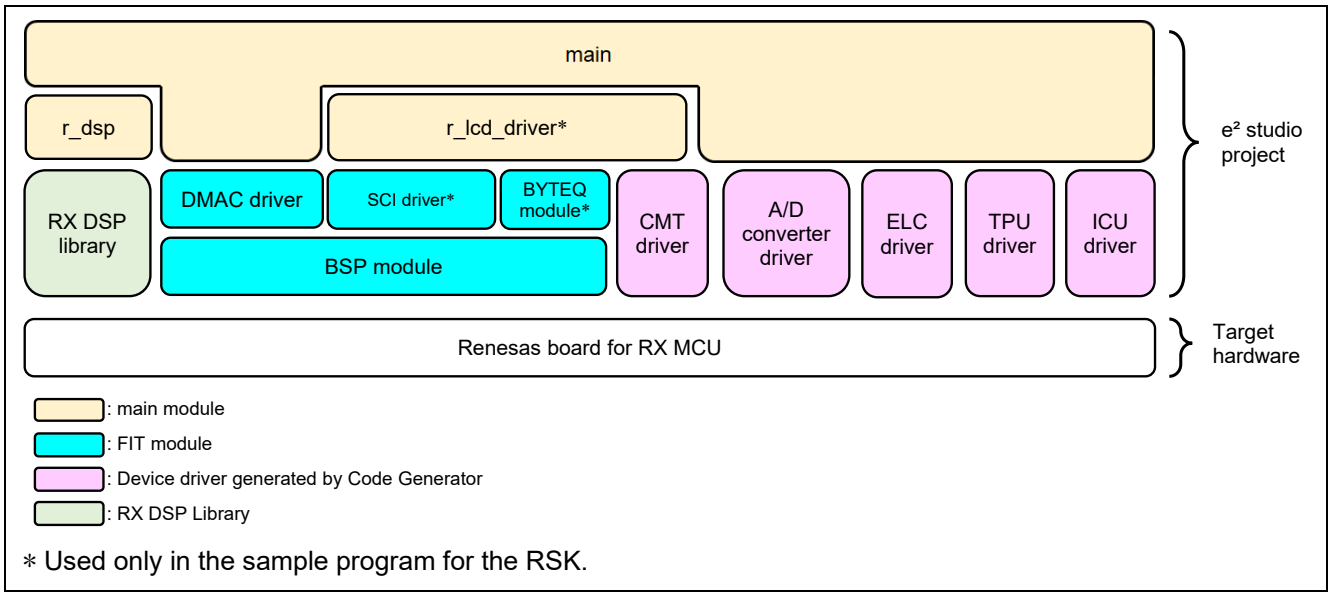


Figure 1.6 Structure of Sample Program

Table 1.1 List of Software Modules Used

Module	Document Title	Document No.	Category
main	—	—	Module containing the main function developed for the program described in this application note
r_dsp	—	—	DSP library operation module developed for the program described in this application note
r_lcd_driver	—	—	Debug LCD control module developed for the program described in this application note (Used only in the sample program for the RSK.)
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
SCI	RX Family SCI Multi-Mode Module Using Firmware Integration Technology	R01AN1815EJ	FIT module (Used only in the sample program for the RSK.)
BYTEQ	RX Family BYTEQ Module Using Firmware Integration Technology	R01AN1683EJ	FIT module (Used only in the sample program for the RSK.)
S12AD	—	—	Driver function generated by Code Generator
CMT	—	—	
ELC	—	—	
TPU	—	—	
ICU	—	—	
RX DSP Library	RX Family DSP Library version 5.0 (CC-RX)	R01AN4359EJ	DSP library

1.3 Operating Environment

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

Table 1.2 Operation Confirmation Conditions (For RSK: dsp_demo_rx231_rsk)

Item		Description
MCU		R5F52318ADFP (RX231 Group)
	Operating frequency	• Main clock: 8 MHz • PLL circuit output: 54 MHz (main clock × 1/2 × 13.5) • System clock (ICLK): 54 MHz (PLL circuit output × 1) • Peripheral module clock (PCLKB): 27 MHz (PLL circuit output × 1/2)
	Operating voltage	3.3 V
	Operating mode	Single-chip mode
	Processor mode	Supervisor mode
Integrated development environment		Renesas Electronics e^2 studio 2022-04
C compiler		Renesas Electronics RX Compiler CC-RX V3.04.00
		Compiler options • -lang = c99 • -fpu • -save_acc
Endian order		• Data: Little endian • Debug tool setting: Little endian
iodefine.h		Version 1.0I
Sample program		Version 1.60
Evaluation board		Renesas Electronics: Renesas Starter Kit for RX231 (R0K505231S000BE) CPU board • On-board MCU: See above. • Board settings: Default • Power supply: Supplied by emulator. • Debug LCDs connected.
Emulator		Renesas Electronics E2 emulator Lite
Function generator		Signal generator with analog signal output terminal to output sine waveforms. Output signal set to 1.65 V bias relative to ground and amplitude of 3.0 Vpp. See Figure 1.2. • Analog signal output (+) is connected to the JA1.10 pin on the evaluation board. The signal applied to the JA1.10 pin is 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 GROUND on the evaluation board.

Table 1.3 Operation Confirmation Conditions (For Target Board: dsp_demo_rx231_tb)

Item		Description
MCU		R5F52318ADFP (RX231 Group)
	Operating frequency*	• HOCO clock: 54 MHz • System clock (ICLK): 54 MHz (HOCO output × 1) • Peripheral module clock (PCLKB): 27 MHz (HOCO output × 1/2)
	Operating voltage	3.3 V
	Operating mode	Single-chip mode
	Processor mode	Supervisor mode
Integrated development environment		Renesas Electronics e^2 studio 2022-01
C compiler		Renesas Electronics RX Compiler CC-RX V3.04.00
		Compiler options • -lang = c99 • -fpu • -save_acc
Endian order		• Data: Little endian • Debug tool setting: Little endian
iodefine.h		Version 1.0I
Sample program		Version 1.60
Evaluation board		Renesas Electronics: Target Board for RX231 (RTK5RX2310C00000BR) CPU board • On-board MCU: See above. • Board settings: Default • Power supply: Supplied via USB.
Emulator		Renesas Electronics E2 emulator Lite (on-board)
Function generator		Signal generator with analog signal output terminal to output sine waveforms. Output signal set to 1.65 V bias relative to ground and amplitude of 3.0 Vpp. See Figure 1.2. • Analog signal output (+) is connected to the J2.43 pin on the evaluation board. The signal applied to the J2.43 pin is input to AN001 of S12AD. • Analog signal output (GND) is connected to the J2.12 pin on the evaluation board. The J2.12 pin is connected to GROUND on the evaluation board.

* The sample program for the Target Board uses HOCO as the clock source. For details, refer to 4.1, Frequency Values of FFT Processing Results.

2. Running the Sample Program

The procedure for running the program described in this application note is shown below.

2.1 Launching the Workspace

Extract the ZIP file containing the project described in this application note to a location of your choice, and make sure the path of the destination does not contain any Japanese or other double-byte characters. Next, launch e^2 studio and, when Eclipse Launcher window appears, select the workspace (workspace_dsp_example) described in this application note.

If Eclipse Launcher window does not appear when e^2 studio is launched, make the following selection after launching e^2 studio:

[File] >> [Switch Workspace] >> [Other]

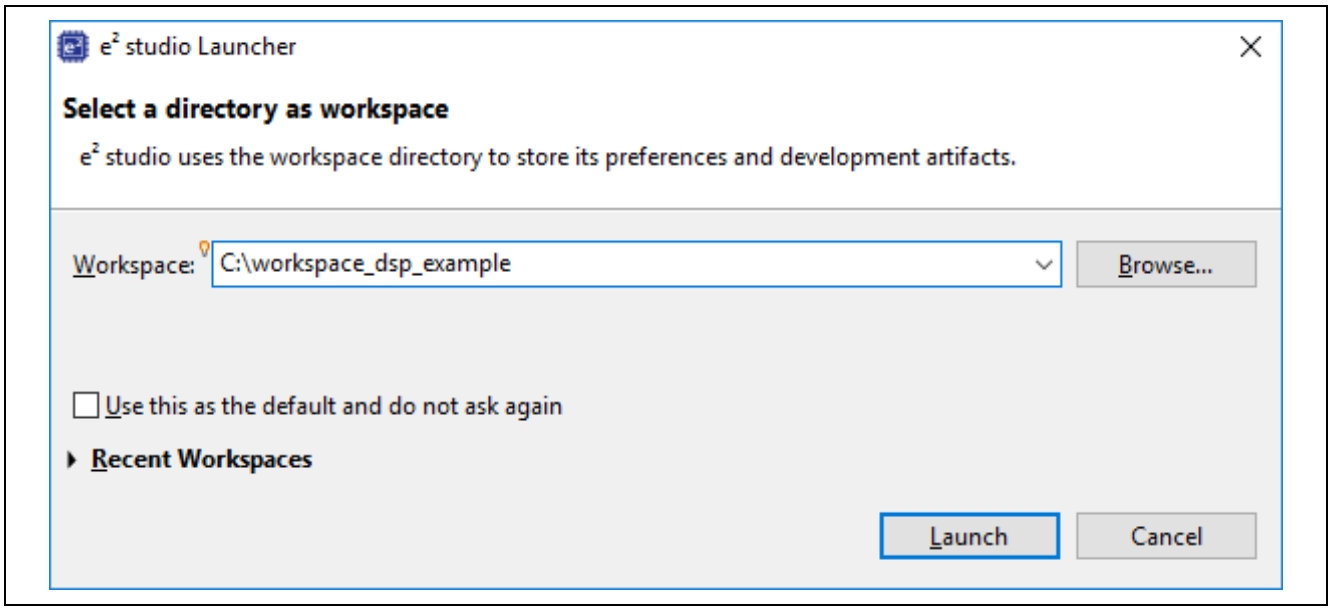
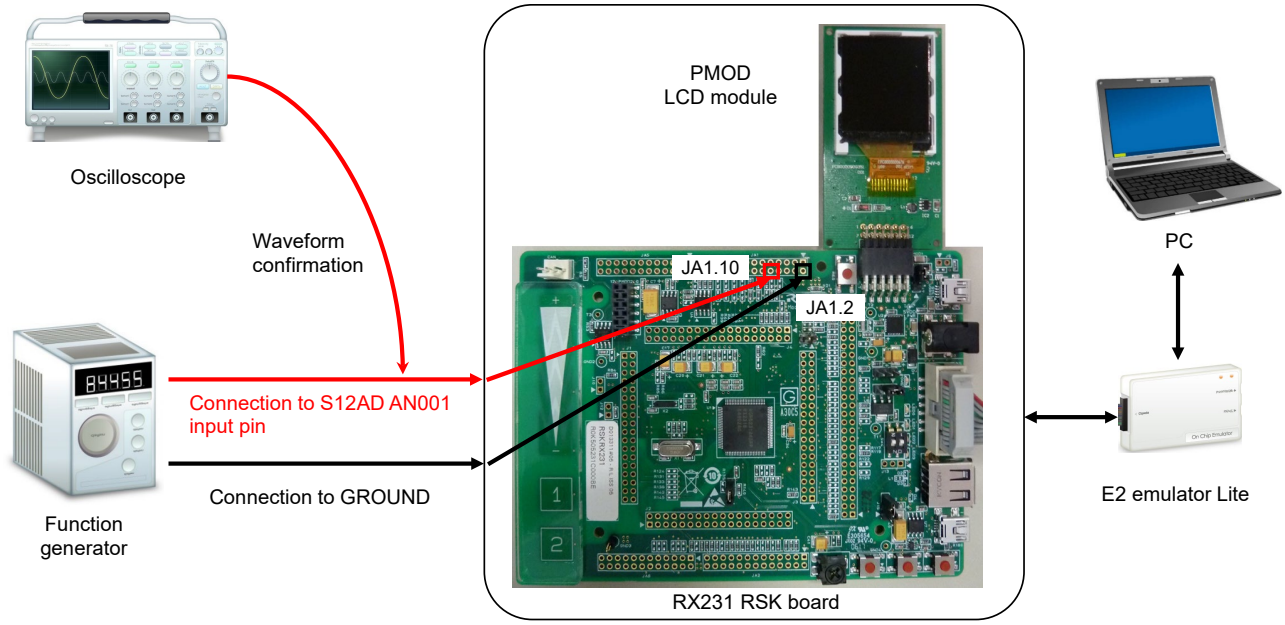


Figure 2.1 Selecting a Workspace

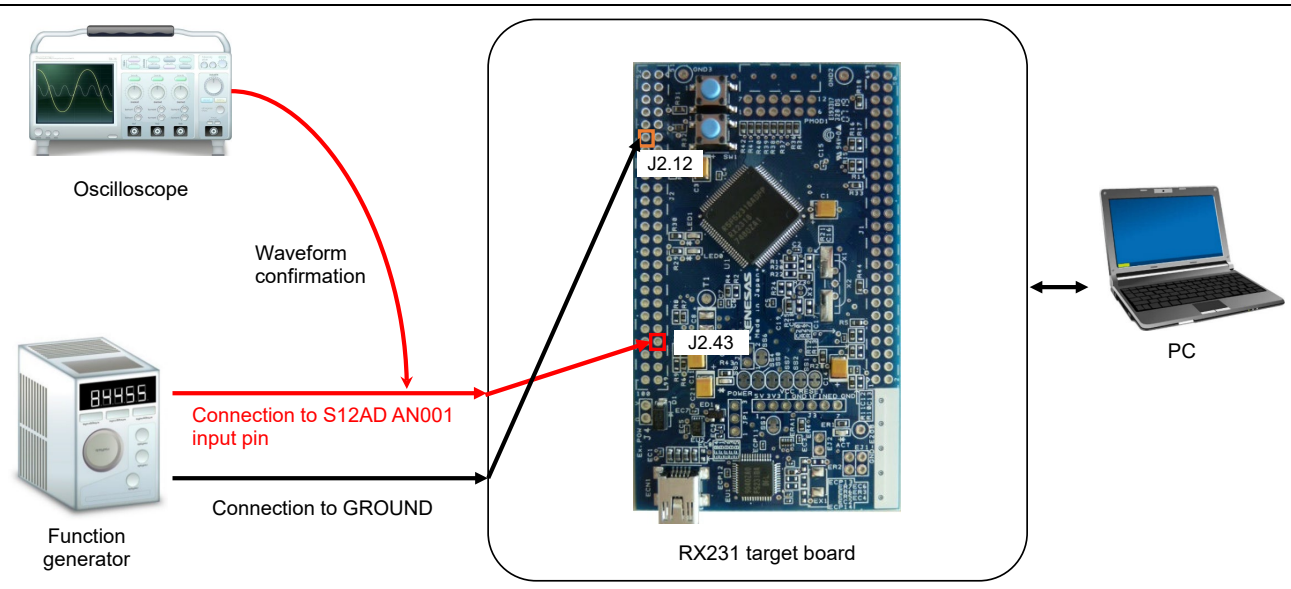
2.2 Connecting Equipment

Make connections as shown in Figure 2.2.



Note: The signal output terminals of some function generators may require terminating resistors. Refer to the manual of the function generator for confirmation.

Figure 2.2 Connections to RSK



Note: The signal output terminals of some function generators may require terminating resistors. Refer to the manual of the function generator for confirmation.

Figure 2.3 Connections to Target Board

2.3 Running the Sample Program and Checking Operation

Connect the debug tool and run the sample program.

2.3.1 Running the Sample Program

When you open the workspace, two projects are listed in Project Explorer, one for the RSK and one for the Target Board.

In the example below, the method of running the sample program for the RSK is shown.

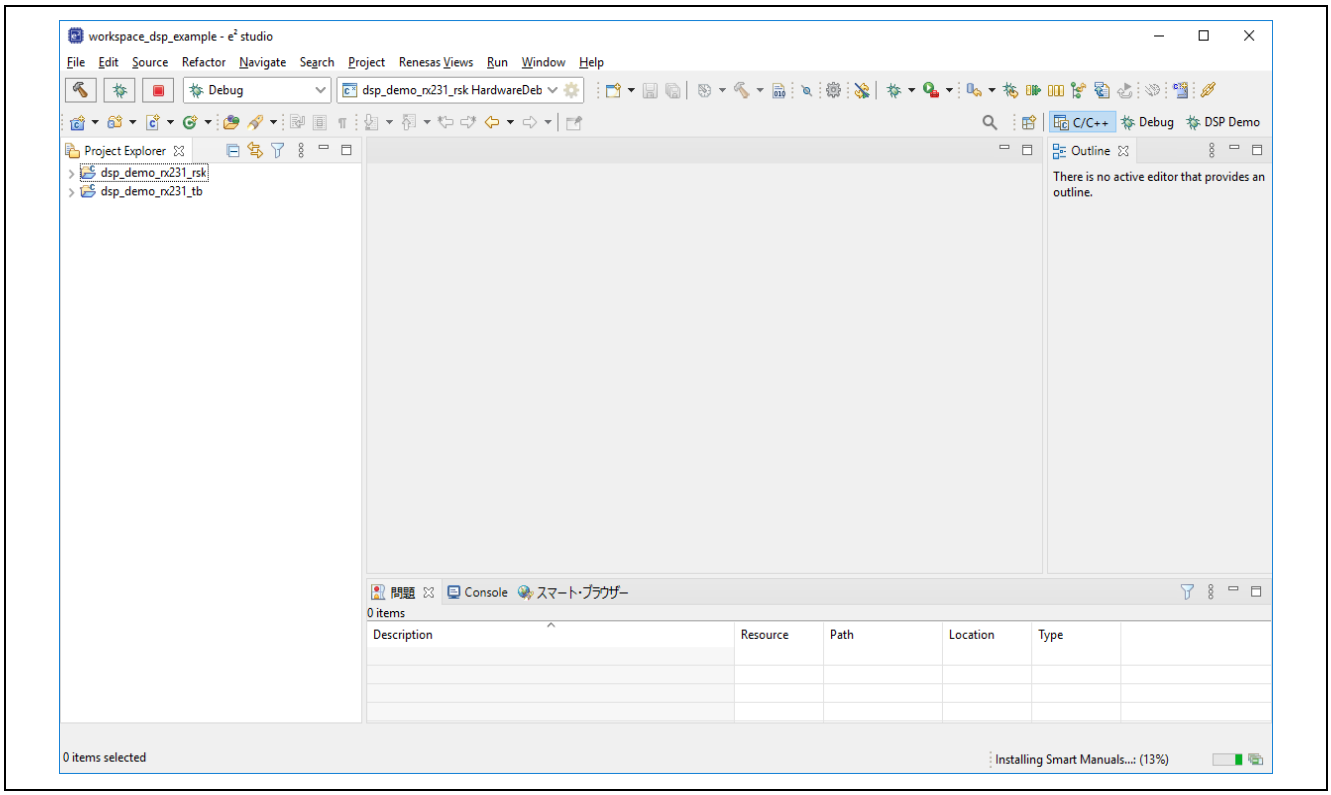


Figure 2.4 Workspace Immediately after Launch

From the **Launch Configuration** dropdown menu, select the project to connect to the debugger. After selecting the project, click the **Build** button and wait for building of the project to finish. When building of the project finishes, click the **Launch in Debug Mode** button to open the connection to the debugger.

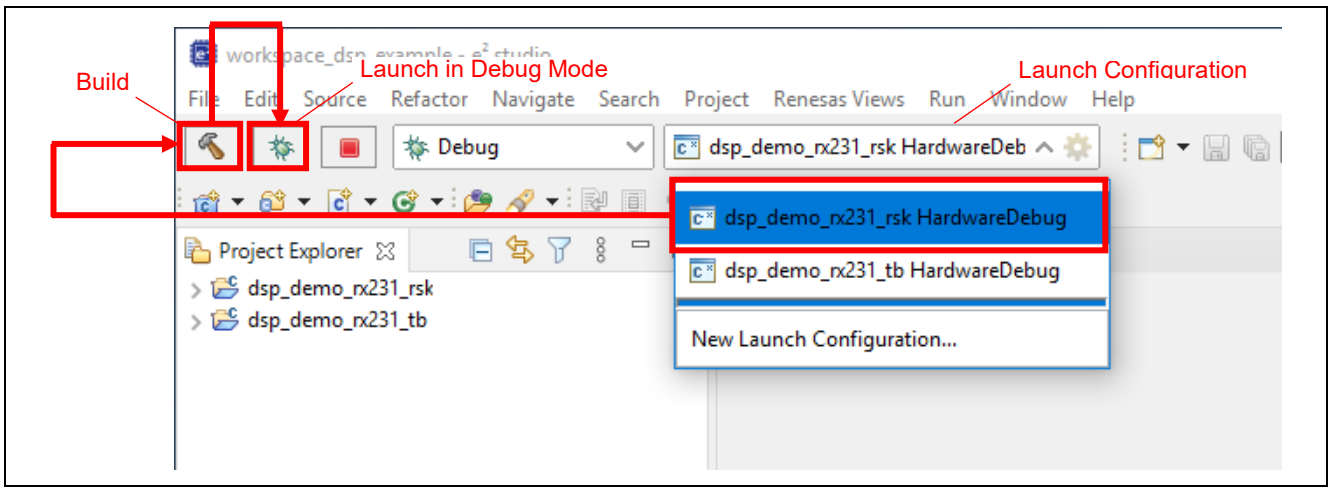


Figure 2.5 Operations

Shortly after the connection to the debugger is established, the **Confirm Perspective Switch** window appears. Click the **Switch** button.

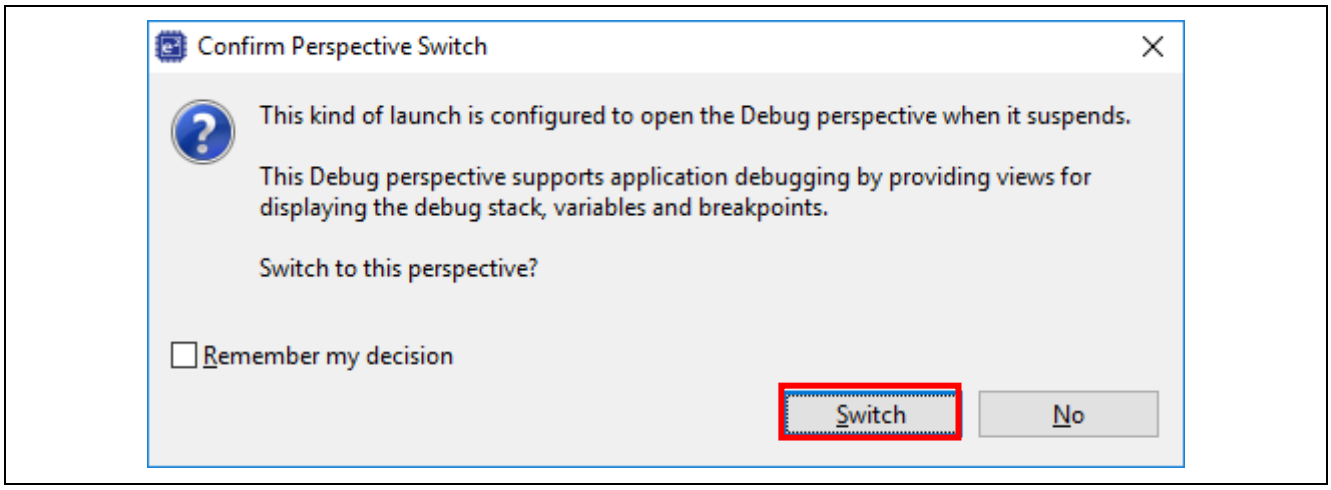


Figure 2.6 Confirm Perspective Switch

After the focus switches to the **Debug** perspective, click the **Resume** button. A break occurs at the start of the main function, so click the **Resume** button again to run the program. (A break occurs at the start of the main function only the first time the program is run after connecting to the debugger.)

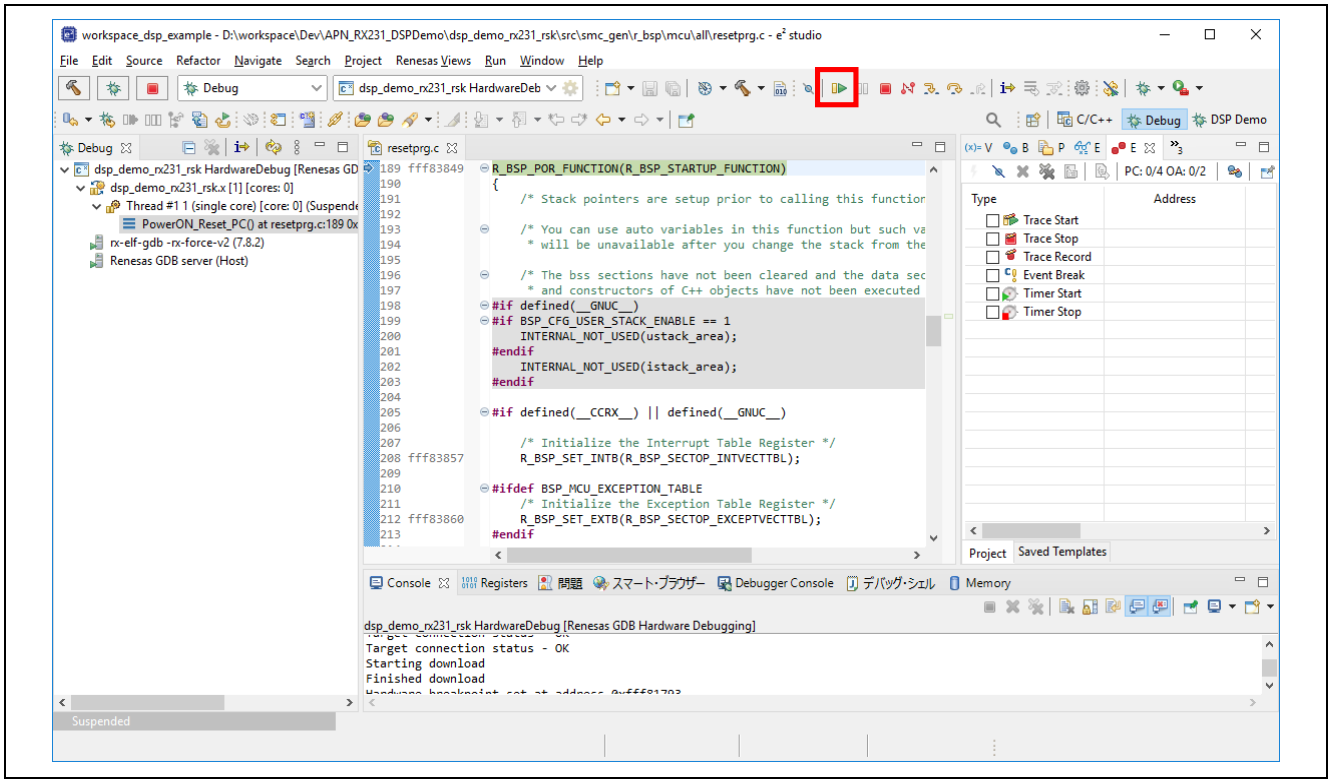


Figure 2.7 Workspace after Connecting to Debugger

After confirming that the program is running, proceed to inputting a signal waveform from the function generator.

2.3.2 LCD Display Contents and Changing Filter Characteristics

When you run the sample program, the processing results are displayed on the LCD module. For details of the display contents, refer to 3.4.5, Judgement of FFT Processing Results. The LCD module display is updated at fixed intervals (of approximately 1 second).

You can change the IIR filter characteristics by pressing a switch on the evaluation board (SW1) while the sample program is running. An indication of the filter characteristics currently being applied is displayed on the LCD module, and each press of the switch (SW1) causes the indication on the LCD module to change to the next item in the following sequence: FLAT (initial setting) → HPF → LPF → FLAT (back to the beginning), with the corresponding IIR filter processing applied.

The version of the sample program for the Target Board does not include the functionality to display information on the LCD module, but it is possible to use the debugging functions of e^2 studio to confirm the variables that are the basis for the information displayed on the LCD module.

2.3.3 Using e^2 studio Functions to Monitor System Operation

e^2 studio has many debugging functions including perspectives like monitoring system operation using the following functions.

- The Waveform function in the Memory view for waveform display
- Monitoring of global variable values in the Visual Expression view

After connecting the debug tool, click the **DSPDemo** button among the available perspectives (Figure 2.8). You can switch among the perspectives in this way.

Note that this perspective is included in the workspace setting information. Follow the procedure described in 2.1, Launching the Workspace, to use the workspace included in the distribution package with this application note.

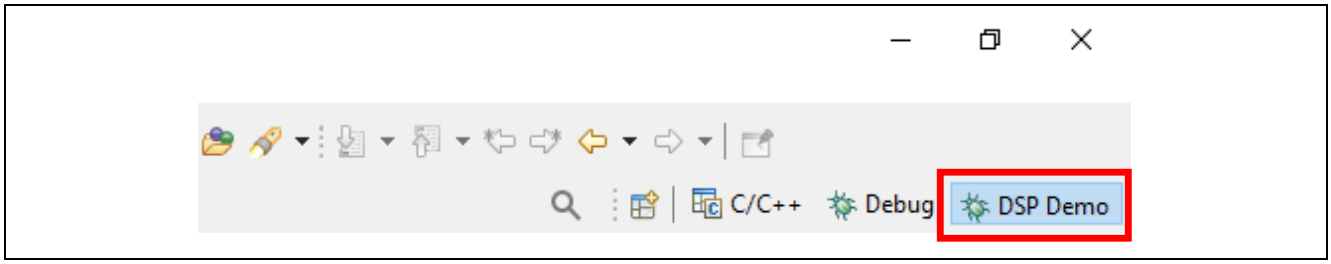


Figure 2.8 Switching among Perspectives

The perspective (e^2 studio screen configuration) used for the demo appears as shown in Figure 2.9. You can set expected values for frequency and level by dragging the sliders. Changes made using the sliders are reflected in the indications on the LCD module.

Figure 2.9 DSP Demo Perspective

Enable **Real-time Refresh** in the **Memory** view to update the display at the specified interval (Figure 2.10).

Figure 2.10 Enabling Real-Time Refresh

For information on how to use the **Waveform** function in the **Memory** view and the **Visual Expression** view, which are used in the demo perspective, refer to 5.1, Monitoring Signal Processing in e^2 studio, and the help system in e^2 studio.

Help → Help Contents → e^2 studio User Guide → Debugging Projects → Views

- Memory → Waveform Memory Rendering
- Visual Expression

2.4 User Changeable Settings

Table 2.1 lists the system settings that can be changed by the user. These settings are defined in the main.h file.

Table 2.1 Changeable Settings (main.h)

Function/Definition Name	Description	Default Value
Input data switching		
SAMPLE_DATA_MODE	Selects the signal source for IIR filter processing and FFT processing. 0: External input signal 1: Sample data 2: Sine wave generation processing	0
SELECT_SAMPLE_DATA	Selects the sample data used. Setting value: 1: 40 Hz and 400 Hz (mixed sine wave) 2: 250 Hz and 7.8125 Hz (mixed sine wave)	1
Judgement conditions for processing results		
EXPECTED_FREQUENCY	Expected value of peak frequency (the frequency of maximum magnitude according to FFT processing result) Setting unit: Hz Setting range: 1 to 499	150
EXPECTED_MAGNITUDE	Judgement threshold for frequency magnitude value set by EXPECTED_FREQUENCY Setting unit: Decibels (dB) Setting range*: –60.0 to 0.0	–6.6

* Make settings such that the value is 0 dB when the output signal from the S12AD is at maximum magnitude. For the decibel conversion method used by the sample program, refer to 3.4.6, Judgement of FFT Processing Results.

3. Description of Sample Program

3.1 Overview of Sample Program

Table 3.1 lists the processing performed by the sample program.

Table 3.1 Roles of Processing

Processing	Role
Main processing	• Initializes peripheral modules and DSP-related processing. • Starts the first DMA transfer. • Notifies DMA transfer end interrupt processing of next DMA destination address • Performs normalization processing of input data. • Performs DSP-related processing such as IIR filter processing and FFT processing. • Changes IIR filter characteristics in response to switch (SW1) input. • Judges FFT processing results and updates LCD module display. • Counts the number of cycles required for DSP processing and for inputting 1,024 samples of the input signal.
DMA transfer end interrupt processing	Makes settings for second and subsequent DMA transfers and starts transfer.
CMT0 interrupt processing	Redraws the LCD module display (controlled by the r_lcd_driver module).

3.2 Processing Sequence

The processing sequence of the sample program consists of two parts that operate independently: on the one hand the main processing and the DMA transfer end interrupt processing, and on the other the CMT0 compare match interrupt processing. Figure 3.1 shows the sequence for the main processing and DMA transfer end interrupt processing, and Figure 3.2 shows the sequence for the CMT0 compare match interrupt processing.

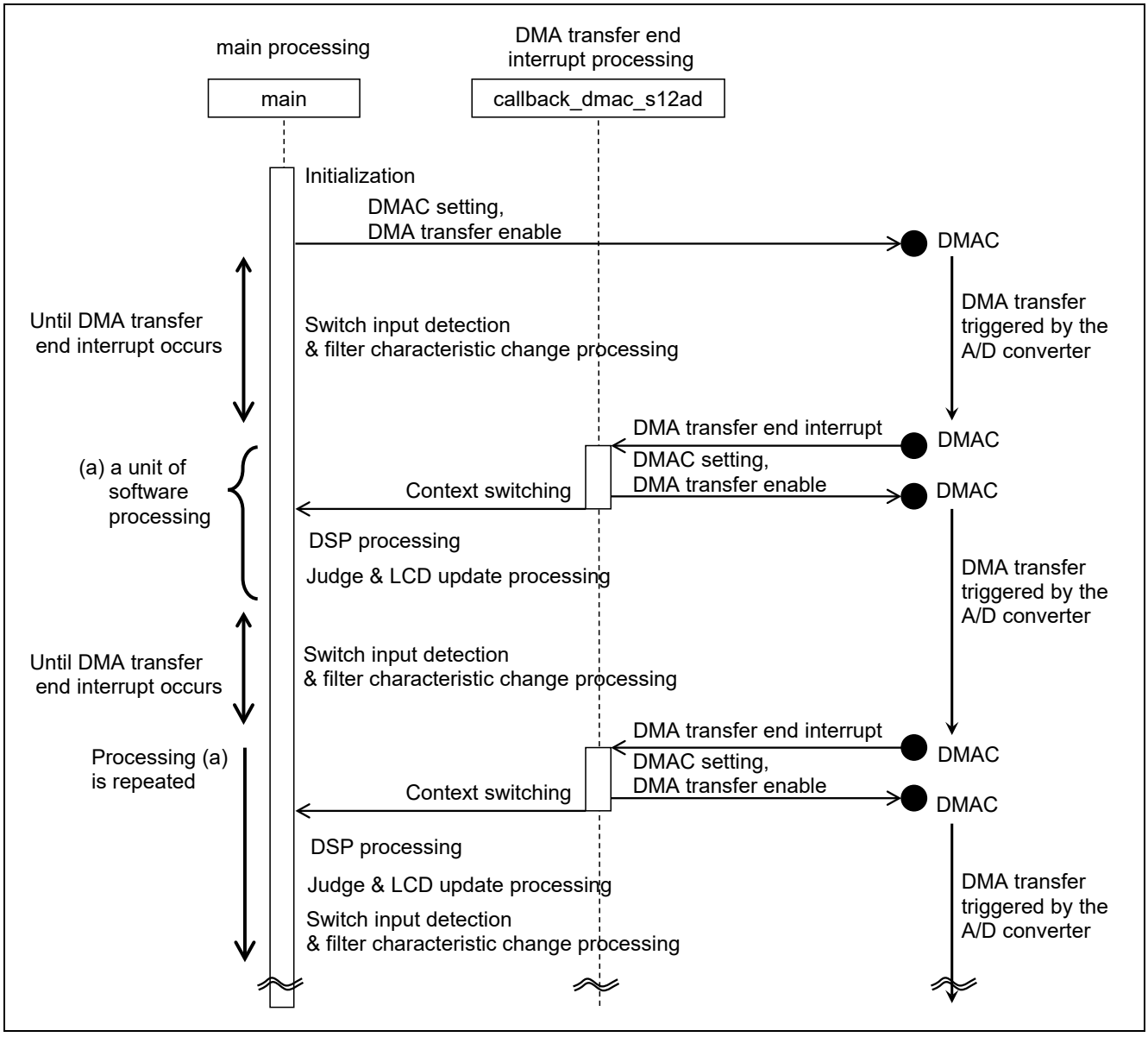


Figure 3.1 Sample Program Processing Sequence (Main Processing and DMA Transfer End Interrupt Processing)

As shown in the figure, the first DMA transfer is enabled when A/D conversion ends. When DMA transfer of the specified number of data units finishes, a DMA transfer-end interrupt request occurs. This triggers execution of the DMA transfer end interrupt processing and main processing, in that order. Thereafter, the same processing sequence is executed repeatedly.

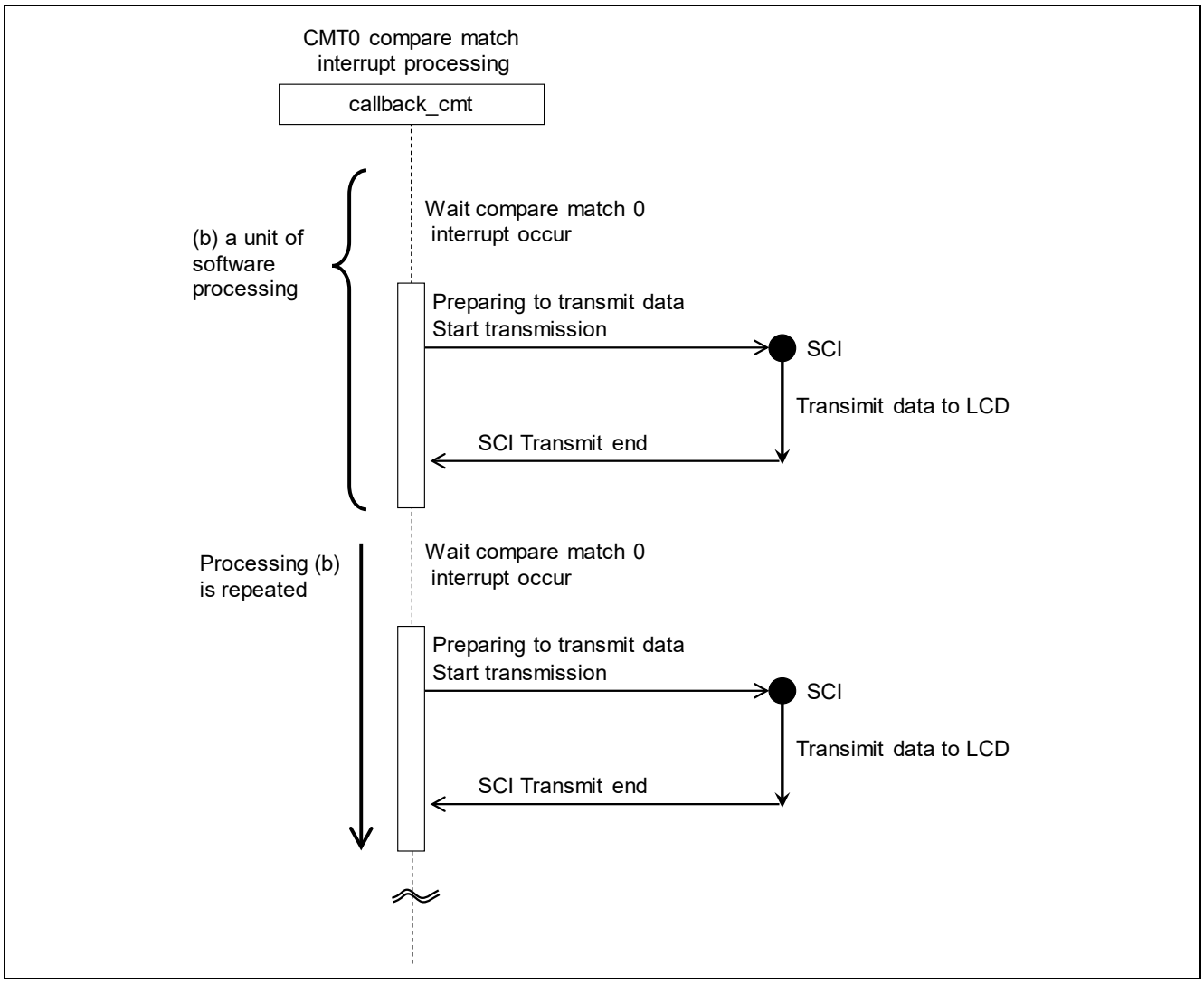
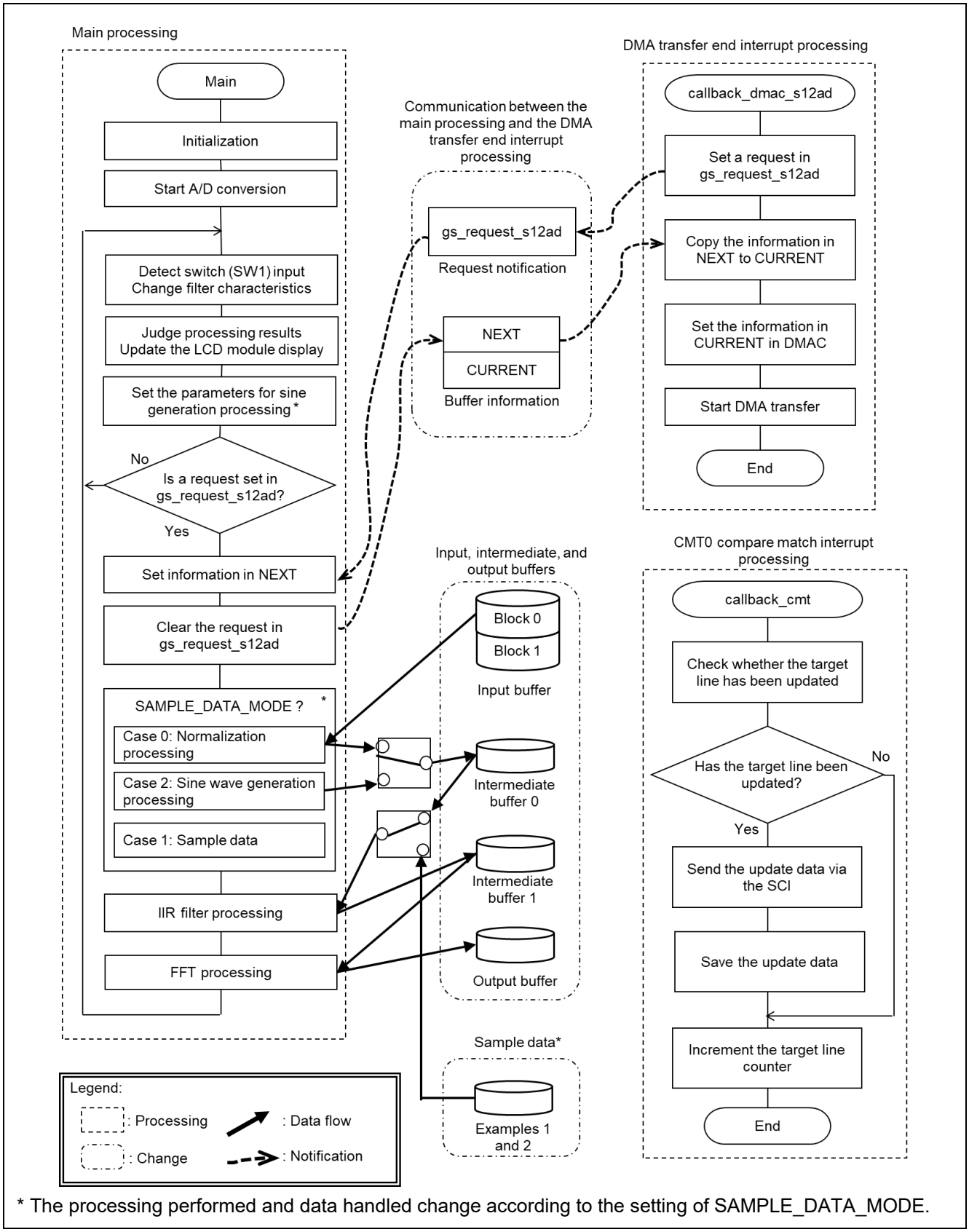


Figure 3.2 Sample Program Processing Sequence (CMT0 Compare Match Interrupt Processing)

After initializing the peripheral function, the main processing uses CMT0 and the SCI to process displaying indications on the LCD module. The CMT0 compare match interrupt processing uses the SCI to send one line of data to the LCD module. The display is updated, one line at a time, each time a CMT0 interrupt request occurs. The height of the LCD module is 128 pixels, so it takes 128 CMT0 interrupt requests to update the entire screen.

3.3 Processing Flow

Figure 3.3 shows the Processing Flow of the Sample Program.



* The processing performed and data handled change according to the setting of SAMPLE_DATA_MODE.

Figure 3.3 Sample Program Processing Flow

The elements of Figure 3.3 are described below.

- Main processing
  Initialization is performed first. After this, processing for switch (SW1) input detection and updating the filter characteristics takes place until the A/D conversion result is stored in a buffer by the DMAC. When a DMA transfer-end notification is received, the next DMA transfer destination is set, the input buffer data is normalized, DSP processing (IIR filter processing and FFT processing) is performed, and the information is updated based on the FFT processing results. Thereafter, the same processing sequence is executed repeatedly.
- DMA transfer end interrupt processing
  Sends a request to the main processing to update the next DMA transfer destination, and starts the current DMA transfer.
- Communication between main processing 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 3.2 lists variables used in the communication.
- Input buffer
  The DMAC stores the A/D conversion result here, and it is read as input data for normalization processing.
  It is configured as two blocks to avoid access conflicts between the CPU (normalization processing) and the DMAC. As shown in Figure 3.4, the DMA transfer-end interrupt is used as a trigger for switching between the buffer blocks accessed by the CPU and DMAC, respectively. Figure 3.5 shows the configuration of the input buffer.
- Intermediate buffer
  Stores the results of normalization processing and of IIR filter processing. Figure 3.5 shows the configuration of the intermediate buffer.
- Output buffer
  Stores the results of FFT processing. Figure 3.5 shows the configuration of the output buffer.
- Sample data
  Used when macro definition SAMPLE_DATA_MODE is set to 1. Sample data is used as the input for IIR filter processing instead of external signal input data.
- CMT0 compare match interrupt processing
  Updates what is displayed by the LCD module. The LCD module is 128 pixels wide and 128 pixels high, and one line is updated each time the CMT0 compare match interrupt processing runs. Each time a line is updated, it is displayed with the information for the currently displayed line and, if they differ, one line of updated data is sent to the LCD module. The updated data is then saved for comparison with the next data to be used to update the same line. If there is no difference, updating of the line is skipped.
- Sine wave generation processing
  Used when macro definition SAMPLE_DATA_MODE is set to 2. The sample program generates signal data that substitutes for external signal input data, and the generated data is used as the input of the IIR filter processing.

Table 3.2 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 processing. 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: 1,024
	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: 1,024

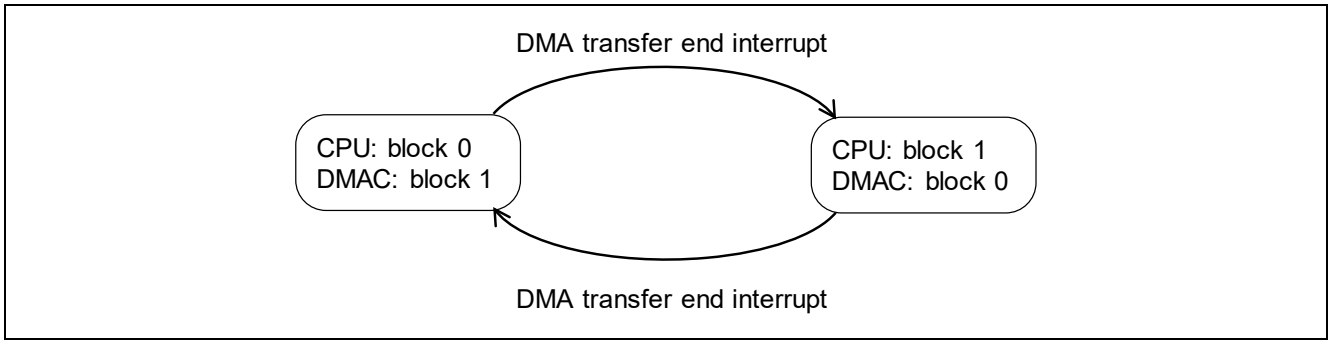
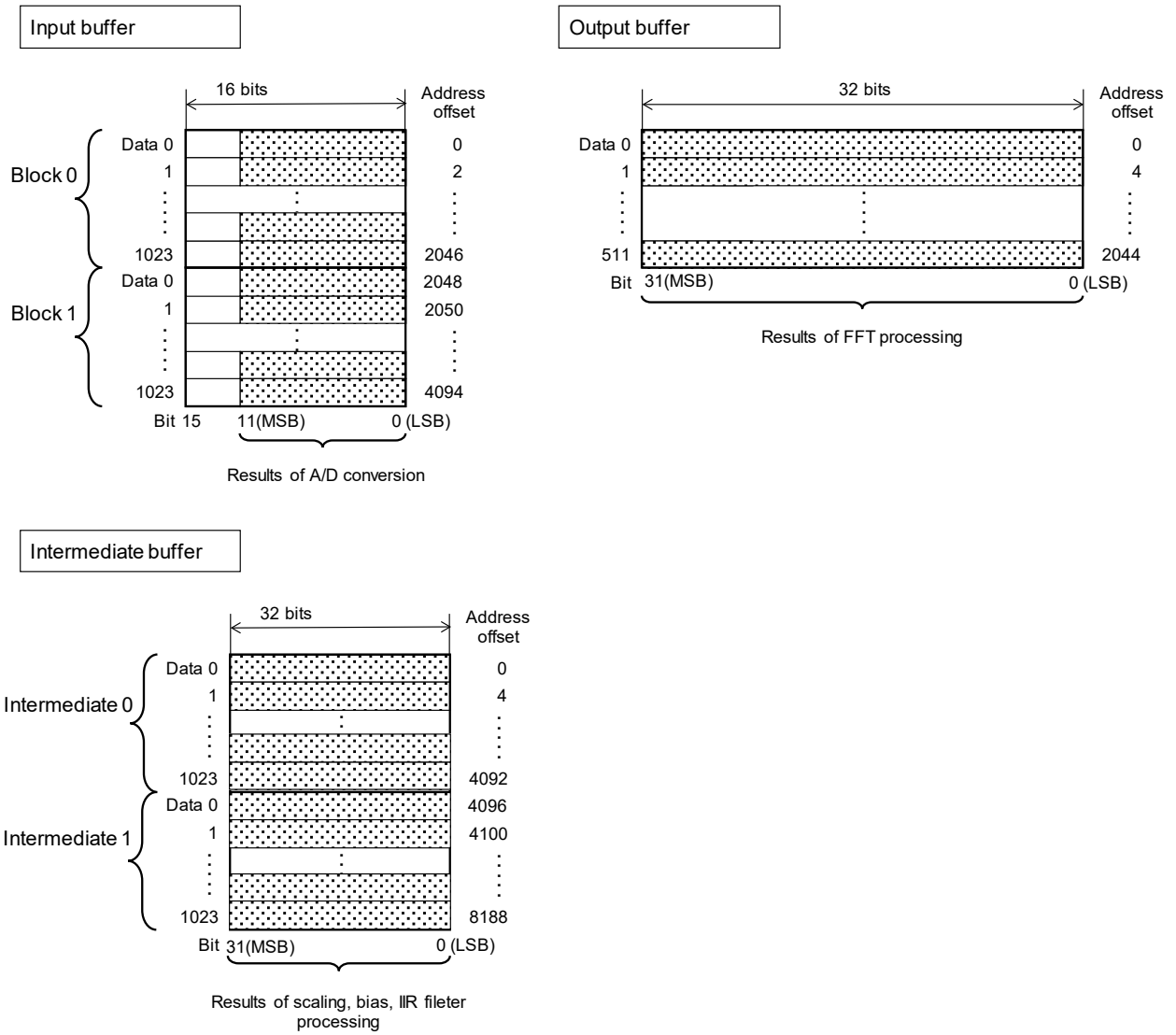


Figure 3.4 Switching Input Buffer Blocks Accessed by CPU and DMAC



Note: Due to the specifications of the real number FFT API in the DSP library, the point count of the FFT result is half that of the input. Therefore, the data count of the output buffer is one-half that of the input and intermediate buffers.

Figure 3.5 Input, Intermediate, and Output Buffers

3.4 Details

3.4.1 Signal Processing Flow

Figure 3.6 shows the signal processing flow of the sample program. Details of the portion of this processing performed by the DSP are detailed in 3.4.2 to 3.4.5.

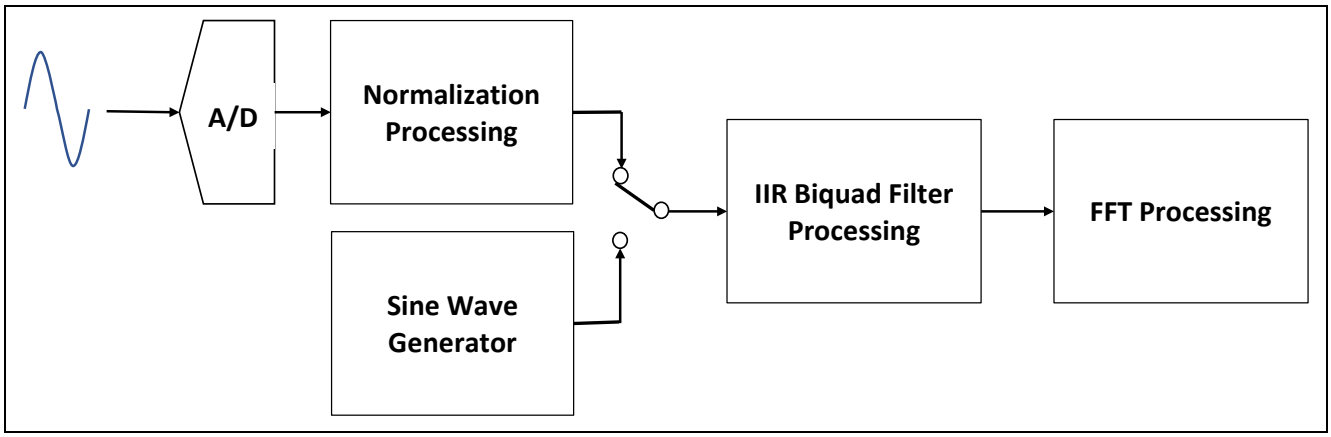


Figure 3.6 Signal Processing Flow

Table 3.3 Signal Processing Ranges

Analog Signal Input (A/D)	Normalization Processing	IIR Biquad Filter Processing	FFT Processing
0 to 2^{12}-1	-2^{30} to 2^{30}-1	-2^{30} to 2^{30}-1	-2^{29} to 2^{29}-1
Sine wave generation processing (Sine Wave Generator)			
-2^{30} to 2^{30}-1			

3.4.2 Normalization Processing

The sample program performs the normalization processing (bias processing and scaling) shown in Figure 3.7.

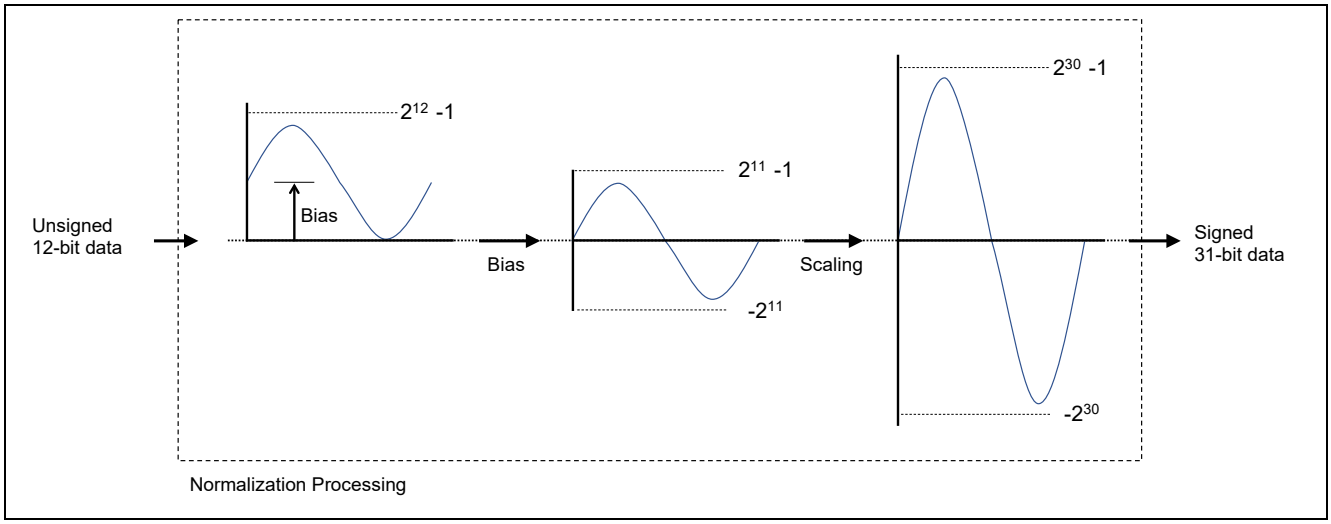


Figure 3.7 Normalization Processing

- Normalization processing
  The data input by the S12AD is in 12-bit (unsigned) format, so it must be normalized to 31-bit (signed) format in order to obtain adequate operation results from IIR filter processing and FFT processing. The normalization processing consists of bias processing and scaling.

3.4.3 Sine Wave Generation Processing

The sample program can generate sine waves by using the sine wave generator shown in Figure 3.8.

Examples of generating output signals are also shown later.

- Two oscillators for sine wave generation processing are installed.
  The frequencies and amplitudes of these oscillators can be adjusted and mixed. (For the adjustable ranges, see Table 3.4.)
- The oscillators are generated by the signal processing to which the secondary IIR filter shown in Figure 3.9 is applied.
- The range of possible values for the output signal of each oscillator is from -1073741824 to 1073741823 (31-bit signed integer)
- The signal resulting from mixing the two oscillators is limited in the range from -1073741824 to 1073741823 is output as 31-bit signed integer type.
- For details about the adjustment method, refer to 5.1.3, Adjusting the Parameters for Sine Wave Generation Processing.

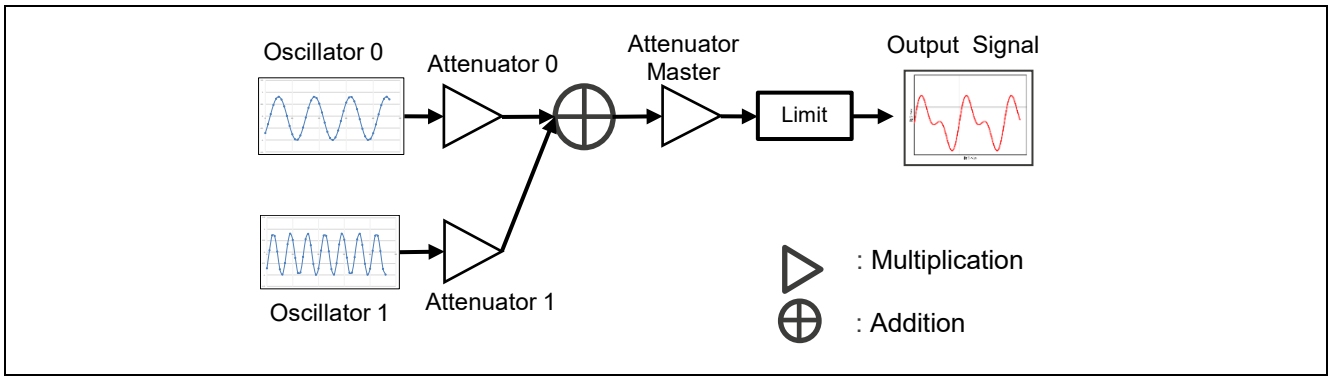


Figure 3.8 Sine Wave Generation Processing

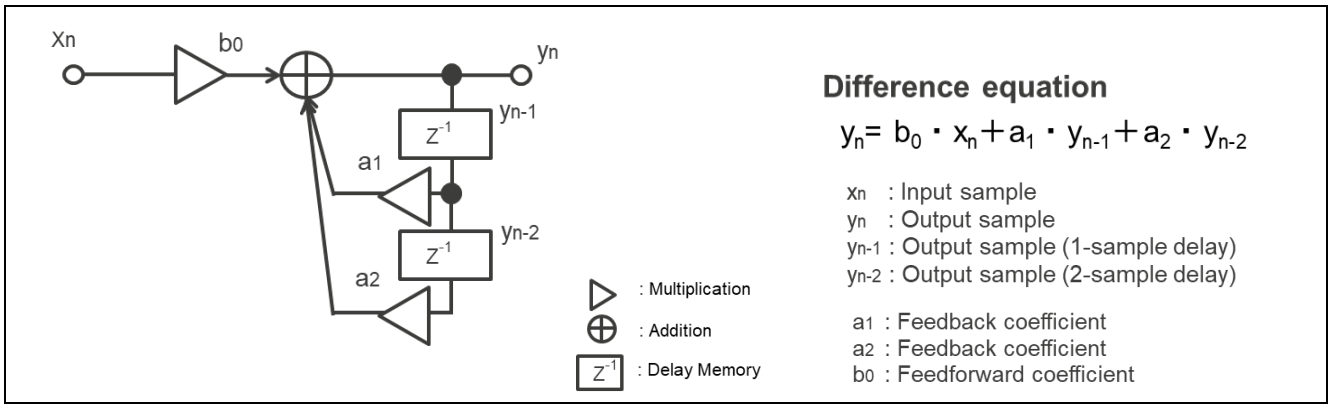


Figure 3.9 Signal Flow of Oscillator Processing to Which the IIR Filter Is Applied

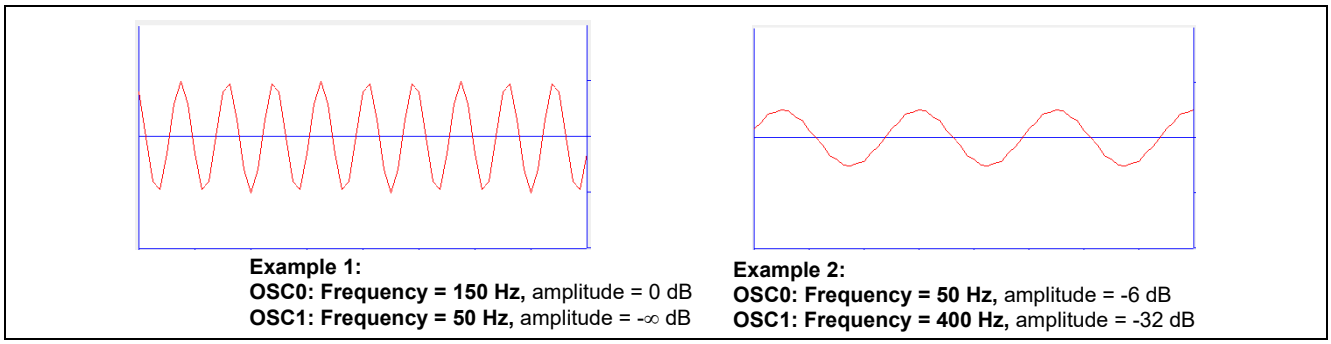


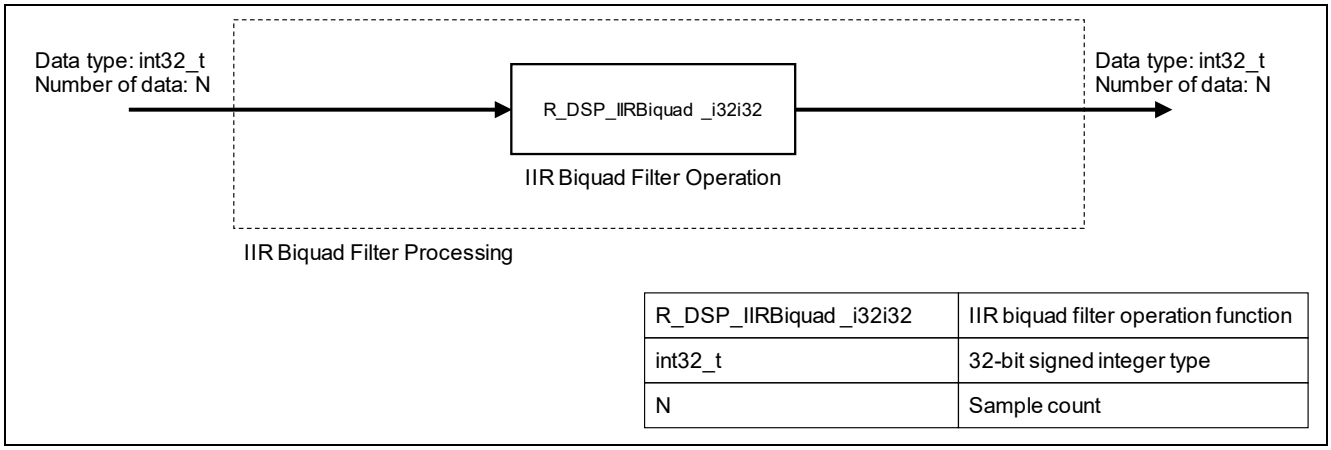
Figure 3.10 Signal Generation Examples

Table 3.4 Sine Wave Output Range

Item	Adjustable Range	Units of Adjustment
Sine wave frequency	0 to 490 Hz	10 Hz
Sine wave output level Output level of a mixed signal	Negative infinity (-∞), or -98 to 0 dB	-6 to 0 dB: 0.2 dB -24 to -6 dB: 0.5 dB -44 to -24 dB: 1 dB -56 to -44 dB: 2 dB -98 to -56 dB: 6 dB (Negative infinity is assumed to be 0.0.)

3.4.4 IIR Filter Processing

The sample program uses an API function from the DSP library to perform filter processing on the input signal shown in Figure 3.11.



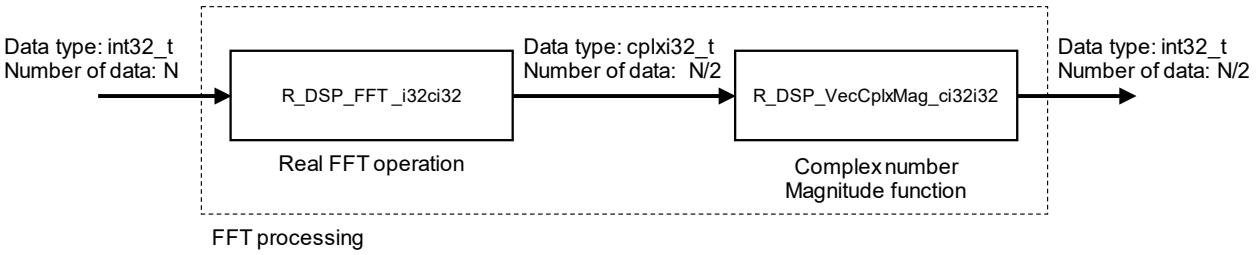
R_DSP_IIRBiquad _i32i32	IIR biquad filter operation function
int32_t	32-bit signed integer type
N	Sample count

Figure 3.11 Flow of Data in IIR Filter Processing

- API function comprising IIR filter processing
  The IIR filter processing is performed by the IIR biquad filter operation function R_DSP_IIRBiquad_i32i32. R_DSP_ IIRBiquad_i32i32 outputs operation results according to the filter coefficient settings. The sample program has three presets of filter characteristics pass-through (FLAT), low-pass filter (LPF), and high-pass filter (HPF). It is possible for software to change the filter characteristics set in the function during operation.
- Input and output data
  The sample program inputs 1,024 samples of data as type int32_t for IIR filter processing, and also outputs data as type int32_t.
  For details of the API specifications, refer to RX DSP Library Version 5.0 API User's Manual: Software.

3.4.5 FFT Processing

The sample program uses the DSP library API functions shown in Figure 3.12 to output the frequency spectrum magnitude characteristics of the input signal.



R_DSP_FFT _i32ci32	Real FFT Operation Function
R_DSP_VecCplxMag_ci32i32	Complex Number Magnitude Function
int32_t	32-bit signed integer type
cplxi32_t	32-bit complex number type
N	FFT point count

Figure 3.12 Flow of Data in FFT Processing

- APIs used for FFT processing
  FFT processing consists of the real FFT operation function R_DSP_FFT_i32ci32 and the complex number magnitude function R_DSP_VecCplxMag_ci32i32. R_DSP_FFT_i32ci32 outputs FFT results as complex number.
  In order to obtain the frequency spectrum magnitude characteristics of the input signal, the sample program converts the complex numbers output by R_DSP_FFT_i32ci32 into magnitudes expressed as integer values 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_i32ci32. The point count of results is 1/2 of the input in compliance with the specification of R_DSP_FFT_i32ci32. The sample program inputs data for 1,024 points and outputs results for 512 points. Also, since the output specification of R_DSP_VecCplxMag_ci32i32 is Q2.30 format, the output magnitude values are one-half the input values. 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 3.5, the magnitudes corresponding to frequencies are stored in the output buffer shown in Figure 3.5.
  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 / 1024 = 0.9765625 Hz

Table 3.5 Output Data corresponding to Frequencies

Data No.	Frequency [Hz]
0	0.97... × 0 = 0
1	0.97... × 1 = 0.97
2	0.97... × 2 = 1.95
:	:
510	0.97... × 510 = 498.04
511	0.97... × 511 = 499.02

3.4.6 Judgement of FFT Processing Results

A judgement is made based on the FFT processing results stored in the output buffer. The judgement is “pass” (OK) if the following two conditions are met:

- Peak frequency = expected frequency (definition EXPECTED_FREQUENCY)
- Peak frequency magnitude value (dB) ≥ expected magnitude value (dB) (definition EXPECTED_MAGNITUDE)

Peak frequency is the frequency of maximum magnitude according to the FFT processing result. The expected frequency is set in increments of 1 Hz, but the FFT processing output data and frequency are delineated in increments of approximately 0.97 Hz, as indicated in Table 3.5. Therefore, the expected frequency is converted to the corresponding data number to perform the comparison. The acceptable range is specified as extending one data number before and after the converted data number.

The magnitude value comparison is performed by converting the peak frequency magnitude value into decibel (dB) units.

The following formula is used for conversion:

$$\text{Magnitude value (unit: dB)} = 20 * \log_{10} (\text{magnitude value after FFT processing} / 2^{29} - 1)$$

Note that the measurement result may not be exactly 0 dB due to factors such as the computational precision of A/D conversion and signal processing. For information on computational precision, refer to 3.4.1, Signal Processing Signal Processing Flow.

3.4.7 Processing to Change Display on LCD Module

Table 3.6 lists the information displayed on the LCD module. Processing to change the display is divided between the following two processing:

- Main processing
  Calculates data needed to change the display based on the contents of the output buffer and other information. Converts the calculated data to display data and writes it to the display update buffer.
- CMT0 interrupt processing
  Reads the display data from the display update buffer and uses it to update the LCD module, one line at a time each time a CMT0 interrupt occurs, as described in 3.2, Processing Sequence.

Table 3.6 Information Displayed on LCD Module

Item	Display Example	Description
IIR filter characteristics	IIR: FLAT	Filter characteristics FLAT, LPF, or HPF is displayed.
Peak frequency	Meas.Freq: 250 Hz	Measurement value of peak frequency
Peak frequency magnitude value	Meas.Mag: –6.6 dB	Measurement value peak frequency magnitude Expressed in dB to the first decimal place
Expected frequency	Exp.Freq: 250 Hz	Expected value of peak frequency
Expected magnitude value	Exp.Mag: –65.7 dB	Threshold at which peak frequency magnitude value is judged OK (pass) Expressed in dB to the first decimal place
Judgement	Judge: NG	Judgement result OK (pass) or NG (fail) is displayed.*
DSP processing cycle count	DSP Proc: 89,528 Cyc	DSP processing (normalization processing, IIR filter processing, and FFT processing) cycle count
Input signal input cycle count	Unit Proc: 27,647,946 Cyc	Cycle count for inputting 1,024 samples of the input signal

DSP processing CPU occupancy rate	CPU Occupy: 0.324%	Displays (DSP-related processing cycle count / cycle count for inputting 1,024 samples of the input signal) as a percentage (%).

* Even if the expected frequency and the peak frequency do not match, OK (pass) may be displayed because the judgment explained in 3.4.6, Judgement of FFT Processing Results, has an acceptable range.

3.4.8 Processing for Changing IIR Filter Characteristics

You can change the IIR filter characteristics by pressing a switch on the evaluation board (SW1) while the sample program is running. When the switch (SW1) is depressed, an IRQ interrupt request is generated and a flag used for detection judgement is set to 1 in the interrupt processing. The detection judgement flag is checked in the main processing, and, if it has been set, the IIR filter characteristics are changed and the flag is cleared to 0. After the change, the IIR filter characteristics are displayed as shown in Table 3.6.

3.4.9 Processing Cycle Count Measurement

The processing cycle counts for DSP processing (normalization processing, IIR filter processing, and FFT processing), and for inputting 1,024 samples of the input signal, are measured. These measurement results are then used to calculate the CPU occupancy rate associated with DSP processing by the sample program. The result of this calculation is displayed on the LCD module. Figure 3.13 shows the measurement periods.

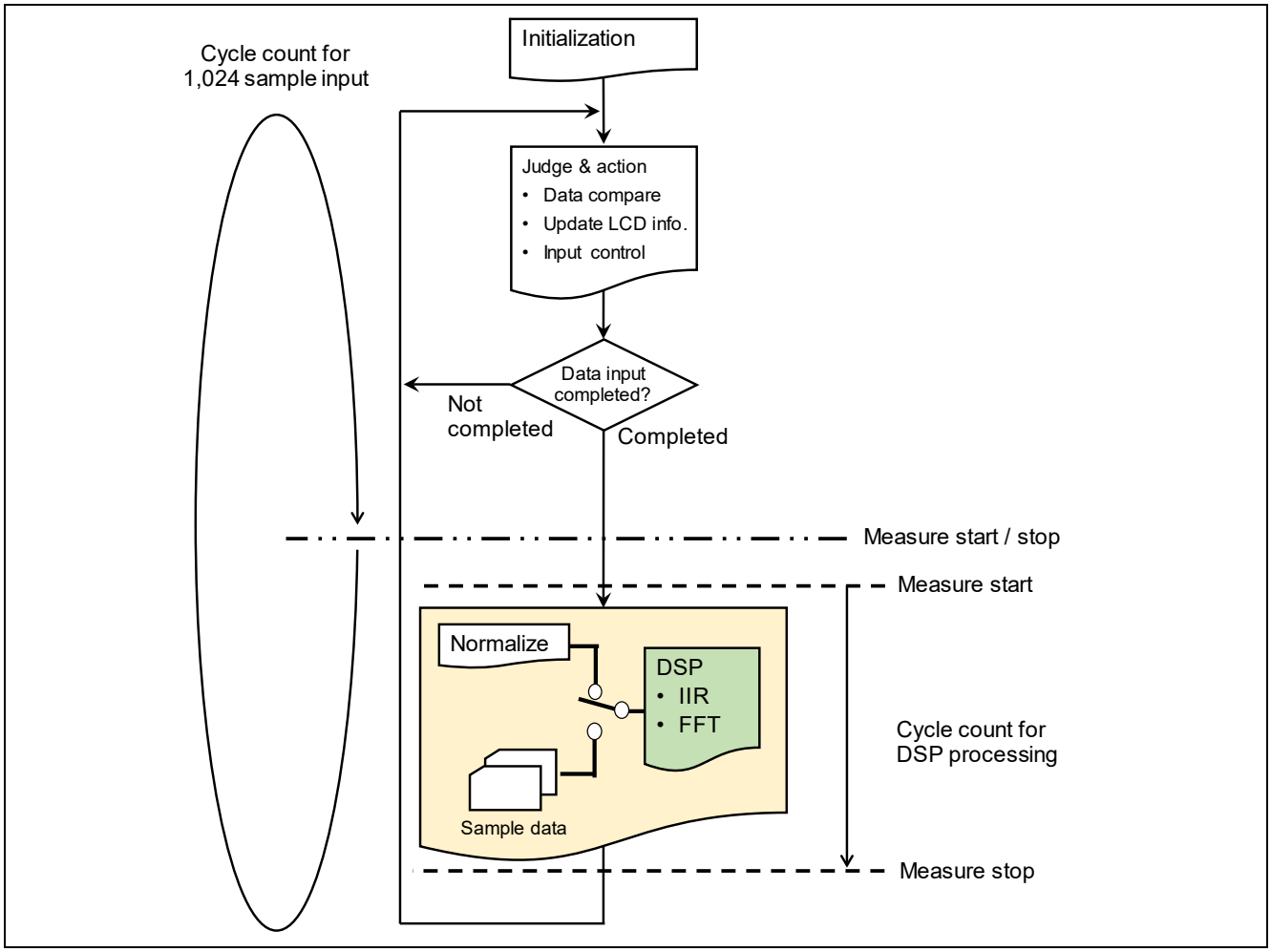


Figure 3.13 Cycle Count Measurement Periods

3.5 File Structure

Table 3.7 shows the file structure of the modules described in this application note. Table 3.8 to Table 3.12 list the variables used in the source files. For details of the other modules, refer to their respective application notes, and for settings, see 5.3, Software Module Settings.

Table 3.7 File Structure of Modules

Module/File		Description
main		Main processing of sample program
	main.c	Main processing, DMA transfer end interrupt processing, etc.
	main.h	Header file for the main.c file
r_dsp		DSP-related processing
	r_normalize.c	Normalization processing
	r_normalize.h	Header file for the r_normalize.c file
	r_dsp_iirbiquad.c	IIR filter processing initialization, IIR filter processing
	r_dsp_iirbiquad.h	Header file for the r_dsp_iirbiquad.c file
	r_dsp_real_fft.c	FFT processing initialization, FFT processing
	r_dsp_real_fft.h	Header file for the r_dsp_real_fft.c file
	r_signal_gen.c	Sine wave generation processing
	r_signal_gen.h	Header file for the r_signal_gen.c file
	r_signal_gen_att_coef.c	Frequency table for sine wave generation processing
	r_signal_gen_freq_coef.c	Output level table for the sine wave generator
r_lcd_driver		LCD module control processing (for RSK)
	r_ascii.c	Character table
	r_ascii.h	Header file for the r_ascii.c file
	r_lcd_driver.c	LCD module control processing
	r_lcd_driver_private.h	Private header file
	r_lcd_driver_if.h	Interface header file

Table 3.8 main.c File Functions

Function Name	Description
main	• Initializes peripheral devices. • Starts first DMA transfer. • Notifies next DMA transfer destination to DMA transfer end interrupt processing. • Performs DSP processing. • Displays judgement of processing results on LCD module.
set_buf_info	Sets the next DMA transfer destination address and data count in the variables used for communication between the main processing and the DMA transfer end interrupt processing.
init_dmac_s12ad	Initializes DMAC channel 0 with S12AD as the transfer source.
callback_dmac_s12ad	The callback function registered in the DMAC module by the main processing • DMAC channel 0 DMA transfer end interrupt processing • Sets the next transfer destination in the DMAC and enables the DMA transfer. • Requests the next transfer destination information from the main processing.
init_lcd_display	Initializes the LCD module and sets the initial display information. (for RSK)
change_coef	Performs processing to change the filter characteristics.
judge_and_update	Makes a judgement based on the FFT results and updates the information displayed on the LCD module.

cycle_measure_start	Starts measurement of the processing cycle counts for DSP processing and for inputting 1,024 samples of the input signal.
cycle_measure_stop	Stops measurement of the processing cycle counts for DSP processing and for inputting 1,024 samples of the input signal, calculates the cycle counts, and returns the values.

Table 3.9 r_normalize.c File Functions

Function Name	Description
R_Normalize_Operation	Normalization processing

Table 3.10 r_dsp_iirbiquad.c File Functions

Function Name	Description
R_DSP_IIRBiquad_Init	Initializes IIR filter processing.
R_DSP_IIRBiquad_Operation	Executes IIR filter processing.
R_DSP_IIRBiquad_UpdateCoef	Sets the IIR filter characteristics and clears the delay data array.

Table 3.11 r_dsp_real_fft.c File Functions

Function Name	Description
R_DSP_REAL_FFT_Init	Initializes FFT processing.
R_DSP_REAL_FFT_Operation	Executes FFT processing.

Table 3.12 r_lcd_driver.c File Functions

Function Name	Description
R_LCD_Driver_Open	Initializes the LCD module and driver.
R_LCD_Driver_PrintString	Writes character strings to the update buffer.
R_LCD_Driver_PeriodicCallback	Processing called from the CMT0 interrupt processing to update one line of the LCD module.

Table 3.13 r_signal_gen.c File Functions

Function Name	Description
R_Signal_Gen_Init	Initializes the sine wave generation processing.
R_Signal_Gen_Operation	Performs the sine wave generation processing.
R_Signal_Gen_osc_freq	Changes the sine wave frequency.
R_Signal_Gen_osc_att	Changes the sine wave output level.
R_Signal_Gen_master_att	Changes the output level of the mixed signal.

4. Usage Notes

4.1 Frequency Values of FFT Processing Results

The RX231 on the Target Board operates using HOCO as the clock source. HOCO has a maximum error of ±2%, and this is the sampling frequency error of the A/D conversion processing performed by the sample program. Since the sampling frequency error shows up as frequency error in the FFT processing results, the demo system's peak frequency display may be off by a few Hz from the actual input signal frequency. For applications requiring a system capable of more accurate frequency analysis than the demo system, use a highly accurate oscillator as the clock source of the RX231 and make corresponding changes to the clock settings of the sample program.

4.2 Aliasing

For the evaluation board used in this document, no countermeasures are taken for the aliasing of signals that are used as the input of A/D conversion of the RX231. Note that aliasing occurs if the frequency of the signal input for A/D conversion exceeds 1/2 of the sampling frequency. If you use this document for reference when designing a system, take preventive measures for aliasing as needed. For example, add an external anti-aliasing filter to the RX231.

4.3 Chronological Change in Oscillator Output Signals

The sine wave generation processing of the sample program was designed on the assumption that it would be used for simple operational verification or demonstration. Continuous long-time use of the processing arises a chronological change in the output signals of the oscillators, which is an unavoidable structural problem with the oscillators. Before you apply the oscillators in this document to your system, consider the applicability in light of the system requirements.

5. Reference

5.1 Monitoring Signal Processing in e² studio

5.1.1 Waveform Rendering

You can use the waveform rendering function of e² studio to monitor the signal input to the RX231 and the FFT processing results.

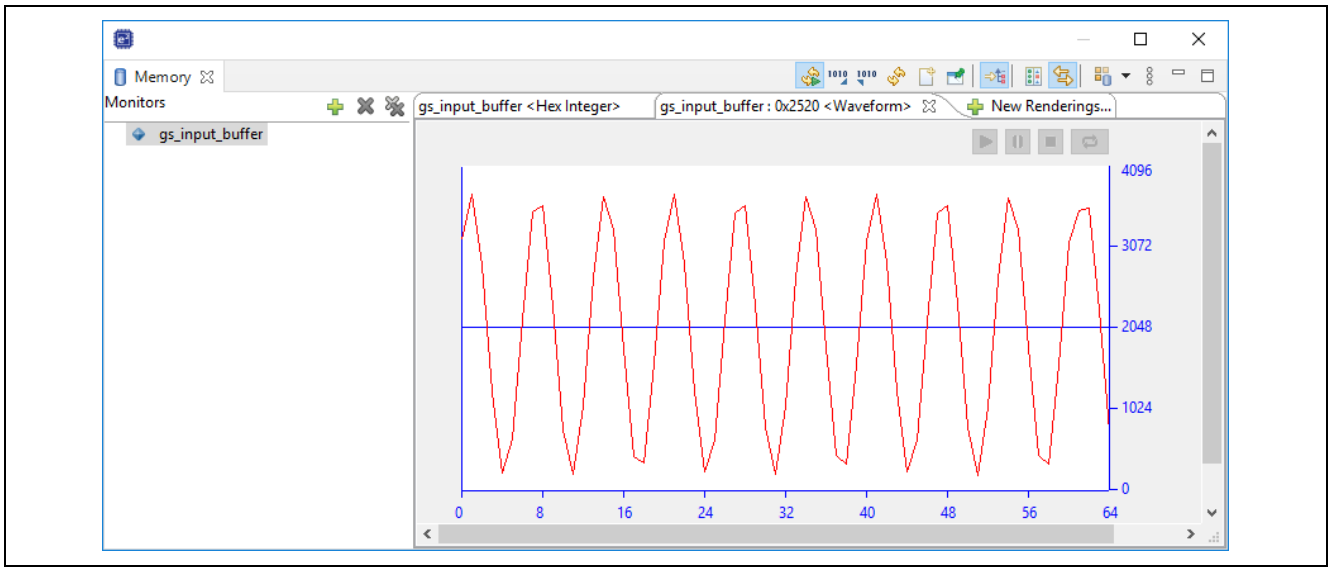


Figure 5.1 Waveform Rendering Display Example (Data Stored in Input Buffer)

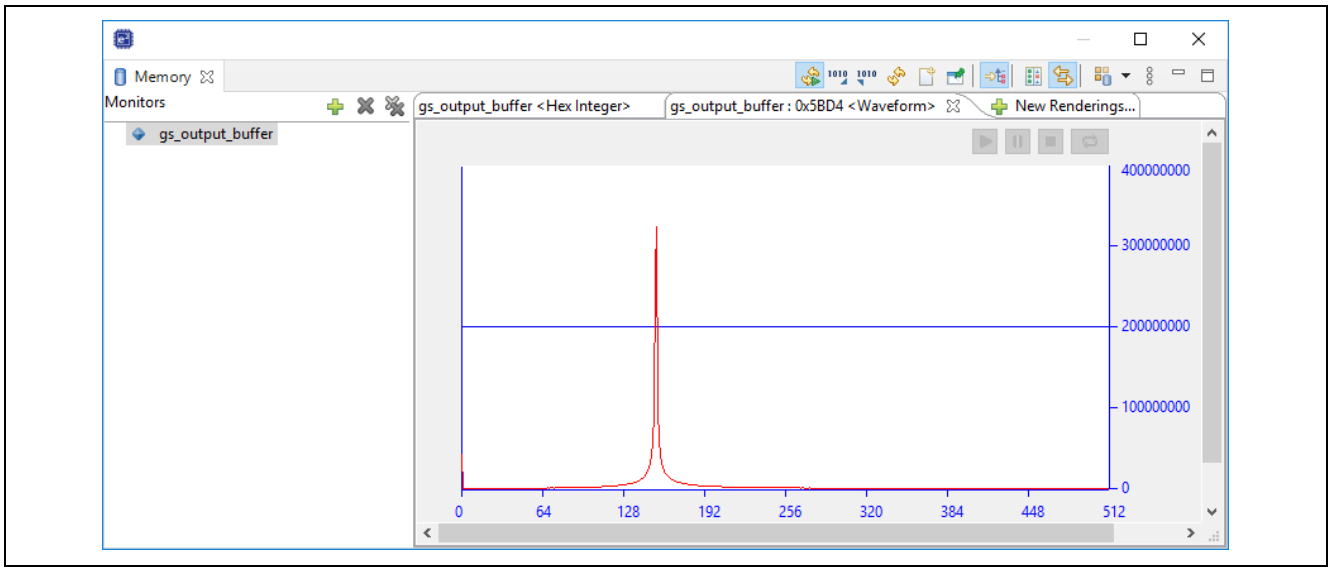


Figure 5.2 Waveform Rendering Display Example (Frequency Magnitude Characteristics Produced by FFT Processing)

After connecting the RX231 to the debugger, select **Window** → **Show View** → **Memory** to display the Memory view. When the **Memory** view appears, specify the input buffer (gs_input_buffer) as the variable to monitor. After adding the input buffer, specify the output buffer (gs_output_buffer) in the same manner.

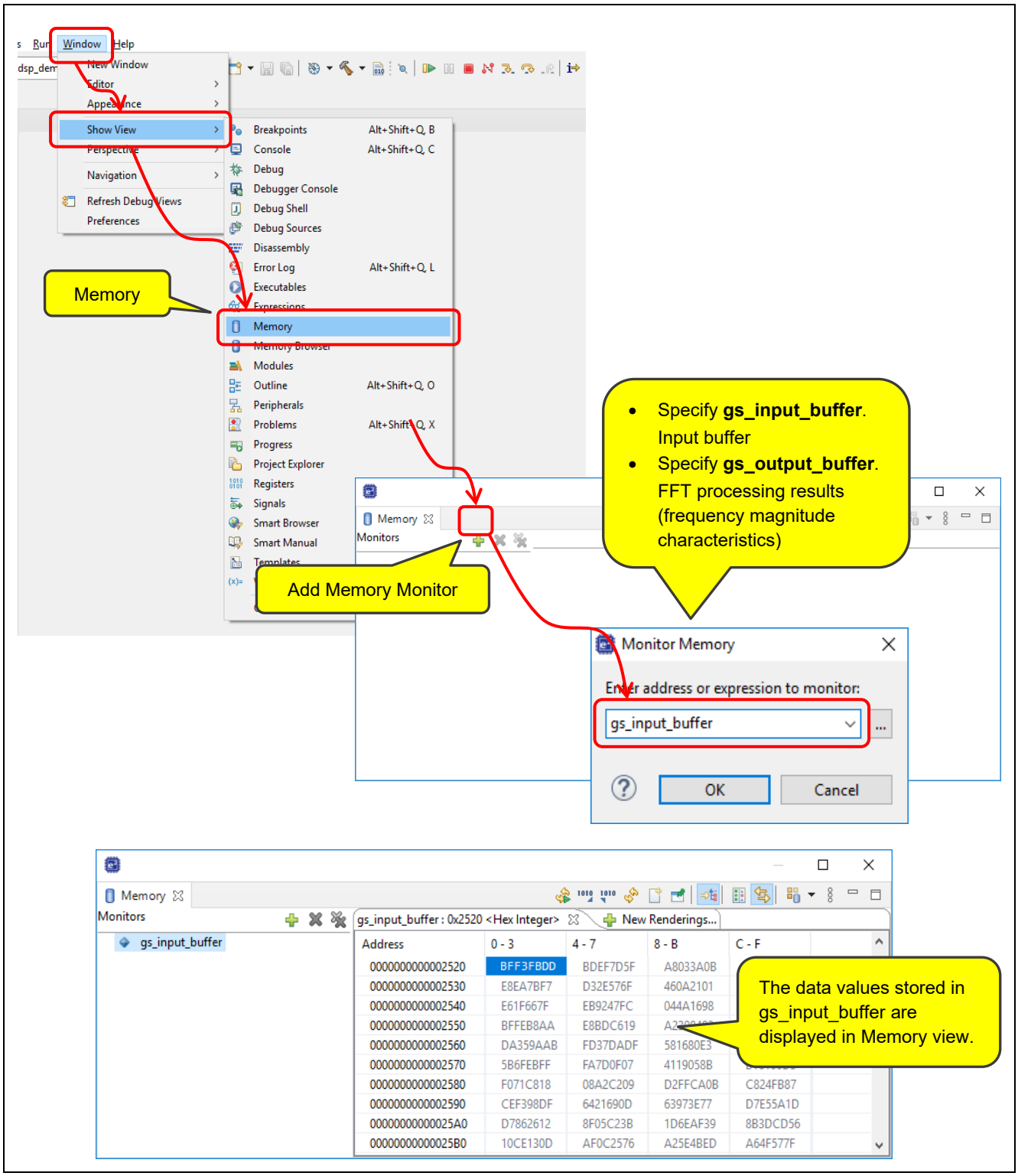


Figure 5.3 Memory View Display Procedure

Next, select **New Renderings...** → **Waveform** → **Add Rendering(s)** to choose a variable to be displayed graphically. When the **Waveform Properties** window appears, enter the necessary settings for the variable, and finally click the **OK** button to show the graphical display.

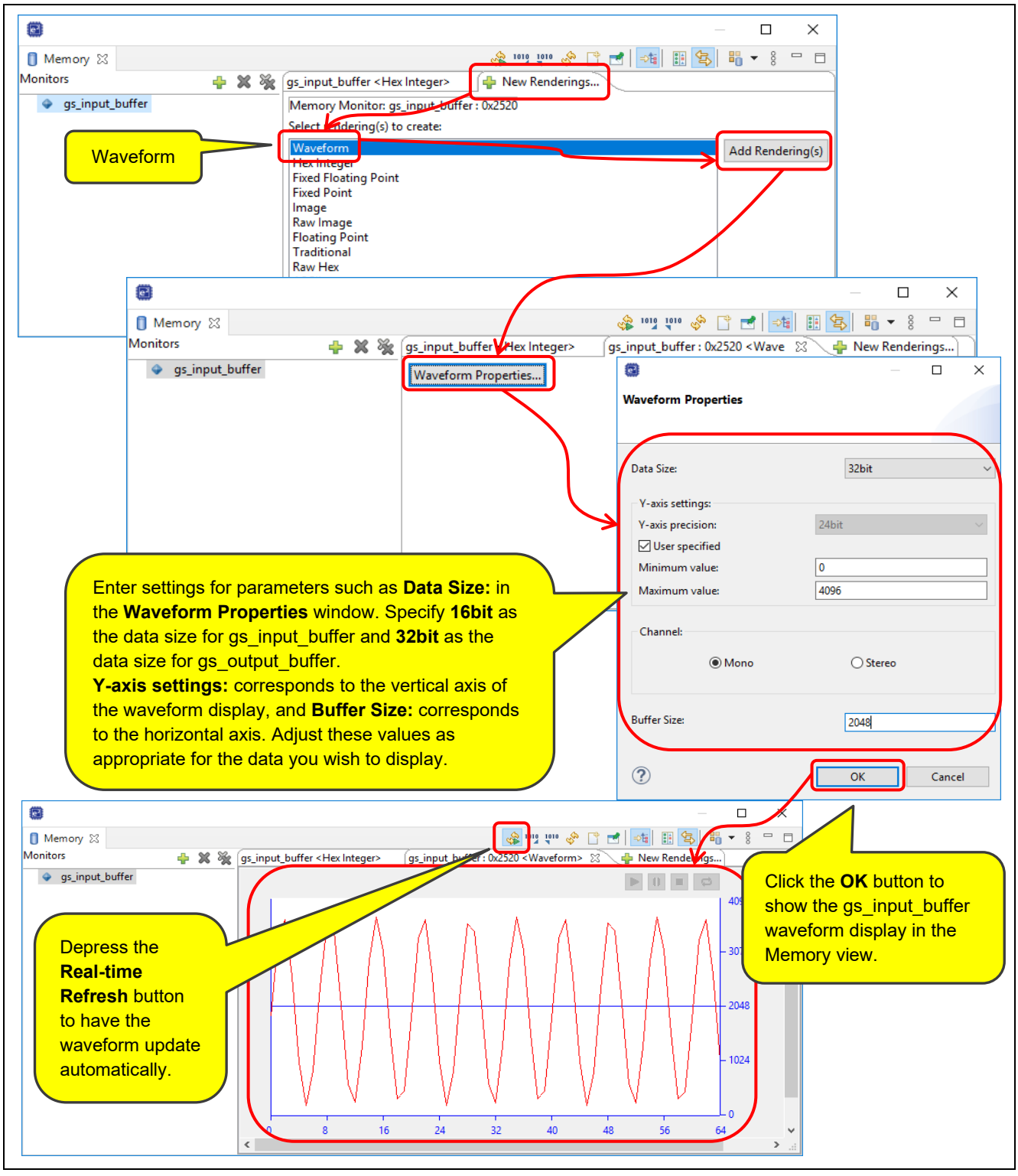


Figure 5.4 Waveform Rendering Display Procedure

5.1.2 Visual Expression

You can use the Visual Expression function of e^2 studio to monitor formulas in the code visually. Formulas that are expressed visually are called “visual elements” and are of three types: **Gauges and meters**, **Controllers**, and **Indicators**.

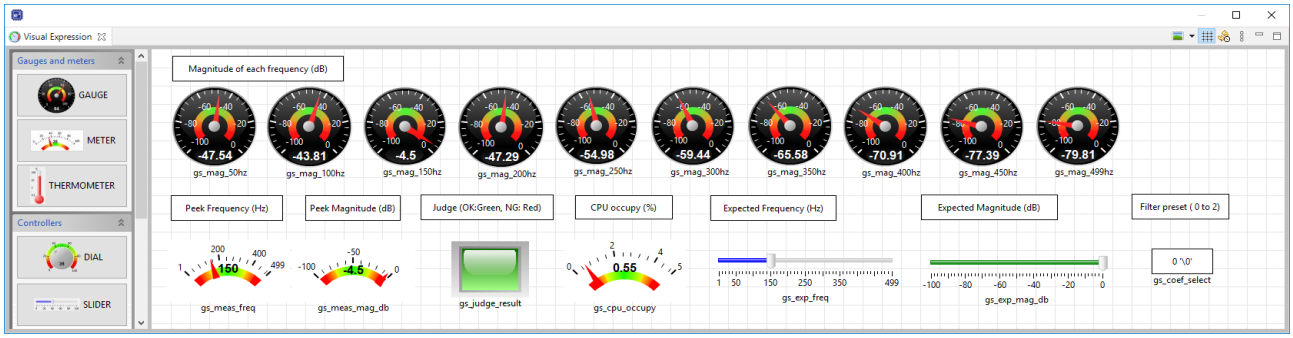


Figure 5.5 Visual Expression View Display Example

The Visual Expression view can only be manipulated when connected to the debugger. Therefore, you first must connect the RX231 to the debugger.

Select **Window** → **Show View** → **Other...**. When the **Show View** window appears, select **Visual Expression** under **Debug**, and click the **Open** button.

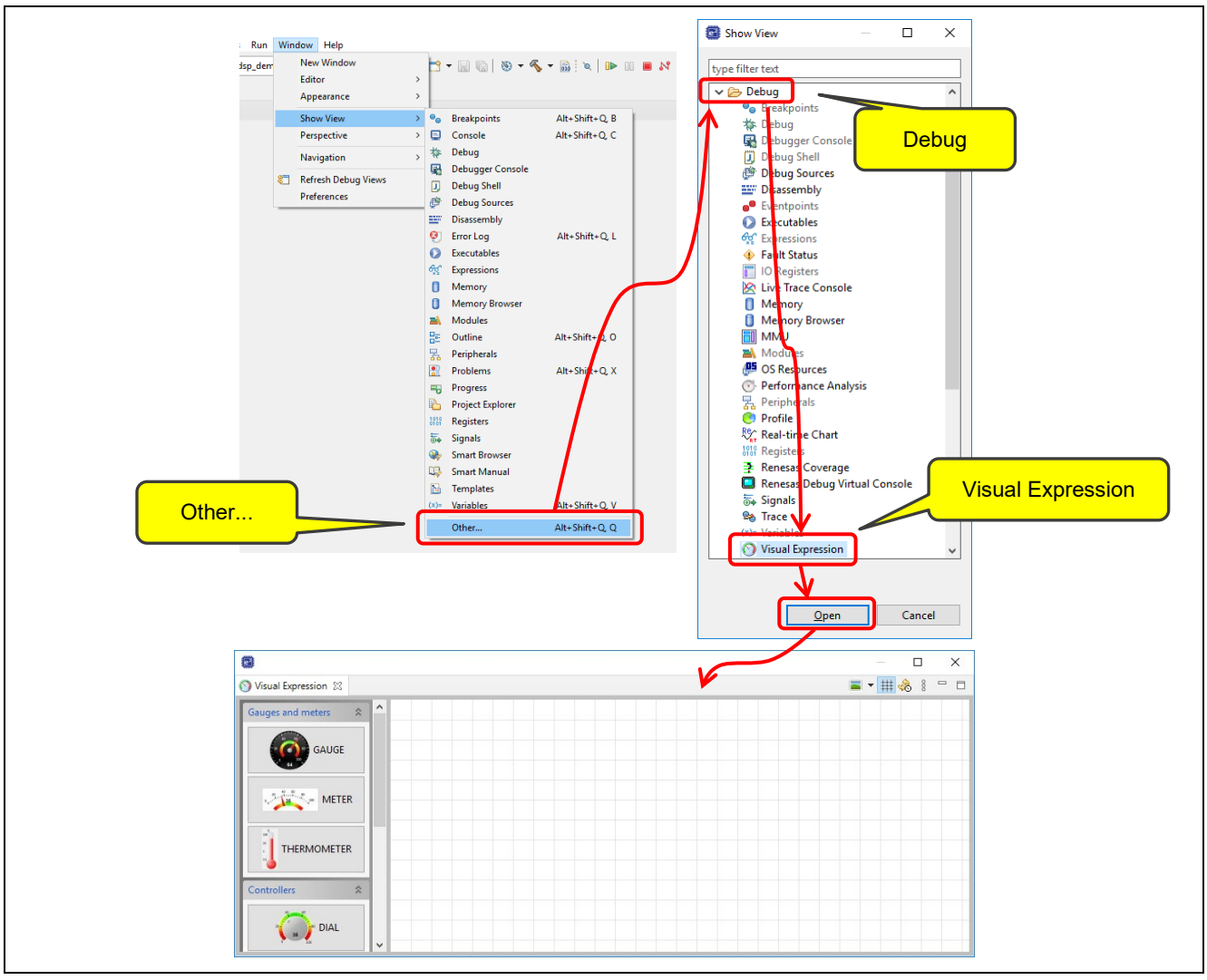


Figure 5.6 Visual Expression View Display Procedure

(a) Gauges and meters

The example below shows the procedure for entering settings for a gauge.

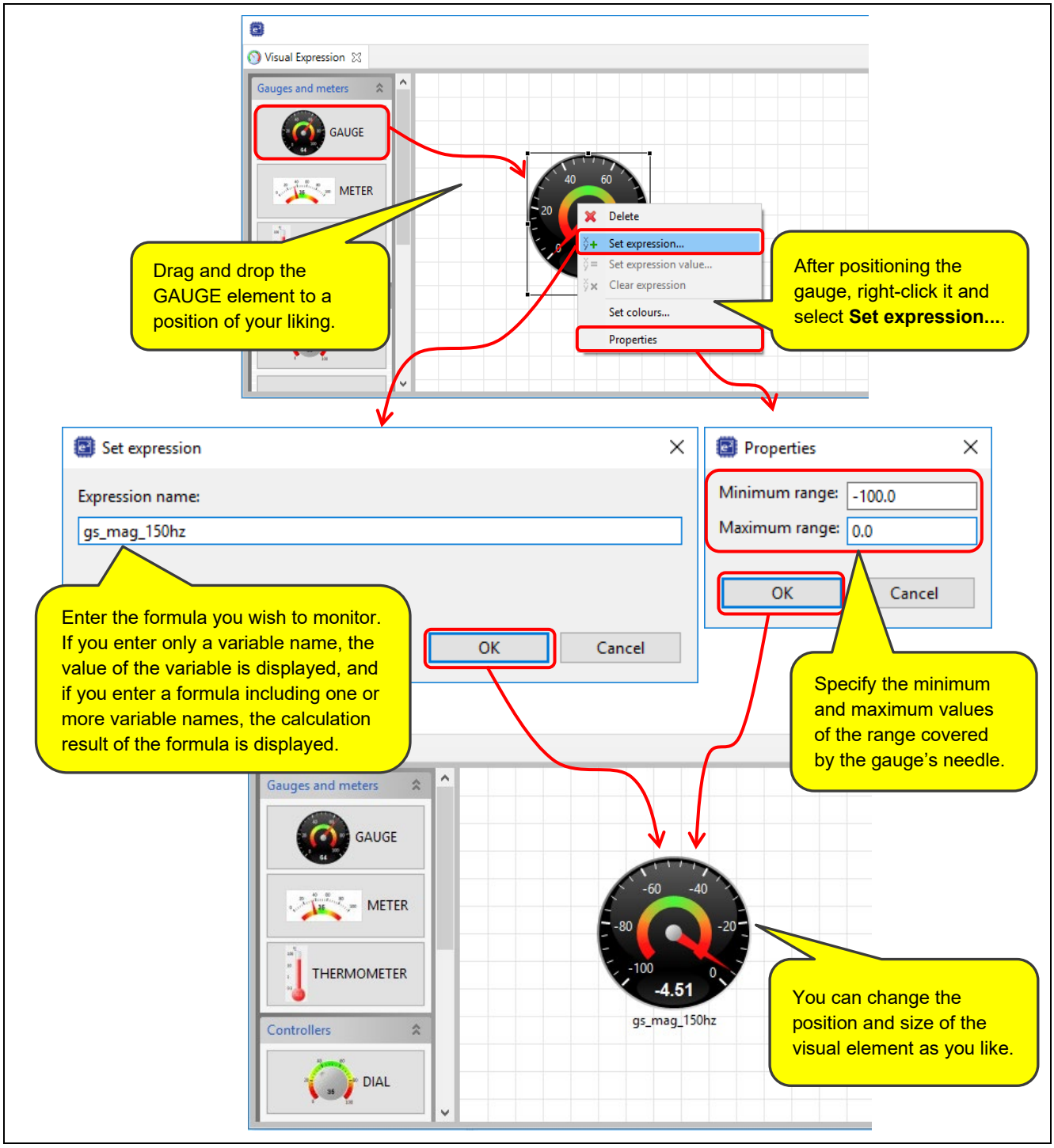


Figure 5.7 Gauge Setting Procedure

(b) Controllers

The example below shows the procedure for entering settings for a slider.

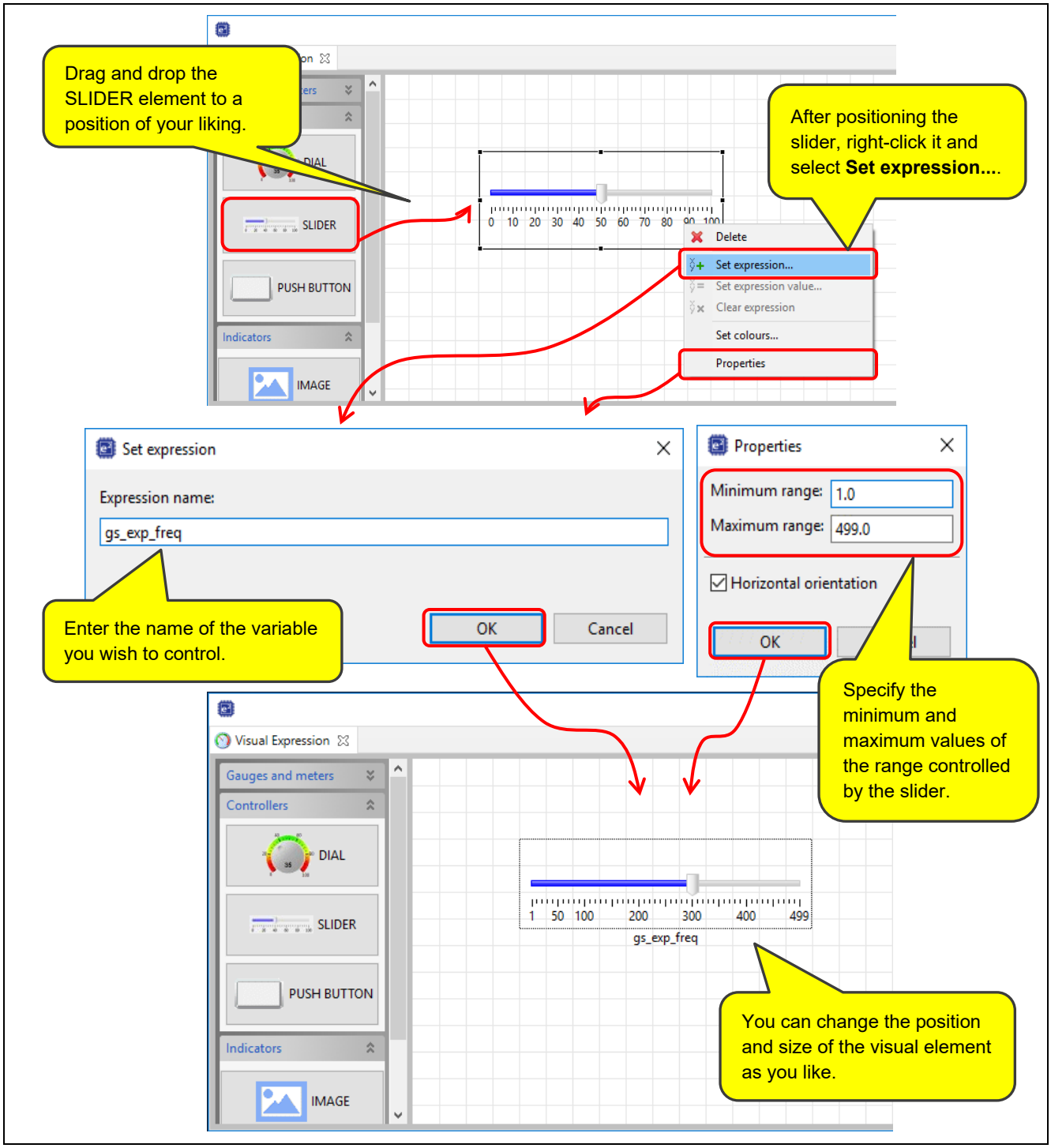


Figure 5.8 Slider Setting Procedure

(c) Indicators

The example below shows the procedure for entering settings for an image.

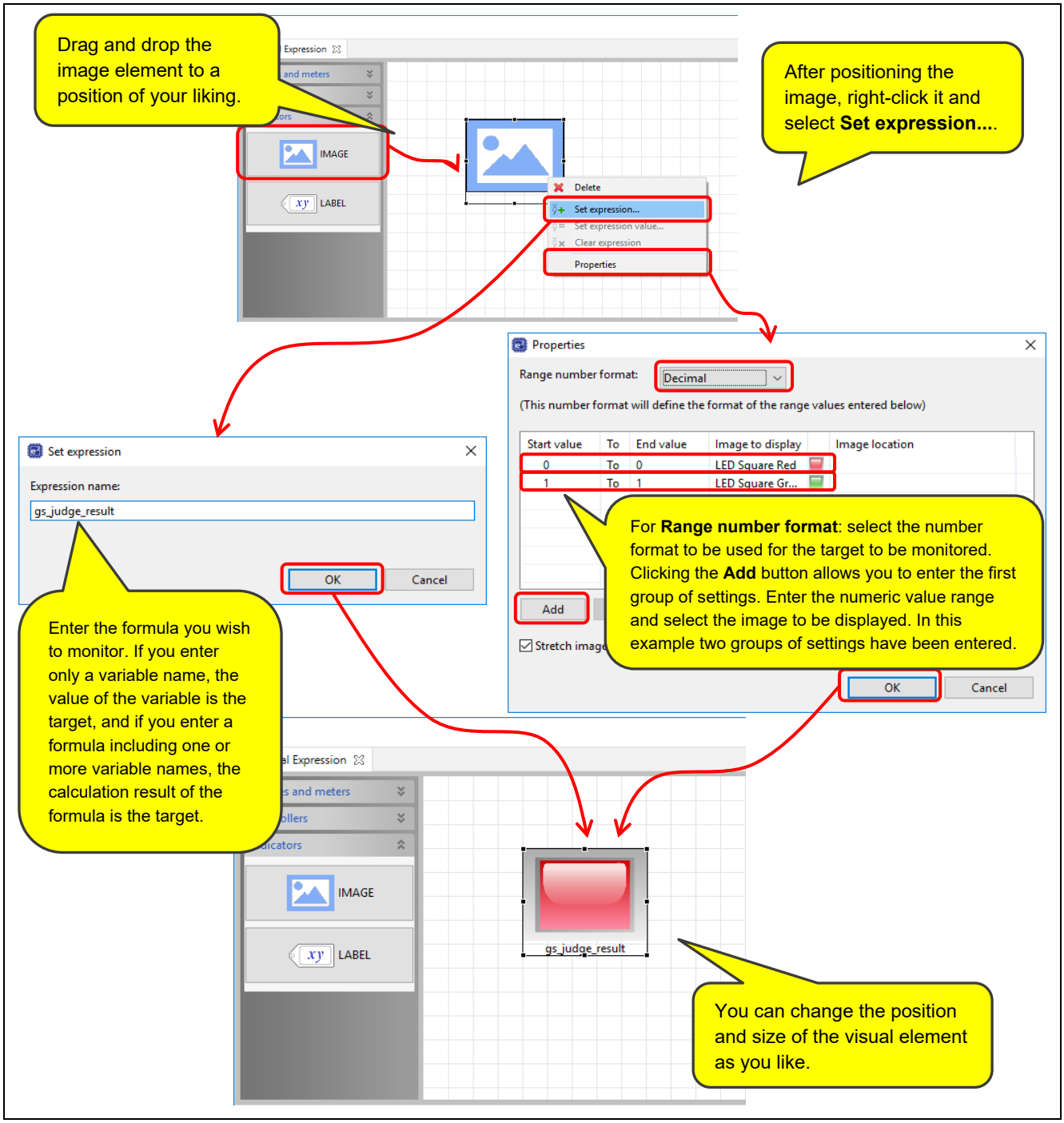


Figure 5.9 Image Setting Procedure

5.1.3 Adjusting the Parameters for Sine Wave Generation Processing

For the sine wave generation processing, the frequency and output level can be adjusted by using the controllers shown in Figure 5.10. In the figure, the left pair of the knob and slider (**gs_osc_freq_0** and **gs_osc_att_0**) can be used to adjust the frequency and output level of the sine wave for Oscillator 0. The right pair of the knob and slider (**gs_osc_freq_1** and **gs_osc_att_1**) can be used to adjust the frequency and output level of the sine wave for Oscillator 1. The rightmost slider (**gs_osc_att_master**) can be used to change the output level of the mixed signal.

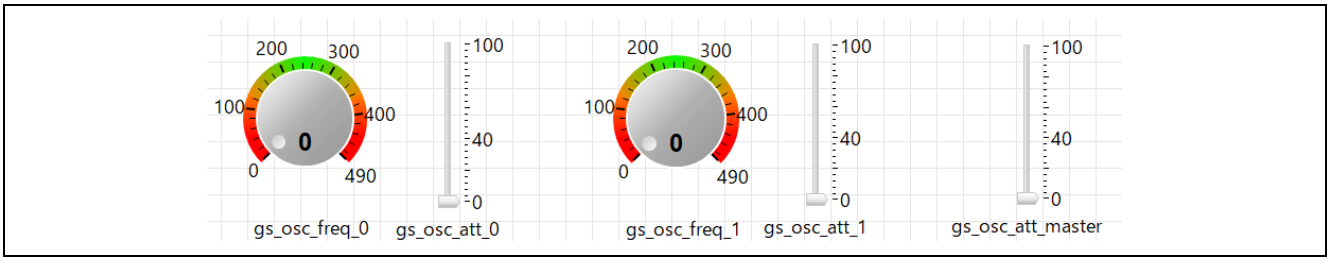


Figure 5.10 Sine Wave Generation Processing Controllers of Visual Expression

5.2 Memory Usage

Table 5.1 and Table 5.2 list the memory usage of the sample program.

The figures for ROM and RAM were calculated based on a .map file generated under the conditions listed in Table 1.2. The figures for user stack and interrupt stack were obtained by measuring the actual stack memory usage while the sample program was running.

Table 5.1 Sample Program Memory Usage (dsp_demo_rx231_rsk) (Reference)

Item	Measured Value [Bytes]	Description
ROM	45870	Sample program ROM usage
RAM	34195	Sample program RAM usage
User stack	467	Sample program stack memory usage
Interrupt stack	251	

Table 5.2 Sample Program Memory Usage (dsp_demo_rx231_tb) (Reference)

Item	Measured Value [Bytes]	Description
ROM	31512	Sample program ROM usage
RAM	27911	Sample program RAM usage
User stack	160	Sample program stack memory usage
Interrupt stack	88	

5.3 Software Module Settings

Table 5.3 to Table 5.13 list the FIT module settings, e^2 studio Smart Configurator settings, and DSP library settings used in the sample program. For Smart Configurator settings, the items and setting details match those displayed on the setting menu. For details of the software modules, refer to the application notes listed in 8, Reference Document.

Table 5.3 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	Sample project for RSK	Operation: Checked. Oscillation source: Resonator Frequency: 8 (MHz) Wait time: 8,192 cycles, 2,048 (µs)
		Sample project for Target Board	Operation: Unchecked.
	PLL circuit settings		Frequency division: ×1/2 Frequency multiplication: ×13.5
	Sub-clock setting		Operation: Unchecked.
	HOCO clock settings	Sample project for RSK	Operation: Unchecked.
		Sample project for Target Board	Operation: Checked. Frequency: 54 (MHz)
	LOCO clock settings		Operation: Unchecked.
	System clock settings	Sample project for RSK	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): ×1/2 27 (MHz) FlashIF clock (FCLK): ×1/2 27 (MHz)
		Sample project for Target Board	Clock source: HOCO 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): ×1/2 27 (MHz) FlashIF clock (FCLK): ×1/2 27 (MHz)
	IWDT-dedicated low-speed clock settings		Operation: Unchecked.
	USB-PLL circuit settings		Frequency division: ×1/2 Frequency multiplication: ×12
	External bus clock pin (BCLK pin) settings		Operation: Unchecked.

Table 5.4 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	This setting specifies whether BSP parameter checking is used. Changed to Disabled (Omit parameter checking when compiling code.)

Table 5.5 SCI Module Settings

Category	Item	Setting/Description
r_sci_rx_config.h		Other than the changes listed below, default settings are used.
	SCI_CFG_SYNC_INCLUDED	Changed to 1 (clock synchronous mode enabled)
	SCI_CFG_ASYNC_INCLUDE	Changed to 0 (asynchronous mode disabled)
	SCI_CFG_CH1_INCLUDED	Changed to 0 (channel 1 not used)
	SCI_CFG_CH8_INCLUDED	Changed to 1 (channel 8 used)
	SCI_CFG_PARAM_CHECKING_ENABLE	This setting specifies whether BSP parameter checking is used. Changed to Disabled (Omit parameter checking when compiling code.)

Table 5.6 BYTEQ Module Settings

Category	Item	Setting/Description
Smart Configurator >> Components >> r_byteq		Other than the changes listed below, default settings are used.
	Parameter check	This setting specifies whether BSP parameter checking is used. Changed to Disabled (Omit parameter checking when compiling code.)
	Number of static queue control blocks	Changed to 2

Table 5.7 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.

Category	Item	Setting
	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 5.8 Smart Configurator Settings (CMT0)

Category	Item	Setting/Description
Smart Configurator >> Components >> Compare Match Timer (Config_CMT0)		Code is generated using the settings below.
	Count clock setting	PCLK/8
	Compare match setting	Interval value: 10 ms (Actual value: 10) Register value (CMCOR): 33,749 Compare match interrupt (CMI0): Checked. Priority: Level 13
Elements added after code generation		
Config_CMT0_user.c	Addition by include	Include r_lcd_driver_if.h
	Additional processing (r_Config_CMT0_cmi0_interrupt)	Call R_LCD_Driver_PeriodicCallback function

Table 5.9 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 setting	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 5.10 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 5.11 Smart Configurator Settings (TPU1, TPU2, TPU4, and TPU5)

Category	Item	Setting/Description
Smart Configurator >> Components >> Normal Mode Timer (Config_TPU1, Config_TPU2, Config_TPU4, Config_TPU5)		Code is generated using the settings below.
	Synchronous mode setting	Unchecked.
	TCNT1 counter setting	Counter clear source: Disabled counter clear Counter clock selection: TPU2 overflow/underflow
	TCNT2 counter setting	Counter clear source: Disabled counter clear Counter clock selection: PCLK
	TCNT4 counter setting	Counter clear source: Disabled counter clear Counter clock selection: TPU5 overflow/underflow
	TCNT5 counter setting	Counter clear source: Disabled counter clear Counter clock selection: PCLK
	General register setting	TGRAx: Output compare register 100 count TGRBx: Output compare register 100 count
	Input/Output setting	TIOCAx pin: Output disabled TIOCBx pin: Output disabled
	A/D converter start trigger setting	Unchecked.
	Interrupt setting	All unchecked.

Table 5.12 Smart Configurator Settings (ICU)

Category	Item	Setting/Description
Smart Configurator >> Components >> Interrupt Controller (Config_ICU)		Code is generated using the settings below.
	(Sample project for RSK) IRQ1	IRQ1: Checked. Detection type: Falling edge Digital filter: PCLK/64 Priority: Level 10
	(Sample project for Target Board) IRQ4	IRQ4: Checked. Detection type: Falling edge Digital filter: PCLK/64 Priority: Level 10

Table 5.13 DSP Library Settings

Category	Item	Setting/Description
Files used		The files in the DSP library dsplib-rxv2 folder listed below are used.
	.lib file	RX_DSP_FPU_LE.lib
	.h file	r_dsp_complex.h r_dsp_transform.h r_dsp_filters.h r_dsp_typedefs.h r_dsp_types.h r_dsp_ver_info.h

6. Obtaining the Development Environment

6.1 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 version 2022-04 or later of e² studio is used. If a version earlier than 2022-04 is used, some e² studio functions may not be supported. Make sure to download the latest version of e² studio on the website.

6.2 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

7. Others

7.1 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

8. Reference Documents

- RX Family Board Support Package Module Using Firmware Integration Technology (R01AN1685)
- RX Family DMA Controller DMACA Control Module Firmware Integration Technology (R01AN2063)
- RX Family SCI Multi-Mode Module Using Firmware Integration Technology (R01AN1815)
- e² studio Code Generator Integrated Development Environment User's Manual: RX API Reference (R20UT2864)
- Renesas e² studio Smart Configurator User Guide (R20AN0451)
- RX Family RX DSP Library Version 5.0 (CC-RX) (R01AN4359EJ0100)
- RX230 Group, RX231 Group User's Manual: Hardware (R01UH0496)
- RX231 Group Renesas Starter Kit User's Manual (R20UT3027)
- Renesas Starter Kit for RX231 CPU Board Schematics (R20UT3026)
- Target Board for RX231 User's Manual (R20UT4168)
- Target Board for RX231 Schematic (R20UT4165)

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Nov. 20, 18	—	First edition issued
1.50	Apr. 26, 21	1	Operation confirmation board Target Board for RX231 added Explanation related to sample program added
		4	1. System Overview DSP library version updated to 5.0 Figures updated
		6, 7	1.1 File Structure Associated with This Application Note Explanation related to sample program for Target Board for RX231 added Figures updated
		8, 9	1.2 Structure of Sample Program Explanation related to sample program for Target Board for RX231 and explanation related to ICU (IRQ) added Figure and table updated
		10, 11	1.3 Operating Environment Operation confirmation environment updated Confirmation environment for Target Board for RX231 added Tables updated
		12	2.1 Launching the Workspace Figure 2.1 added
		13	2.2 Connecting Equipment Figure updated
		14	2.3.1 Running the Sample Program Added
		16	2.3.2 LCD Display Contents and Changing Filter Characteristics Explanation related to sample program for Target Board for RX231 added
			2.3.3 Using e^2 studio Functions to Monitor System Operation Explanation related to Visual Expression view added Figures updated
		18	2.4 User Changeable Settings EXPECTED_MAGNITUDE explanation updated Note added to table
		26	3.4.1 Signal Processing Flow Added
		28	3.4.4 FFT Processing DSP library version updated to 5.0 and accompanying explanation added
		29	3.4.5 Judgement of FFT Processing Results Judgement processing procedure updated
		30	3.4.6 Processing to Change Display on LCD Module Contents displayed on LCD updated
		31	3.4.7 Processing for Changing IIR Filter Characteristics Processing procedure to change filter characteristics updated
		34	4. Usage Notes Added
		35	5.1 Monitoring Signal Processing in e^2 studio Added

Rev.	Date	Description	
		Page	Summary
1.50	Apr. 26, 21	42	5.2 Memory Usage Information on sample program for RSK updated Memory usage information on sample program for Target Board added
		43	5.3 Software Module Settings Operation confirmation environment updated and accompanying information updated Explanation related to sample program for Target Board for RX231 added
		48	6. Obtaining the Development Environment Explanation updated
			8. Reference Documents Reference documents for Target Board for RX231 added
		Program	• Operating environment (e^2 studio, compiler, FIT modules, DSP library, etc.) updated • Method of detecting switch input to change IIR filter coefficient settings changed to use of IRQ interrupt (main.c and Config_ICU_user.c) • Changed so that output magnitude value after FFT processing is converted to decibels and displayed on LCD and in Visual Expression view (main.c) • Changed several local variables to global variables to allow monitoring in Visual Expression view (main.c) • Changed DSP processing specifications as follows: — Changed output format of FFT processing result frequency magnitude characteristics from Q1.31 to Q2.30 to accommodate updated DSP library — Changed method of deriving element [0] of FFT processing result frequency magnitude characteristics to obtaining absolute value of $R_{N/2}$ and converting it to Q2.30 format (r_dsp_iirbiquad.c and r_dsp_real_fft.c) • Sample project for Target Board added (dsp_demo_rx231_tb)
1.60	Mar.28, 22	5, 6	1. System Overview Figures were updated. A description of sine wave generation processing was added.
		10, 11	1.3 Operating Environment The environment in which to confirm the operation of the sample program was updated. Tables were updated.
		17	2.3.3 Using e^2 studio Functions to Monitor System Operation Figures were updated.
		18	2.4 User Changeable Settings The table was updated.
		22, 23	3.3 Processing Flow Figures were updated. A description of sine wave generation processing was added.
		26	3.4.1 Signal Processing Flow Figures and tables were updated.
		28, 29	"3.4.3 Sine Wave Generation Processing" was added.

Rev.	Date	Description	
		Page	Summary
1.60	Mar.28, 22	33, 34	3.5 File Structure Tables were updated. "Table 3.13 r_signal_gen.c File Functions" was added.
		35	"4.2 Aliasing" was added.
			"4.3 Chronological Change in Oscillator Output Signals" was added.
		43	"5.1.3 Adjusting the Parameters for Sine Wave Generation Processing" was added.
		Program	• The operating environment (such as e^2 studio, compiler, FIT module, and DSP library) was updated. • Some global variables were defined so that the settings can be changed in Visual Expression. (main.c) • "Sine wave generation processing" was added in "Input data switching". (main.h) • Processing to be performed if sine wave generation processing is selected as the input data source was added. (main.c) • Files related to sine wave generation processing were added: r_signal_gen.c, r_signal_gen.h, r_signal_gen_att_coef.c, and r_signal_gen_freq_coef.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/.