RX ファミリ

FFT によるデジタル信号解析・判定デモンストレーション

要旨

本書は、RX マイコンのセンサ分野への応用を想定したデモのアプリケーションノートです。アナログ信号入力、デジタル信号解析、解析結果の表示をひとつの RX マイコンで行うことができます。

本アプリケーションノートのサンプルプログラムは、図 1 のセンサシステムのモデルに基づき構成されています。モデルの各要素に対応するサンプルプログラムの処理を以下に示します。

- A/D（データ収集部）

  RX マイコンの A/D 変換機能によるアナログ信号入力
- Digital Signal Processing（解析部）

  IIR フィルタ、FFT（Fast Fourier Transform）を応用した周波数解析
- Check & Judge（判断部）

  周波数解析結果に対する OK/NG 判定
- Action（制御部）

  判定結果などの LCD 表示

以降、本アプリケーションノートは、デモの環境と手順、サンプルプログラムについて説明します。

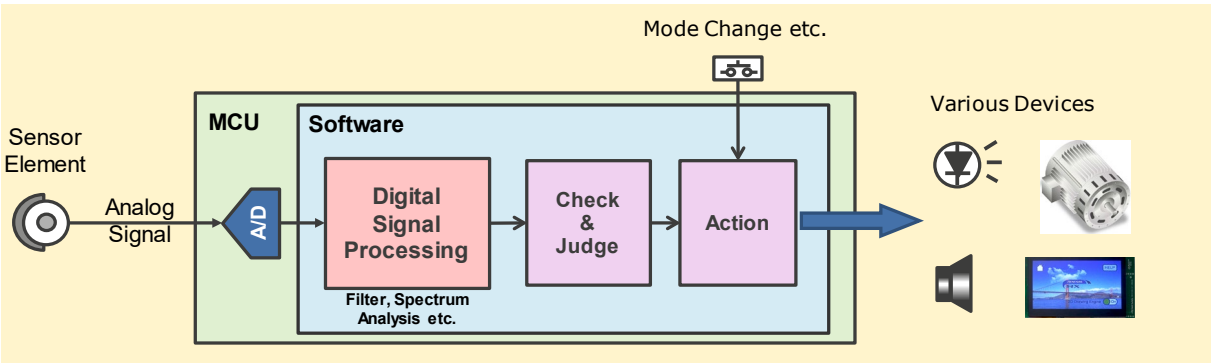


図 1　センサシステムのモデル

動作確認デバイス

RX231 グループ

動作確認ボード

Renesas Starter Kit for RX231

Target Board for RX231

本アプリケーションノートには Renesas Starter Kit for RX231（以後、RSK）または Target Board for RX231（以後、Target Board）で動作するサンプルプログラムを添付しています。なお Target Board のサンプルプログラムには LCD 表示の機能を実装していません。

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様に合わせて変更し、十分評価してください。

目次

1. システムの概要 4
1.1 アプリケーションノートの構成 6
1.2 サンプルプログラムの構成 8
1.3 動作環境 10

2. サンプルプログラムの実行方法 12
2.1 ワークスペースの起動 12
2.2 機器の接続 13
2.3 サンプルプログラムの実行と動作確認 14
2.3.1 サンプルプログラムの実行方法 14
2.3.2 LCD の表示内容とフィルタ特性の変更方法 16
2.3.3 e² studio の機能を使ったシステム動作のモニタ 16
2.4 ユーザ変更可能な設定 18

3. サンプルプログラムの説明 19
3.1 サンプルプログラムの概要 19
3.2 処理シーケンス 20
3.3 処理フロー 22
3.4 詳細(信号発生機能の説明を追加) 26
3.4.1 信号処理フロー 26
3.4.2 正規化処理 27
3.4.3 正弦波生成処理 28
3.4.4 IIR フィルタ処理 29
3.4.5 FFT 処理 30
3.4.6 FFT 処理結果の判定 31
3.4.7 LCD モジュールの表示内容更新処理 31
3.4.8 IIR フィルタ特性の変更処理 32
3.4.9 サイクル数のカウント処理 32
3.5 ファイル構成 33

4. 注意事項 35
4.1 FFT 処理結果の周波数値 35
4.2 エイリアシング 35
4.3 オシレータ出力信号の経時変化 35

5. 参考 36
5.1 e² studio による信号処理のモニタ 36
5.1.1 Waveform レンダリング 36
5.1.2 Visual Expression 39
5.1.3 正弦波生成処理のパラメータ調整 43
5.2 使用メモリ(再計測) 44
5.3 ソフトウェアモジュールの設定 45

6. 開発環境の入手 49
6.1 e² studio の入手方法 49

6.2　コンパイラパッケージの入手方法 49

7.　補足 49
7.1　無償評価版の「RX ファミリ用 C/C++コンパイラパッケージ」を利用する場合の注意事項 49

8.　参考資料 49

改訂記録 50

1.　システムの概要

本アプリケーションノートのシステムの概要を図 2 に示します。

このシステムでは入力信号のサンプリングから判定結果の出力制御まで RX231 マイコンひとつで行っています。

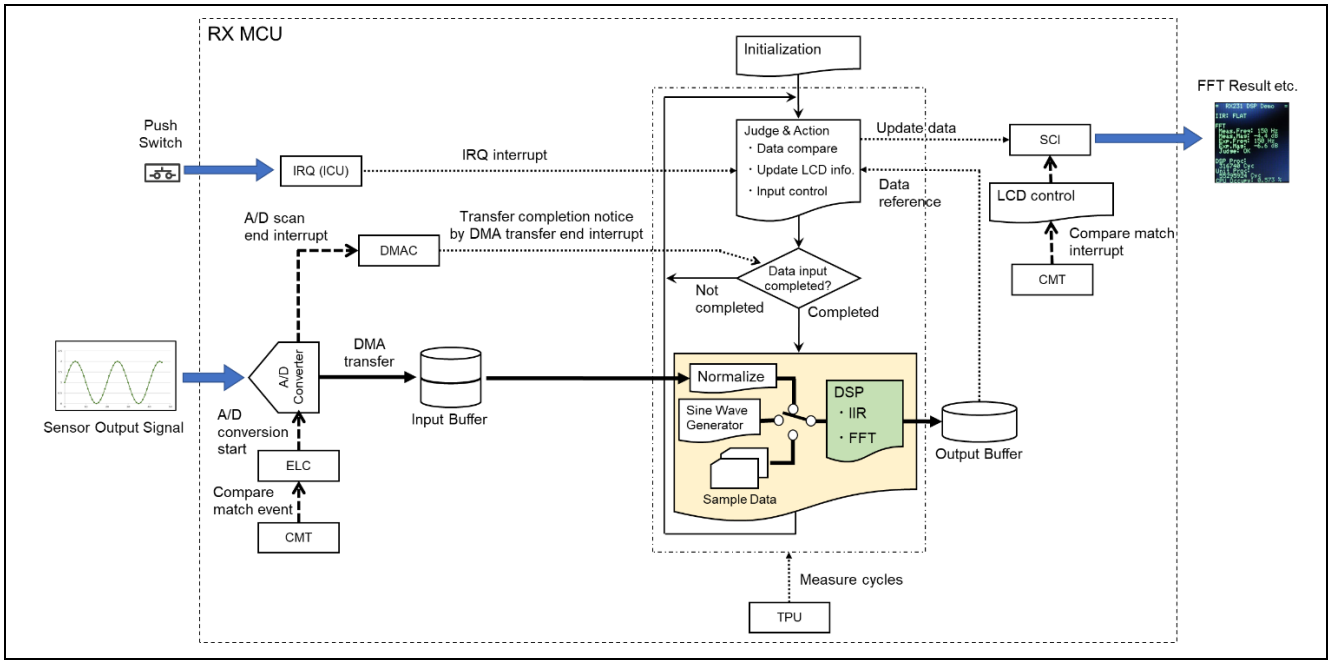


図 2　システムの概要

このシステムは次の処理を行います。

- 入力信号のサンプリング
  12 ビット A/D コンバータ（S12AD）、コンペアマッチタイマ（CMT）、イベントリンクコントローラ（ELC）を使って約 1kHz のサンプリング周波数で A/D 変換します。まず、CMT にて約 1μs の周期でコンペアマッチイベントを生成、このイベントを ELC にて S12AD の A/D 変換開始トリガとして与えます。変換後のデータは DMA コントローラ（DMAC）により入力バッファに転送されます。
- 入力データの正規化
  S12AD で A/D 変換した入力信号は 12 ビット（符号なし）のデータとして入力バッファに格納されます。入力バッファに格納されている 12 ビットのデータを 31 ビット（符号付き）で正規化（バイアス処理とスケーリング）します。
- IIR フィルタ処理と FFT 処理
  RX DSP ライブラリ API Version 5.0 を使用し、IIR フィルタ処理を行います。フィルタ特性は、スイッチ入力によりパススルー（FLAT）、ローパスフィルタ（LPF）、ハイパスフィルタ（HPF）の 3 つを切り替えることができます。IIR フィルタを通過したデータに 1024 ポイントの FFT 処理を行い、結果を出力バッファに格納します。
- 処理結果の判定と判定結果に基づく処理
  FFT 処理で得られた周波数スペクトルの情報を期待値と比較します。比較結果から OK/NG 判定を行い、LCD モジュールに情報を表示します。LCD モジュールへの表示は、コンペアマッチタイマ（CMT）とシリアルコミュニケーションインタフェース（SCI）を使って行います。
- 処理サイクル数の計測
  以下の処理サイクルを計測します。計測には 16 ビットタイマパルスユニット（TPU）を使用します。
  - DSP 処理（正規化処理、IIR フィルタ処理と FFT 処理）のサイクル数
  - 入力信号 1024 サンプルの取り込みにかかるサイクル数

- 正弦波生成処理

  正弦波を発生させる処理です。0Hz～490Hz の範囲で周波数、-Infinity, -96～0dB の範囲で振幅を調整することができます。この処理の出力をテスト信号として、IIR フィルタ処理、FFT 処理を評価することができます。

このシステムに正弦波を入力したときに得られる周波数スペクトルの振幅特性の例を図 3 に示します。

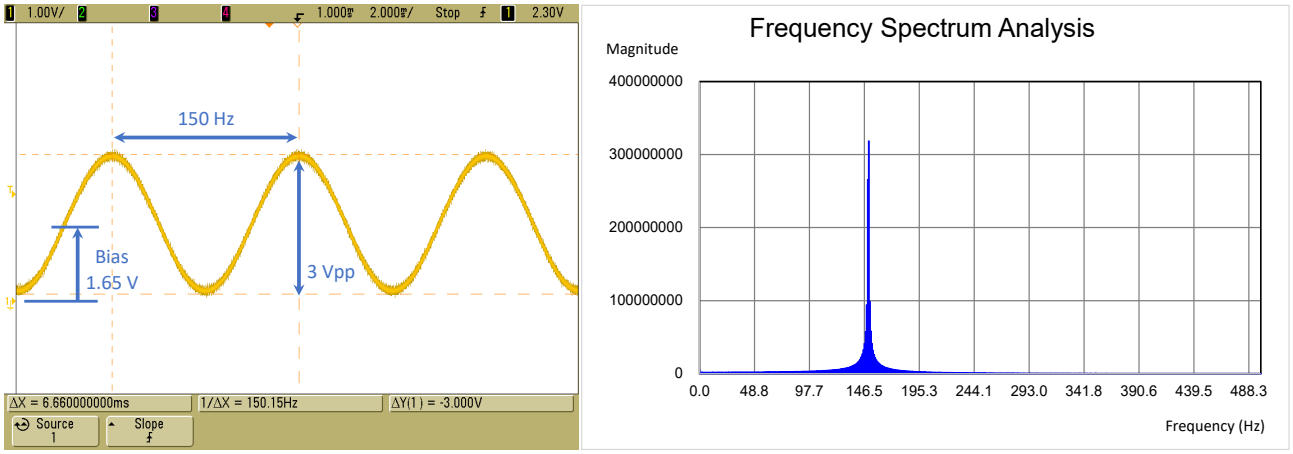


図 3　入力信号（周波数 150Hz、振幅 3Vpp の正弦波）（左）と FFT 処理結果（右）

図 4　LCD モジュールの表示画面

1.1　アプリケーションノートの構成

図 5 に本アプリケーションノートの構成を示します。本アプリケーションノート提供 ZIP ファイルを展開すると、ZIP ファイルと同名のフォルダが作成されます。その下の「workspace_dsp_example」フォルダは e² studio 形式のプロジェクト「dsp_demo_rx231_rsk（RSK 用）」「dsp_demo_rx231_tb（Target Board 用）」を含む e² studio のワークスペースです。プロジェクトのフォルダには、図 6 に示すようにサンプルプログラムのソースコードのほか、e² studio の設定ファイル、本アプリケーションノートを含みます。

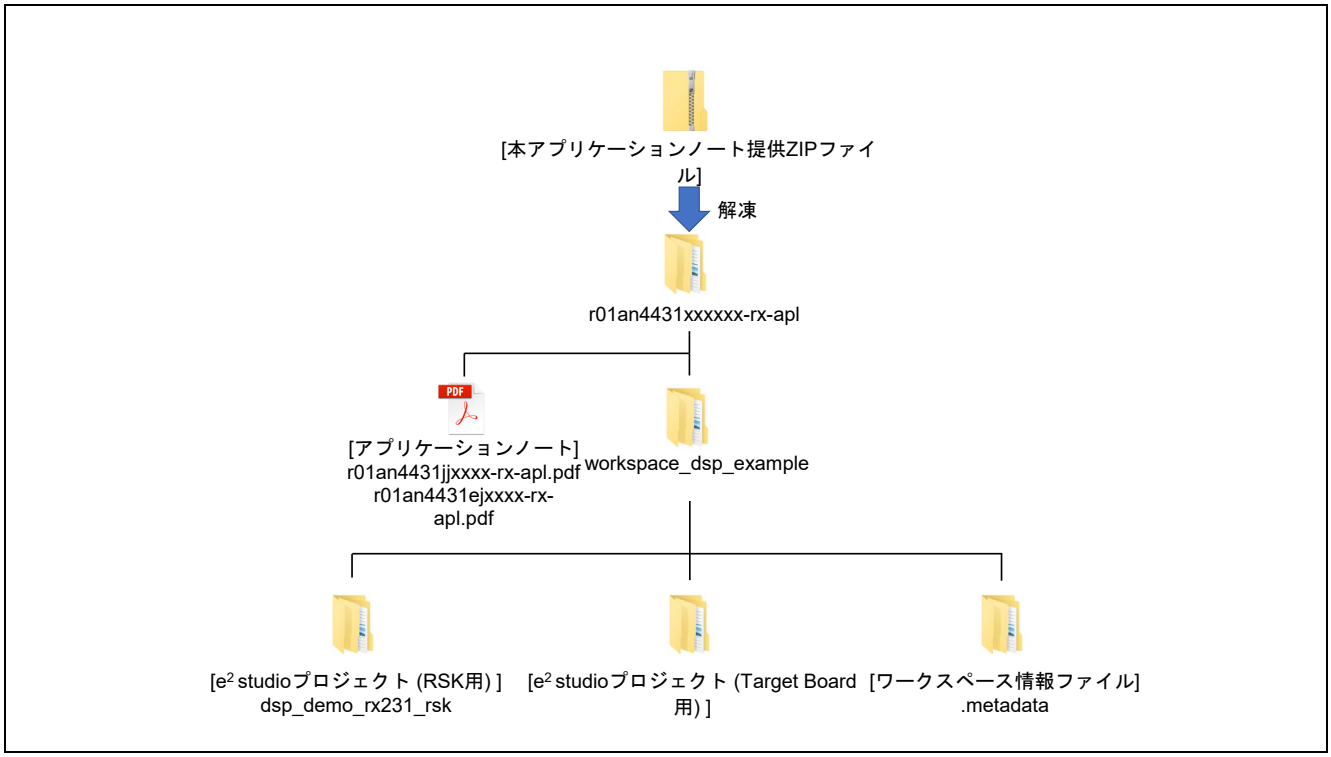


図 5 アプリケーションノートの構成

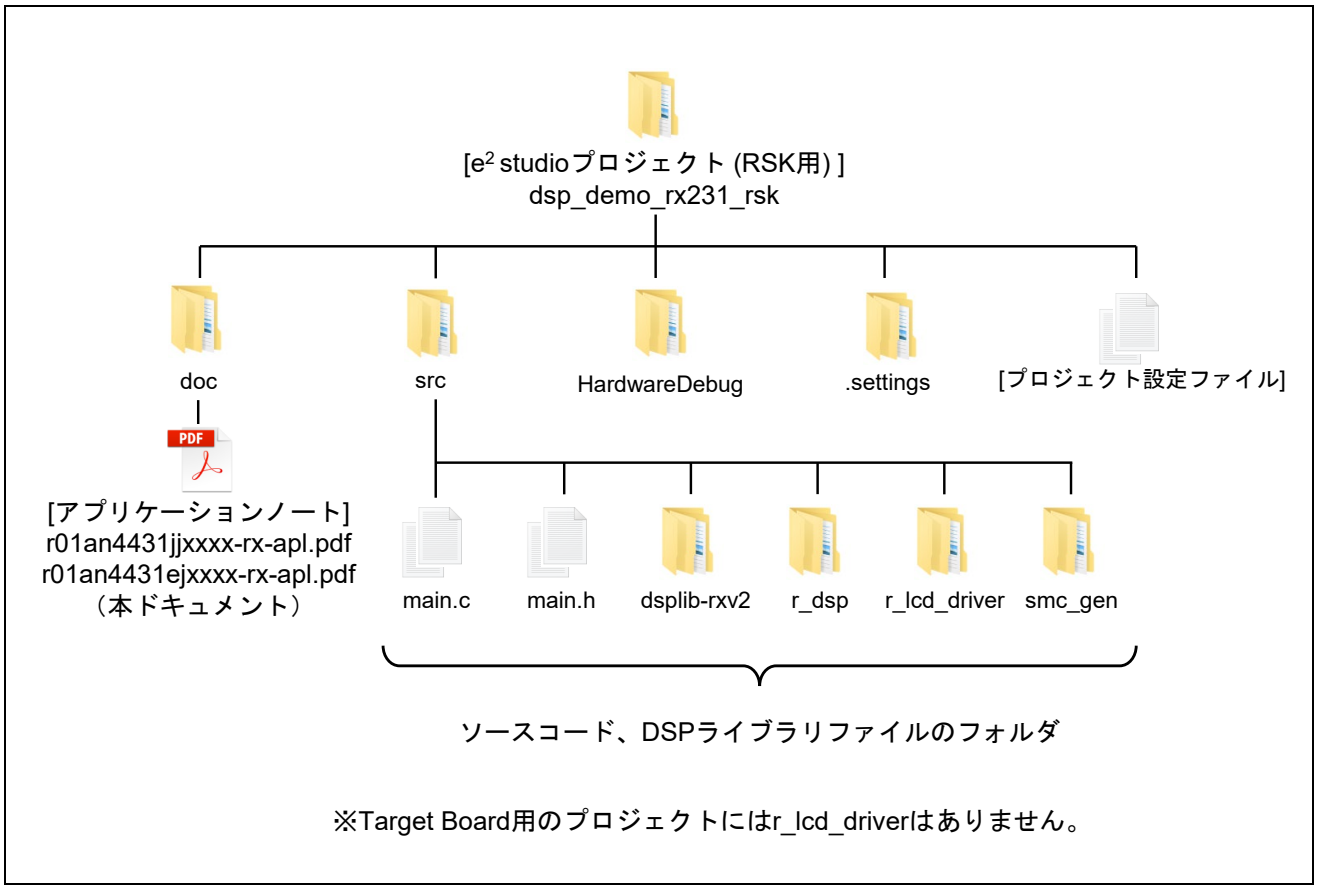


図 6　サンプルプロジェクトのフォルダ構成

1.2　サンプルプログラムの構成

サンプルプログラムの構成を図 7 に、使用しているソフトウェアモジュールを表 1 に示します。Target Board 用のサンプルプログラムには LCD 表示の機能を実装していないため、RSK 用と Target Board 用で使用するソフトウェアモジュールの構成が一部異なります。FIT モジュールと DSP ライブラリはルネサスの Web サイトから入手可能です。その他の周辺機能のドライバソフトウェアは e² studio のコード生成機能により生成しています。各ソフトウェアモジュールの詳細は、各アプリケーションノートや e² studio のヘルプを参照してください。

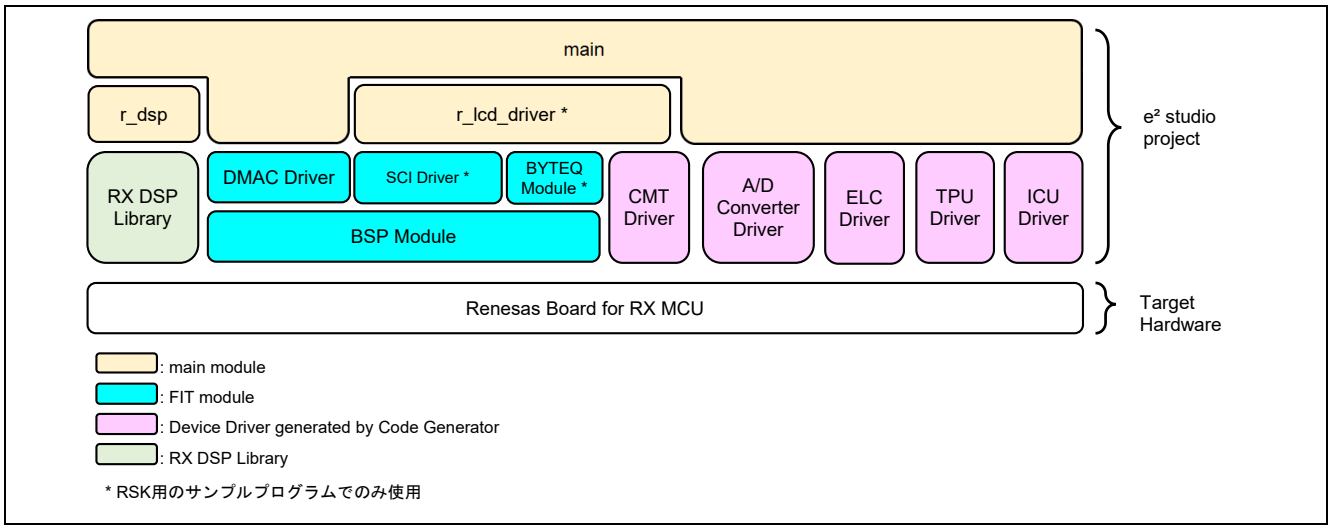


図 7　サンプルプログラムの構成

表 1 使用ソフトウェアモジュール一覧

モジュール	ドキュメントタイトル	ドキュメント番号	種類
main	-	-	本アプリケーションノートで開発した main 関数を含むモジュール
r_dsp	-	-	本アプリケーションノートで開発した DSP ライブラリの操作モジュール
r_lcd_driver	-	-	本アプリケーションノートで開発したデバッグ LCD の制御モジュール（RSK 用のサンプルプログラムでのみ使用）
BSP	RX ファミリ ボードサポートパッケージモジュール Firmware Integration Technology	R01AN1685JJ	FIT モジュール
DMAC	RX ファミリ DMA コントローラ DMACA 制御モジュール Firmware Integration Technology	R01AN2063JJ	FIT モジュール
SCI	RX ファミリ SCI モジュール Firmware Integration Technology	R01AN1815JJ	FIT モジュール（RSK 用のサンプルプログラムでのみ使用）
BYTEQ	RX ファミリ バイト型キューバッファ (BYTEQ) モジュール Firmware Integration Technology	R01AN1683JJ	FIT モジュール（RSK 用のサンプルプログラムでのみ使用）
S12AD	-	-	コード生成機能で生成したドライバ関数
CMT	-	-	
ELC	-	-	
TPU	-	-	
ICU	-	-	
RX DSP Library	RX Family DSP ライブラリ Version 5.0 (CCRX)	R01AN4359JJ	DSP ライブラリ

1.3　動作環境

本アプリケーションノートのサンプルプログラムは、下記の条件で動作を確認しています。

表 2 動作確認条件（RSK 用：dsp_demo_rx231_rsk）

項目	説明
MCU	R5F52318ADFP（RX231 グループ）
動作周波数	・ メインクロック: 8 MHz ・ PLL 回路出力: 54 MHz（メインクロックを 2 分周し 13.5 逓倍する） ・ システムクロック（ICLK）: 54 MHz（PLL 回路出力を 1 分周する） ・ 周辺モジュールクロック（PCLKB）: 27 MHz（PLL 回路出力を 2 分周する）
動作電圧	3.3V
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
統合開発環境	ルネサス エレクトロニクス e^2 studio 2022-04
C コンパイラ	ルネサス エレクトロニクス RX Compiler CC-RX V3.04.00
	コンパイルオプション ・ -lang = c99 ・ -fpu ・ -save_acc
エンディアン	・ データ・エンディアン：リトル・エンディアン ・ デバッグツール設定：リトル・エンディアン
iodefine.h	Version 1.0l
サンプルプログラム	Version 1.60
Evaluation board	ルネサス エレクトロニクス Renesas Starter Kit for RX231 (R0K505231S000BE) の CPU ボード ・ 搭載 MCU：上記 ・ ボード設定：デフォルト ・ 電源：エミュレータから電源を供給する ・ デバッグ LCD を接続
Emulator	ルネサス エレクトロニクス E2 エミュレータ Lite
ファンクション ジェネレータ	正弦波を出力できるアナログ信号出力端子を備えた信号発生器。 出力信号は GND に対しバイアス 1.65V、振幅 3.0Vpp に設定。図 3 参照。 ・ アナログ信号出力(+)を Evaluation board の JA1.10 端子に接続します。 JA1.10 端子に印可された信号は S12AD の AN001 の入力となります。 ・ アナログ信号出力(GND)を Evaluation board の JA1.2 端子に接続します。 JA1.2 端子は Evaluation board の GROUND に接続されています。

表 3 動作確認条件（Target Board 用：dsp_demo_rx231_tb）

項目	説明
MCU	R5F52318ADFP（RX231 グループ）
動作周波数（注）	・ HOCO: 54 MHz ・ システムクロック（ICLK）: 54 MHz（HOCO 出力を 1 分周する） ・ 周辺モジュールクロック（PCLKB）: 27 MHz（HOCO 出力を 2 分周する）
動作電圧	3.3V
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
統合開発環境	ルネサス エレクトロニクス e^2 studio 2022-04
C コンパイラ	ルネサス エレクトロニクス RX Compiler CC-RX V3.04.00
	コンパイルオプション ・ -lang = c99 ・ -fpu ・ -save_acc
エンディアン	・ データ・エンディアン：リトル・エンディアン ・ デバッグツール設定：リトル・エンディアン
iodefine.h	Version 1.0l
サンプルプログラム	Version 1.60
Evaluation board	ルネサス エレクトロニクス Target Board for RX231 (RTK5RX2310C00000BR) の CPU ボード ・ 搭載 MCU：上記 ・ ボード設定：デフォルト ・ 電源：USB 経由で供給
Emulator	ルネサス エレクトロニクス E2 エミュレータ Lite（オンボード）
ファンクション ジェネレータ	正弦波を出力できるアナログ信号出力端子を備えた信号発生器。 出力信号は GND に対しバイアス 1.65V、振幅 3.0Vpp に設定。図 3 参照。 ・ アナログ信号出力(+)を Evaluation board の J2.43 端子に接続します。 J2.43 端子に印可された信号は S12AD の AN001 の入力となります。 ・ アナログ信号出力(GND)を Evaluation board の J2.12 端子に接続します。 J2.12 端子は Evaluation board の GROUND に接続されています。

注. Target Board 用のサンプルでは HOCO をクロックソースに使用しています。詳細は 4.1 FFT 処理結果の周波数値を参照してください。

2.　サンプルプログラムの実行方法

本アプリケーションノートの実行方法を以降に示します。

2.1　ワークスペースの起動

アプリケーションノートの zip ファイルをパスに日本語が含まれない場所に展開してください。次に e² studio を起動し、ワークスペースの選択画面が表示されたら本アプリケーションノートに同梱されているワークスペース（workspace_dsp_example）を選択してください。

e² studio の起動時にワークスペースの選択画面が出ない場合は、e² studio の起動後に

「ファイル」>>「ワークスペースの切り替え(W)」>>「その他(O)」

からワークスペースを選択しなおしてください。

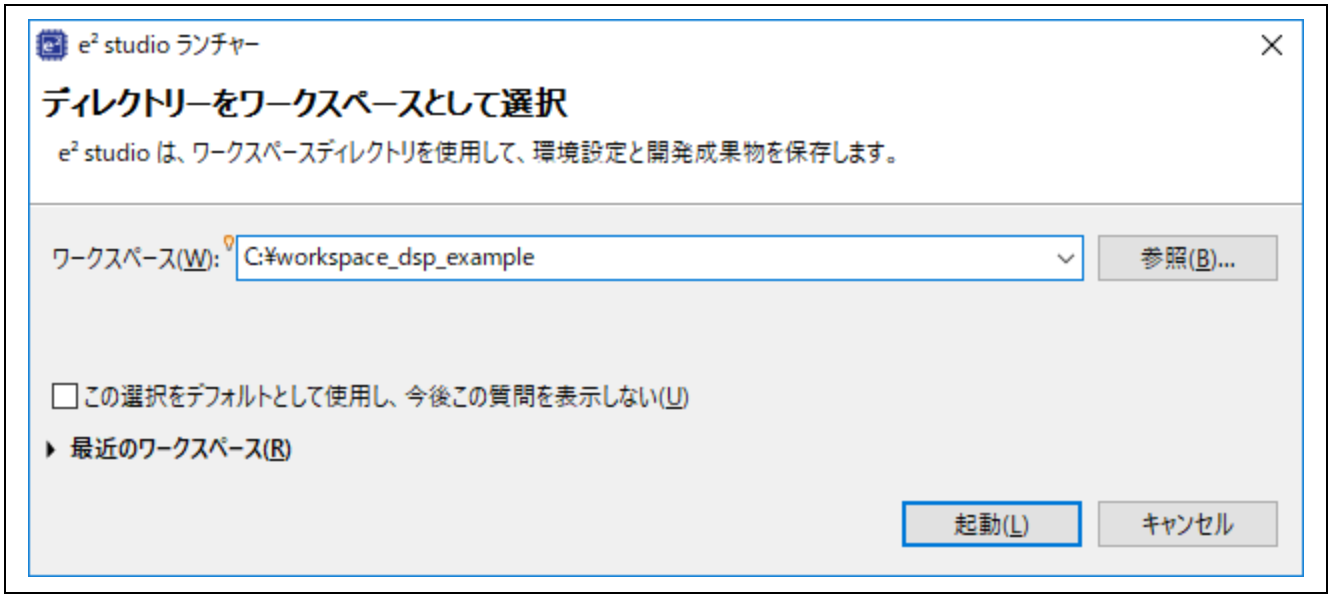


図 8　ワークスペースの選択

2.2　機器の接続

必要な機器を接続します。図 9 や図 10 のように各機器を接続してください。

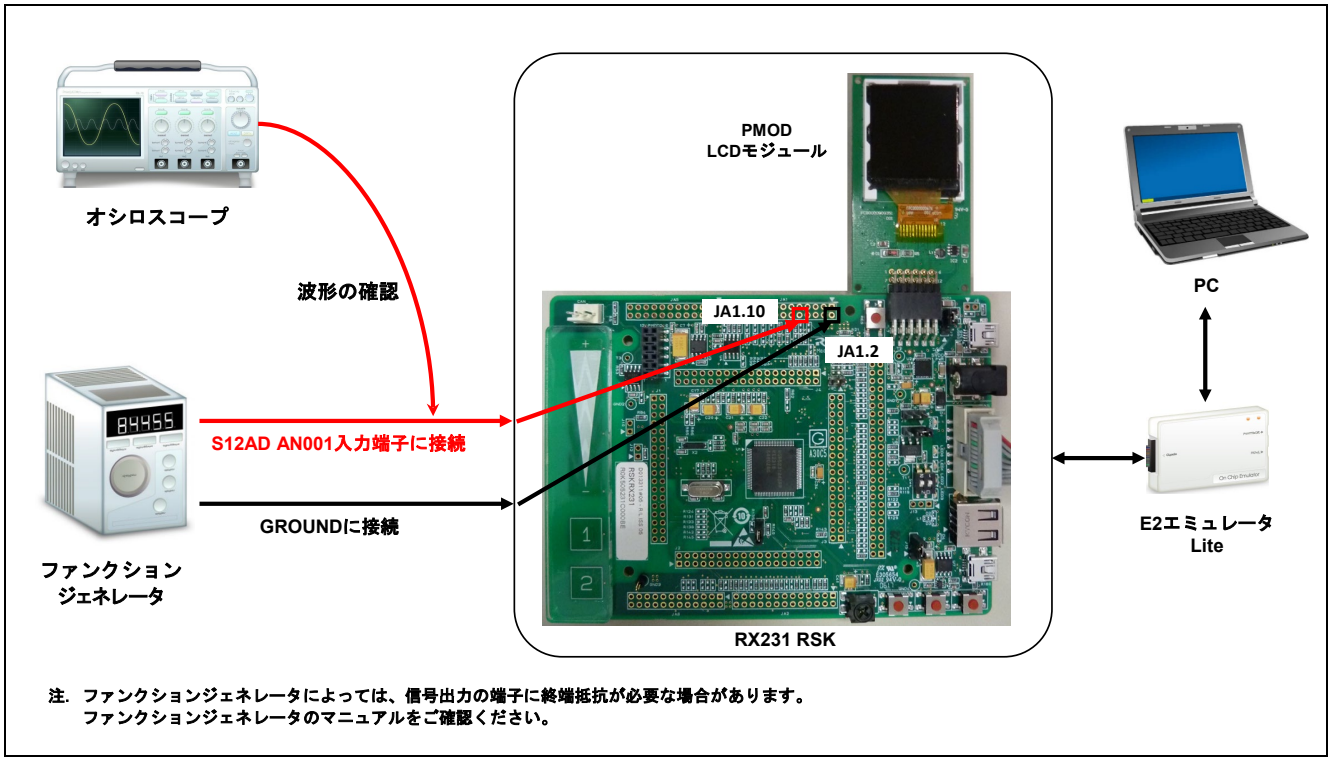


図 9　RSK の場合の接続

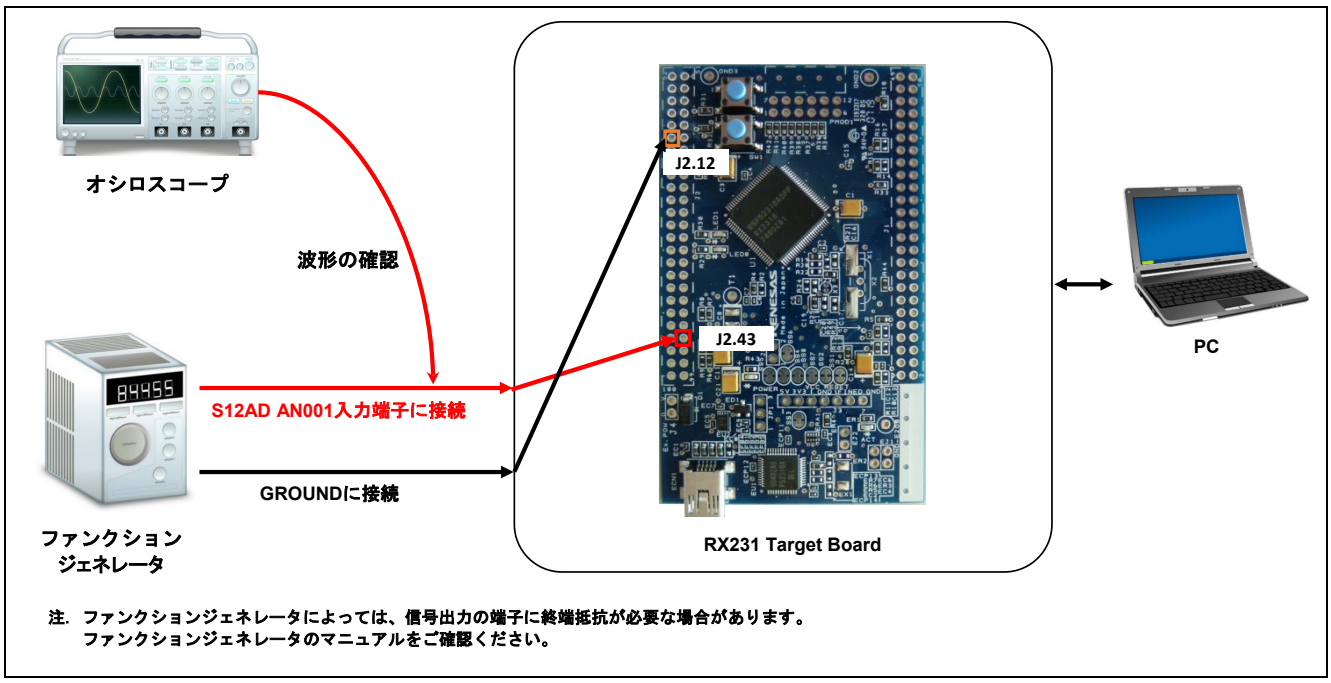


図 10　Target Board の場合の接続

2.3　サンプルプログラムの実行と動作確認

デバッグ接続をしてサンプルプログラムを実行します。

2.3.1　サンプルプログラムの実行方法

ワークスペースを起動すると、プロジェクト・エクスプローラ上に RSK 用と Target Board 用の 2 つのプロジェクトがあります。

ここでは RSK 用を例にサンプルプログラムの実行方法を示します。

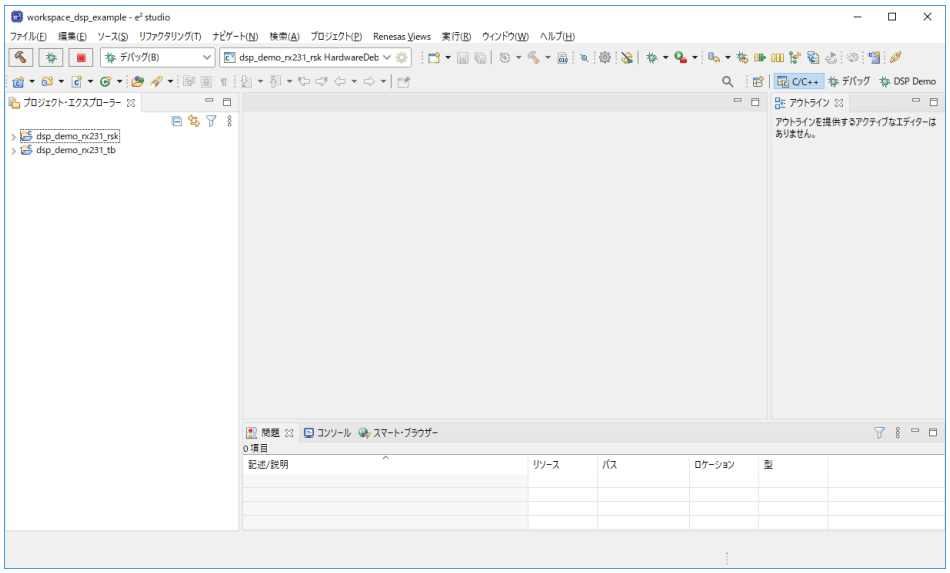


図 11　起動直後のワークスペース

「Launch Configuration」からデバッグ接続するプロジェクトを選択します。選択後、「Build」ボタンをクリックしてビルドが完了するのを待ちます。ビルドが完了したら「Launch in ‘デバッグ(D)’ mode」ボタンをクリックしてデバッグ接続を開始します。

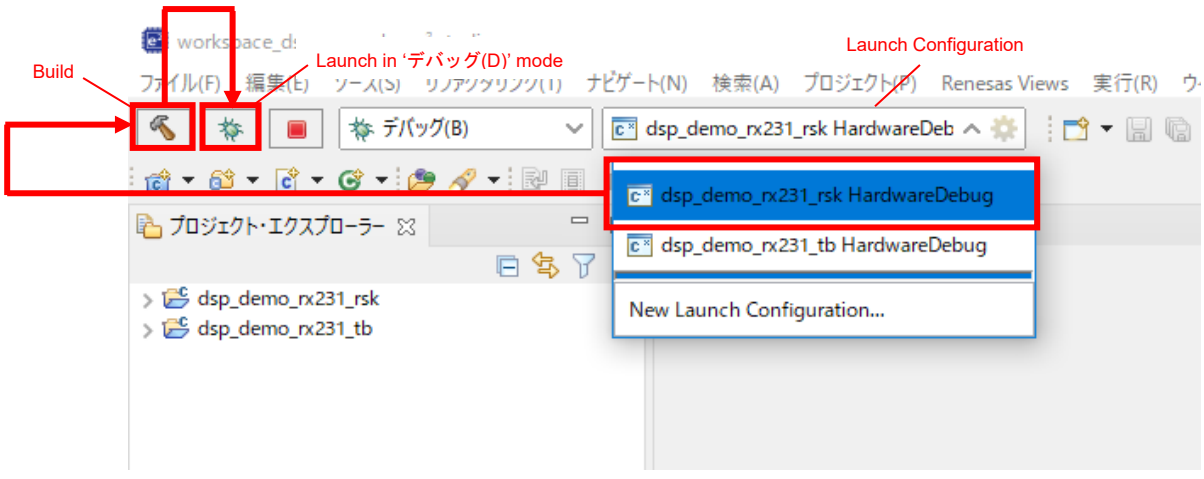


図 12　操作方法

デバッグ接続を開始してしばらくすると、「パースペクティブ切り替えの確認」ウィンドウが表示されますので、「切り替え(S)」ボタンをクリックします。

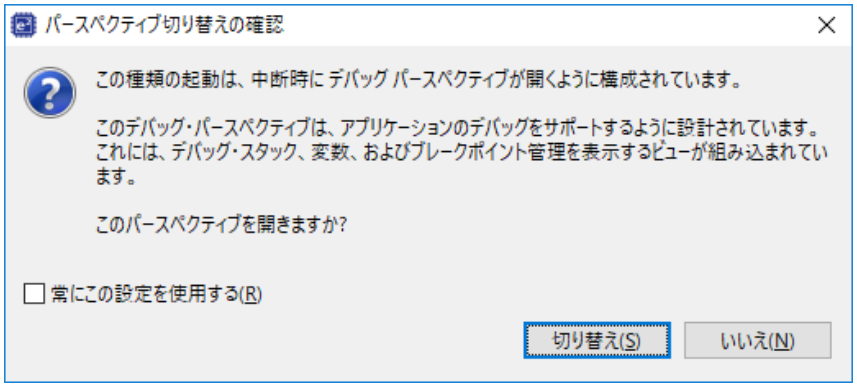


図 13　パースペクティブの切り替え確認

デバッグパースペクティブに切り替わりますので、「再開(M)」ボタンをクリックします。main 関数の先頭でブレークしますので、再度「再開(M)」ボタンをクリックしてプログラムを実行します。（main 関数の先頭でブレークするのは、デバッグ接続直後の 1 回目のみです。）

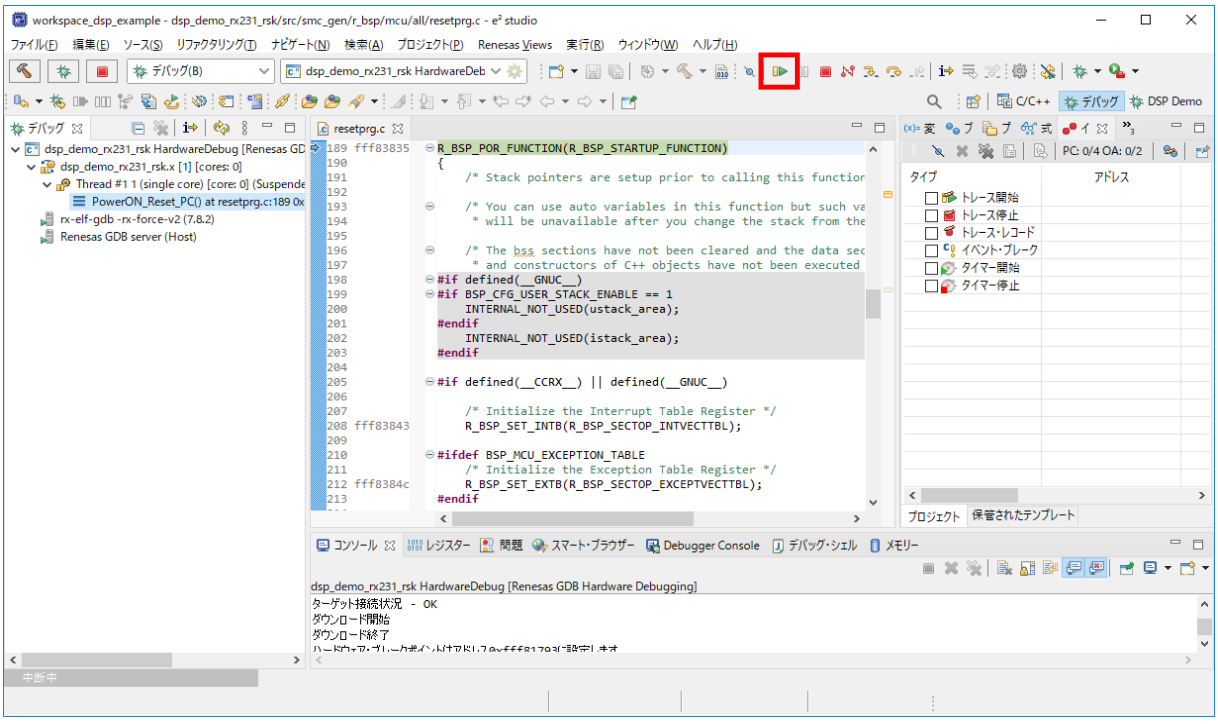


図 14　デバッグ接続後のワークスペース

プログラムが実行されたことを確認後、ファンクションジェネレータから信号波形を入力してください。

2.3.2　LCD の表示内容とフィルタ特性の変更方法

サンプルプログラムを実行すると、処理結果が LCD モジュールに表示されます。表示内容の詳細は 3.4.4 FFT 処理結果の判定を参照してください。一定周期（約 1 秒）で LCD モジュールの表示が更新されます。

サンプルプログラム実行中に Evaluation board のスイッチ（SW1）を押下すると、IIR フィルタの特性を切り替えることができます。適用中のフィルタ特性は LCD モジュールに表示されており、スイッチ（SW1）を押下するたびに「FLAT」（初期値）→「HPF」→「LPF」→「FLAT」（最初に戻る）と LCD モジュールの表示が切り替わり、IIR フィルタ処理に適用されます。

Target Board 用のサンプルプログラムには LCD モジュールへの表示機能はありませんが、LCD モジュールに表示している情報の元となる変数を e² studio のデバッグ機能で確認することができます。

2.3.3　e² studio の機能を使ったシステム動作のモニタ

e² studio には多くのデバッグ機能が搭載されています。本アプリケーションノートでは、下記の機能を使用してシステムの動作をモニタするデモ用パースペクティブ（e² studio の画面構成）を用意しています。

- 「メモリ」ビューの「Waveform」機能による波形表示
- 「Visual Expression」ビューによる各グローバル変数値のモニタ

デバッグ接続後、パースペクティブの中から「DSP Demo」（図 15）をクリックしてください。パースペクティブを切り替えることができます。

なお、このパースペクティブはワークスペースの情報に含まれておりますので、2.1 ワークスペースの起動に記載のとおり、本アプリケーションノートに同梱しているワークスペースをご使用ください。

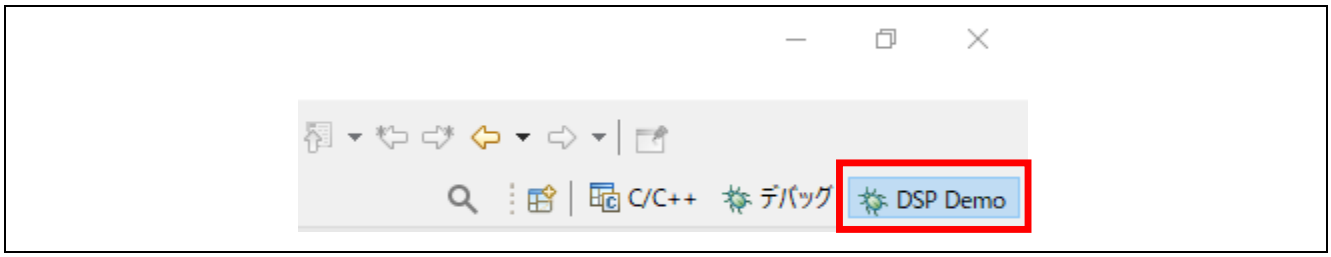


図 15　パースペクティブの切り替え

デモ用パースペクティブ（e² studio の画面構成）は図 16 のとおりです。スライダーをドラッグすると、周波数とレベルの期待値を設定することができます。スライダーからの変更は LCD モジュールにも反映されます。

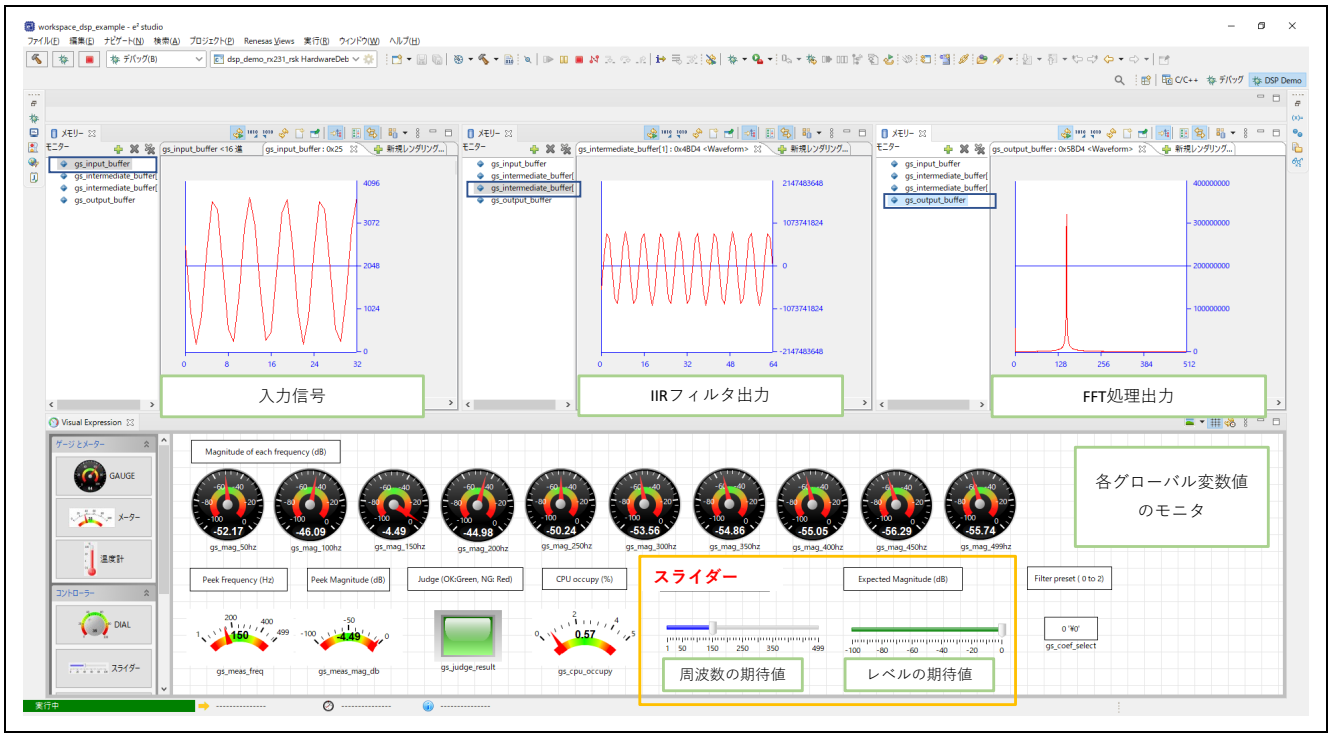


図 16　DSP Demo パースペクティブの内容

「メモリ」ビューは「リアルタイムリフレッシュ」を有効にすることで、指定した周期で更新されます。（図 17）

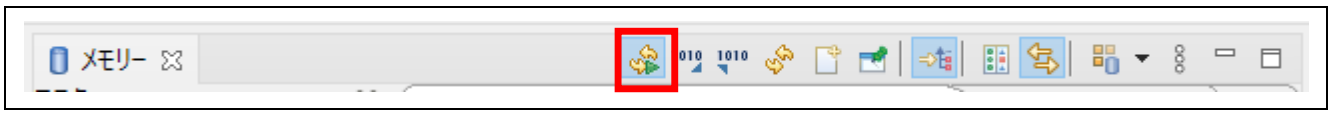


図 17　リアルタイムリフレッシュの有効

デモ用パースペクティブで使用している「メモリ」ビューの「Waveform」機能と「Visual Expression」ビューの使用方法については 5.1 e^2 studio による信号処理のモニタ および e^2 studio のヘルプを参照してください。

ヘルプ(H) → 「e2 studio ユーザーガイド」→「デバッグに関する機能」→「ビュー」

- 「メモリー」→「Waveform メモリー・レンダリング」
- ビジュアル・エクスプレッション

2.4　ユーザ変更可能な設定

本システムでユーザが変更可能な設定を表 4 に示します。これらの定義は main.h ファイルで定義されています。

表 4 変更可能な設定（main.h）

機能/定義名	説明	デフォルト値
入力データの切り替え		
SAMPLE_DATA_MODE	IIR フィルタ処理、FFT 処理する信号ソースを選択します。 “0”: 外部入力信号 “1”: サンプルデータ “2”: 正弦波生成処理	0
SELECT_SAMPLE_DATA	使用するサンプルデータを選択します。 設定値： 1：40Hz と 400Hz の正弦波を重ねた信号 2：250Hz と 7.8125Hz の正弦波を重ねた信号	1
処理結果の判定条件		
EXPECTED_FREQUENCY	ピーク周波数（FFT 処理の結果による振幅値が最大の周波数）の期待値 設定単位：Hz 設定範囲：1～499	150
EXPECTED_MAGNITUDE	EXPECTED_FREQUENCY で指定したピーク周波数の振幅値を OK と判定する閾値 設定単位： dB（デシベル） 設定範囲(注)：-60.0～0.0	-6.6

注. S12AD の出力信号の振幅値が最大のときを 0dB とし設定を行って下さい。本サンプルプログラムのデシベル変換については 3.4.5 FFT 処理結果の判定 を参照してください。

3. サンプルプログラムの説明

3.1 サンプルプログラムの概要

本サンプルプログラムは、表 5 に示す処理から構成されています。

表 5 各処理の役割

処理	役割
main 処理	・ 周辺機能と DSP 関連処理の初期化 ・ 最初の DMA 転送の開始 ・ DMA 転送終了割り込み処理への DMA 転送先の通知 ・ 入力データの正規化処理 ・ IIR フィルタ処理、FFT 処理などの DSP 関連処理 ・ スイッチ（SW1）入力による IIR フィルタ特性の変更処理 ・ FFT 処理結果の判定と LCD モジュールの表示内容更新処理 ・ DSP 処理と入力信号 1024 サンプルの取り込みにかかるサイクル数のカウント処理
DMA 転送終了割り込み処理	・ 2 回目以降の DMA 転送の設定と転送の開始
CMT0 割り込み処理	・ LCD モジュールの描画更新処理（r_lcd_driver モジュールによる制御）

3.2　処理シーケンス

サンプルプログラムの処理シーケンスは主に main 処理と DMA 転送終了割り込み処理、CMT0 割り込み処理の 2 つに分けられ、この 2 つの処理は独立して動作しています。main 処理と DMA 転送終了割り込み処理のシーケンスを図 18 に、CMT0 割り込み処理のシーケンスを図 19 に示します。

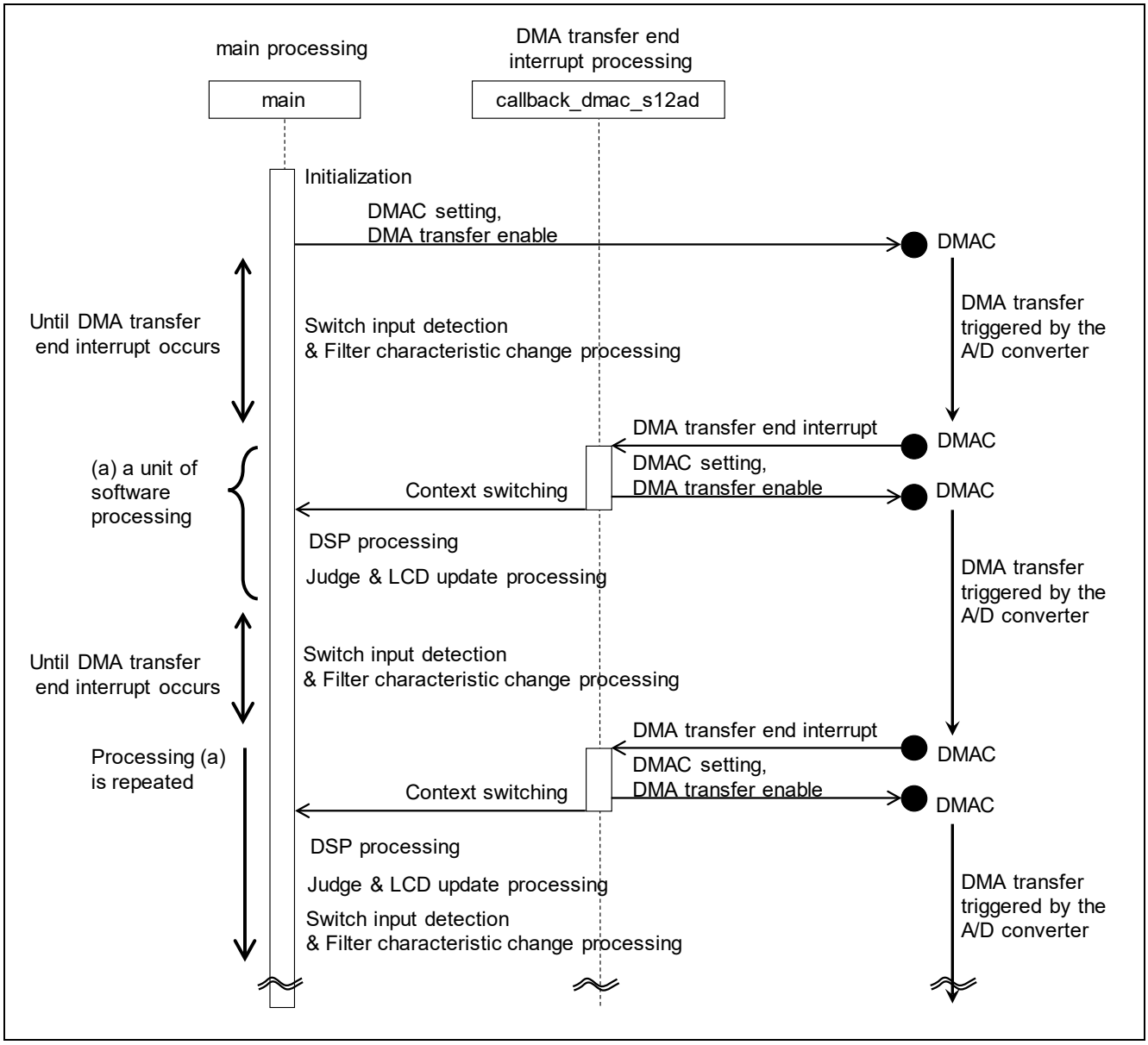


図 18　サンプルプログラムの処理シーケンス（main 処理と DMA 転送終了割り込み処理）

図中、最初の DMA 転送は main 処理で A/D 変換完了による転送を許可します。指定データ数の DMA 転送の終了による DMA 転送終了割り込み要求が発生します。これをトリガとして DMA 転送終了割り込み処理と main 処理を順次実行します。以後同じ処理を繰り返し実行します。

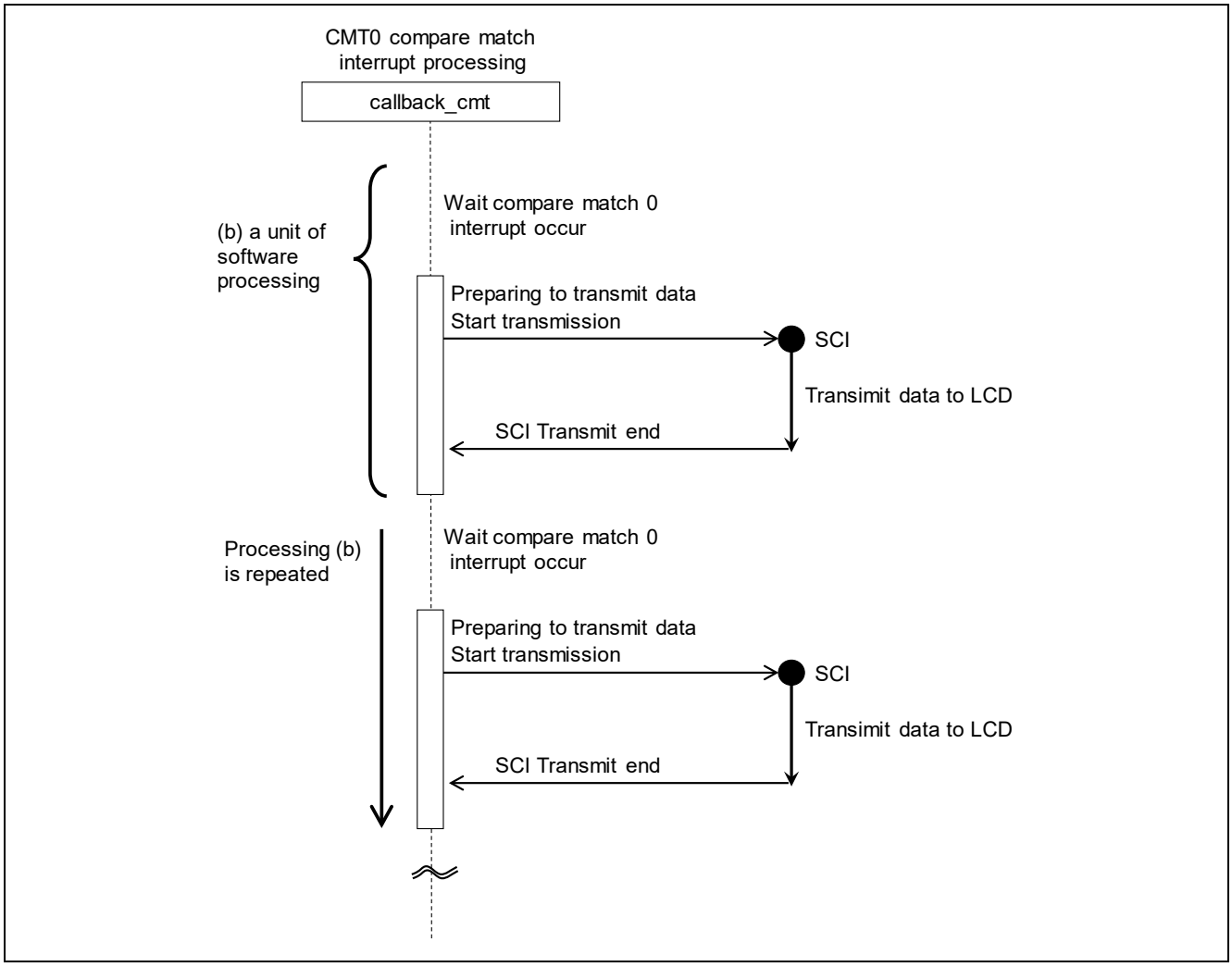
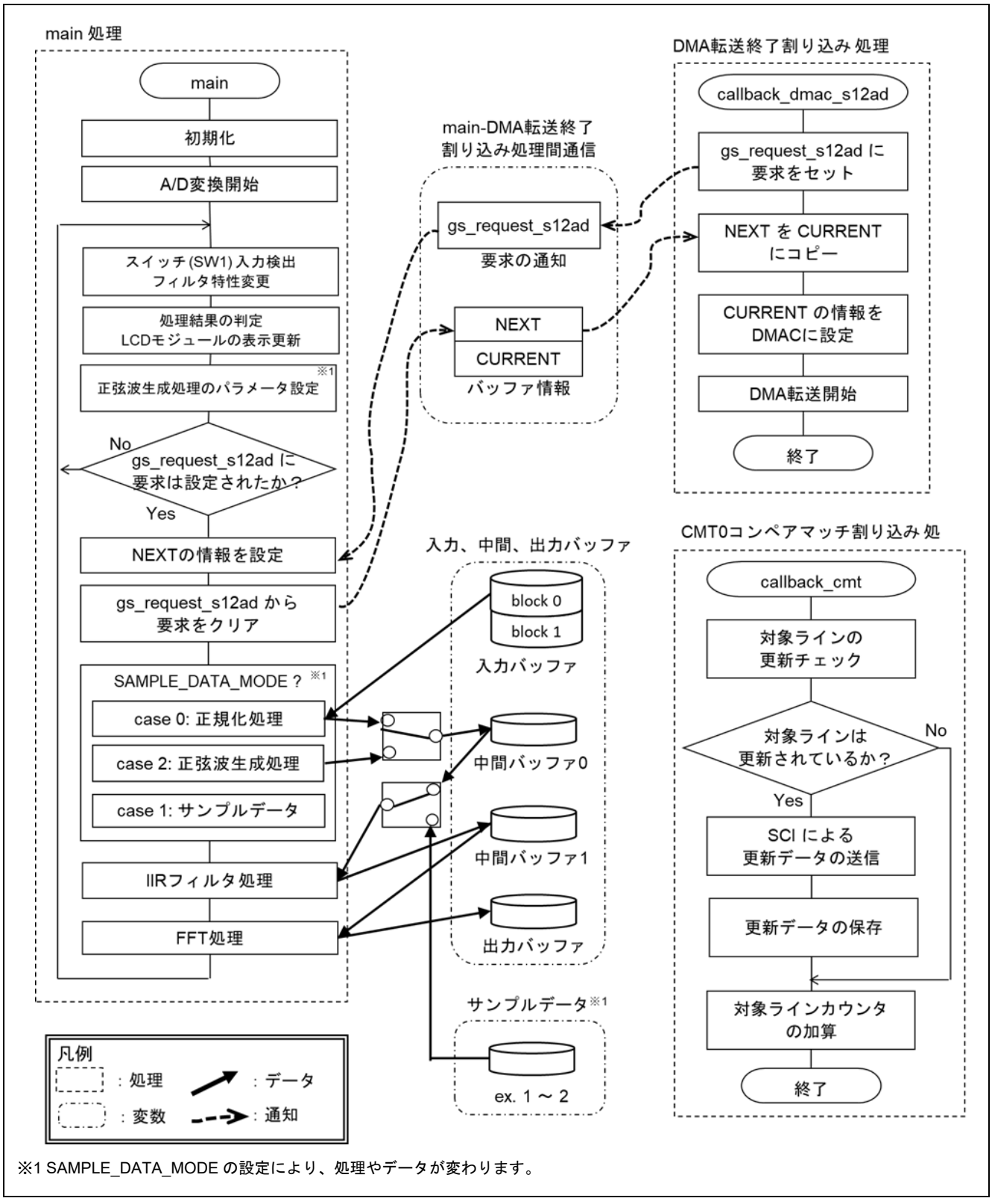


図 19　サンプルプログラムの処理シーケンス（CMT0 割り込み処理）

main 処理で周辺機能の初期化後、CMT0 と SCI により LCD モジュールへの表示処理を行います。CMT0 の割り込み処理内では、SCI により LCD モジュールに 1 ライン分のデータを送信します。CMT0 割り込み要求が発生するたびに 1 ラインずつ更新します。LCD モジュールの縦幅は 128 ピクセルのため、128 回分の CMT0 割り込み要求で 1 画面更新します。

3.3　処理フロー

サンプルプログラムの処理フローを図 20 に示します。



※1 SAMPLE_DATA_MODE の設定により、処理やデータが変わります。

図 20　サンプルプログラムの処理フロー

図 20 の要素について以下に説明します。

- main 処理
  最初に初期化します。その後、DMAC により A/D 変換結果がバッファに格納されるまで、スイッチ（SW1）入力の検出とフィルタ特性の更新処理を行います。DMA 転送終了の通知がくると、次の DMA 転送先の設定と入力バッファのデータの正規化処理、および DSP 処理（IIR フィルタ処理、FFT 処理）を行い、FFT 処理結果による情報を更新します。以後同じ処理を繰り返し実行します。
- DMA 転送終了割り込み処理
  次の DMA 転送先の更新を main 処理に要求し、今回の DMA 転送を開始させます。
- main-DMA 転送終了割り込み処理間通信
  main 処理から DMA 転送終了割り込み処理への次の DMA 転送先の通知、DMA 転送終了割り込み処理から main 処理への次の DMA 転送先の更新の要求は、本処理間通信の変数を介して行います。表 6 に各変数について説明します。
- 入力バッファ
  DMAC が A/D 変換結果を格納し、正規化処理が入力データとして読み出します。
  CPU（正規化処理）と DMAC のアクセス競合を避けるため 2 ブロック構成とします。CPU、DMAC それぞれがアクセスするバッファは、図 21 に示すように DMA 転送終了割り込みをトリガとして切り替えます。図 22 に入力バッファの構成を示します。
- 中間バッファ
  正規化処理の結果、および IIR フィルタ処理の結果を格納します。図 22 に中間バッファの構成を示します。
- 出力バッファ
  FFT 処理が処理結果を格納します。図 22 に出力バッファの構成を示します。
- サンプルデータ
  マクロ定義 SAMPLE_DATA_MODE が“1”の場合に使用します。外部信号入力のデータの代わりにサンプルデータを IIR フィルタ処理の入力にします。
- CMT0 コンペアマッチ割り込み処理
  LCD モジュールの表示内容を更新します。LCD モジュールの幅と高さはそれぞれ 128 ピクセルとなっており、CMT0 コンペアマッチ割り込み処理ごとに 1 ラインずつ更新します。各ラインを更新するとき、表示中のライン情報と比較して差分があれば LCD モジュールに 1 ライン分の更新データを送信します。更新データは次に同じラインを更新する際の比較データとして保存します。また、差分が無ければそのラインの更新はスキップします。
- 正弦波生成処理
  マクロ定義 SAMPLE_DATA_MODE が“2”の場合に使用します。外部信号入力のデータの代わりにサンプルプログラム上で生成した信号のデータを IIR フィルタ処理の入力にします。

表 6 main-DMA 転送終了割り込み処理間通信の変数

種類		説明
(a)　要求の通知		DMA 転送終了割り込み処理 が、(b)のバッファの情報の更新を main 処理に要求するための変数。DMA 転送終了割り込み処理が要求をセットする。セットされている場合 main 処理は、(b)　の NEXT 面の情報を更新し、要求をクリアする。 初期化時、main 処理がクリアする。
(b)　バッファ情報		DMA 転送先のバッファの先頭アドレスとデータ数を格納する変数。 処理間のアクセス競合を避けるため NEXT と CURRENT の 2 面で構成する。初期値は 2 面共 main 処理が設定する。
	NEXT	main 処理が、(a)に応じ先頭アドレスとデータ数を格納する面。 以下に初期値を示す。 ・　先頭アドレス：　Input Buffer の block 1 の Data 0 ・　データ数：1024
	CURRENT	DMA 転送終了割り込み処理が、DMAC に設定する先頭アドレスとデータ数を参照する面。DMA 転送終了割り込み処理自身が NEXT 面の情報を CURRENT 面にコピーする。 以下に初期値を示す。 ・　先頭アドレス：　Input Buffer の block 0 の Data 0 ・　データ数：1024

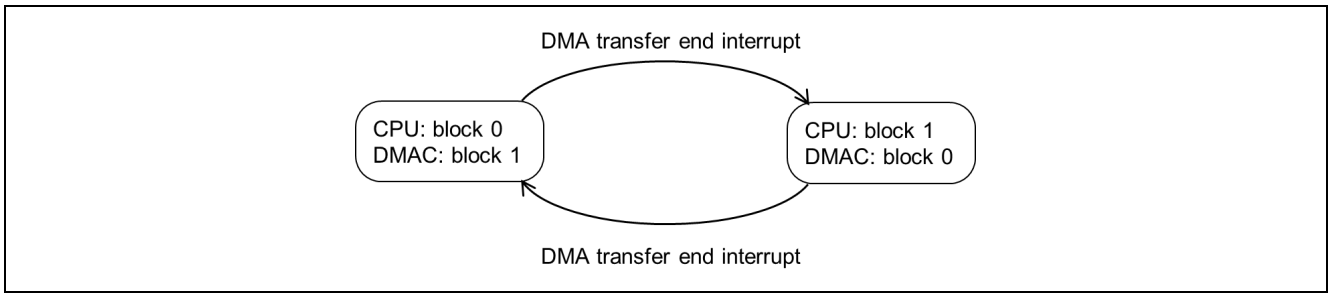
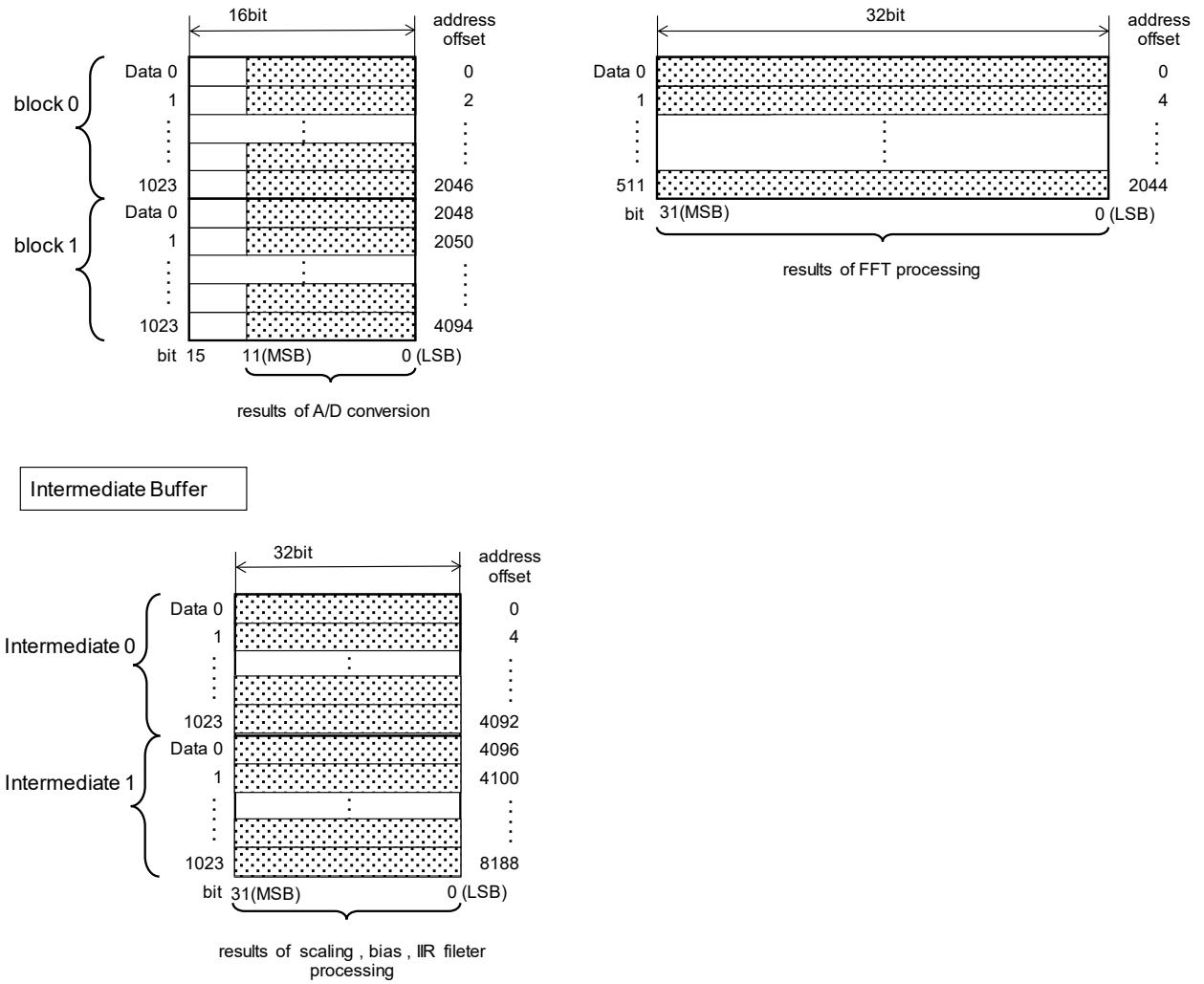


図 21　CPU と DMAC がアクセスする入力バッファの切り替え



注. DSP ライブラリの実数 FFT の API 仕様により、FFT の結果のポイント数は入力の 1/2 となります。
　　このため出力バッファのデータ数は入力および中間バッファの 1/2 とします。

図 22　入力、中間、出力バッファ

3.4　詳細

3.4.1　信号処理フロー

本サンプルプログラムの信号処理フローを図 23 に示します。このうち DSP 処理の詳細を 3.4.2 ～ 3.4.4 で説明します。

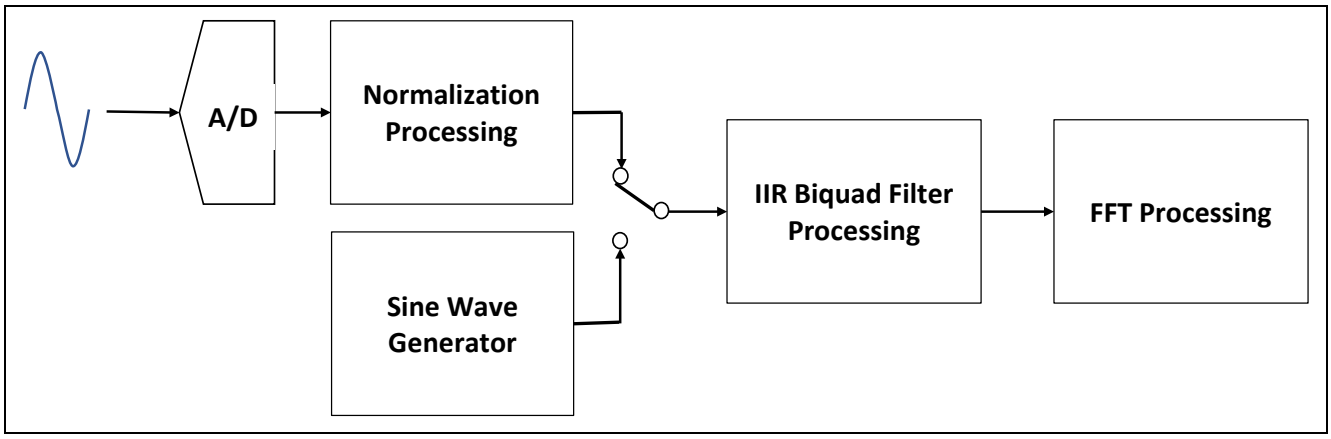


図 23　信号処理フロー

表 7 信号処理の値域

アナログ信号入力 (A/D)	正規化処理 (Normalization)	IIR フィルタ処理 (IIR Biquad Filter)	FFT 処理 (FFT)
0 ～ 2^{12}-1	-2^{30} ～ 2^{30}-1	-2^{30} ～ 2^{30}-1	-2^{29} ～ 2^{29}-1
正弦波生成処理 (Sine Wave Generator)			
-2^{30} ～ 2^{30}-1			

3.4.2　正規化処理

サンプルプログラムでは、図 24 に示す正規化処理（バイアス処理とスケーリング）を行っています。

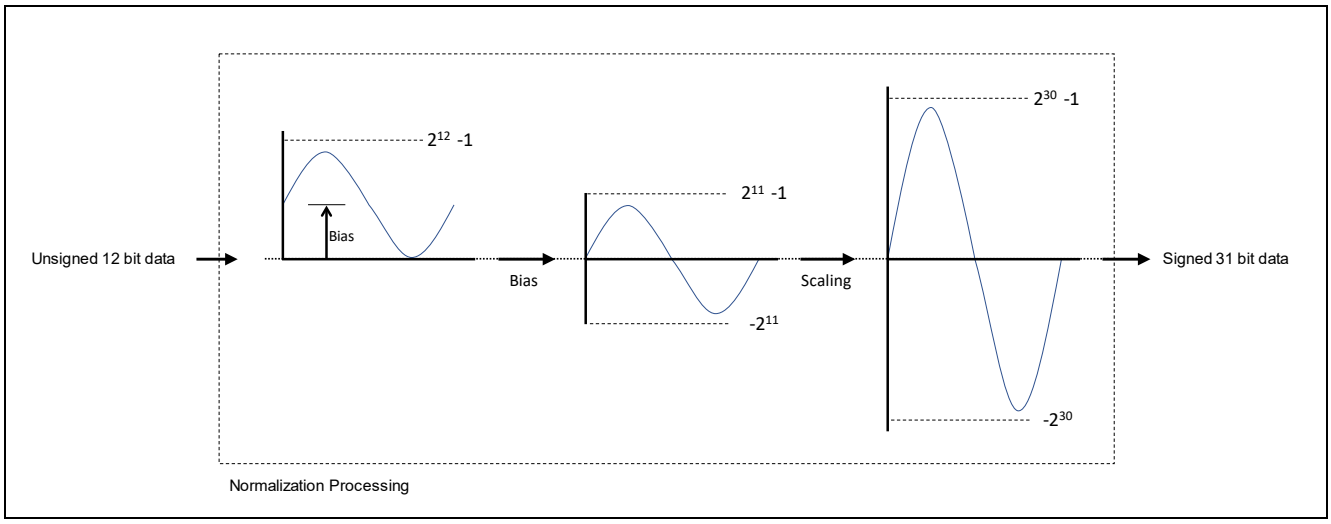


図 24　正規化処理

- 正規化処理の構成
  S12AD から入力されるデータは 12 ビット（符号なし）のため、IIR フィルタ処理や FFT 処理において十分な演算結果から得られるよう 31 ビット（符号付き）に正規化します。正規化処理の内容はバイアス処理とスケーリングです。

3.4.3　正弦波生成処理

サンプルプログラムは、図 25 に示す正弦波発生機能で正弦波を生成することができます。

図に出力信号の生成例を示します。

- 2 個のオシレータ（正弦波生成処理）を実装
  各オシレータの周波数と振幅を調整しミックスすることが可能（表 8 に調整範囲を示す）
- オシレータは図 26 に示す 2 次の IIR フィルタを応用した信号処理で生成
- 各オシレータの出力信号の値域：-1073741824～1073741823（31 ビット符号付き整数）
- 2 つのオシレータをミックスした信号を-1073741824～1073741823 の範囲でリミットし、31 ビット符号付き整数型で出力
- 調整方法は「5.1.3 信号発生機能」を参照

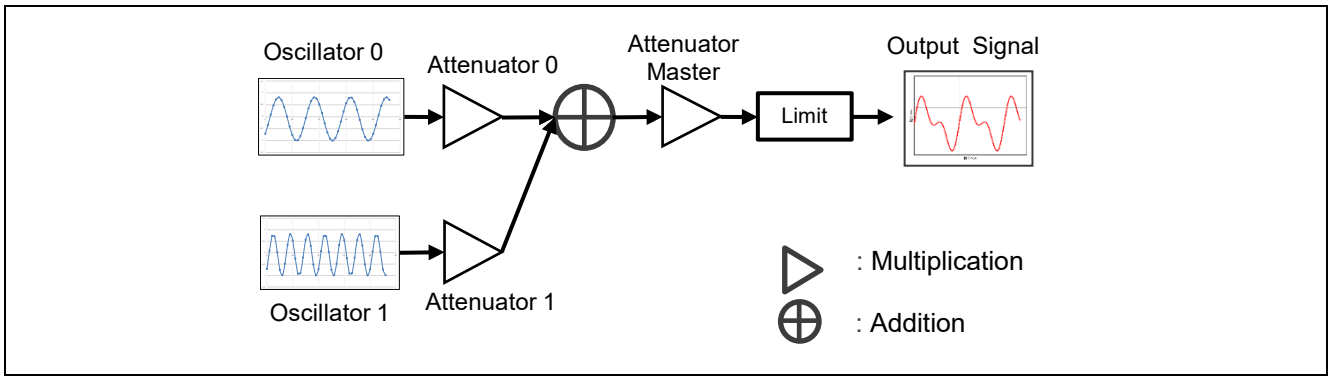


図 25　正弦波生成処理

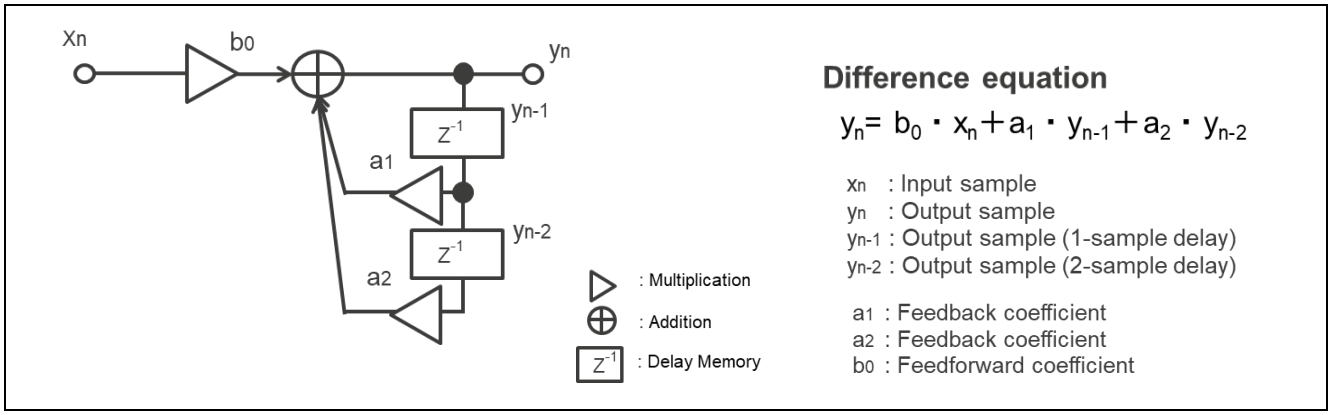


図 26　IIR フィルタを応用したオシレータ処理のシグナルフロー

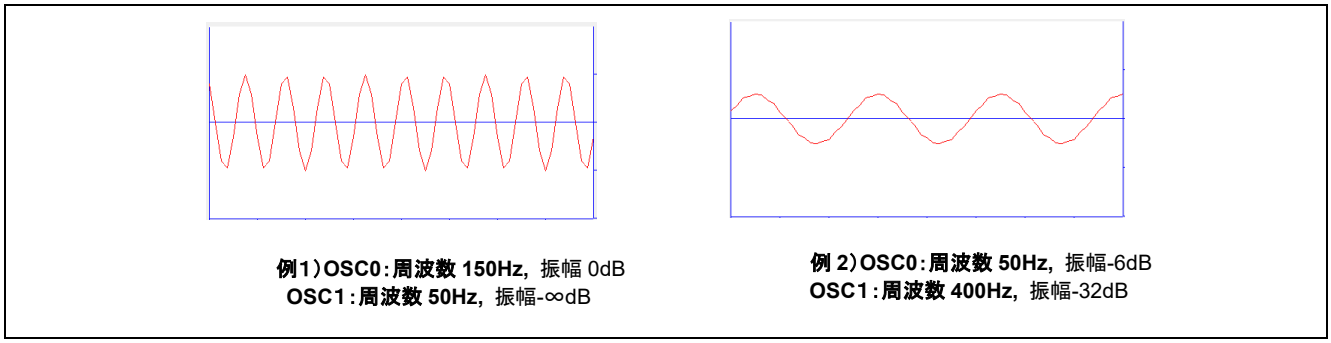


図 27　信号の生成例

表 8 正弦波の出力範囲

項目	調整範囲	調整単位
正弦波の周波数	0Hz~490Hz	10Hz
正弦波の出力レベル	-Infinity, -98dB ~ 0dB	-6 ～ 0 dB: 0.2 dB -24 ～ -6 dB: 0.5 dB -44 ～ -24 dB: 1 dB -56 ～ -44 dB: 2 dB -98 ～ -56 dB: 6 dB (-Infinity は 0.0 とします)
ミックスした信号の出力レベル		

3.4.4 IIR フィルタ処理

サンプルプログラムでは、図 28 に示す DSP ライブラリの API 関数を使って入力信号にフィルタ処理を行っています。

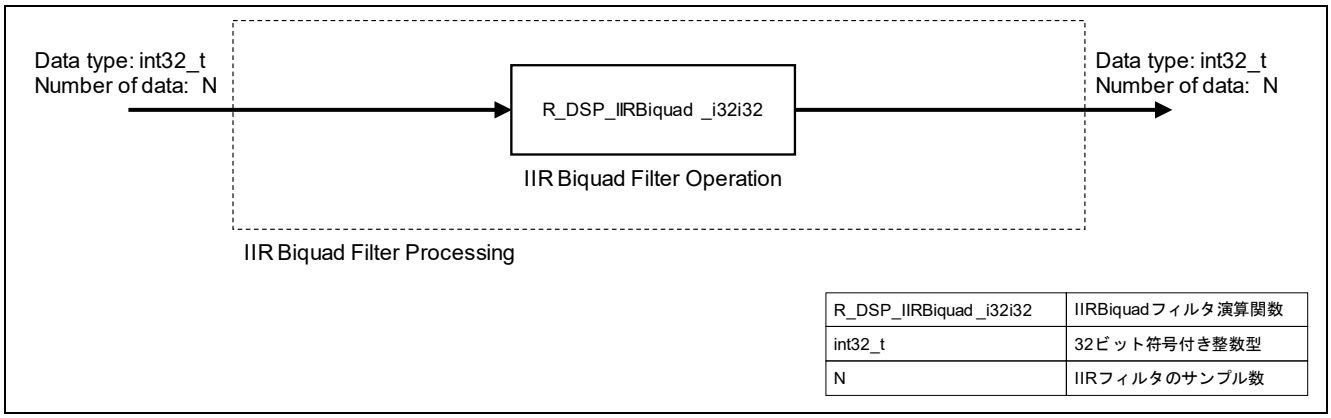


図 28　IIR フィルタ処理のデータフロー

- IIR フィルタ処理を構成する API
  IIR フィルタ処理は、IIR Biquad フィルタ演算関数 R_DSP_IIRBiquad_i32i32 からなります。
  R_DSP_ IIRBiquad_i32i32 は、関数に与えるフィルタ係数に応じた演算結果を出力します。サンプルプログラムでは、パススルー（FLAT）、ローパスフィルタ（LPF）、ハイパスフィルタ（HPF）の 3 つのフィルタ特性を用意しています。演算時に関数に与えるフィルタ特性はソフトウェアで切り替えることができるようにしています。
- 入力、出力データについて

  サンプルプログラムでは、IIR フィルタ処理において 1024 サンプルのデータを int32_t 型で入力し、同じく int32_t 型で出力します。

  API 仕様の詳細は「RX DSP ライブラリ API Version 5.0 ユーザーズマニュアル ソフトウェア編」を参照してください。

3.4.5　FFT 処理

サンプルプログラムは、図 29 に示す DSP ライブラリの API 関数を使って入力信号の周波数スペクトルの振幅特性を出力します。

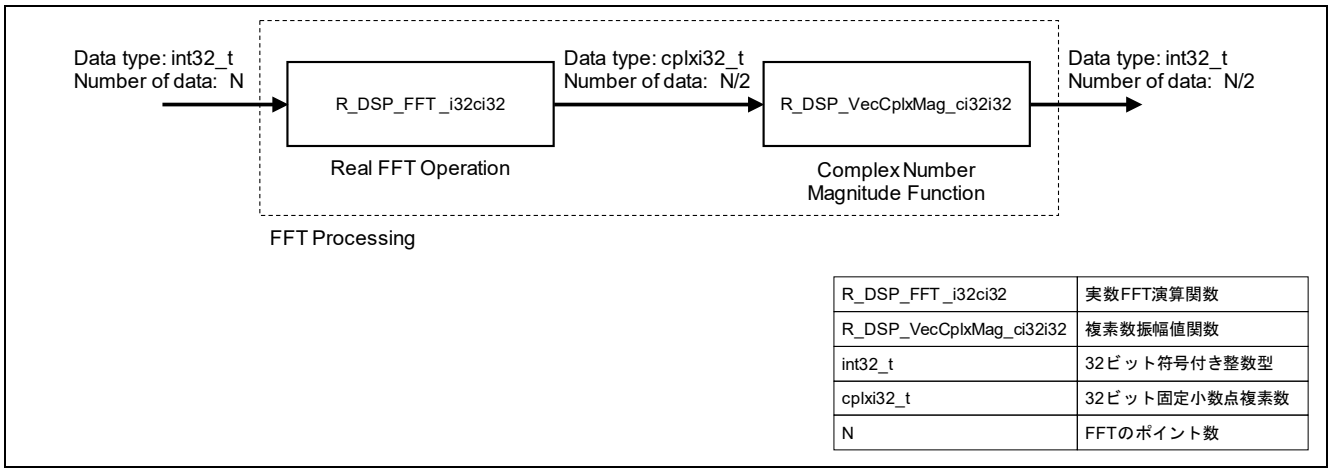


図 29　FFT 処理のデータフロー

- FFT 処理を構成する API
  FFT 処理は、実数 FFT 演算関数 R_DSP_FFT_i32ci32 と複素数振幅値関数 R_DSP_VecCplxMag_ci32i32 からなります。
  R_DSP_FFT_i32ci32 は、FFT 結果を複素数で出力します。サンプルプログラムは、入力信号の周波数スペクトルの振幅特性を得ることを目的としたため、R_DSP_FFT_i32ci32 の複素数出力を R_DSP_VecCplxMag_ci32i32 で整数による振幅値に変換しています。

- 入力、出力データについて
  FFT 処理は、N 個の入力データに対し、N/2 個の結果データを出力します。N は FFT のポイント数に相当し、R_DSP_FFT_i32ci32 の 1 回の処理単位です。R_DSP_FFT_i32ci32 の API 仕様により、結果のポイント数は入力の 1/2 となります。サンプルプログラムでは、1024 ポイントのデータを入力し、512 ポイントの結果が出力されます。また、R_DSP_VecCplxMag_ci32i32 は Q2.30 フォーマットで出力する仕様のため、出力される振幅値は入力に対し 1/2 となります。

  API 仕様の詳細は「RX DSP ライブラリ API Version 5.0 ユーザーズマニュアル ソフトウェア編」を参照してください。

- 出力データと周波数の関係
  FFT 処理の出力データは、周波数に対する振幅を示します。図 22 に示す出力バッファに表 9 のように周波数に対応する振幅値が格納されます。
  周波数の刻みは、サンプリング周波数と FFT のポイント数により得られます。サンプルプログラムの場合、以下の式により約 0.97Hz 刻みとなります。

  サンプリング周波数/FFT のポイント数 = 1000/1024 = 0.9765625Hz

表 9 出力データと周波数の関係

データ番号	周波数 [Hz]
0	0.97… x 0 = 0
1	0.97… x 1 = 0.97
2	0.97… x 2 = 1.95
:	:
510	0.97… x 510 = 498.04
511	0.97… x 511 = 499.02

3.4.6　FFT 処理結果の判定

出力バッファに格納された FFT 処理結果を判定します。下記 2 点がともに成立する場合に OK と判断します。

- ピーク周波数 = 期待周波数 （定義 EXPECTED_FREQUENCY）であること
- ピーク周波数の振幅値（dB） ≧ 期待振幅値（dB）（定義 EXPECTED_MAGNITUDE）であること

ピーク周波数は、FFT 処理の結果による振幅値が最大の周波数です。期待周波数は 1Hz 刻みで設定しますが、FFT 処理の出力データと周波数は表 7 のように約 0.97Hz 刻みとなっています。このため、期待周波数は対応するデータ番号に変換して比較します。変換されたデータ番号の前後 1 データを許容範囲としています。

振幅値の比較はピーク周波数の振幅値をデシベル（dB）単位に変換して行っています。

デシベル変換は以下に示す数式で行っています。

振幅値（単位：dB）= $20 * \log_{10}(\text{FFT 処理後の振幅値} / 2^{29}-1)$

なお、A/D 変換や信号処理の演算精度などにより、測定結果は 0dB 丁度にならない場合があります。演算精度については 3.4.1 信号処理フローを参考にしてください。

3.4.7　LCD モジュールの表示内容更新処理

LCD モジュールに表示される内容を表 10 に示します。表示内容の更新は以下 2 つの処理に分けて行っています。

- main 処理
  表示内容の更新に必要なデータを出力バッファやその他の情報を基に算出し、算出した数値を表示用データに変換して表示更新用バッファに書き込みます。
- CMT0 割り込み処理
  表示更新用バッファから表示用データを読み出し、3.2 処理シーケンスで説明したように CMT0 割り込みごとに 1 ラインずつ LCD モジュールの表示内容を更新します。

表 10 LCD モジュールの表示内容

項目	表示例	説明
IIR フィルタ特性	IIR : FLAT	フィルタ特性 FLAT, LPF, HPF を表示
ピーク周波数	Meas.Freq: 250 Hz	ピーク周波数の測定値
ピーク周波数の振幅値	Meas.Mag: -6.6 dB	ピーク周波数の振幅の測定値 dB で小数第 1 位まで表示
期待周波数	Exp.Freq: 250 Hz	ピーク周波数の期待値
期待振幅値	Exp.Mag: -65.7 dB	ピーク周波数の振幅値を OK と判定する閾値 dB で小数第 1 位まで表示
判定	Judge: NG	判定結果 OK または NG を表示(注)
DSP 処理サイクル数	DSP Proc: 89528 Cyc	DSP 処理（正規化処理、IIR フィルタ処理、FFT 処理）のサイクル数
入力信号の取り込みサイクル数	Unit Proc: 27647946 Cyc	入力信号 1024 サンプルの取り込みにかかるサイクル数

DSP 処理の CPU 占有率	CPU Occupy: 0.324 %	(DSP 関連処理サイクル数 / 入力信号 1024 サンプルの取り込みにかかるサイクル数) をパーセント（%）で表示

注.　期待周波数とピーク周波数が不一致であっても OK と表示される場合があります。3.4.4 FFT 処理結果の判定に示す判定処理が許容範囲を持つためです。

3.4.8　IIR フィルタ特性の変更処理

サンプルプログラム実行中に Evaluation board のスイッチ（SW1）を押下すると、IIR フィルタの特性を切り替えることができます。スイッチ（SW1）を押下すると IRQ 割り込み要求が発生して割り込み処理内で検出判定用のフラグを“1”にセットします。main 処理内で検出判定用のフラグを確認し、フラグがセットされていれば IIR フィルタの特性を変更してフラグを“0”にクリアします。変更後、表 10 で示す IIR フィルタ特性の表示を更新します。

3.4.9　サイクル数のカウント処理

DSP 処理（正規化処理、IIR フィルタ処理と FFT 処理）と、入力信号 1024 サンプルの取り込みにかかるサイクル数を計測しています。これらの計測結果より、サンプルプログラムにおける DSP 処理の CPU 占有率を算出します。算出した結果は LCD に表示しています。計測区間を図 30 に示します。

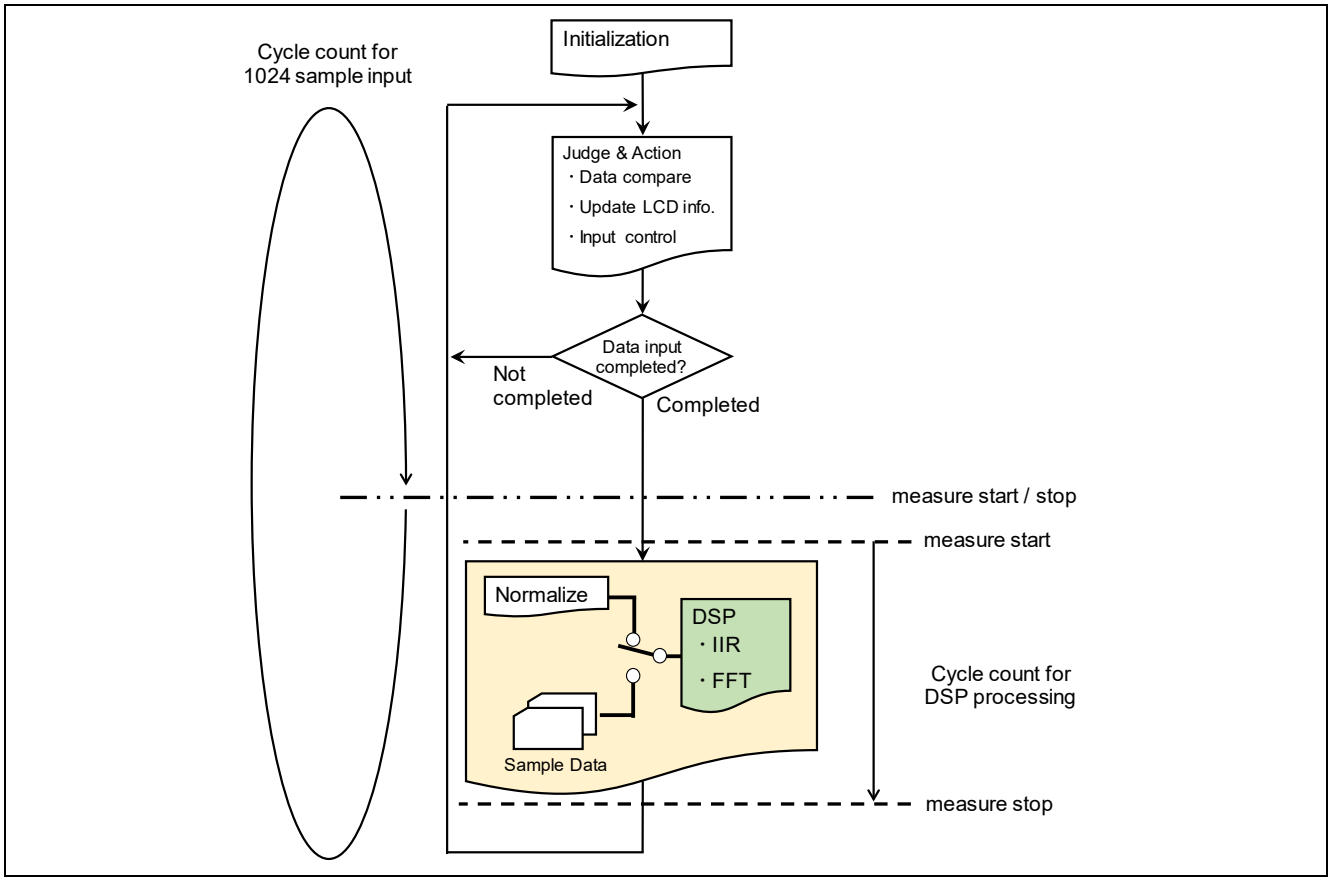


図 30　サイクル数の計測区間

3.5　ファイル構成

本アプリケーションノートで用意したモジュールのファイル構成を表 11 に示します。また、各ソースファイルの関数を表 12～表 16 に示します。その他のモジュールの詳細は各アプリケーションノートを、設定は「5.3 ソフトウェアモジュールの設定」を参照してください。

表 11　モジュールのファイル構成

モジュール/ファイル		説明
main		サンプルプログラムのメイン処理
	main.c	main 処理と DMA 転送終了割り込み処理など
	main.h	Main.c のヘッダファイル
r_dsp		DSP 関連処理
	r_normalize.c	正規化処理
	r_normalize.h	r_normalize.c のヘッダファイル
	r_dsp_iirbiquad.c	IIR フィルタ処理の初期化、IIR フィルタ処理
	r_dsp_iirbiquad.h	r_dsp_iirbiquad.c のヘッダファイル
	r_dsp_real_fft.c	FFT 処理の初期化、FFT 処理
	r_dsp_real_fft.h	r_dsp_real_fft.c のヘッダファイル
	r_signal_gen.c	正弦波生成処理
	r_signal_gen.h	r_signal_gen.c のヘッダファイル
	r_signal_gen_att_coef.c	正弦波生成処理の周波数テーブル
	r_signal_gen_freq_coef.c	正弦波生成機能の出力レベルテーブル
r_lcd_driver		LCD モジュールの制御処理 （RSK 用）
	r_ascii.c	文字テーブル
	r_ascii.h	r_ascii.c のヘッダファイル
	r_lcd_driver.c	LCD モジュールの制御処理
	r_lcd_driver_private.h	プライベートヘッダファイル
	r_lcd_driver_if.h	インタフェースヘッダファイル

表 12 main.c ファイルの関数一覧

関数名	説明
main	・　周辺デバイスを初期化 ・　最初の DMA 転送の開始 ・　DMA 転送終了割り込み処理 に次の DMA 転送先を通知 ・　DSP 処理の実行 ・　処理結果の判定と LCD モジュールへの表示
set_buf_info	main-DMA 転送終了割り込み処理間通信の変数に次の DMA 転送先アドレスとデータ数を設定
init_dmac_s12ad	S12AD を転送元として DMAC のチャネル 0 を初期化
callback_dmac_s12ad	main 処理により DMAC モジュールに登録されるコールバック関数。 ・　DMAC チャネル 0 の DMA 転送終了割り込み処理 ・　次の転送先を DMAC に設定し、DMA 転送を許可 ・　次の転送先の情報を main 処理に要求
init_lcd_display	LCD モジュールの初期化と初期表示の設定（RSK 用）
change_coef	フィルタ特性の変更処理
judge_and_update	・　FFT 結果の判定と LCD モジュールに表示する情報の更新

cycle_measure_start	DSP 処理または入力信号の 1024 サンプル取り込みサイクル数の測定を開始する。
cycle_measure_stop	DSP 処理または入力信号の 1024 サンプル取り込みサイクル数の測定を停止し、サイクル数を算出して戻す。

表 13 r_normalize.c ファイルの関数一覧

関数名	説明
R_Normalize_Operation	・　正規化処理

表 14 r_dsp_iirbiquad.c ファイルの関数一覧

関数名	説明
R_DSP_IIRBiquad_Init	・　IIR フィルタ処理の初期化
R_DSP_IIRBiquad_Operation	・　IIR フィルタ処理の実行
R_DSP_IIRBiquad_UpdateCoef	・　IIR フィルタ特性の設定と遅延データ配列のクリア

表 15 r_dsp_real_fft.c ファイルの関数一覧

関数名	説明
R_DSP_REAL_FFT_Init	・　FFT 処理の初期化
R_DSP_REAL_FFT_Operation	・　FFT 処理の実行

表 16 r_lcd_driver.c ファイルの関数一覧

関数名	説明
R_LCD_Driver_Open	・　LCD モジュールとドライバの初期化
R_LCD_Driver_PrintString	・　更新用バッファへの文字列書き込み
R_LCD_Driver_PeriodicCallback	・　CMT0 の割り込み処理内でコールされる LCD モジュールの 1 ライン更新処理

表 17 r_signal_gen.c ファイルの関数一覧

関数名	説明
R_Signal_Gen_Init	・　正弦波生成処理の初期化
R_Signal_Gen_Operation	・　正弦波生成処理の実行
R_Signal_Gen_osc_freq	・　正弦波の周波数の変更
R_Signal_Gen_osc_att	・　正弦波の出力レベルの変更
R_Signal_Gen_master_att	・　ミックスした信号の出力レベルの変更

4. 注意事項

4.1 FFT 処理結果の周波数値

Target Board 上の RX231 は HOCO をクロックソースとして動作します。HOCO は最大±2%の誤差を含んでおり、サンプルプログラムの A/D 変換処理のサンプリング周波数の誤差となります。サンプリング周波数の誤差は FFT 処理結果の周波数の誤差となるため、デモシステムのピーク周波数表示は入力信号の周波数より数 Hz ずれる場合があります。デモシステムをより正確な周波数解析が必要なシステムに応用する場合、RX231 のクロックソースとして精度の高い発振子を使用し、サンプルプログラムのクロック設定の変更を行ってください。

4.2 エイリアシング

本書で使用する評価ボードは、RX231 の A/D 変換に入力する信号のエイリアシングへの対策を行っていません。サンプリング周波数の 1/2 を超える周波数の信号を A/D 変換に入力するとエイリアシングが発生します。本書を参考にシステムを設計する場合、必要に応じアンチエイリアシングフィルタを RX231 に外付けするなどの対応を行ってください。

4.3 オシレータ出力信号の経時変化

サンプルプログラムの正弦波生成処理は、簡易的な動作確認やデモンストレーションを想定して設計しています。長時間の連続動作ではオシレータの構造により出力信号に経時変化を伴います。お客様のシステムに本書のオシレータを応用する場合、要求に照らし妥当性を検討したうえでご使用下さい。

5.　参考

5.1　e² studio による信号処理のモニタ

5.1.1　Waveform レンダリング

e² studio の Waveform レンダリング機能を活用することで、RX231 に取り込まれた入力信号や FFT 処理結果をモニタすることができます。

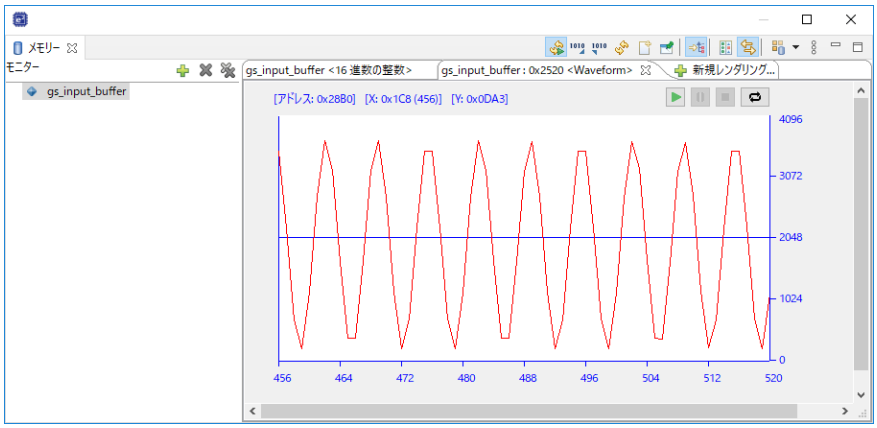


図 31 Waveform レンダリングの表示例（入力バッファに格納されたデータ）

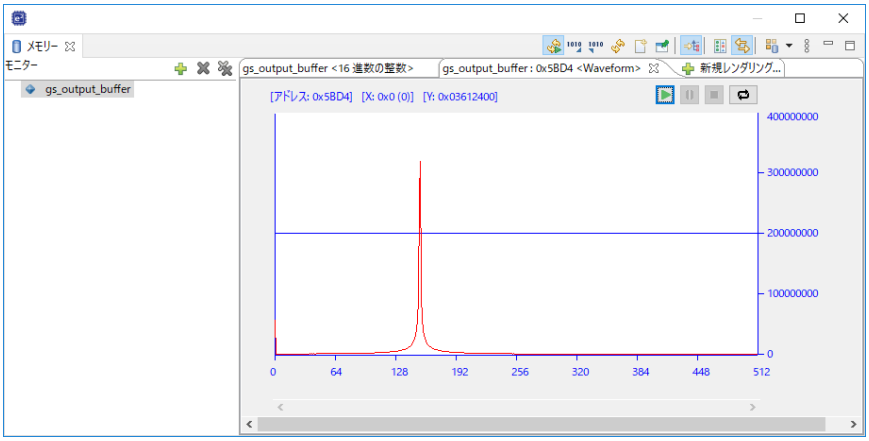


図 32 Waveform レンダリングの表示例（FFT 処理による周波数振幅特性）

RX231 とのデバッグ接続後、「ウィンドウ(W)」→「ビューの表示(V)」から「メモリー」ビューを選択して表示します。「メモリー」ビューが表示されたら、モニタする変数として入力バッファ（gs_input_buffer）を指定します。追加されたら同様に出力バッファ（gs_output_buffer）も指定します。

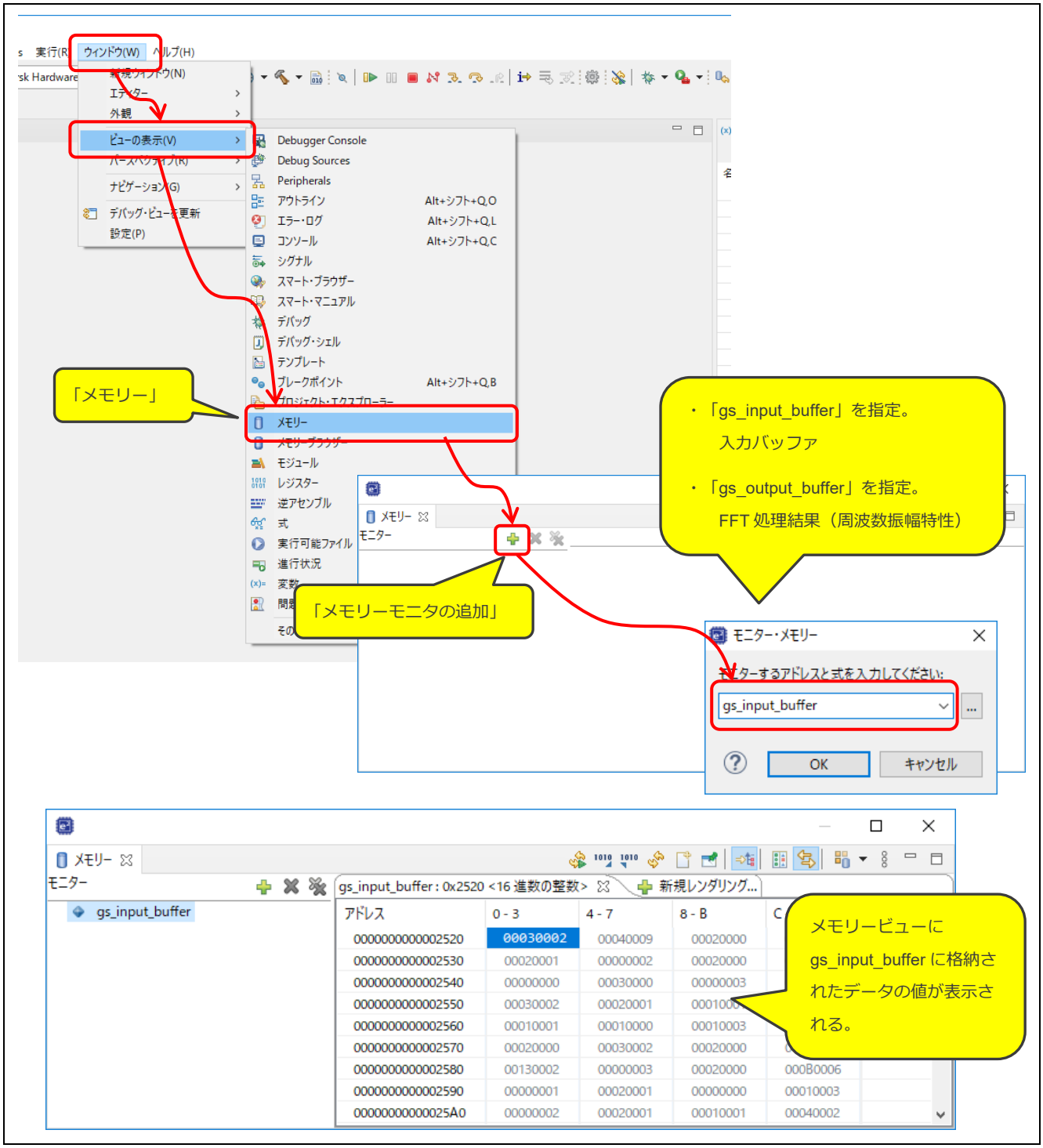


図 33 メモリビューの表示手順

次に、グラフ表示したい変数を選択し、「新規レンダリング」→「Waveform」→「レンダリング」の順にクリックします。Waveform プロパティーが表示されますので、変数に応じて必要な設定を行い、最後に「OK」をクリックするとグラフが表示されます。

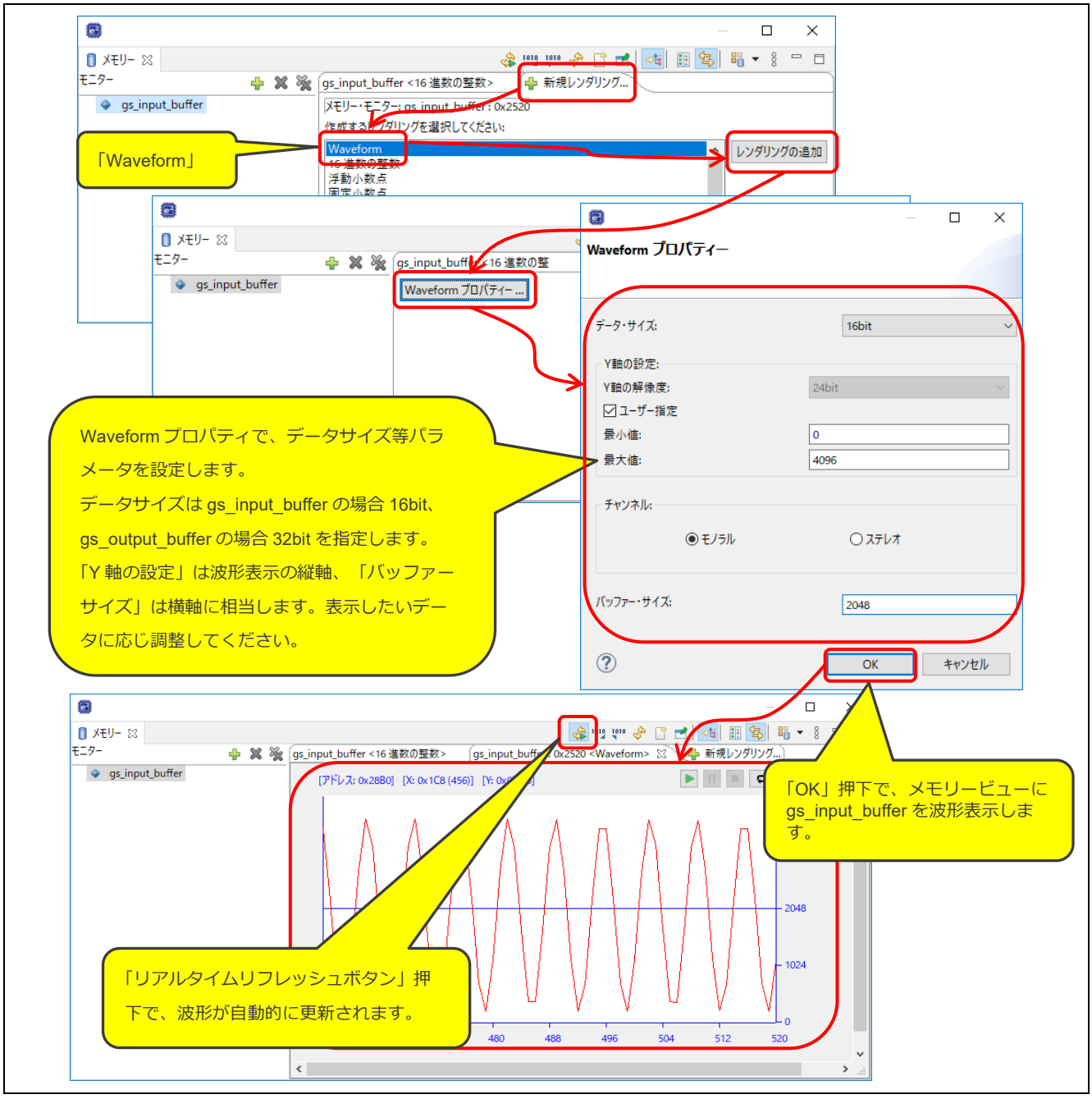


図 34 Waveform レンダリングの表示手順

5.1.2　Visual Expression

e^2 studio の Visual Expression 機能を活用することで、コード上の式をビジュアルにモニタすることができます。ビジュアルに表現された式はビジュアル・エレメントと呼ばれ、「ゲージとメーター」、「コントローラー」および「インジケーター」の 3 種類があります。

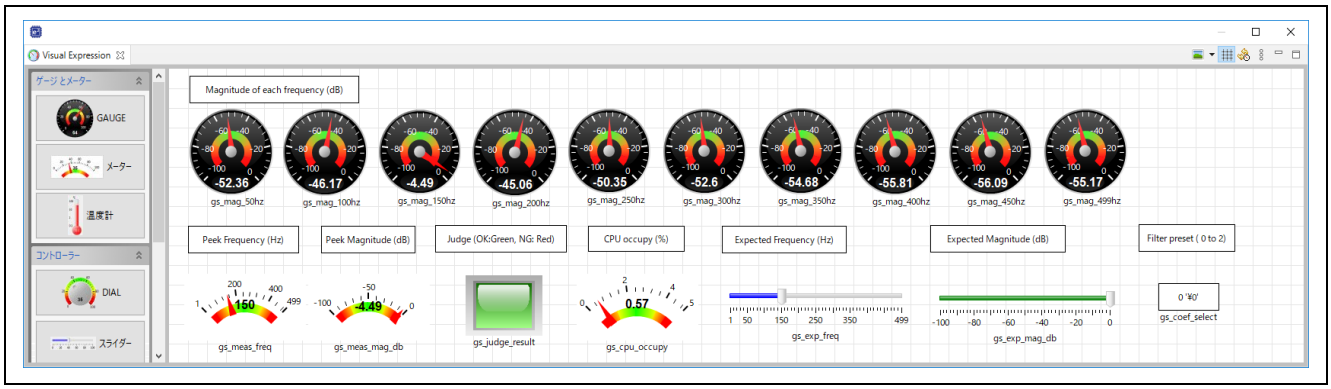


図 35　Visual Expression ビューの表示例

Visual Expression ビューはデバッグ接続中にしか操作できません。先に RX231 とのデバッグ接続を行ってください。

「ウィンドウ(W)」→「ビューの表示(V)」→「その他(O)」をクリックします。「ビューの表示」ウィンドウが表示されますので、「デバッグ」から「Visual Expression」を選択して、「開く(o)」をクリックします。

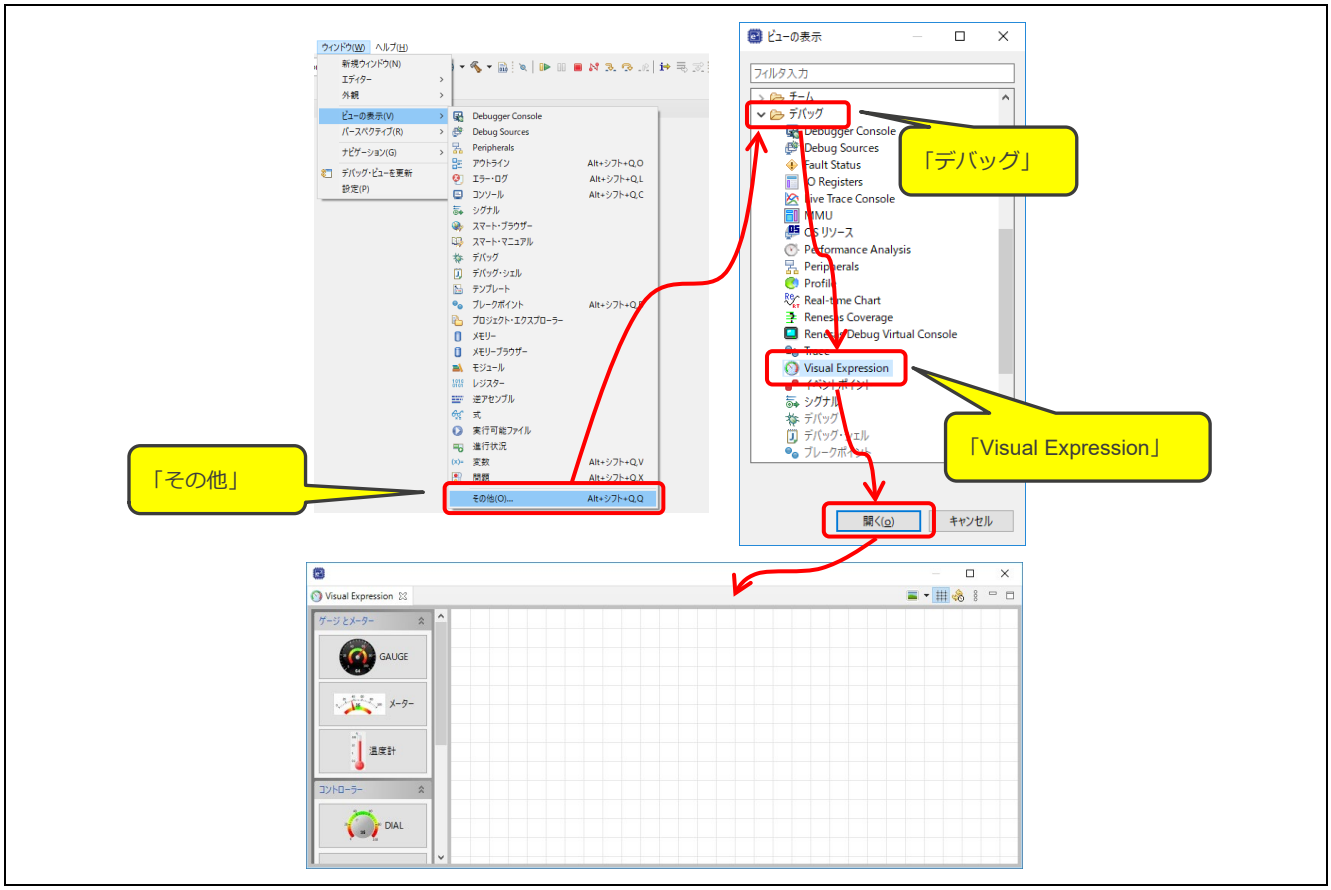


図 36　Visual Expression ビューの表示手順

(a)　ゲージとメーター

例として、ゲージの設定方法を示します。

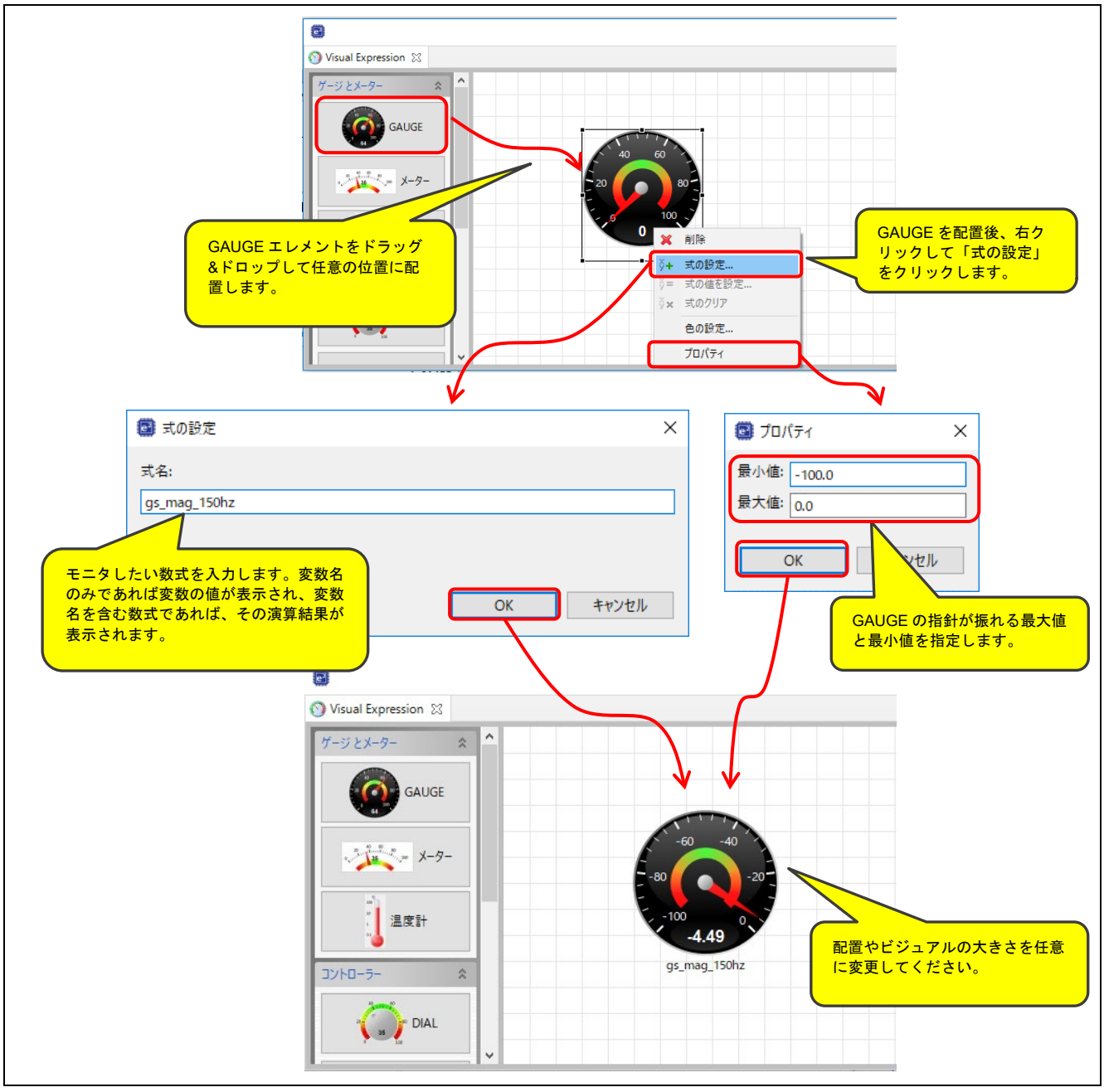


図 37　ゲージの設定手順

(b)　コントローラー

例として、スライダーの設定方法を示します。

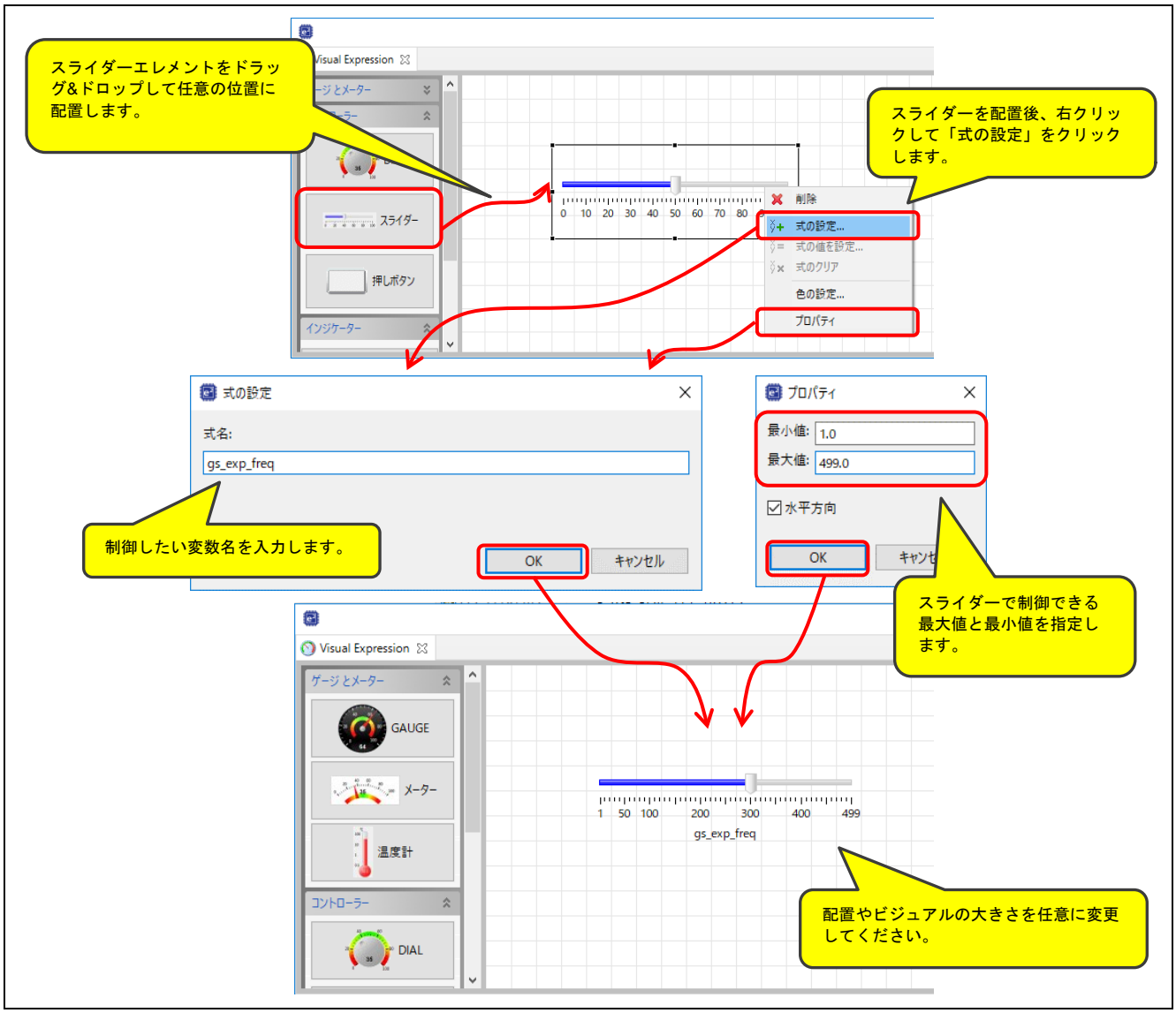


図 38　スライダーの設定手順

(c)　インジケーター

例として、イメージの設定方法を示します。

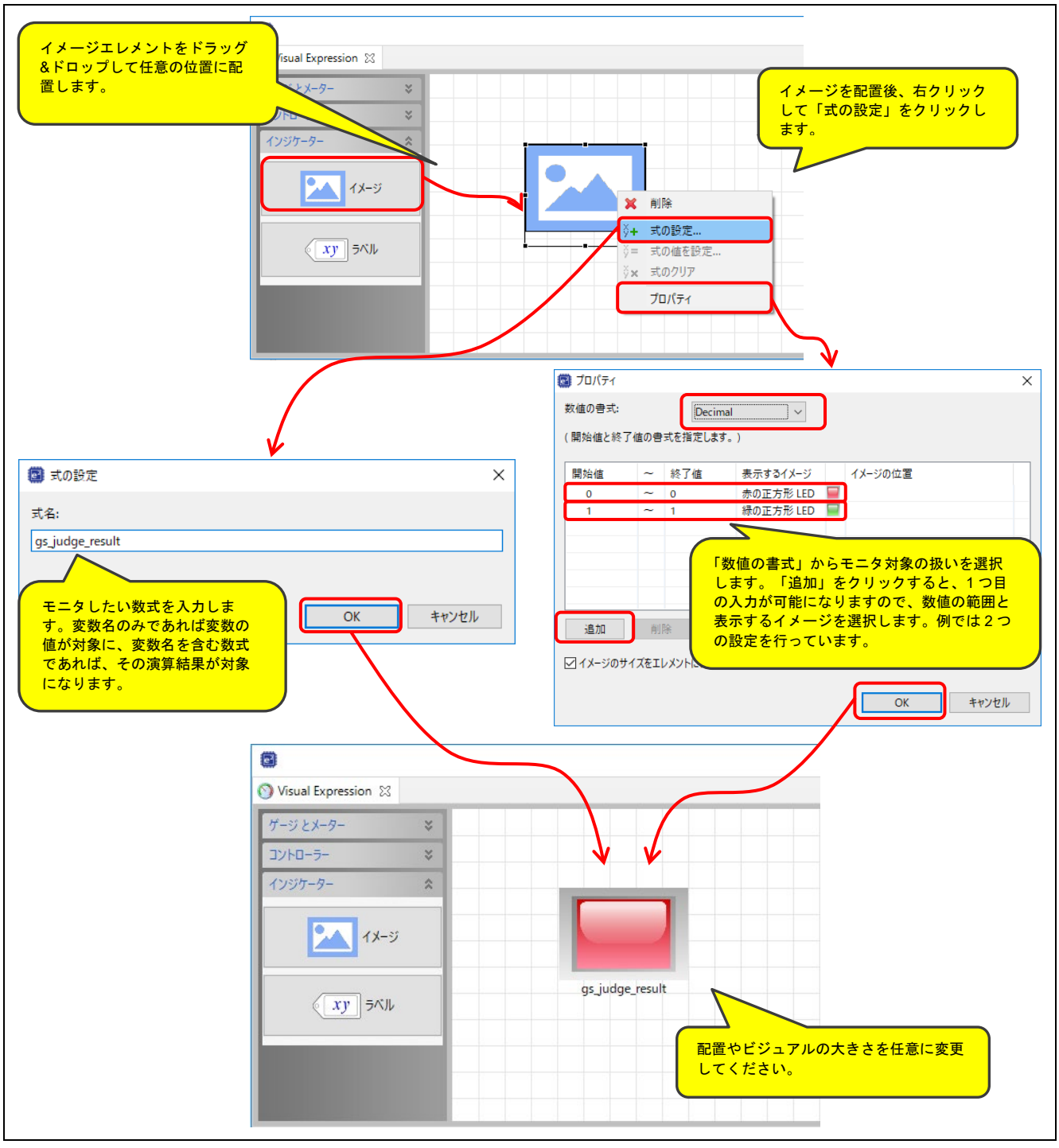


図 39　イメージの設定手順

5.1.3　正弦波生成処理のパラメータ調整

正弦波生成処理は図 40 のコントローラによって周波数と出力レベルを調整することができます。図中“gs_osc_freq_0”、“gs_osc_att_0”で Oscillator 0 の正弦波の周波数、出力レベルの変更を行い、“gs_osc_freq_1”、“gs_osc_att_1”で Oscillator 1 の正弦波の周波数、出力レベルの変更を行うことができます。“gs_osc_att_master”ではミックスした信号の出力レベルの変更を行うことができます。

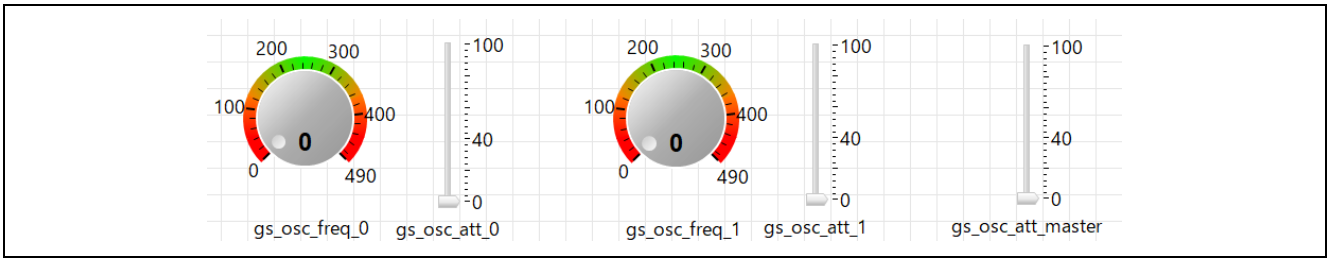


図 40　Visual Expression の正弦波生成処理コントローラ

5.2　使用メモリ

サンプルプログラムのメモリ使用量を表 18、表 19 に示します。

ROM と RAM は表 2 の条件で生成した.map ファイルを元に算出、ユーザスタックと割り込みスタックはサンプルプログラムを実行し実際に使用されたスタックメモリを測定しています。

表 18 サンプルプログラムの使用メモリ（dsp_demo_rx231_rsk）（参考）

項目	測定値 [byte]	説明
ROM	45870	サンプルプログラムの ROM 使用量。
RAM	34195	サンプルプログラムの RAM 使用量。
ユーザスタック	467	サンプルプログラムのスタック使用量。
割り込みスタック	251	

表 19 サンプルプログラムの使用メモリ（dsp_demo_rx231_tb）（参考）

項目	測定値 [byte]	説明
ROM	31512	サンプルプログラムの ROM 使用量。
RAM	27911	サンプルプログラムの RAM 使用量。
ユーザスタック	160	サンプルプログラムのスタック使用量。
割り込みスタック	88	

5.3　ソフトウェアモジュールの設定

サンプルプログラムで使用した FIT モジュールおよび e² studio のスマート・コンフィグレータの設定、DSP ライブラリの設定を表 20～表 30 に示します。スマート・コンフィグレータの設定における各表の項目、設定内容は設定画面の表記で記載しています。各ソフトウェアモジュールの詳細は、8.参考資料に記載のアプリケーションノートを参照してください。

表 20 BSP モジュールの設定

分類	項目	設定、説明
スマート・コンフィグレータ >> コンポーネント >> r_bsp		プロパティは下記の変更以外はデフォルトの設定とする。
	Parameter checking	Disabled
スマート・コンフィグレータ >> クロック		「クロック」タブを下記の設定とし r_bsp_config.h に反映させる。
	VCC 設定	3.3(V)
	(RSK 用サンプルプロジェクトの場合) メインクロック設定	動作：チェックする 発振源：発振子 周波数：8 (MHz) 発振安定時間：8192 サイクル、2048 (μs)
	(Target Board 用サンプルプロジェクトの場合) メインクロック設定	動作：チェックを外す
	PLL 回路設定	分周比： x 1/2 逓倍比： x 13.5
	サブクロック発振器設定	動作：チェックを外す
	(RSK 用サンプルプロジェクトの場合) HOCO クロック設定	動作：チェックを外す
	(Target Board 用サンプルプロジェクトの場合) HOCO クロック設定	動作：チェックする 周波数：54 (MHz)
	LOCO クロック設定	動作：チェックを外す
	(RSK 用サンプルプロジェクトの場合) システムクロック設定	クロックソース：PLL 回路 システムクロック(ICLK)：x1　54 (MHz) 周辺モジュールクロック(PCLKA)：x1　54 (MHz) 周辺モジュールクロック(PCLKB)：x1/2　27 (MHz) 周辺モジュールクロック(PCLKD)：x1　54 (MHz) 外部バスクロック(BCLK)：x1/2　27 (MHz) FlashIF クロック(FCLK)：x1/2　27 (MHz)
	(Target Board 用サンプルプロジェクトの場合) システムクロック設定	クロックソース：HOCO システムクロック(ICLK)：x1　54 (MHz) 周辺モジュールクロック(PCLKA)：x1　54 (MHz) 周辺モジュールクロック(PCLKB)：x1/2　27 (MHz) 周辺モジュールクロック(PCLKD)：x1　54 (MHz) 外部バスクロック(BCLK)：x1/2　27 (MHz) FlashIF クロック(FCLK)：x1/2　27 (MHz)
	IWDT 専用クロック設定	動作：チェックを外す
	USB-PLL 回路設定	分周比： x 1/2 逓倍比： x 12
	外部バスクロック端子(BCLK 端子)設定	動作：チェックを外す

表 21 DMAC モジュールの設定

分類	項目	設定、説明
r_dmaca_rx_config.h		下記の変更以外はデフォルトの設定とする。
	DMACA_CFG_PARAM_CHECKING_ENABLE	BSP のパラメータチェックに従う設定となっており、BSP でパラメータチェックを Disabled に変更（コンパイル時にパラメータチェック処理をコードから省略する）

表 22 SCI モジュールの設定

分類	項目	設定、説明
r_sci_rx_config.h		下記の変更以外はデフォルトの設定とする。
	SCI_CFG_SYNC_INCLUDED	1 に変更（クロック同期式モードを有効）
	SCI_CFG_ASYNC_INCLUDE	0 に変更（調歩同期式モードを無効）
	SCI_CFG_CH1_INCLUDED	0 に変更（ch1 は未使用）
	SCI_CFG_CH8_INCLUDED	1 に変更（ch8 を使用する）
	SCI_CFG_PARAM_CHECKING_ENABLE	BSP のパラメータチェックに従う設定となっており、BSP でパラメータチェックを Disabled に変更（コンパイル時にパラメータチェック処理をコードから省略する）

表 23 BYTEQ モジュールの設定

分類	項目	設定、説明
スマート・コンフィグレータ >> コンポーネント >> r_byteq		プロパティは下記の変更以外はデフォルトの設定とする。
	Parameter check	BSP のパラメータチェックに従う設定となっており、BSP でパラメータチェックを Disabled に変更（コンパイル時にパラメータチェック処理をコードから省略する）
	Number of static queue control blocks	2 に変更

表 24 スマート・コンフィグレータの設定（S12AD）

分類	項目	設定
スマート・コンフィグレータ>> コンポーネント >> シングルスキャンモード S12AD（Config_S12AD0）		下記の設定とし、コードを生成する。
	アナログ入力モード設定	ダブルトリガモード：チェックを外す
	アナログ入力チャネル設定	AN001 のみチェックする
	変換開始トリガ設定	開始トリガソース：ELC からのトリガ
	割り込み設定	AD 変換終了割り込みを許可（S12ADI0）にチェックする。 優先順位：レベル 0（割り込み禁止）
	AD 変換値を加算/平均	AN001 のチェックを外す
	A/D 変換動作選択ビット	高速変換動作
	高電位側基準電圧選択ビット	VREFH0
	低電位側基準電圧選択ビット	VREFL0
	自己診断設定	モード：未使用
	断線検出アシスト設定	チャージ設定：未使用
	データレジスタ設定	データレジスタフォーマット：右詰にする 自動クリアイネーブル：自動クリアを禁止 加算/平均モード選択：加算モード 加算回数：1 回変換
	データ格納バッファ設定	禁止
	ウィンドウ機能設定	禁止

	ウィンドウ A/B 動作設定	比較ウィンドウ A 有効：チェックを外す 比較ウィンドウ B 有効：チェックを外す
	入力サンプリング時間設定	AN001：3.667 (μs)
	イベントリンクコントロールビット設定	ELC 用スキャン終了イベント：すべてのスキャン終了時にイベント発生

表 25 スマート・コンフィグレータの設定（CMT0）

分類	項目	設定、説明
スマート・コンフィグレータ>> コンポーネント >> コンペアマッチタイマ （Config_CMT0）		下記の設定とし、コードを生成する。
	クロック設定	PCLK/8
	コンペアマッチ設定	インターバル時間：10 ms (実際の値:10) レジスタ（CMCOR）：33749 コンペアマッチ割り込みを許可（CMI0）：チェックする 優先順位：レベル 13
コード生成後の追加要素		
Config_CMT0_user.c	インクルード追加	r_lcd_driver_if.h のインクルード
	処理追加 （r_Config_CMT0_cmi0_interrupt）	R_LCD_Driver_PeriodicCallback 関数の呼び出し

表 26 スマート・コンフィグレータの設定（CMT1）

分類	項目	設定、説明
スマート・コンフィグレータ>> コンポーネント >> コンペアマッチタイマ （Config_CMT1）		下記の設定とし、コードを生成する。
	クロック設定	PCLK/8
	コンペアマッチ設定	インターバル時間：1000 μs (実際の値:1000) レジスタ（CMCOR）：3374 コンペアマッチ割り込みを許可（CMI1）：チェックを外す

表 27 スマート・コンフィグレータの設定（ELC）

分類	項目	設定、説明
スマート・コンフィグレータ>> コンポーネント >> イベントリンクコントローラ（Config_ELC）		下記の設定とし、コードを生成する。
	ソース	設定：Config_CMT1 リソース：CMT1 イベント：CMT1・コンペアマッチ 1
	送先	設定：Config_S12AD0 リソース：S12AD0 動作：A/D 変換開始

表 28 スマート・コンフィグレータの設定（TPU1、TPU2、TPU4、TPU5）

分類	項目	設定、説明
スマート・コンフィグレータ>> コンポーネント >> ノーマルモード（Config_TPU1、Config_TPU2、Config_TPU4、Config_TPU5）		下記の設定とし、コードを生成する。
	同期動作設定	チェックを外す
	TCNT1 カウンタ設定	カウンタクリア要因：カウンタクリアなし カウンタクロックの選択：TPU2 のオーバフロー/アンダフロー
	TCNT2 カウンタ設定	カウンタクリア要因：カウンタクリアなし カウンタクロックの選択：PCLK
	TCNT4 カウンタ設定	カウンタクリア要因：カウンタクリアなし カウンタクロックの選択：TPU5 のオーバフロー/アンダフロー
	TCNT5 カウンタ設定	カウンタクリア要因：カウンタクリアなし カウンタクロックの選択：PCLK
	ジェネラルレジスタの設定	TGRAx：アウトプットコンペアレジスタ 100 カウント TGRBx：アウトプットコンペアレジスタ 100 カウント
	入出力端子の設定	TIOCAx 端子：端子出力は無効 TIOCBx 端子：端子出力は無効
	A/D 変換開始トリガ設定	チェックを外す
	割り込み設定	全てチェックを外す

表 29 スマート・コンフィグレータの設定（ICU）

分類	項目	設定、説明
スマート・コンフィグレータ>> コンポーネント >> 割り込みコントローラ（Config_ICU）		下記の設定とし、コードを生成する。
	(RSK 用サンプルプロジェクトの場合) IRQ1	IRQ1：チェックする 検出タイプ：立ち下がりエッジ デジタルフィルタ：PCLK/64 優先順位：レベル 10
	(Target Board 用サンプルプロジェクトの場合) IRQ4	IRQ4：チェックする 検出タイプ：立ち下がりエッジ デジタルフィルタ：PCLK/64 優先順位：レベル 10

表 30 DSP ライブラリの設定

分類	項目	設定、説明
使用するファイル		DSP ライブラリの dsplib-rxv2 フォルダにある以下のファイルを使用する。
	.lib ファイル	RX_DSP_FPU_LE.lib
	.h ファイル	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. 開発環境の入手

6.1 e² studio の入手方法

以下の URL にアクセスし、e² studio をダウンロードしてください。

https://www.renesas.com/ja-jp/products/software-tools/tools/ide/e2studio.html

なお、本ドキュメントは e² studio 2022-04 以降を使用することを前提としています。2022-04 よりも古い Ver.を使用した場合、e² studio の一部機能を使用できない可能性があります。ダウンロードする場合、ホームページに掲載されている最新 Ver.の e² studio を入手してください。

6.2 コンパイラパッケージの入手方法

以下の URL にアクセスして、RX ファミリ用 C/C++コンパイラパッケージをダウンロードしてください。

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

7. 補足

7.1 無償評価版の「RX ファミリ用 C/C++コンパイラパッケージ」を利用する場合の注意事項

無償評価版の「RX ファミリ用 C/C++コンパイラパッケージ」には、使用期限と使用制限があります。使用期限が過ぎた場合、リンクサイズが 128K バイト以内に制限されるためロードモジュールが正しく生成されなくなる場合があります。

詳しくは、ルネサスのホームページにある、無償版ソフトウェアツールのページを参照してください。

https://www.renesas.com/ja-jp/products/software-tools/evaluation-software-tools.html

8. 参考資料

- RX ファミリ ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685)
- RX ファミリ DMA コントローラ DMACA 制御モジュール Firmware Integration Technology (R01AN2063)
- RX ファミリ SCI モジュール Firmware Integration Technology (R01AN1815)e² studio コード生成ツール ユーザーズマニュアル RX API リファレンス編 (R20UT2864)
- Renesas e² studio スマート・コンフィグレータ ユーザーガイド(R20AN0451)
- RX DSP ライブラリ Version 5.0 (CC-RX) (R01AN4359JJ0100)
- RX230 グループ、RX231 グループ ユーザーズマニュアル ハードウェア編 (R01UH0496)
- Renesas Starter Kit for RX231 ユーザーズマニュアル (R20UT3027)
- Renesas Starter Kit for RX231 CPU ボード回路図 (R20UT3026)
- Target Board for RX231 ユーザーズマニュアル (R20UT4168)
- Target Board for RX231 CPU ボード回路図 (R20UT4165)

最新版をルネサス エレクトロニクスホームページから入手してください。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Nov.20.18	-	初版発行
1.50	Apr.26.21	1	動作確認ボード Target Board for RX231 を追加 サンプルログラムに関する注記を追加
		4	1. システムの概要 DSP ライブラリのバージョンを 5.0 に更新 図を更新
		6、7	1.1 アプリケーションノートの構成 Target Board for RX231 用のサンプルプログラムについての説明を追加 図を更新
		8、9	1.2 サンプルプログラムの構成 Target Board for RX231 用のサンプルプログラムについての説明、および ICU(IRQ)に関する説明を追加 図表を更新
		10、11	1.3 動作環境 動作確認環境の更新 Target Board for RX231 用の確認環境を追加 表を更新
		12	2.1 ワークスペースの起動 図 8 追加
		13	2.2 機器の接続 図を更新
		14	2.3.1 サンプルプログラムの実行方法 追加
		16	2.3.2 LCD の表示内容とフィルタ特性の変更方法 Target Board for RX231 用のサンプルプログラムについての説明を追加 2.3.3 e^2 studio の機能を使ったシステム動作のモニタ VisualExpression に関する説明を追加 図を更新
		18	2.4 ユーザ変更可能な設定 EXPECTED_MAGNITUDE の記載内容を更新 表に注記を追加
		26	3.4.1 信号処理フロー 追加
		28	3.4.4 FFT 処理 DSP ライブラリのバージョンを 5.0 へ更新と、それに伴う説明の追加
		29	3.4.5 FFT 処理結果の判定 判定処理の方法を更新 3.4.6 LCD モジュールの表示内容更新処理 LCD への表示内容を更新
		30	3.4.7 IIR フィルタ特性の変更処理 変更処理の方法を更新
		33	4. 注意事項 追加

		34	5.1 e^2 studio による信号処理のモニタ 追加
		41	5.2 使用メモリ RSK 用のサンプルプログラムの情報を更新 Target Board 用のサンプルプログラムの使用メモリを追加
		42	5.3 ソフトウェアモジュールの設定 動作確認環境の更新にともない、情報を更新 Target Board for RX231 用のサンプルプログラムについての説明を追加
		46	6. 開発環境の入手 記載内容の更新 8.参考資料 Target Board for RX231 用の参考資料を追加
		プログラム	・動作環境（e^2 studio, コンパイラ、FIT モジュール、DSP ライブラリなど）の更新 ・IIR フィルタ係数変更で使用するスイッチ入力の検出方法を、IRQ 割り込みを使用する方法に変更 (main.c、Config_ICU_user.c) ・FFT 処理後の出力振幅値をデシベルに変換して LCD、VisualExpression に表示するよう変更 (main.c) ・Visual Expression でモニタできるよういくつかのローカル変数をグローバル変数化 (main.c) ・DSP 処理の仕様を以下のように変更 - DSP ライブラリの更新にともない FFT 処理結果の周波数振幅特性の出力フォーマットを Q1.31 から Q2.30 に変更 - FFT 処理結果の周波数振幅特性の要素[0]の導出の方法を $R_{N/2}$ の絶対値を取得して Q2.30 フォーマットにするように変更 (r_dsp_iirbiquad.c、r_dsp_real_fft.c) ・TargetBoard 用のサンプルプロジェクト追加 (dsp_demo_rx231_tb)
1.60	Mar.28.22	4	1. システムの概要 図を更新 正弦波生成処理の説明を追加
		10、11	1.3 動作環境 動作確認環境の更新 表を更新
		17	2.3.3 e^2 studio の機能を使ったシステム動作のモニタ 図を更新
		18	2.4 ユーザ変更可能な設定 表を更新
		22	3.3 処理フロー 図を更新 正弦波生成処理の説明を追加
		26	3.4.1 信号処理フロー 図表を更新
		28	3.4.3 正弦波生成処理を追加

		33 35 35 43 プログラム	3.5 ファイル構成 表を更新 表 17 r_signal_gen.c ファイルの関数一覧を追加 4.2 エイリアシングを追加 4.3 オシレータ出力信号の経時変化を追加 5.1.3 正弦波生成処理のパラメータ調整を追加 ・動作環境（e^2 studio, コンパイラ、FIT モジュール、DSP ライブラリなど）の更新 ・Visual Expression で変更できるよういくつかのグローバル変数を定義 (main.c) ・入力データの切り替えに正弦波生成処理を追加 (main.h) ・入力データを正弦波生成処理にした場合の処理を追加 (main.c) ・正弦波生成処理に関するファイルを追加 (r_signal_gen.c、r_signal_gen.h、r_signal_gen_att_coef.c、r_signal_gen_freq_coef.c)

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。
3. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。
5. クロックについて
   リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。
8. 製品間の相違について
   型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
   標準水準：　コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等
   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1　2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。

© 2022 Renesas Electronics Corporation. All rights reserved.