

本資料は英語版を翻訳した参考資料です。内容に相違がある場合には英語版を優先します。資料によっては英語版のバージョンが更新され、内容が変わっている場合があります。日本語版は、参考用としてご使用のうえ、最新および正式な内容については英語版のドキュメントを参照ください。

要旨

本シンプルトーンジェネレータ例は正弦波 1 サイクルに必要なデータを動的に作成します。そのデータをボード上の S124 プロセッサの D/A コンバータで繰り返し動作させることで 1 つの音声トーンを作成し、イヤホンジャックに接続されている増幅器を動作させます。正弦波の周波数は、データ出力に使われるサンプリングレートをプログラムで変更することにより変化します。

DK-S124 には PMOD LCD ディスプレイが付属で含まれています (図 1.1 参照)、本プログラムでは、制御用のシンプルな 2 個のボタン式インターフェースとオペレータフィードバック用のオンボード LED を使用しています。PMOD LCD は本アプリケーション例では使用しません。

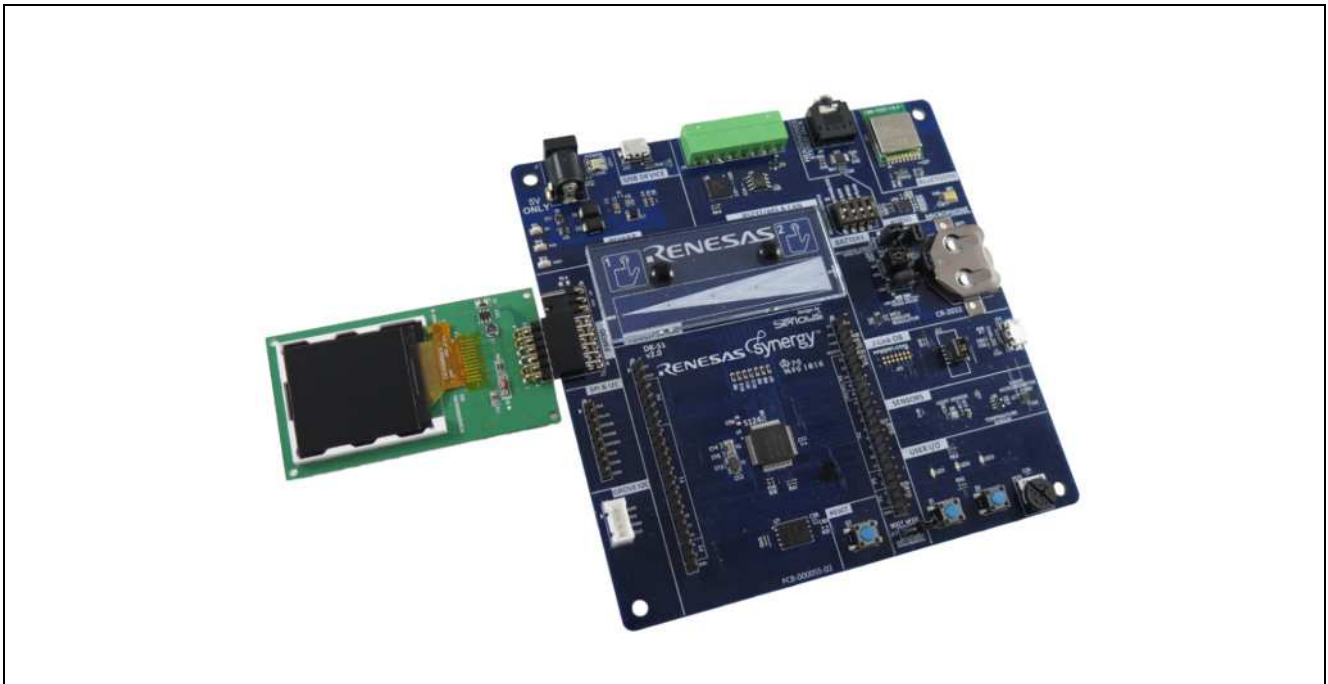


図 1.1 DK-S124 と PMOD LCD (接続時)

前提条件

本アプリケーションノートは、Renesas e²studio ISDE または IAR IDE の使用経験のあるユーザを対象としています。本プログラムは、このいずれか一方のツールチェーンを使用してコンパイル、実行が可能です。

Synergy Software Package (SSP) の使用経験も必要になります。本アプリケーションノートの手順に入る前に、ご使用のボードのクイックスタートガイド記載の手順に従って、Blinky プロジェクトのビルドおよび実行をすることをお勧めします。そうすることで、SSP アプリケーションのビルドおよび実行に慣れ、さらにご使用のボードへのデバッグ接続が正しく機能することが確認できます。

必要動作環境

サンプルアプリケーションは、Renesas Synergy S124 デバイスを対象としています。アプリケーションプログラムのビルド、実行のためには以下が必要です。

- Renesas Synergy DK-S124 ボード v3.0 以降
- Microsoft® Windows® 7 以降搭載かつ以下の Renesas ソフトウェアをインストール済の PC
 - Synergy e²studio 5.3.1.002 以降
 - Synergy Software Package (SSP) v1.2.0 以降
 - IAR for Synergy 7.71.1.11989
 - Synergy Standalone Configurator 5.3.1.002.

必要な Renesas ソフトウェアは、Renesas Synergy Gallery (<https://synergygallery.renesas.com>)よりダウンロードできます。

本ドキュメントの目的

本アプリケーションノートは、ユーザが S124 プロセッサの機能に慣れることを目的としています。

所要時間

アプリケーションサンプルのインストール、ビルドおよび実行は約 30 分で完了します。この所要時間で行われる手順は以下のとおりです。

1. DK-S124 ボードを設定しトーンジェネレータアプリケーション例を実行する。
2. プロジェクトをインポート、ビルド、デバッグする。
3. イヤホンジャックまたは外部スピーカーより音声トーンを再生する。
4. モメンタリ（押し時のみ有効）押しボタン S1 を使用して、設定されている音声トーンの一通り再生する。
5. モメンタリ（押し時のみ有効）押しボタン S2 を使用して音声出力を有効 / 無効にする。
6. POT1 を使い、音量を変化させる。

目次

1. 概要およびデザイン時の考慮事項	4
2. Synergy プラットフォームの性能.....	4
3. デザイン概要	6
3.1 スイッチ入力のデバウンス	6
3.2 アクセラレータの初期化.....	8
3.3 スレッド	9
3.4 オーディオスレッドロジック.....	10
3.5 スレッドおよびモジュール	11
3.6 周期的 ADC フレームワークの使用.....	12
3.7 ADC ユニットの設定.....	13
3.7.1 周期的 ADC フレームワークのコールバック	15
3.8 ソースコードレイアウト.....	15
4. デザインデモンストレーション.....	16
4.1 トーンジェネレータサンプル用の DK-S124 設定.....	16
4.2 テストジェネレータサンプルプログラムの扱い方	17
5. プロジェクトのインポートとビルド.....	17
6. e ² studio の技法 - 変数のテスト	17
6.1 グラフィック表示.....	19
7. 次のステップ.....	21

1. 概要およびデザイン時の考慮事項

本アプリケーションプログラムの目的は、澄んだ音声トーンを生成し、イヤホンジャックへ接続されている増幅回路の忠実性をテストすることです。本アプリケーションは、正弦波 1 サイクルに必要なデータポイントを作成し、データをバッファに保存します。このデータは、イヤホンジャックを駆動する増幅器に接続されている 12 ビット D/A コンバータを繰り返し動作させるのに使用されます。

本アプリケーションでは、この正弦波データを異なる周波数で再生させる必要があります。モメンタリ（押し時のみ有効）押しボタン S1 が押される度に、タイマの周期を変更するタイマ API コールの 1 つによって実現できます。

ボード上のポテンシオメータ（POT1）を使い、アプリケーションプログラムによって出力されるトーンの音量を変える必要もあります。音量を変えるには、アプリケーションで POT1 にあるアナログ電圧を周期的に読み、その電圧に従って出力音量を調整する必要があります。これを達成するためには、いくつかの方法があります。

1. メインスレッドループを実行する度に、手動でデータのシングルスキャンを行います。
2. 周期的 ADC フレームワークとコールバック関数を使い、ADC スキャン専用のタイマモジュールによって設定された速度でアナログチャンネルをスキャンします。

このアプリケーションでは、2 番目の手法を使います。他の、サーモスタットのようなアプリケーションでは、1 番目の手法が使われています。

2. Synergy プラットフォームの性能

本アプリケーションでは、イヤホン増幅器（U9）を動作させるオンボード D/A コンバータへのデータ書き込みを主に扱います。汎用タイマモジュールを使って、制御された速度で再生が行なわれます。この制御された速度で作成されるトーンの周波数を決定します。SSP は、このようなタイプのハードウェア周辺機器とインターフェースを設定するため、高レベルのフレームワークモジュールだけでなく低レベルのドライバモジュールも提供します。ユーザは低レベルドライバコードを書く必要がありません。

図 2.1 に、トーンジェネレータアプリケーションサンプルに使われる S1 ハードウェア周辺機器を一覧にまとめました。



図 2.1 トーンジェネレータサンプルが使用する S1 周辺機器

表 2.1 は、本サンプルで使用する周辺機器を、DK-S124 上の多様な外部コンポーネントに接続するために使われるプロセッサ端子接続を定義しています。さらに詳細な情報については、DK-S124 回路図を参照ください。

表 2.1 トーンジェネレータが使用する周辺接続におけるポート一覧

回路構成要素	説明	S1 ポート接続	端子機能
S1	モメンタリスイッチ	ポート 2 端子 6	I/O ポート端子
S2	モメンタリスイッチ	ポート 0 端子 4	I/O ポート端子
POT1	10K ポテンシオメータ	ポート 0 端子 12	アナログチャンネル 7
U13	BMA250 3軸加速度計	ポート 2 端子 6	IRQ
U13	BMA250 3軸加速度計	ポート 4 端子 1	SDA0 (I2C クロック)
U13	BMA250 3軸加速度計	ポート 0 端子 0	SCL0 (I2C データ)
U11	音声パワー増幅器	ポート 0 端子 14	DAC 出力

3. デザイン概要

DK-S124 ボードは、モメンタリ（押し時のみ有効）スイッチ（S1、S2）2つと、オペレータの入力として使用できるポテンシオメータ(POT1)を備えています。PMOD LCD ディスプレイが使用されない代わりに、オペレータフィードバック用として DK-S124 には 3 つの LED を備わっています。ボードにはさらに、標準的なイヤホン用標準 3.5 mm コネクタに対応するイヤホンジャックが含まれています。図 3.1 は、S124 プロセッサへの入力および出力の接続状態を簡易ブロック図で表しています。

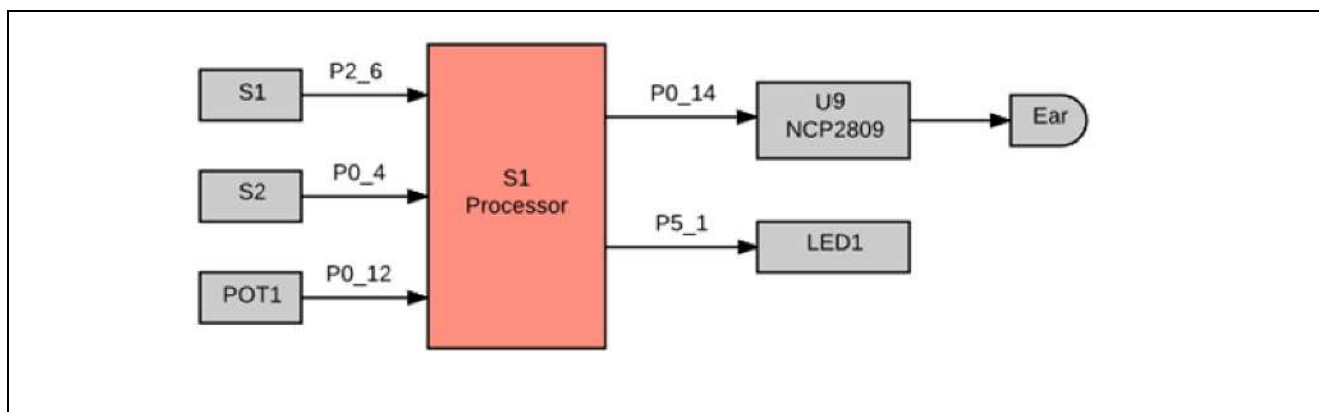


図 3.1 ハードウェア接続の簡易ブロック図

本トーンジェネレータアプリケーション例では使える動作モードは 1 つです。DK-S124 上の S1 および S2 の 2 個の押しボタンによりアプリケーションは制御されます。ボード上の 3 個の LED のうち、1 個のみがオペレータフィードバックに使われます。オペレータインターフェースについての詳細は、第 4 章、デザインデモンストレーションを参照してください。

3.1 スイッチ入力のデバウンス

スイッチ入力スイッチバウンスは常に問題になります。ほとんどの機械式スイッチでコンタクトバウンスとして知られる現象があります。これは、スイッチが押された後、スイッチが一つの状態から次の状態に変わる間、ON/OFF を繰り返してしまう時間のことです。これによる誤トリガを防ぐには、このような正しくない遷移をハードウェアまたはファームウェアが無視する必要があります。標準的なファームウェアの技術として、一定の間隔、スイッチの状態をサンプリングして、有効な状態と判断するまで、あらかじめ設定したサンプル数の間、スイッチ状態が一定になるまで待つ方法があります。

DK-S124 上の、両方の押しボタンはハードウェア端子に接続され、割り込み機能を有します。アプリケーションの中には Simple Record ように、これら端子に対し g_ioport ドライバを使用してポーリングするものもあります。通常このドライバは、ユーザがスレッドを定義する前に HAL/Common 領域で自動的にユーザアプリケーションに追加されます。本アプリケーションでは、ボタンが同時に押されたのを検知し、さらにスイッチバウンスを除去する Synergy 外部 IRQ ドライバ(x_icu)の使い方を説明します。

図 3.2 では、S2 でのコンタクトバウンスを調べるための、オシロスコープトレースの遅延掃引を示します。オシロスコープはシングルキャプチャモードに設定され、立ち下がりエッジでトリガされます。トップトレースは 200ms 刻みで、複数のボタンの押下がキャプチャされます。ボトムトレースでは、遅延掃引（ズーム）モードが使われ、コンタクトバウンスが存在する場所の 1 つのエッジを拡大します。図 3.2 のとおり、スイッチは 1 回のみバウンスし、またそのバウンスは短い期間になります。DK-S124 で使用されているポー

ドに実装されたモメンタリ（押し時のみ有効）スイッチのバウンスは極めて小さく、図 3.2 のようなバウンスを捉えるのに多くのサンプルをとる必要があります。機械式スイッチは、バウンスがさらに良くないものが多くあり、より長いデバウンス期間が必要です。

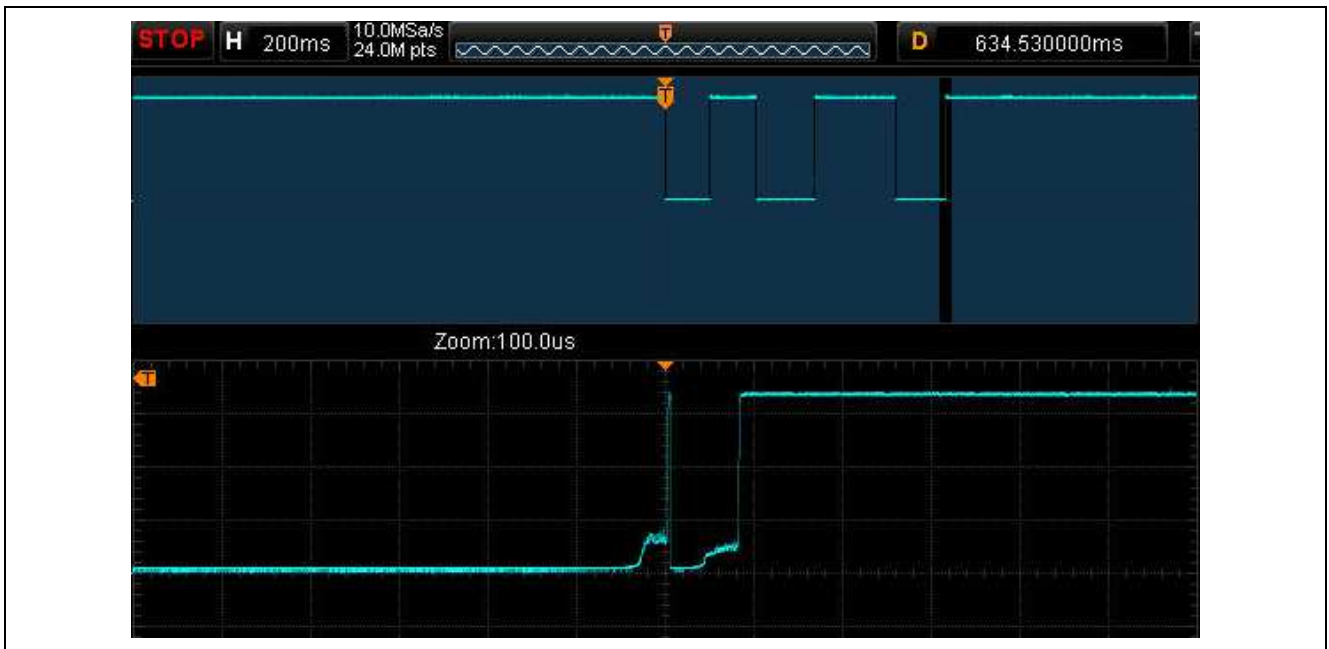


図 3.2 Contact bounce on S2

S124 プロセッサ上の外部割り込み端子は、デジタルフィルタリング機能を備えており、ファームウェアでデバウンス入力をする必要がありません。図 3.3 でハイライトした `r_icu` スタックの 2 つのプロパティを使って、このフィルタリングを設定することができます。デジタルフィルタは入力信号をサンプリングして、3 サンプルサイクルより短いパルス幅の信号を無視します。サンプルのサイクルは、[Properties] タブに設定した値によって分周された PCLKB の現在の設定に基づいています。この場合は 64 です。

Property	Value
Parameter Checking	Disabled
Module g_sf_irq_s1 External IRQ Driver on r_icu	
Name	g_sf_irq_s1
Channel	0
Trigger	Falling
Digital Filtering	Disabled
Digital Filtering Sample Clock (Only valid when Digital Filtering)	PCLK / 64
Interrupt enabled after initialization	True
Callback	g_irq_sw1_callback
Interrupt Priority	Priority 2

図 3.3 r_icu ドライバのデジタルフィルタリングのプロパティ

PCLKB の値は、Synergy コンフィグレータの[Clocks]タブを使って設定することができます。図 3.4 に示すとおり、トーンジェネレータアプリケーション用の PCLKB が高速発振回路(HOCO)の 2 分周で設定され、周波数は 24MHz となります。この周波数 24MHz を 64 で分周するとデジタルフィルタ周波数 375kHz となり、サンプリング期間は 1/375,000 または 2.66 μ s となります。このサンプリングの 3 期間は 8 μ s 弱となります。これは、8 μ s に満たないパルス幅はデジタルフィルタにより無視されることを意味します。

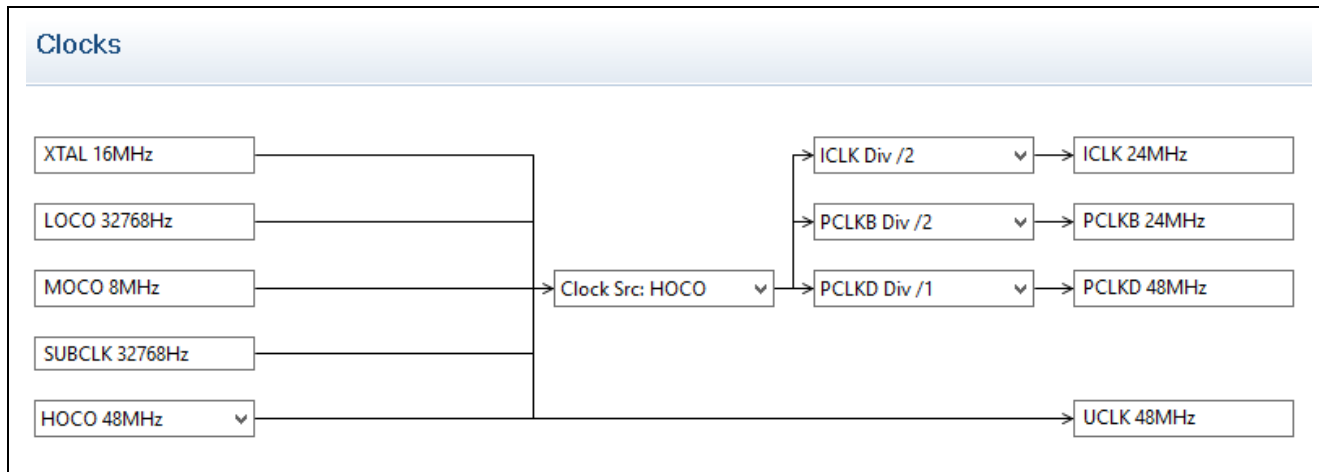


図 3.4 トーンジェネレータサンプルの Clocks タブに表示されたクロックツリー

前述したとおり、DK-S124 ボード上のスイッチのバウンスは最小限の大きさのため、デジタルフィルタリングの 10 μ s は、トーンジェネレータアプリケーションではよく機能します。しかしながら、バウンス時間のより長いスイッチも存在します。その場合、誤トリガを防ぐためにはデジタルフィルタ時間を長くする必要があります。

デジタルフィルタプロパティで(図 3.3)設定可能な分周比は最大で 64 なので、[Clocks]タブで PCLKB の分周率を上げる必要があります。その結果、PCLKB の周波数が下がります。例えば、PCLKB4 分周に分周比を上げ、実効デバウンス時間をデジタルフィルタリングで 2 倍にするか、PCLKB8 分周に分周比を上げて 4 倍 (デバウンス時間計最大で 32 μ s) することができます。

注： PCLKB 分周を下げてデジタルフィルタリング時間を長くする場合、PCLKB が S124 プロセッサの他の周辺回路(SCI、IIC、SPI 等)にも使用されることを注意してください。SSP ユーザマニュアルの第 8 章では、S124 クロックを説明し、PCLKB によって動作する周辺回路の一覧があります。それらの周辺回路では PCLKB 周波数が遅くなるとスループットの減少がみられることがあり、アプリケーションに悪影響が及ぼす可能性があります。その場合は PCLKB を高い周波数にするする必要があります。ファームウェア・デバウンスプログラムのデジタルフィルタリングを変更するか増やす必要があります。

3.2 アクセラレータの初期化

DK-S124 開発ボード v2.0 では、加速度センサ IRQ ラインは S1(図 3.5)と共通です。

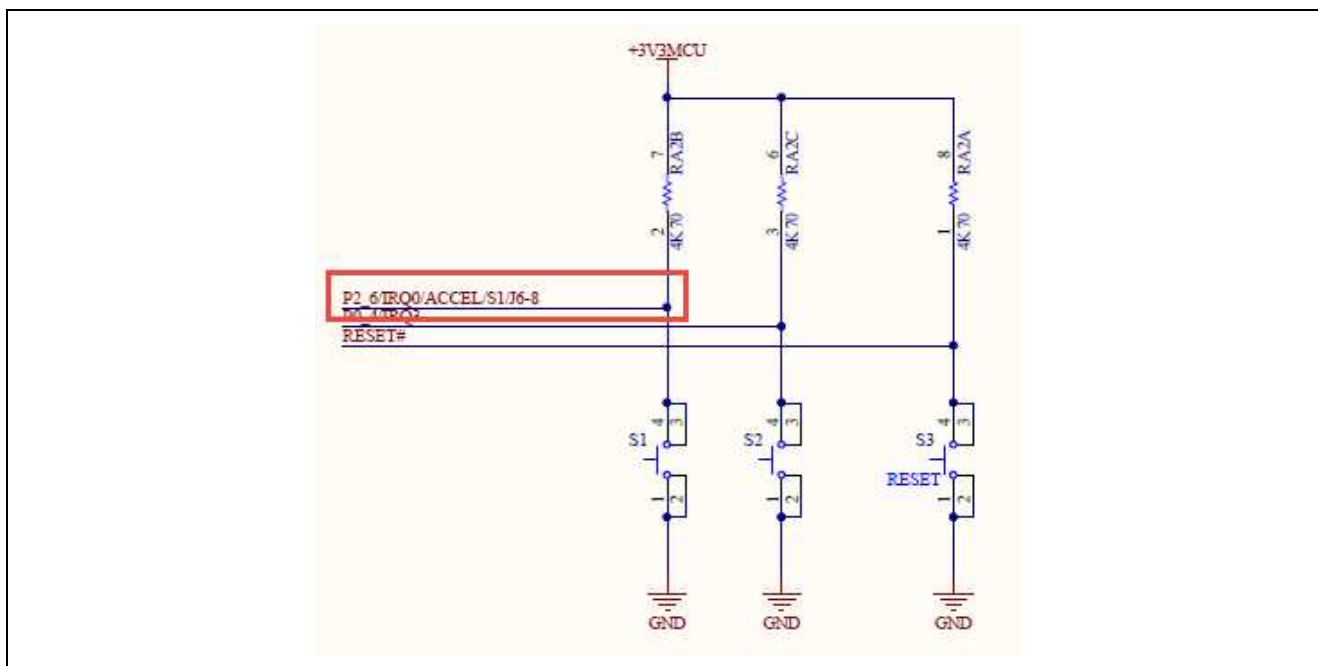


図 3.5 加速度センサの IRQ ラインと共通の S1 入力

加速度センサ IRQ 端子のデフォルト設定はプッシュ/プル設定でかつアクティブハイ IRQ 状態です。これは、S1 に接続されているラインが常にプルダウンされているということの意味し、S1 の状態をリードするためにこのラインを使用することを妨げます。S1 を使用するには、加速度センサの割り込み設定を変更する必要があります。変更するには、I²C フレームワークをオーディオスレッドに追加し、オーディオスレッド初期化シーケンス中に、I²C コマンドを加速度センサへ複数発行します。

3.3 スレッド

表 3.1 のとおり、単純化のため、本アプリケーションは 1 つのスレッドで設計されています。

表 3.1 アプリケーションスレッド名と機能

スレッド名	スレッド機能
オーディオスレッド	トーンジェネレータに関するすべての論理の処理

図 3.6 はトーンジェネレータアプリケーション例の概念図です。S124 プロセッサの内部周辺回路はブルーで、外部周辺機器（加速度センサ等）はグレーで示しています。各周辺回路/機器へのアクセスに使用される構成要素は、接続線上にテキストで示しています。

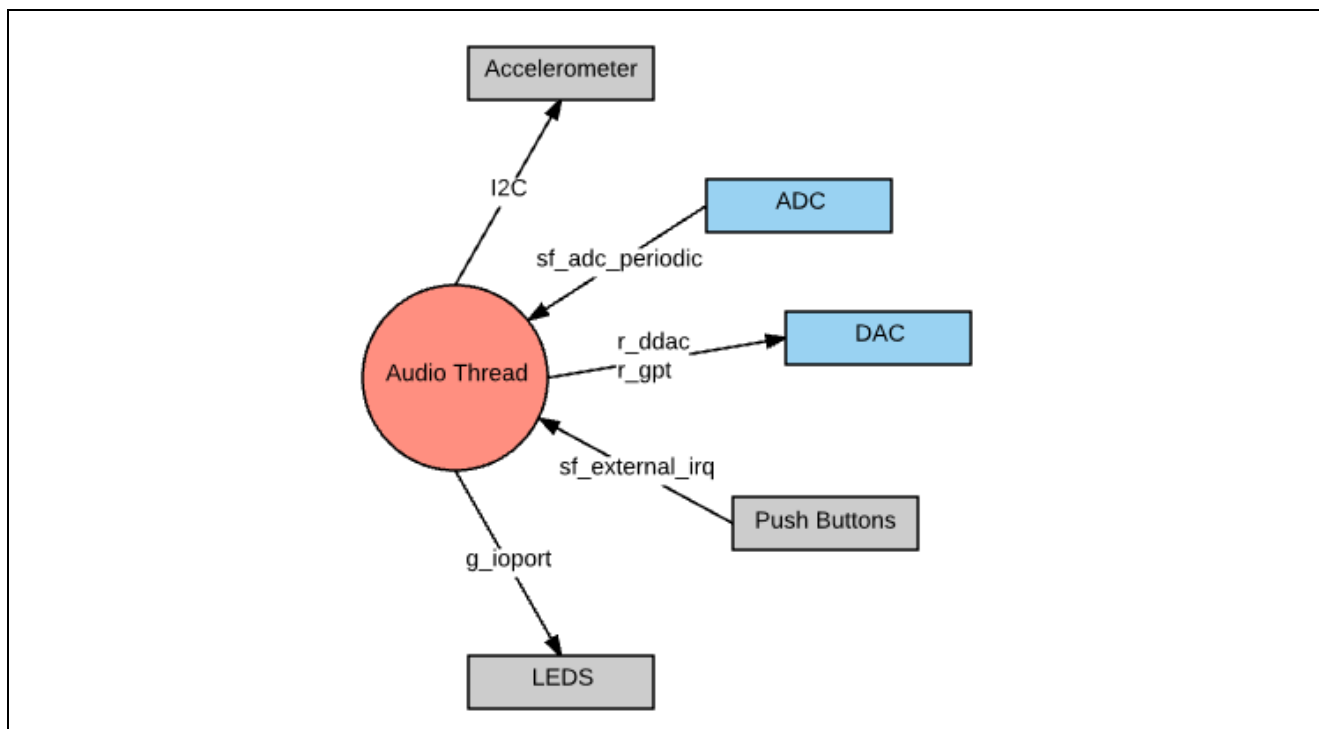


図 3.6 トーンジェネレータアプリケーションの概念図

3.4 オーディオスレッドロジック

オーディオスレッドまたはその他のスレッド内のコードは、3つの別々のセクションに分割できます。

- メインスレッドループに入る前に実行する初期化コード
- メインスレッドループを実行する度に実行するコード
- 割り込みが発生した時にコールバックルーティン内で実行するコード。

メインスレッドループ内で実際に実行するコードは `tx_thread_sleep()` だけのため、トーンジェネレータアプリケーション例は独特です。このコールによりスレッド実行を休止させながら、その他のスレッドの実行を可能にします。

オーディオスレッドは4つコールバックルーティンを含んでおり、全ての作業がそれらのルーティンのみで実行されます。

- 各スイッチ(S1およびS2)のための独立したコールバックルーティン
- 周期的ADCフレームワーク用のコールバックルーティン。アナログデータをボード上のポテンシオメータ(POT1)から一定間隔でサンプリングし、データを使って出力量を決定します。
- 汎用タイマ(GPT)割り込みに関連する別のコールバックルーティン。これは設定された速度で正弦波データを再生するのに使われ、この速度はユーザが聞くトーンの周波数を決定します。スイッチ割り込みの1つにより速度が変わり、事前に定義された周波数を、すべて順に入れ替えていきます。

図 3.7 はメインスレッドループで実行されるロジックを表しています。反応を良くするため、オーディオスレッドは 50ns 毎にボタン状態をサンプリングします。S1 および S2 の通常のボタン押下でいくつかの変数(例: `cur_freq` または `amp_enable`) が設定されます。これらの変数は必要であれば、他のコールバックルーティン内で、テストされます。

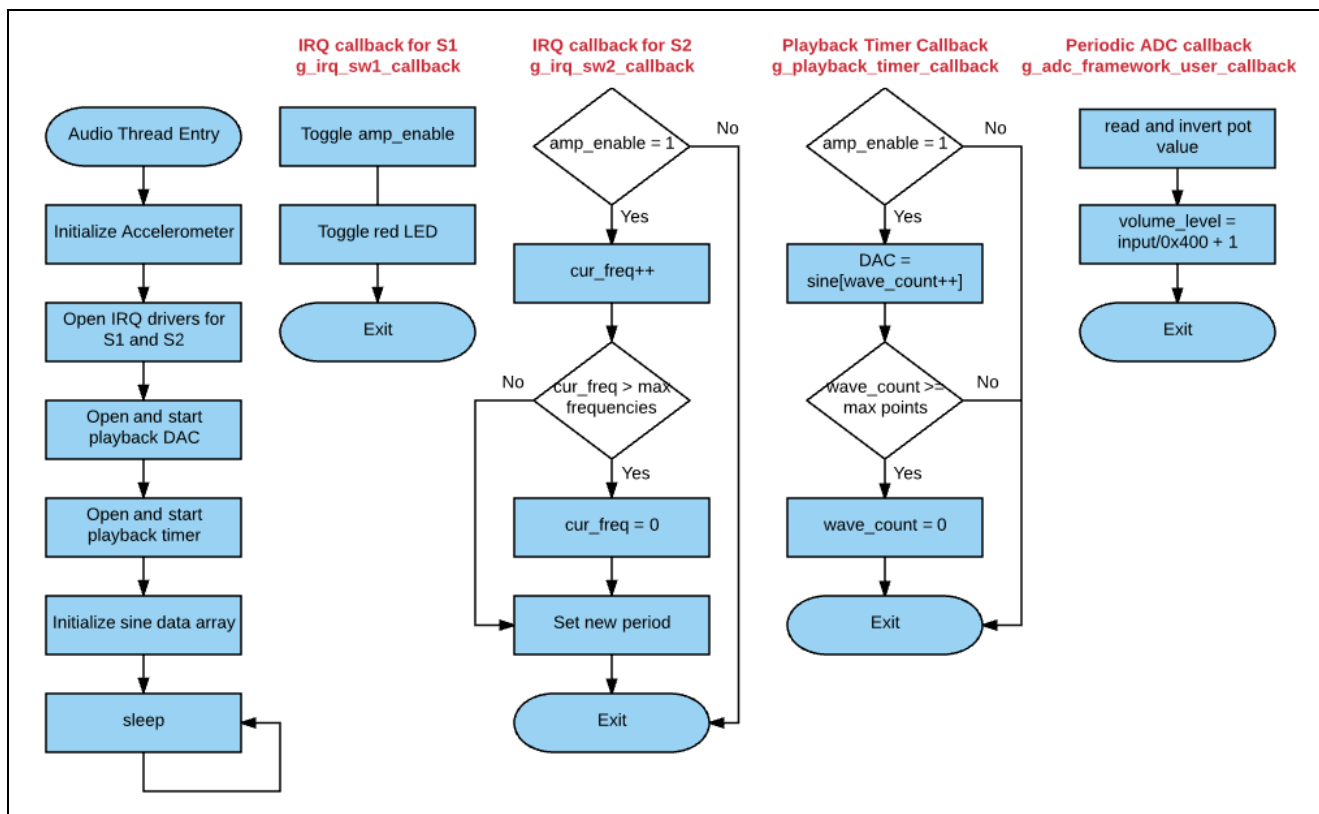


図 3.7 入力スレッド論理の簡潔フローチャート

3.5 スレッドおよびモジュール

全ての Synergy アプリケーションでは、モジュールドライバがスレッドに追加され、そのプロパティは ISDE コンフィグレータを使って設定されます。e²studio 内の [project explorer] から configuration.xml ファイルをダブルクリックすると、コンフィグレータは開きます。Synergy モジュールおよびドライバの詳細については、SSP ユーザーズマニュアルを参照ください。

外部周辺機器へインターフェースを設定するには、周辺機器が実際どのように S124 プロセッサへ接続されるかを理解する必要があります。例えば前述のとおり、S1 を開放するために加速度センサを初期化する必要があります。S124 プロセッサは I²C バスを介して加速度センサと通信します。このため加速度センサと通信するスレッドに I²C ドライバを追加する必要があります。図 3.8 のとおり、r_{riic} ドライバが追加されています。

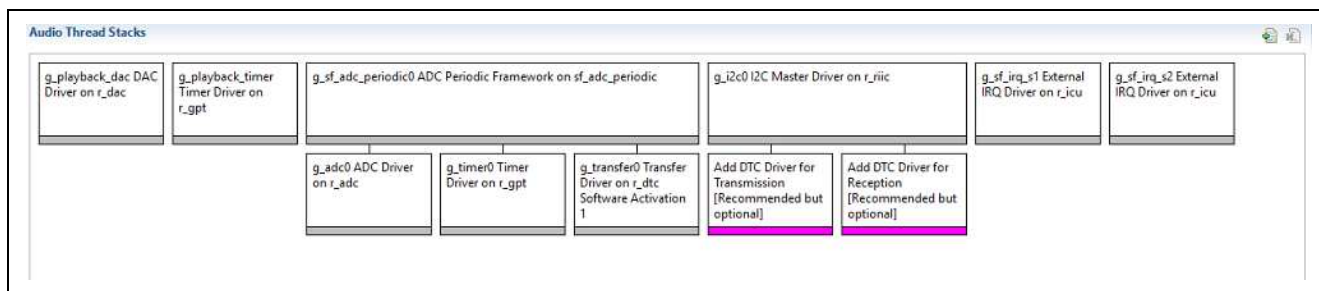


図 3.8 オーディオスレッドスタック

Synergy プロセッサは I²C 通信用に 2 種類の周辺回路、すなわち専用 I²C 周辺回路とシリアルコミュニケーションズインターフェース (SCI) を備えています。SCI はシンプル I²C、シンプル SPI、または UART (同期と非同期) として動作します。SSP は I²C 周辺回路の両方のタイプに対応しています。SPI 周辺回路に関しても同様です。

周辺回路の選択は使用するシステムの複雑さに依存します。多くの場合、専用 SPI または I²C 周辺回路で利用できる機能をすべて使用する必要はありません。専用 I²C 周辺回路に割り当てられた端子を他の用途で使

わなくてはならない可能性もあります。周辺回路の選択はハードウェア設計フェーズで決定するのが通常です。なぜなら周辺機器に接続されている物理プロセッサの端子にも影響を及ぼすからです。

システムを設計する際には、ユーザは設計に応じて周辺回路の種類と必要な数を検討する必要があります。S124 プロセッサは SPI および I²C の 2 個の専用周辺回路を有していますが、シンプル SPI または I²C として設定可能な SCI 周辺回路は 9 個あります。

本トーンジェネレータアプリケーションが、9 個ある SCI 周辺回路のうち 1 個ではなく、専用 I²C 周辺回路を使用する理由は、S124 プロセッサを加速度センサへ接続する際にハードウェア設計者が端子を選択したためです。

3.6 周期的 ADC フレームワークの使用

トーンジェネレータアプリケーションは、ポテンシオメータ (POT1) 上のアナログ値をサンプリングするために周期的 ADC フレームワークコンポーネントを使用します。ポテンシオメータを回していくと読み込み値が適切にスケールされ、4 つの音量レベルのうちの 1 つのレベルに設定されます。周期的 ADC フレームワークコンポーネントはあらかじめ設定された速度でアナログデータをサンプリングします。新しいサンプルが完了する度、データを処理するためにコールバック関数を定義します。

周期的 ADC フレームワークは動作が多少複雑です。このフレームワークコンポーネントの使用経験のあるユーザは、次章へ進んでください。まだ使用経験がないユーザは、本章を詳細に読み、関係のあるコールバック関数のアプリケーションコードを復習してください。

注： ADC Periodic フレームワークコンポーネントの強みの 1 つは、複数のアナログチャンネルをサンプリングできることです。ADC Periodic フレームワークのコンポーネントは反復中でも各チャンネルの複数のサンプルをとることもできます。本書では、その機能性には触れませんが、その他のアプリケーションノート、Simple Record サンプル等では、Periodic ADC フレームワークの追加機能を使用しています。

周期的 ADC フレームワークコンポーネント(図 3.9)は、基本的には r_adc ADC ドライバ、an_r_gpt タイマドライバ、r_dtc 転送ドライバの上にビルドされたフレームワークのレイヤです。転送ドライバ (r_dtc) 以外の各コンポーネントにいくつかのパラメータを設定しなくてはなりません、各コンポーネントの役割はユーザに完全に明確になっています。

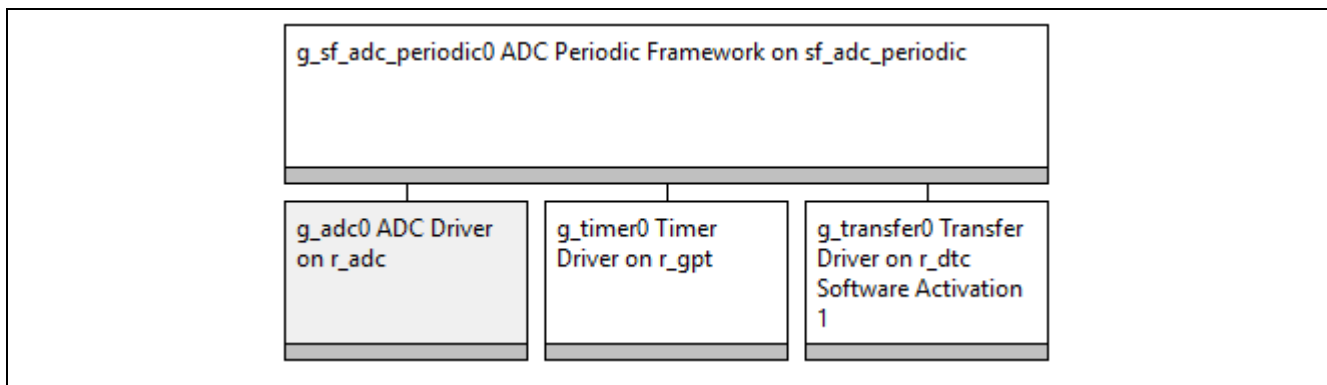


図 3.9 ADC Periodic フレームワークスタック

全ての SSP コンポーネントを設定する場合、適切なボックスの上でクリックした後、[Properties] タブを使いながらプロパティを定義していきます。周期的 ADC フレームワークのコンポーネントは数多くのプロパティを含み、そのデフォルト値はその多くのプロパティで受け付けられます。一度アプリケーションプログラムをインポート、ビルドしたら、第 4 章 デザインデモンストレーションの記述のとおり、アプリケーション内で使用する各コンポーネントのプロパティをすべてブラウズすることができます。本章では、いくつかのプロパティをレビューします。

図 3.10 に示すとおり、サンプルを持つ g_user_buffer バッファは 16 バイトの深さがあります。各スキャンが反復されるたびに 1 つのサンプルが取られ、GPT タイマチャンネル 0 がスキャンのトリガとして使用されます。コールバック名は g_adc_framework_user_callback です。ユーザコード内のコールバック関数がここで定義した関数名と一致する限り、コールバック関数にはどのような名前を付けても構いません。

Property	Value
▼ Common	
Parameter Checking	Disabled
▼ Module g_sf_adc_periodic0 ADC Periodic Framework on sf_adc_pe	
Name	g_sf_adc_periodic0
Name of the data-buffer to store samples	g_user_buffer
Length of the data-buffer	16
Number of sampling iterations	1
GPT Timer channel used to trigger the scan	Channel 0
Callback	g_adc_framework_user_callback

図 3.10 sf_adc_periodic のプロパティ

サンプルを保存するのに使われるデータバッファ(g_user_buffer)は、自動生成されるファイル audio_thread.c 中で、ユーザ向けに定義されます。各スキャンに関連するデータを読むために、コールバックルーティンのなかのこのバッファを参照します。ここで大切なのは、GPT チャンルの選択です。この場合、チャンネル 0 で GPT タイマモジュールが使用するチャンネルがロックされるので、チャンネル 0 を使用しなくてはなりません (図 3.11)。

r_rgpt モジュールの主なパラメータは図 3.11 のとおりです。スキャンを機能させるには、GPT0 COUNTER OVERFLOW IRQ (割り込み)を有効にし、優先順位を選択する必要があります。さらに、50ms のサンプルの速度は[Period Value]を 50、[Period Unit]を[Milliseconds]に設定すると定義できます。これにより、周期的 ADC フレームワークがポテンシオメータにあるアナログ値をサンプリングし、50ms 毎に sf_adc_periodic 構成要素で定義したコールバック関数を呼び出します。

Property	Value
▼ Common	
Parameter Checking	Disabled
▼ Module g_timer0 Timer Driver on r_gpt	
Name	g_timer0
Channel	0
Mode	Periodic
Period Value	10
Period Unit	Milliseconds
Duty Cycle Value	50
Duty Cycle Unit	Unit Raw Counts
Auto Start	False
GTIOCA Output Enabled	False
GTIOCA Stop Level	Pin Level Low
GTIOCB Output Enabled	False
GTIOCB Stop Level	Pin Level Low
Callback	NULL
Interrupt Priority	Priority 2

図 3.11 ADC Periodic フレームワークの r_rgpt プロパティ

注： ユーザのアプリケーションに合わせて、幅広い[Period Unit]から選択ができます。現在オプションには、[Raw Count]、[Nanoseconds]、[Microseconds]、[Milliseconds]、[Seconds]、[Hertz]があります。

3.7 ADC ユニットの設定

構成要素の中で一番プロパティが多いのが r_adc で、ADC ユニットを設定するのに使用されます。ここでもデフォルト値のほとんどが受け付けられます。下半分のプロパティには変更が必要でないため、図 3.12 に

示すプロパティのみに注目してください。SSP 構成要素の大半は、関連する IRQ を有効にしなくてはなりません。ここでは、ADC0 SCAN END 割り込みが Priority 1 に設定されています。実際は、本アプリケーションプログラムのアナログサンプリングは音量制御にしか使用されませんので、この割り込み優先度は低くなります。Simple Record サンプルのような他のアプリケーションプログラムでは、オーディオデータはリアルタイムでサンプリングされますので、より優先度の高い割り込み id を指定する必要があります。

多くの SSP 構成要素では、オプションで構成要素にユーザ自身の名前を指定することがあります。その名前は、ユーザのコード内の構成要素にアクセスするために使用される API コールに使われます。ここでは、g_adc0 のデフォルト名が使われています。コードを読みやすくするために、g_adc_volume_control のような名前も使用できます。名前の付け方は、ユーザまたは使用するコーディング標準によります。

Property	Value
▼ Common	
Parameter Checking	Disabled
▼ Module g_adc0 ADC Driver on r_adc	
Name	g_adc0
Unit	0
Resolution	12-Bit
Alignment	Right
Clear after read	On
Mode	Single Scan
Channel Scan Mask	Select channels below
Channel 0	Unused
Channel 1	Unused
Channel 2	Unused
Channel 3	Unused
Channel 4	Unused
Channel 5	Unused
Channel 6	Unused
Channel 7 (S3A7/S124 Only)	Use in Normal/Group A
Channel 25	Disabled
Channel 26	Disabled
Channel 27	Disabled
Temperature Sensor	Disabled
Voltage Sensor	Disabled
Sample and Hold Mask	Select channels for which individual sample and hold circuit is to be enabled
Channel 0	Disabled
Channel 1	Disabled
Channel 2	Disabled
Sample Hold States (Applies only to the 3 channels selected ab	24
Callback	NULL
Scan End Interrupt Priority	Priority 1
Scan End Group B Interrupt Priority	Disabled

図 3.12 ADC Periodic フレームワークの r_adc コンポーネントのプロパティ

r_adc 構成要素中、正しく設定する必要のある重要なプロパティの 1 つが [Resolution] です。本アプリケーションノート執筆時点では、本フィールドのデフォルトは 8 ビット (S7G2 のみ) のため、S124 プロセッサ(図 3.13)では動作しません。

注：S124 プロセッサで作業をしているときに、12 または 14 ビット (S3A7/S124 のみ) を選択しなかった場合、無効パラメータエラーを実行中に受け取ります。実行中のため、追跡は若干困難です。

Resolution	8-Bit (S7G2 Only)
Alignment	14-Bit (S3A7/S124 Only)
Clear after read	12-Bit
Mode	10-Bit (S7G2 Only)
Channel Scan Mask	8-Bit (S7G2 Only)
	Select channels below

図 3.13 8 ビット(S7G2 のみ) の ADC 解像度デフォルト

最終的には、ADC スキャンが発生する度にサンプリングされる実チャンネルを設定する必要があります。ここでは、選択されているチャンネルは、ポテンシオメータに接続されているアナログチャンネルのチャンネル 7 のみです。Simple Record サンプルのような他のアプリケーションノートでは、複数チャンネル（例：ポテンシオメータおよびマイクロフォン入力）で同時にサンプリングする方法を示しています。

3.7.1 周期的 ADC フレームワークのコールバック

全てのプロパティを定義してコードをビルトすると、下記のようなコードの一部が audio_thread.h ファイル内に見られます。

```
/** Buffer where the sampled data will be stored for application usage */
extern uint16_t g_user_buffer[16];
#ifdef g_adc_framework_user_callback
#define ADC_PERIODIC_ON_ADC_PERIODIC_CALLBACK_USED_g_sf_adc_periodic0 (0)
#else
#define ADC_PERIODIC_ON_ADC_PERIODIC_CALLBACK_USED_g_sf_adc_periodic0 (1)
#endif
#if ADC_PERIODIC_ON_ADC_PERIODIC_CALLBACK_USED_g_sf_adc_periodic0
/** Declaration of user callback function.This function MUST be defined in the user application.*/
void g_adc_framework_user_callback(sf_adc_periodic_callback_args_t * p_args);
#endif
```

このコードは、コールバック関数の原型を定義します。またこのコードは、ユーザアプリケーションプログラムの中で定義されなければなりません。audio_thread_entry.c ファイルの下部に g_adc_framework_user_callback() のコードがあります。フローチャート図 3.7 に、各コールバックルーティンの機能性を示しています。

3.8 ソースコードレイアウト

e²studio 内でトーンジェネレータプロジェクトをビルドした時に表示される、ソースコードレイアウトの概略を本章で説明します。既に Synergy アプリケーションプログラムをプログラミングした経験のあるユーザは、e²studio が標準プロジェクトを体系化する方法を理解されていますので、次の章へ進んでください。

トーンジェネレータアプリケーションプログラムを e²studio にインポートした後、プロジェクトをビルドすると、図 3.14 のようなディレクトリ構成になります。プロジェクトをインポートするのが初めての場合、以下のフォルダ（synergy_gen、synergy、debug、synergy、synergy_cfg）は存在しません。これらのフォルダは、プロジェクトをコンパイルする時に、フレームワークが自動的にビルドしますので、アプリケーションが配布されるときにはサイズを小さくするため、サブフォルダは通常含まれていません。

ユーザのプロジェクトで作業する（コードの追加・削除等）必要があるファイルは緑色で表示されています。

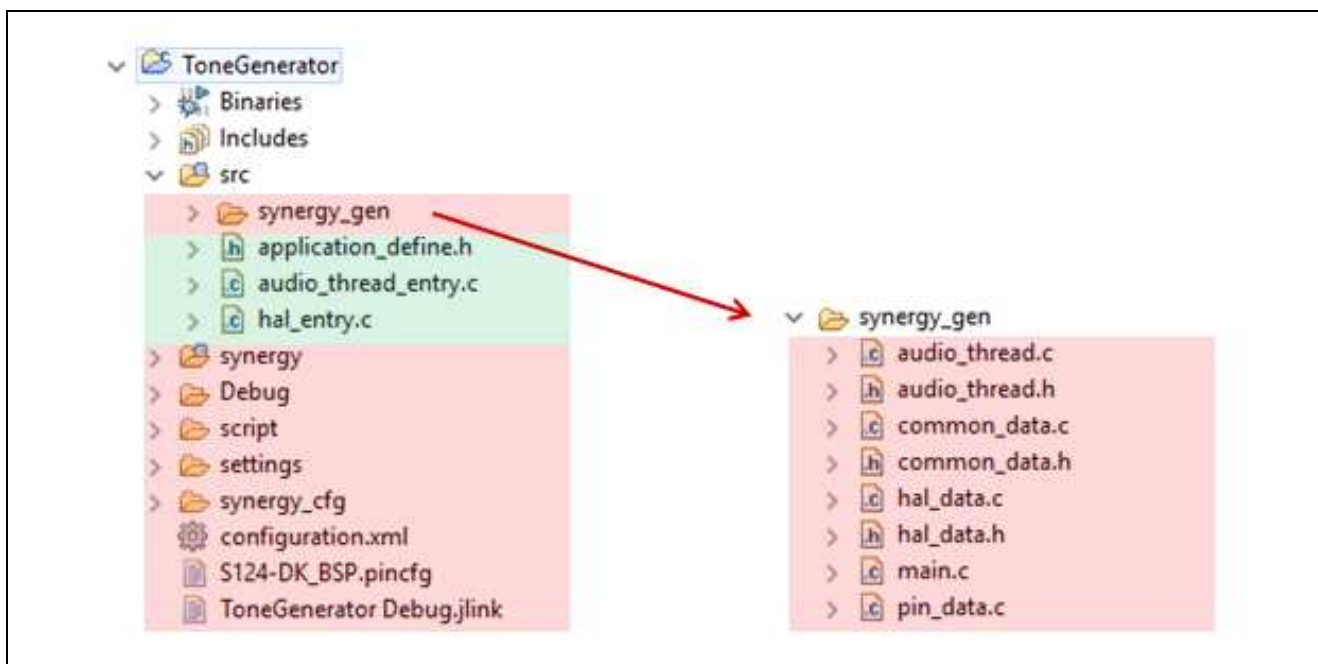


図 3.14 DK-S124 OOB v1.0 プログラムのソースコードレイアウト

4. デザインデモンストレーション

DK-S124 は、Out-of-Box アプリケーションソフトウェアを事前ロードしています。この Out-of-Box アプリケーションソフトウェアは、多様なアナログセンサ値を PMOD LCD スクリーン上に表示します。トーンジェネレータサンプルコードを実行する前に、トーンジェネレータファームウェアでボードを再度書き込みする必要があります。再書き込みする前に、次章に従いボードをまず設定します。

4.1 トーンジェネレータサンプル用の DK-S124 設定

DK-S124 を設定する手順は以下のとおりです。

1. 図 4.1 の上部に示すとおり、バッテリーの近くのヘッダーが全てすべてのジャンパに設定、接続されていることを確認します。
2. マイクロ USB ケーブルを使って、図 4.1 の下部に示す DK-S124 の J18 : JLink-OB を PC に接続します。
3. スピーカーまたはヘッドフォンを、3.5mm ジャックプラグを使って図 4.2 に示す J16 に接続します。

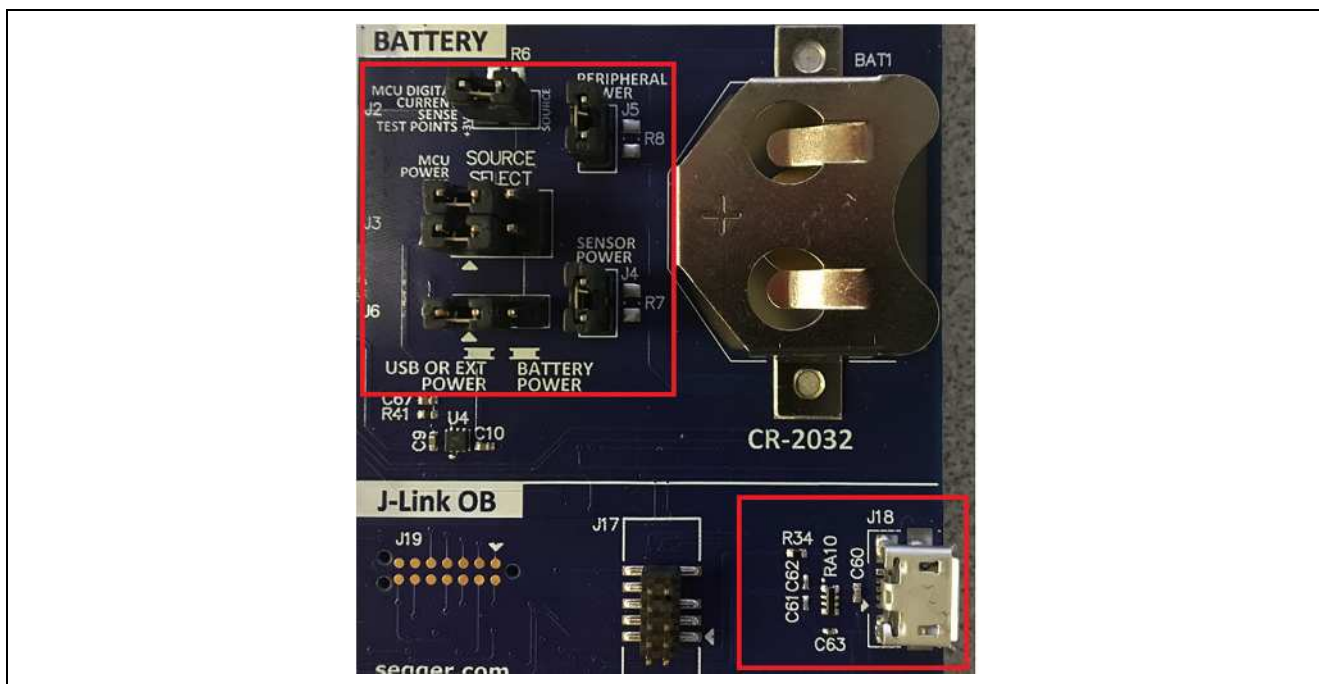


図 4.1 電源設定(上) / J-Link OB 接続(下)



図 4.2 ヘッドフォンまたは外部スピーカーの接続

ボードの設定が完了したら、e²studio にトーンジェネレータアプリケーションをインポート、コンパイルし、DK-S124 ボードにロードします。

電源を投入するとトーンジェネレータプログラムは、トーンを出力しあらかじめ定義された周波数リストの最初のトーンを再生するように、デフォルトで設定されています。第 4.2 章はプログラムの扱い方を説明します。

4.2 テストジェネレータサンプルプログラムの扱い方

テストジェネレータプログラムの扱い方の手順は以下のとおりです。

1. S1 は出力を有効 / 無効にします。有効に設定した場合、赤い LED (LED1) が点灯し、トーンジェネレータが現在選択されている周波数を出力します。無効に設定した場合、赤い LED を消滅し、イヤホンジャックへ音は何も出ません。電源を投入すると、プログラムは出力が有効になるようにデフォルト設定されています。
2. 出力が有効の場合、S2 はトーンジェネレータのプログラムされた周波数リストをインクリメントします。出力が無効の場合 (S1 を使用)、S2 では無視されます。
3. POT1 は音量制御として機能し、時計回りに回すと音量を上げます。
注意事項：過剰な音量設定で聴くと、聴覚を傷つける可能性があります。

5. プロジェクトのインポートとビルド

Synergy Project インポートガイド (r11an0023eu0116-synergy-ssp-import-guide.pdf) の手順に従って、プロジェクトを e²studio ISDE へインポートしてから、ビルドおよびデバッグします。デバッグの設定は、**Tone Generator Debug (Renesas GDB Hardware Debugging の下)** を選択します。

6. e²studio の技法 - 変数のテスト

アプリケーションをデバッグする場合、デバッガでコードを通すときに特定の変数の値をテストする必要があります。Renesas e²studio は、このテスト方法を 2 つ提供します。プロジェクトのコンパイルが完了してボードへロードすると、e²studio は [Debug] ビューに切り替えるかどうかを聞いてきます。下記の複数のイメージは、ユーザが [Debug Perspective] にいることを想定しています。

Window > Perspective > Open Perspective > Other > Debug perspective の順に選択していくことで、いつでも [Debug Perspective] へ変更することができます。

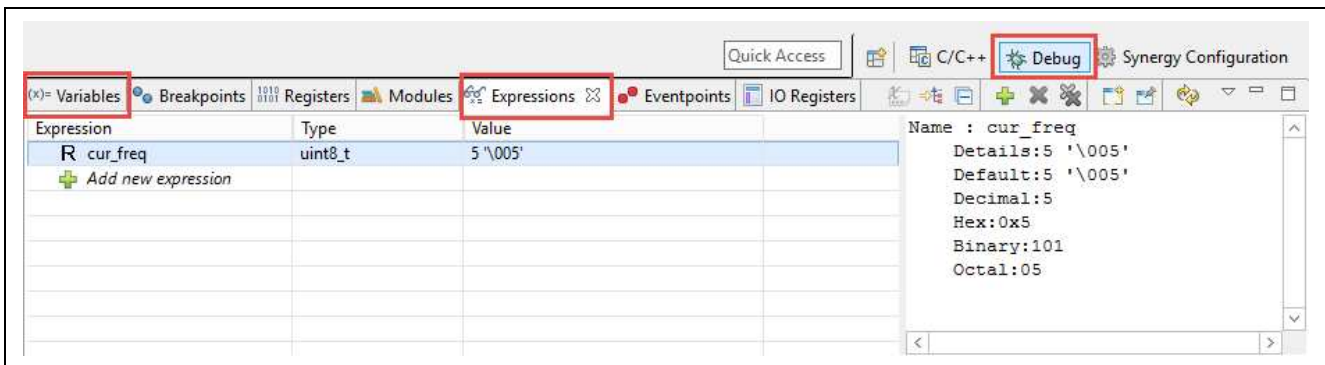


図 6.1 変数および式タブ

最も一般的な変数のテスト方法は、[Variables] または [Expressions] タブ (図 6.1 で赤枠) を使用して行います。この 2 つのタブでしばしば同じ情報が提供されますが、それぞれの機能は異なります。[Variables] タブは、ユーザがブレークポイントをヒットした時に、実行中の関数に従い自動的に変数を加えます。

例えば、ブレークポイントを下記 `g_playback_timer_callback()` ルーティンのハイライト部分に設定する場合、[Variables] タブは自動的に [status] および [wave count] の変数の内容を表示します。本アプリケーションノート執筆時の [Variables] タブの欠点は、グローバル変数を表示しないことです。

```

void g_playback_timer_callback(timer_callback_args_t * p_args)
{
    SSP_PARAMETER_NOT_USED (p_args);
    ssp_err_t status;
    static uint16_t wave_count = 0;

    /* If the output is enabled, write the current data point for the sine wave to the DAC */
    if(amp_enable){
        /* Write out the sine wave data */
        status = g_playback_dac.p_api->write(g_playback_dac.p_ctrl,
                                           (dac_size_t)(sine[wave_count++] << volume_level));
        APP_ERR_TRAP(status);
        if (wave_count >= SINE_POINTS) { wave_count = 0;}
    }
}

```

ここから[Expressions]タブが機能します。[Expressions]タブでは、変数名または変数名を使った式でも入れることができます。現在使用中の関数のローカル変数を見ることだけに制限されませんが、いずれの変数の内容も自動的に追加することはしません。

Name	Type	Value
> p_args	timer_callback_args_t*	0x20001428 <g_main_stack+952>
(x)= status	ssp_err_t	5160
(x)= wave_count	uint16_t	0

図 6.2 関数の変数を自動で表示する Variable タブ

変数を表示するには、[Add New Expression]のプロンプトの隣の緑の矢印をクリックして変数名を入力します。これを図 6.3 に変数 (cur_freq、wave_count、amp_enable) を加えて説明します。画面では式 [Expression]、[Type]、[Value]の欄を表示しています。この場合、式は単に変数名になっていますが、式のいずれかをクリックすると[Expressions]タブの右側にペイン (小窓) が現れ、データを別の形式で表示しますデータ (例: 16 進 10 進)。

Expression	Type	Value	Name
R cur_freq	uint8_t	0 '\0'	cur_freq
(x)= wave_count	uint16_t	0	
(x)= amp_enable	uint8_t	1 '\001'	
+ Add new expression			

Details for cur_freq:
 Details: 0 '\0'
 Default: 0 '\0'
 Decimal: 0
 Hex: 0x0
 Binary: 0
 Octal: 0

図 6.3 Expressions タブによる 3 つの変数の内容表示

図 6.3 では、cur_freq 変数の左側に大文字の R があります。デフォルト設定では、[Expressions]タブ内の表示の値はユーザがブレークポイントをヒットした時のみに更新されます。[Expressions]タブの利点の 1 つに、リアルタイムで変数の内容を表示できることがあります。表示するにはいくつかの方法があります。第 1 の方法は、アプリケーションプログラムを実行中に、見たい変数の横の (x) =の上をクリックします。これは図 6.4 のダイアログボックスのように、この式をリアルタイムで再読み込みするかどうかを聞いてきます。

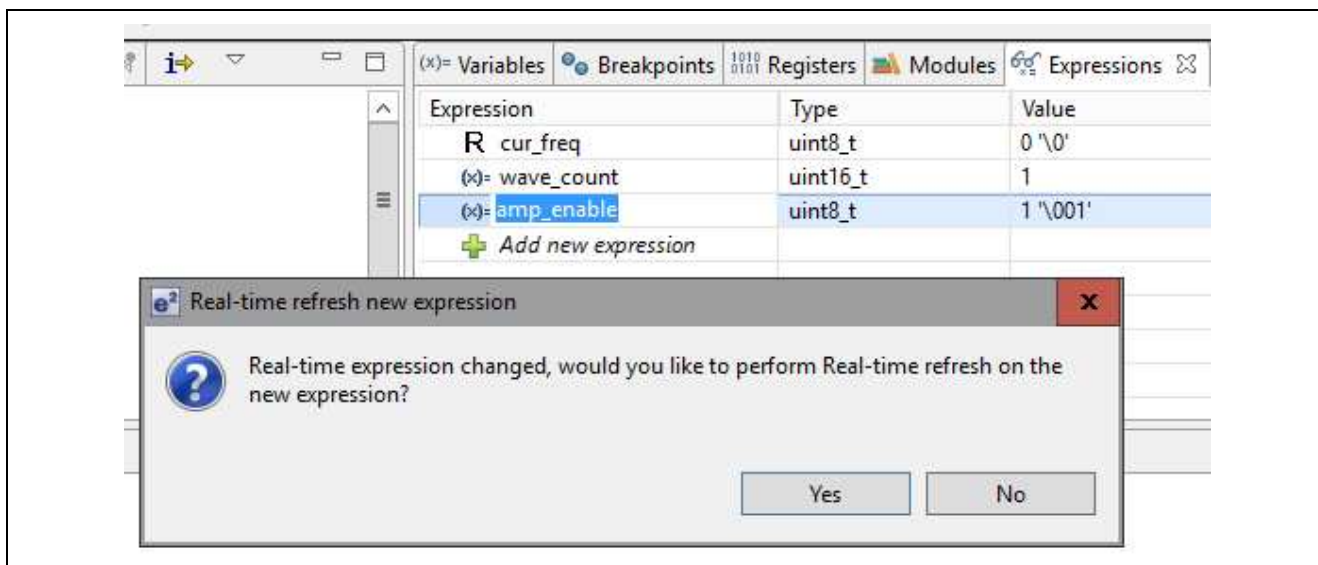


図 6.4 ダイアログボックス：リアルタイム再読み込みのプロンプト

第2の方法は、図 6.5 のとおり、変数の上で右クリックをして、プルダウンメニューより[Real-time refresh]を選択します。またこのドロップダウンメニューは、リアルタイム再読み込みを無効にする場合、現在選択している項目と関連のある式を変更する場合、式を削除する場合等にも使用します。

特定の式に対してリアルタイム再読み込みを有効にした場合、変数の値を見るために、ブレークポイントをヒットしてアプリケーションプログラムを一時停止する必要はありません。

注： いくつかの変数は、他の変数よりも実行時間をテストするのが容易です。変数の変化が速すぎる場合、変化時にすべての値をみることは困難です。その場合はブレークポイントを設定して、変数の値をプログラムの実行中の特定の時点で確かめてください。

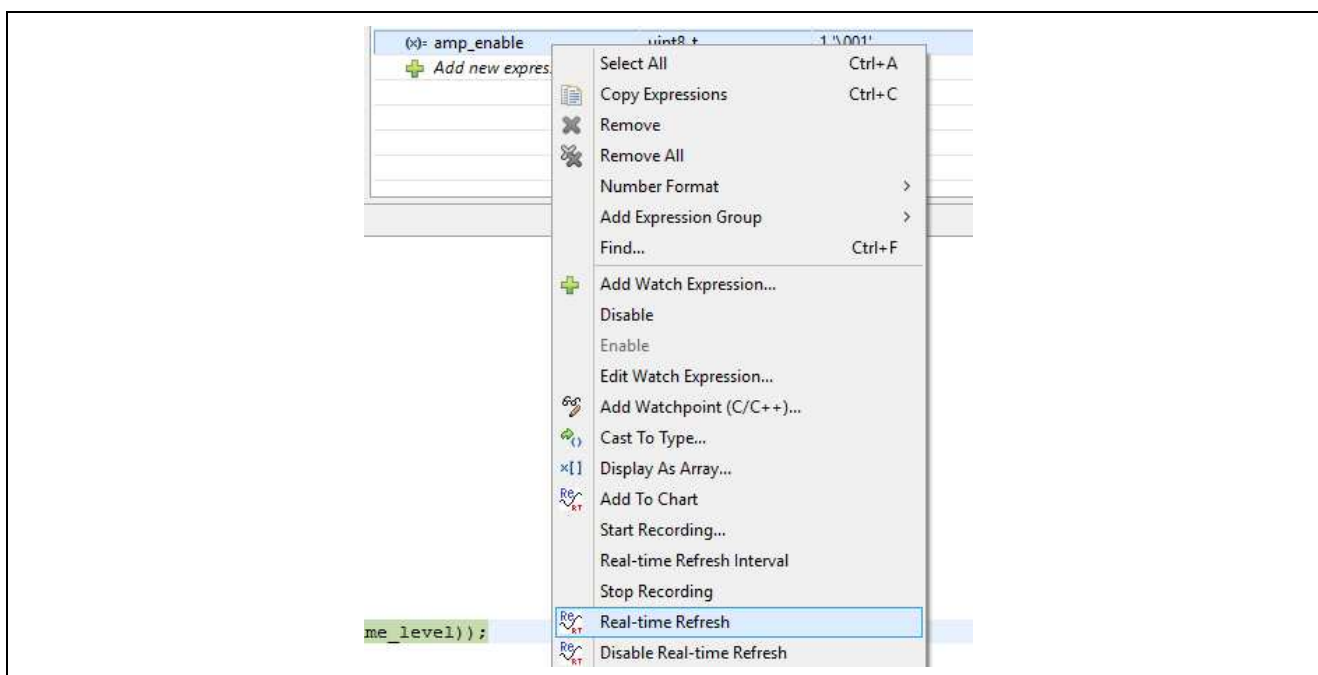


図 6.5 リアルタイム再読み込みの有効化

6.1 グラフィック表示

最後の方法では、[Visual Expressions]を使って変数を確認します。e2studio は、複数のグラフィックがユーザーデータを表示します。この複数のグラフィックは[Visual Expressions]タブから選択できます。[Visual Expressions]タブが表示されていない場合、Window > Show View > Other と選択して、Debug フォルダの中の [Visual Expressions]を選択します。

図 6.6 は、[Visual Expressions]タブ内で、表示できる例を表しています。

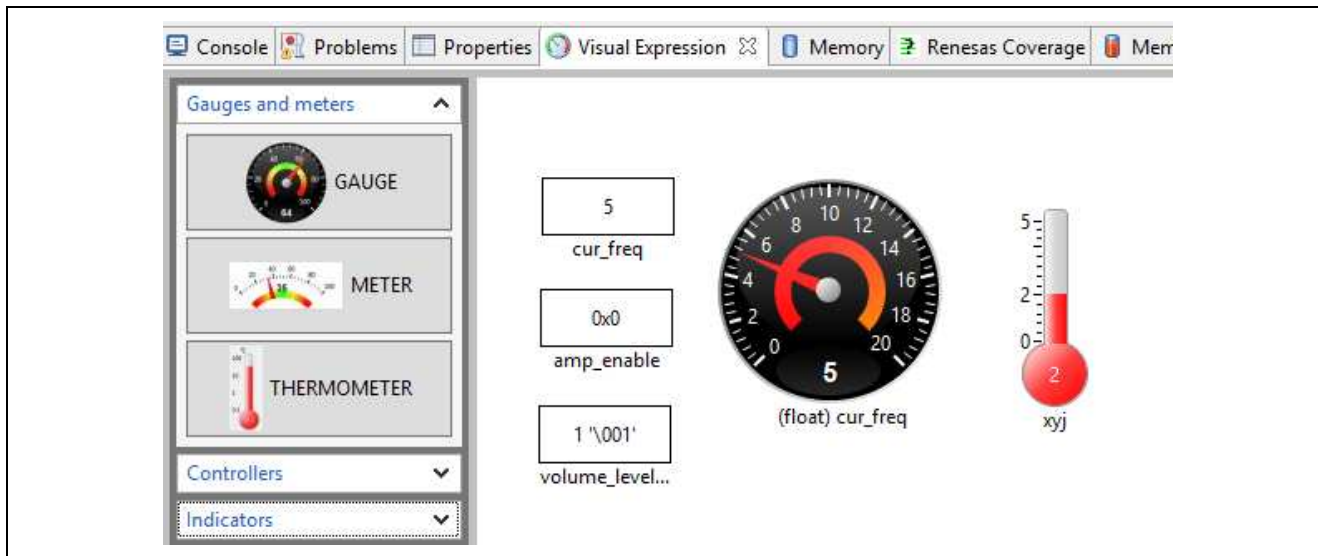


図 6.6 視覚表示

ユーザデータのグラフィック表示はウィジェットとして参照します。左側のペイン（小窓）内の[Gauges]、[Controllers]もしくは[Indicators]から、ウィジェットを右側のペインキャンバスへドラッグ&ドロップします。それからプロパティを設定します。

左から小さい長方形、ゲージ、温度計と3つのラベル並んでいます。1つのラベルの上で右クリックすると設定できる動作のプルダウンメニューリストが表示されます(図 6.7)。

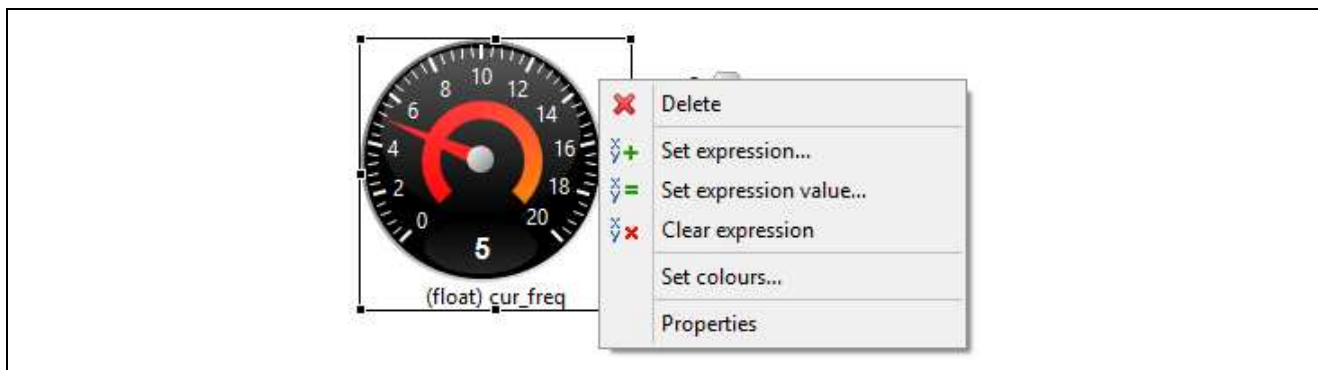


図 6.7 ゲージの設定

変数をグラフィック表現と関連づけるには、[Set expression]をオプションで選択し、変数名または式をタイプします。

注： 本アプリケーションノート執筆時点では、ゲージなどの特定のグラフィック表現は float 式の値のみ認識します。整数の変数を表示するには、表示されるまえに値を float 型に変換する必要があります。

図 6.8 のとおり、ラベルのウィジェットには追加オプションがあり、数字形式を選択できます。

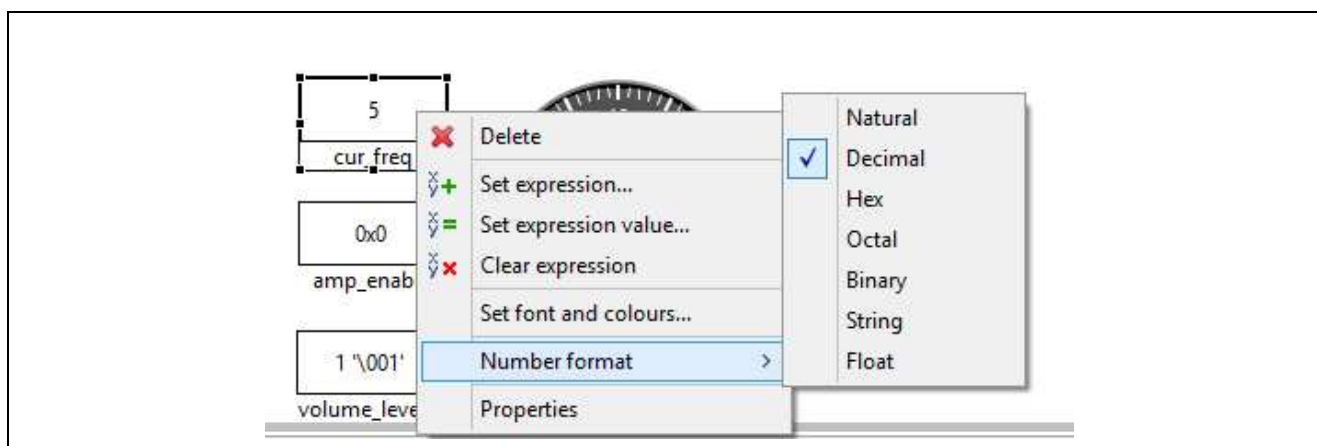


図 6.8 ラベルウィジェットの数字形式の選択

7. 次のステップ

アプリケーション例を実行した後は、アプリケーションの動作や API コールを、API アプリケーションソースコードを調べることでより深く学習することができます。

以下を試験的に実行してみることを推奨します。

1. S1 のデジタルフィルタリングを無効にした結果を見ます(第 4.1 章参照)。デジタルフィルタリングを無効にした場合、S1 を押すと、周波数の増加が 1 インクリメント以上で聞こえることがあります。これは、IRQ がスイッチバウンスに反応して、1 回のスイッチを押すたびに複数回 IRQ コールバック（現在の周波数を増加する）を発生させるためです。
2. 正弦波を表示するために使用するデータポイント数を増減します。これは、audio_thread_entry.c 内の SINE_POINTS の値を変更し、プロジェクトを再度ビルドして行います。正弦波を表示するためのポイント数を増加させると、生成されるトーンの周波数が減少します。逆に、ポイント数を減少させると、ユーザに聞こえるトーンの周波数が増加します。

オシロスコープを使う場合は、ヘッダー J6 の端子 7 に接続されている P0_14 で DAC の出力をモニタすることができます。以下の図は約 1kHz での 2 つの異なる出力を表しています。左側の図は正弦波を表示するためにより多くのデータポイントを使用し、右の図はその約半分のデータポイントを使用しています。

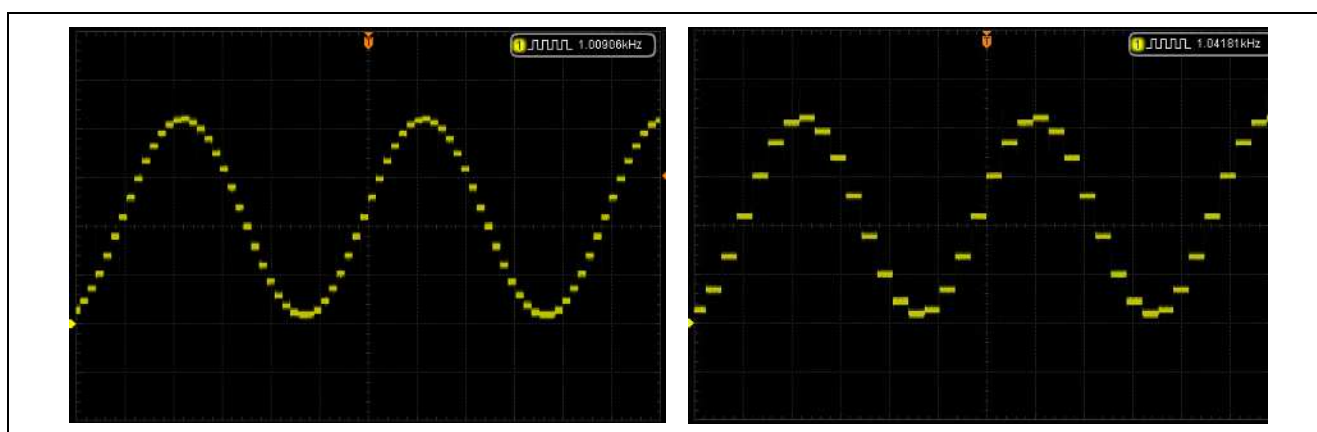


図 7.1 データポイント数を変える効果

注： ナイキスト定理によると、周波数を 2 倍にすると、その周波数を表示するために必要なデータポイントの最低数が出ます。実際には、多くのデザイナーは 4 ~ 10 のデータポイントが必要と考えています。より多くのポイントを取れば、出力される波形の再現性がより向上します。

Renesas のホームページでは、多数の Synergy サンプルを紹介しており、随時新規サンプルが追加されています。追加されたソリューションを見るには、下記 URL から Synergy アプリケーションサンプルをご覧ください。<https://www.renesas.com/en-us/search/keyword-search.html#genre=tooldownload&q=synergy>.

ホームページとサポート窓口

Renesas Synergy™ ギャラリー :

<https://synergygallery.renesas.com/support>

テクニカルサポート窓口 :

- 米国: https://renesas.zendesk.com/anonymous_requests/new
- ヨーロッパ: <https://www.renesas.com/en-eu/support/contact.html>
- 日本: <https://www.renesas.com/ja-jp/support/contact.html>

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2016.08.11	-	初版
1.01	2017.02.16	-	SSP 1.2.0バージョンに更新

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。お客様の機器・システムの設計において、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には、お客様の責任において行ってください。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まれます。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、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、全部または一部を問わず、改造、改変、複製、その他の不適切に使用しないでください。かかる改造、改変、複製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。
 5. 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、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、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り、これらの用途に使用することはできません。たとえ、意図し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。
 6. 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。
 7. 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。
 8. 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。
 9. 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。また、当社製品および技術は、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。
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。
 10. お客様の転売、貸与等により、本書（本ご注意書きを含みます。）記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
 11. 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。
 12. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。
- 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その総株主の議決権の過半数を直接または間接に保有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.3.0-1 2016.11)



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<https://www.renesas.com/contact/>