Application Note

MCU Signal Sight ツール開発者ガイド

Delaney Woodward, Peter Luong, Nima Eskandari

概要

組込みアプリケーションのデバッグや評価の従来の方法は、多くの場合高価な機器を必要とし、内部信号を観測できる範囲も限られています。現在のソフトウェアによる代替手段は、コストは低いものの、リアルタイム制御アプリケーションを正確に評価するために必要な精度を提供できません。この断片的でリソース集約型のデバッグ環境は、顧客にとって障壁となります。MCU Signal Sight ツールは、ソフトウェアベースの高性能なグラフィカル ユーザー インターフェイス (GUI) を提供し、組み込みエンジニアにリアルタイム プロットや高度なデバッグ ツールをもたらします。MCU Signal Sight は、使い慣れたインターフェースで正確なリアルタイム信号の可視化を可能にし、データ エクスポートやリアルタイム統計解析といった強力な機能を提供します。このツールは、簡単に構成可能で軽量なソフトウェアによって動作し、あらゆるソフトウェア プロジェクトにすぐに組み込むことができます。MCU Signal Sight は、あらゆる開発環境にシームレスに統合でき、スタンドアロンツールとして、Code Composer Studio™ (CCS) に統合して使用するか、TI のクラウド ツールを使用してオンラインで利用できます。

目次

図の一覧

商標

Code Composer Studio™ and LaunchPad™ are trademarks of Texas Instruments.
すべての商標は、それぞれの所有者に帰属します。

1 はじめに

MCU Signal Sight は、C2000 マイコン (MCU) アプリケーションからのリアルタイム信号を仮想オシロスコープ上にプロットするためのグラフィカル ユーザー インターフェイス (GUI) です。SysConfig ツールは、組込みアプリケーション内で使用するためのカスタム組み込みソフトウェア ライブラリと、関連するプロット ユーティリティを備えたカスタム GUI アプリケーションを自動生成するために使用されます。このアプリケーション ノートでは、MCU Signal Sight GUI の機能を紹介し、既存の CCS プロジェクトに MCU Signal Sight のサポートを実装する方法を詳しく説明し、さらに C2000ware に含まれる MCU Signal Sight のサンプルの使用方法について概要を示します。

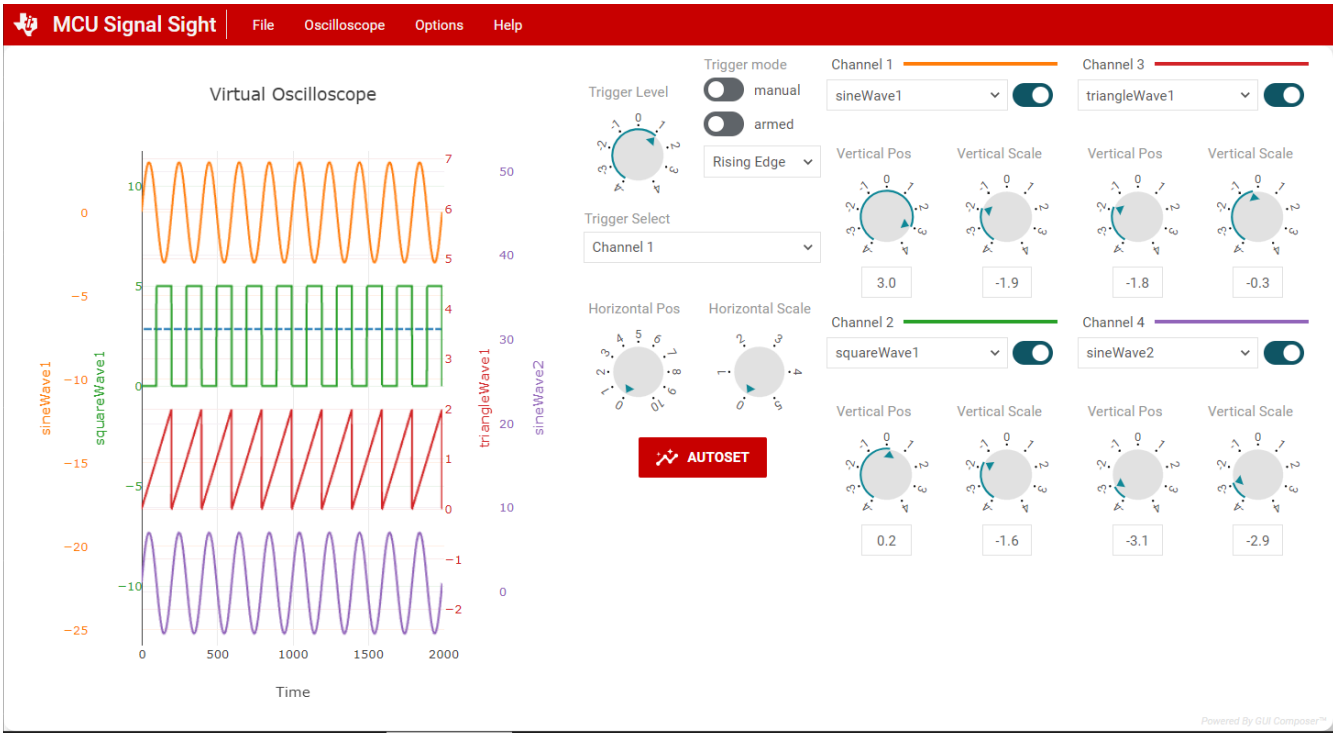

図 1-1. MCU Signal Sight GUI ツールの図

2 特長

MCU Signal Sight ツールは、以下の機能をサポートしています:

- デバッガ接続を必要としない、高速な MCU と GUI 間の通信
  - すべての C2000 LaunchPad™と controlCARD で、UART-USB ブリッジを利用します
- MCU アプリケーション ソフトウェア内部の信号や変数を可視化できる、拡張オシロスコープ機能
- 最大 4 つの同時プロット チャネル
- SysConfig の選択肢に基づいて、組込みコードと GUI コードを自動生成
- ツールの実行に必要な最小限の CPU 帯域幅
  - アプリケーションの非侵入デバッグをイネーブルにします
- プロット信号の解析の強化
  - 手動およびアーム式トリガ モード
  - スコープ表示、バッファ サイズ、サンプル レートの設定をカスタマイズ可能
  - 波形アナライザ
  - グラフ表示の自動設定
- CSV ファイルのエクスポート機能
  - 解析可能な形式で外部プログラムを使用したデータ分析が可能です

2.1 ソフトウェアの設定

MCU 信号サイトのサポートは、SysConfig をサポートし、かつ C2000ware バージョン 5.05.000 以降を使用している既存の C2000 プロジェクトに追加できます。GUI と必要なライブラリ ファイルは、統合された CCS 環境を通じて完全に生成され、インターフェイスできます。必要に応じて、MCU Signal Sight GUI のスタンドアロン版をダウンロードして、CCS インストールとは独立して実行することもできます。

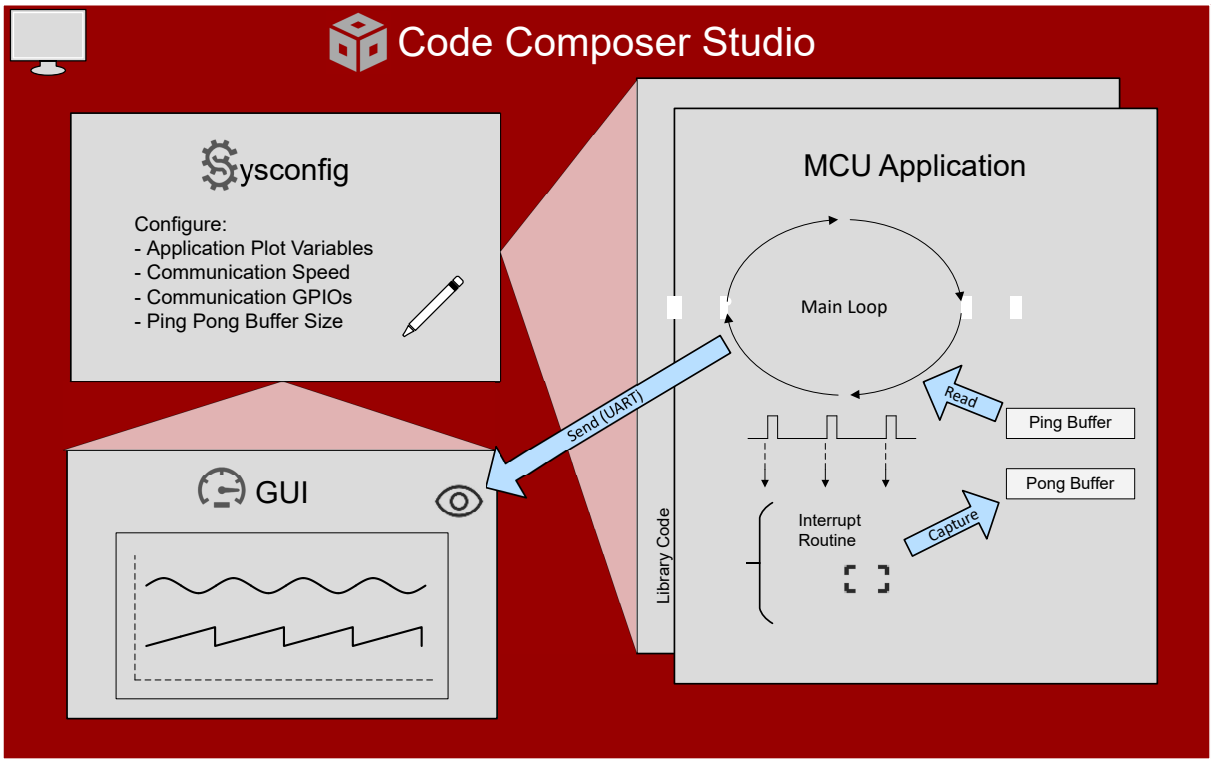

図 2-1. Signal Sight ソフトウェアの図

ユーザーの SysConfig(.syscfg) ファイルに MCU Signal Sight モジュールを追加すると、SysConfig ツールはターゲット アプリケーションで使用するライブラリ ファイルを自動生成し、仮想オシロスコープ部品を備えた統合 GUI Composer アプリケーションを作成します。

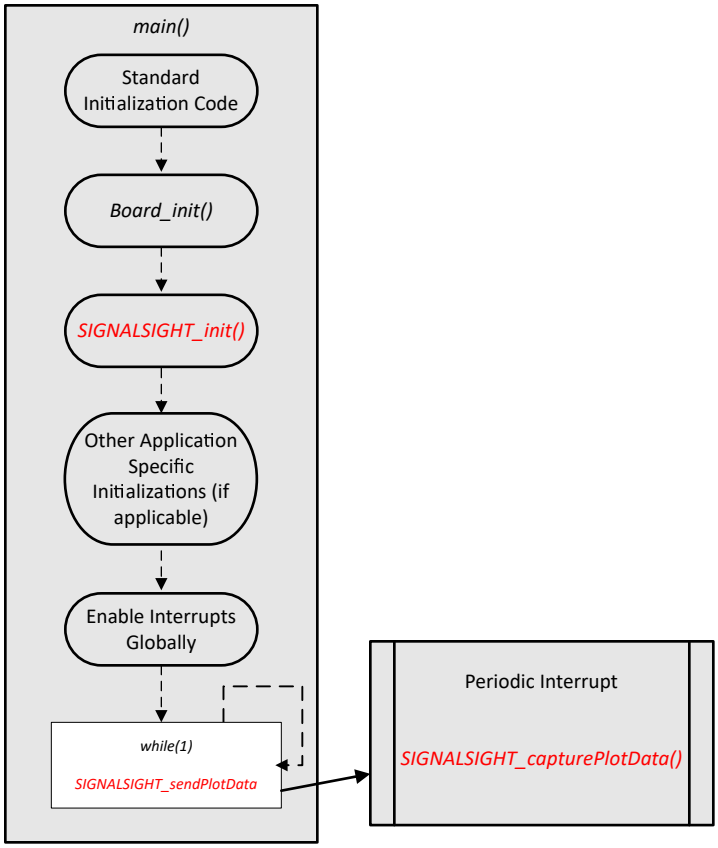

図 2-2. Signal Sight ソフトウェア フロー図

ライブラリ ファイルは、ターゲット アプリケーションで使用するために、ライブラリ内に 3 つのトップレベル関数を生成します。これらは以下のとおりです。ユーザーのアプリケーションでこれらの関数にアクセスするには、次の行を追加して signalsight.h ヘッダー ファイルを含めます: *#include <signalsight/signalsight.h>*

- *SIGNALSIGHT_INIT()* - Signal Sight 初期化関数
  - この関数は、デバイス初期化、割り込み初期化、SysConfig 初期化 (Board_init()) の後、ただしグローバルに割り込みを有効にする前に、初期化コード内で一度だけ呼び出します。
- *SIGNALSIGHT_sendPlotData()*- Signal Sight キャプチャ プロット データ関数
  - この関数は、データをサンプリングする必要があるアプリケーション コードの箇所で呼び出します。プロット変数の値はメモリ バッファに保存され、後続の SIGNALSIGHT_sendPlotData() 呼び出し時に送信されます。通常、この関数は周期タイマまたは ISR 割り込みから呼び出されます。
- *SIGNALSIGHT_capturePlotData()*- Signal Sight データ送信関数
  - この関数は、デバイスが SCI ペリフェラルを介してバッファされたデータを GUI に送信するアプリケーション内で呼び出します。この関数は、CPU が SCI TX FIFO に十分な空きができるまで待機してブロックされる可能性があるため、アプリケーション コード内の低優先度の場所、例えばコードのメイン ループ内に配置することが推奨されます。

2.2 ハードウェア設定

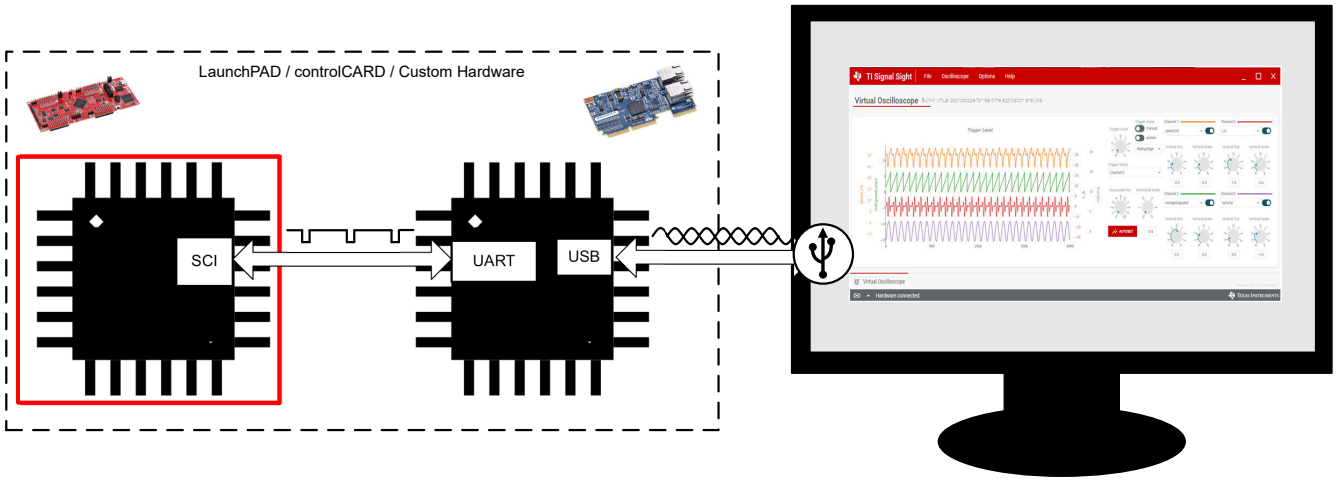

図 2-3. MCU Signal Sight ハードウェア図

MCU Signal Sight ツールを使用するために必要なハードウェアは、SCI ペリフェラルを備えた C2000 デバイスと、ホスト PC 上の GUI アプリケーションと通信するための何らかの UART-to-USB ブリッジだけです。標準的な C2000 LaunchPad や controlCARD には、すでにオンボードのブリッジ デバイスが搭載されており、多くの場合、このデバイスには UART-to-USB ブリッジを実装する XDS110 ソフトウェアが書き込まれています。MCU Signal Sight のハードウェア セットアップでは、ターゲット MCU がオンボードの SCI (UART プロトコル) ペリフェラルを使用して、PC 上の GUI と通信します。

3 C2000ware の例を実行する

MCU Signal Sight ツールを使用した *transfer_ex8_signalsight_basic_demo* の基本的な例が、C2000ware SDK に含まれています。この例を実行して、ハードウェアとソフトウェアの有効な設定を確認し、ツールの機能を学習します。次の手順を実行してください:

1. CCS を起動します
2. *[File] (ファイル) → [Import Projects] (プロジェクトをインポート) →[Browse](参照)* をクリックします
3. C2000WARE のインストール パスの例に移動します:C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/transfer/ folder。
4. [*Finish (完了)*] をクリックします
5. 実行可能ファイルを右クリックし、[Properties] (プロパティ) を選択します
   a. *[General] (一般) → [Variables] (変数)* で、*GUI_SUPPORT* をダブルクリックし、*[Value] (値)* を *1* に変更します
6. プロジェクト名を右クリックし、*[Debug Project] (プロジェクトをデバッグ)* をクリックします
7. Debug (デバッグ) ウィンドウで [Play] (再生) を押します
8. *[View] (表示) → [Plugins] (プラグイン) → myMCUSignalSight0* に移動します
9. GUI がロードされたら、*[Connect] (接続)*をクリックします
10. GUI ウィンドウ、
    a. *[Oscilloscope] (オシロスコープ) → [Enable All Channels] (すべてのチャネルを有効にする)*をクリックします
    b. トリガ モードを *手動*に設定します
11. 信号の正弦、三角波、方形波を表示します

    サイト インターフェイスウィンドウ

これらの手順を実行したときにユーザーが発生する可能性のある問題のデバッグについては、セクション 7 を参照してください。

4 プロジェクトへの Signal Sight の追加

以下のセクションの手順に従って、Signal Sight の仮想オシロスコープで目的のアプリケーションの変数をプロットします。

4.1 SysConfig のステップ

MCU Signal Sight ツールは Sysonfig GUI に組み込まれているため、このツールを使用するには、プロジェクトに SysConfig ファイルを用意する必要があります。ユーザーがすでにプロジェクト内に SysConfig ファイル (*.syscfg) を持っている場合は、既存のプロジェクトに SysConfig ファイル を追加する方法については C2000 SysConfig を参照してください。

1. CCS 内の SysConfig ファイルをダブルクリックして SysConfig を起動します
2. モジュールの *[MCU CONTROL CENTER AND TRANSFER] (MCU コントロール センターと転送)*セクションまでスクロールし、*MCU Signal Sight* モジュールの横にある (+) をクリックします

図 4-1. SysConfig モジュールを追加

3. (オプション) *[Name] (名前)* で GUI モジュール インスタンスの名前をカスタマイズします
4. *ターゲット バッファのサイズ*を入力します。これは、キャプチャしたデータをバッファするためにデバイス メモリ内で使用される浮動小数点配列のサイズを設定します。バッファ サイズを選択すると、データ キャプチャ速度が速くなります。

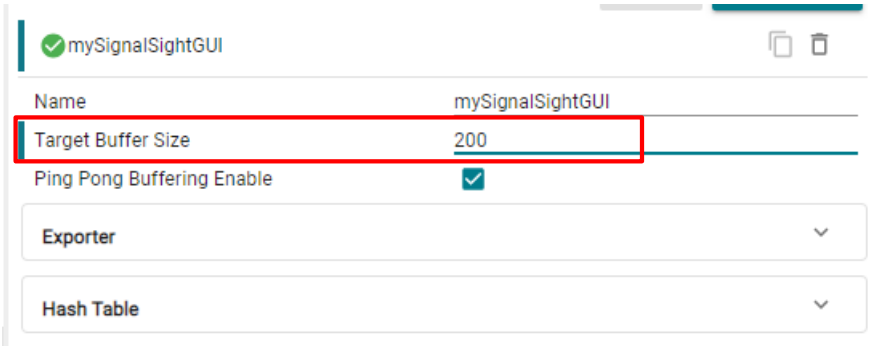

図 4-2. 入力バッファのサイズ

注

オンチップ メモリが限られているデバイスや、メモリ使用量の多いアプリケーションでは、大きなバッファ サイズを設定すると「*利用可能なメモリにプログラムが収まりません*」というビルド エラーが発生し、配列をデバイスの利用可能メモリに収めることができないことを意味します。ビルド エラーが解消されるまで、*ターゲット バッファ サイズ*の値を減らします。

5. *MCU 信号サイト* SysConfig モジュール内の *[Exporter] (エクスポータ)* モジュールのドロップダウン メニューを開きます
   a. (オプション) *[Name (名前)]*でパッケージ レイヤの名前を変更します
6. *[Exporter] (エクスポータ)* モジュール内の*[SCI Transfer Communication Link] (SCI 転送通信リンク)*ドロップダウン メニューを開きます
   a. (オプション)*[Name] (名前)*で通信レイヤの名前を変更します
   b. (オプション) アプリケーションで使用される SCI (UART) ボーレートを *[Baud Rate] (ボーレート)* で変更します。ボーレートを選択すると、データ キャプチャ レートが高速になります。

注

場合によっては、ボーレートが高いと、データの整合性が失われることがあります。このバージョンのツールは、921600 ビット/秒の最大ボーレートで徹底的にテストされています。より高いボーレートを選択する場合は注意が必要です。

7. *[Exporter] (エクスポータ)→[SCI Transfer Communication Link] (SCI 転送通信リンク)* サブモジュール内の PinMux サブモジュール ドロップダウン メニューを開きます

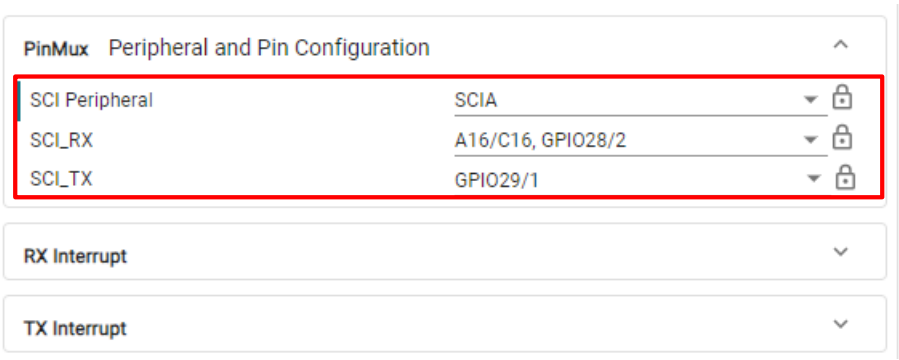

図 4-3. SCI GPIO を構成

   a. UART-USB ブリッジに接続するための TX ピンおよび RX GPIO ピンを選択します。LaunchPad または controlCARD を使用する場合は、プロジェクトにボード サポート (C2000™ SysConfig : [Board Support](基板サポート) を追加し、XDS に接続されているピンを選択します。

8. *[MCU Signal Sight]* SysConfig モジュール内の*[Hash Table] (ハッシュ テーブル)* モジュールのドロップダウン メニューを開きます
   a. (オプション) *Name (名前)*でハッシュ テーブルの名前を変更します

9. ユーザーのメイン アプリケーションで仮想オシロスコープへのプロットを有効にした各変数に対して、[Hash Table Element] (Hash テーブル要素) を (+) 追加します。各要素の *[Name] (名前)* を、アプリケーション内の変数名と一致するように変更します。

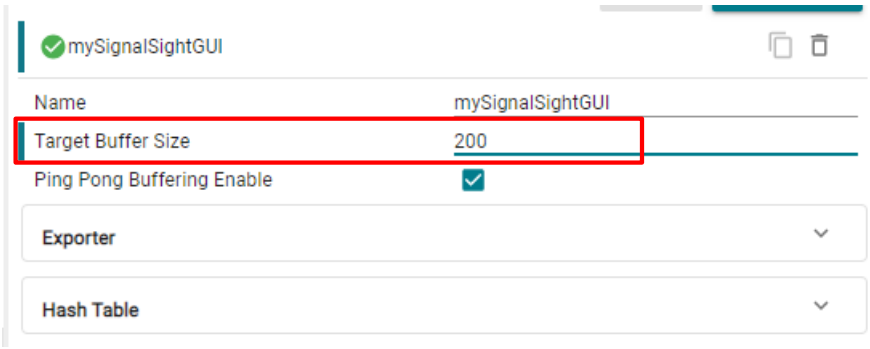

図 4-4. ハッシュ要素を追加し

注

変数名はアプリケーション変数名と完全に一致し、変数はグローバルな 32 ビット浮動小数点である必要があります。

4.2 ターゲット アプリケーションのステップ

SysConfig の選択が完了すると、SysConfig はユーザーのアプリケーション コード内で呼び出せるライブラリ関数を含むファイルを自動生成します。これらの関数を使用してツールをセットアップし、変数データをキャプチャ/バッファし、そのデータを GUI に送信してプロットできます。これらの機能をアプリケーションに実装する手順を以下に示します。

1. 標準の SysConfig board.h ヘッダ ファイルがメイン C ファイルの先頭でインクルードされており、初期化ルーチンの最後に *Board_init()* が呼び出されていることを確認してください。

```
#include "board.h"
```

```
Board_init();
```

2. さらに、*signalsight/signalsight.h* ヘッダ ファイルを含めます

```
#include <signalsight/signalsight.h>
```

```
//
// Included Files
//
#include <signalsight/signalsight.h>
#include "board.h"
```

図 4-5. ターゲット コードに含まれるもの

3. *Board_init()* を呼び出した後、アプリケーションの初期化コード内で *SIGNALSIGHT_init()* を追加で呼び出します。これは、グローバルに割り込みを有効にする前に実行される最後のコードにします。

```
SIGNALSIGHT_init();
```

```
//
// Main
//
void main(void)
{
    //
    // Initialize device clock and peripherals
    //
    Device_init();

    //
    // Disable pin locks and enable internal pull-ups.
    //
    Device_initGPIO();

    //
    // Initialize PIE and clear PIE registers. Disables CPU interrupts.
    //
    Interrupt_initModule();

    //
    // Initialize the PIE vector table with pointers to the shell Interrupt
    // Service Routines (ISR).
    //
    Interrupt_initVectorTable();

    //
    // PinMux and Peripheral Initialization
    //
    Board_init();

    //
    // Initialize Signal Sight tool state
    //
    SIGNALSIGHT_init();
```

図 4-6. ターゲット コードの初期化

4. プログラム内で、プロット変数の現在値を読み取り、その値をバッファに書き込む箇所を選択します。通常、これは PWM または CPU タイマ ISR で定期的に実行します。アプリケーションのこの領域に *SIGNALSIGHT_capturePlotData()*への関数呼び出しを追加します。

```
SIGNALSIGHT_capturePlotData();
```

```
//
// cpuTimer1ISR - CpuTimer1  ISR every 1 ms
//
__interrupt void
cpuTimer1ISR(void)
{

    //
    // Sample current values of the streaming variables and write them to a
    // temporary buffer
    //
    SIGNALSIGHT_capturePlotData();
```

図 4-7. データのキャプチャとバッファ

5. バッファ内のデータを送信するプログラム内の場所を選択します。通常、この処理は低優先度またはバックグラウンドループで行われます。これは、SCI モジュールがデータを送信するまでプログラムが待機状態になり、ブロックされる可能性があるためです。アプリケーションのこの領域に *SIGNALSIGHT_sendPlotData()*への関数呼び出しを追加します。

```
SIGNALSIGHT_sendPlotData();
```

```
//
// Enable Global Interrupt (INTM) and real time interrupt (DBGM)
//
EINT;
ERTM;

while(1)
{
    //
    // Send all streaming data from the temporary buffer when full
    //
    SIGNALSIGHT_sendPlotData();
}
```

図 4-8. バッファ付きデータを送信

4.3 CCS ステップ

1. SysConfig ツールは gui_setup.bat という bat ファイルを自動生成し、自動生成された GUI composer ファイルを CCS インストール内の正しい場所にコピーします。これにより MCU Signal Sight GUI を CCS メニューからプラグインとして起動できるようになります。プロジェクトをビルドするたびにこの bat ファイルを自動的に実行するには、以下の手順に従います:
   a. プロジェクト名を右クリックし、*[Properties] (プロパティ)* を選択します。
   b. *[General] (一般)* → *[Variables] (変数)*に移動します。
   c. (+) をクリックして *GUI_SUPPORT* という変数を追加し、値を 1 に設定します。*[Type] (タイプ)* は、String のままにできます。

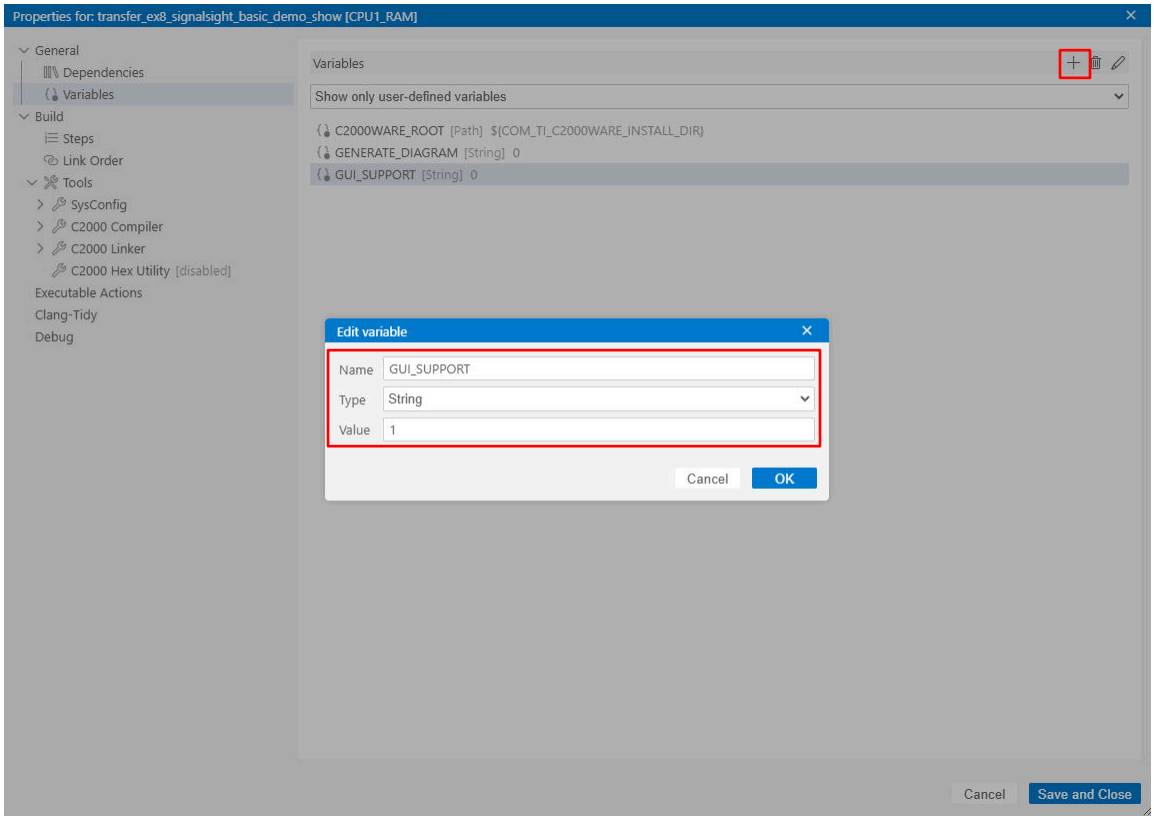

図 4-9. GUI_SUPPORT User_Defined 変数を追加

d. [Properties] (プロパティ) メニューの*[Build] (ビルド)* → *[Steps] (ステップ)*に移動します。

e. 次の行を含むエントリを追加します:

```
if ${GUI_SUPPORT} == 1 ${BuildDirectory}\syscfg\gui_setup.bat
```

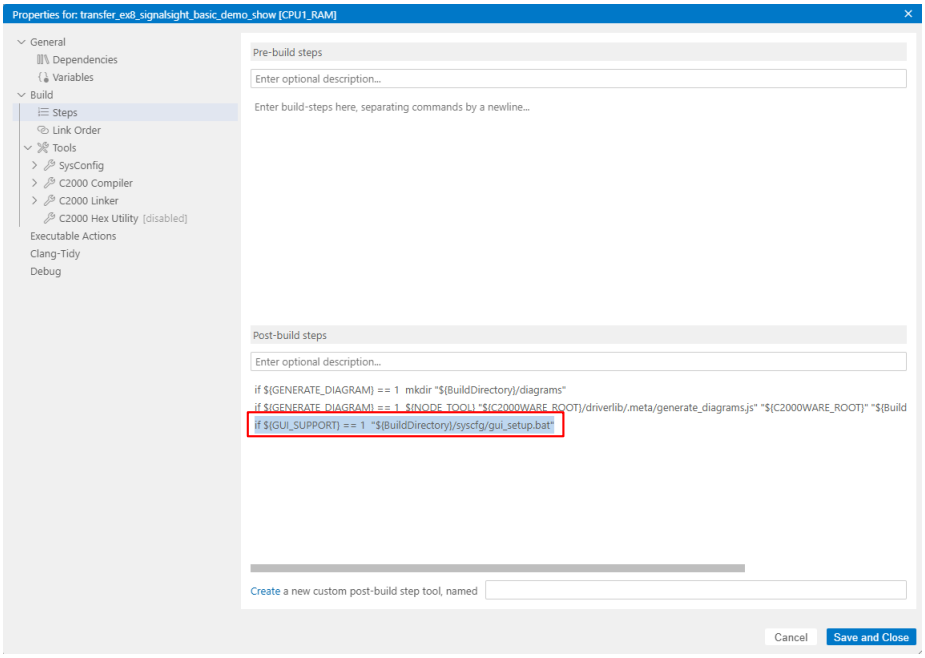

図 4-10. ビルド後のステップを追加

2. この時点で、ユーザーは MCU Signal Sight のサポートを追加した状態でプロジェクトをビルドする準備が整っています。CCS でプロジェクト名を右クリックし、*[Build Projects] (プロジェクトを追加)* を選択します。

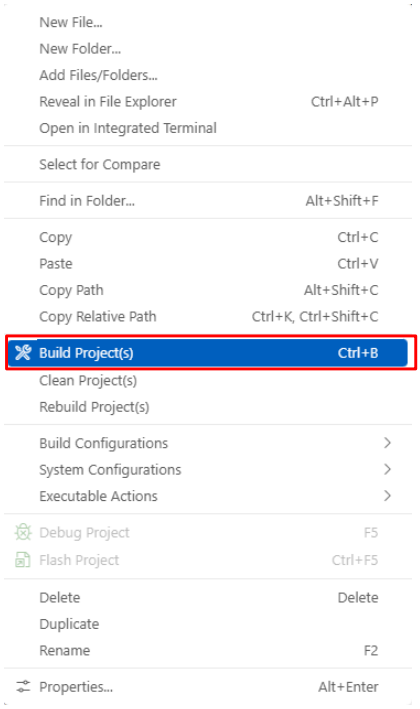

図 4-11. CCS プロジェクトを構築する

3. プロジェクトのビルドが完了したら、*[Help] (ヘルプ)* → *[Reload Window] (ウィンドウの再ロード)*に移動して CCS を更新します。これにより、CCS の*[View] (表示)* → *[Plugins] (プラグイン)*フォルダに入力されているプラグインが更新されます。

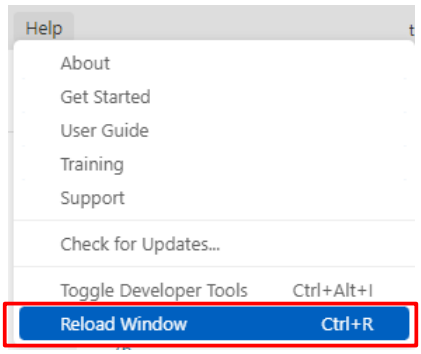

図 4-12. CCS ウィンドウを再ロード

4. アプリケーション コードをデバイスにロードするには、プロジェクトを右クリックして *[Debug Project] (プロジェクトをデバッグ)* を選択して CCS デバッグ セッションを開始するか、*[Run] (実行)* → *[Flash Project] (プロジェクトをフラッシュ)* を選択してデバッグ セッションを開始せずにプログラムをロードします。
5. *[View] (表示)* → *[Plugins] (プラグイン)* → {3 で設定された GUI 名} に移動して、MCU Signal Sight GUI を起動します。
6. *[SELECT COM PORT] (COM ポートの選択)* ボタンをクリックし、以下の手順に従います。
   a. *[Port (ポート)]*ドロップダウン メニューから、PC デバイス マネージャでユーザーのボードで使用される COM ポートと一致する COM ポートを選択します (PC の検索バーに*デバイス マネージャ*を入力します)。ほとんどの LaunchPad や controlCARD では、COM ポート番号は *[Ports (COM & LPT)] (ポート (COM & LPT))* → *[XDS110 Class Application/User UART] (COM#) (XDS110 クラス アプリケーション/ ユーザー UART (COM#))* の下に表示されます。

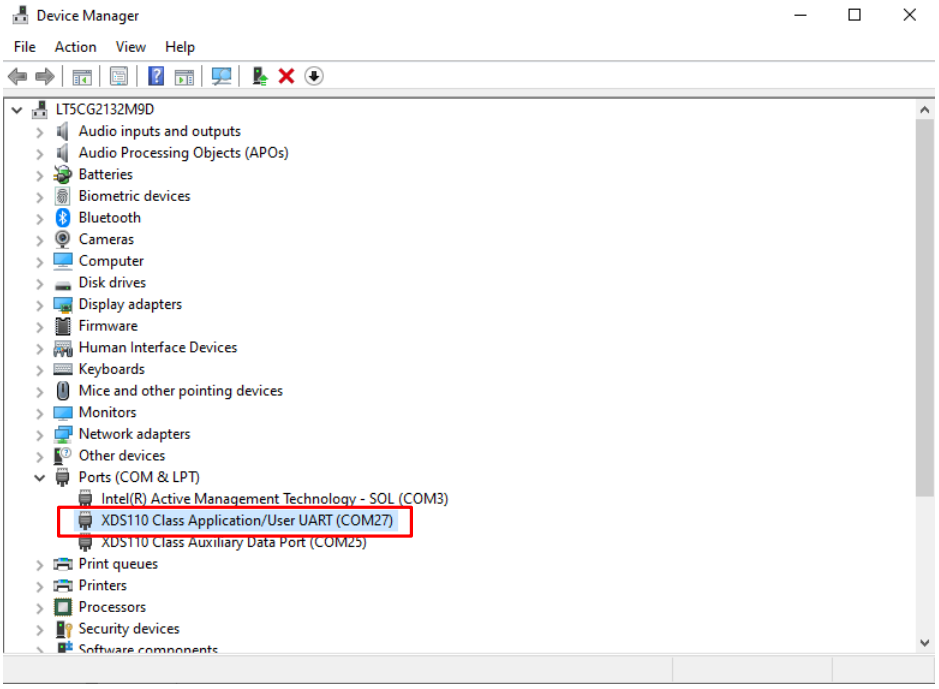

図 4-13. デバイス マネージャからの COM ポート番号

b. 2 で選択した入力と一致するボーレートを選択または入力します。
c. [OK] をクリックします。

7. GUI で *[CONNECT] (接続)* をクリックし、初期通信ハンドシェイクが行われるまで数秒待ちます。
8. チャネルをオンにして、ウィンドウ内でデータのプロットが開始されることを確認します。

注

ウィンドウにプロットされるデータがない場合は、トリガ設定を確認し、コード内の変数値が実際に変化していることを確認します。

これらの手順に問題があるかどうかを、セクション 7 で確認します。

5 Signal Sight GUI の操作

Signal Sight は、標準的なオシロスコープの見慣れたインターフェイスを忠実に再現しつつ、さまざまな拡張機能や特長を備えた、ユーザー フレンドリーな GUI を提供します。以下のセクションでは、ツールの最も一般的に使用される機能を活用する手順について説明します。

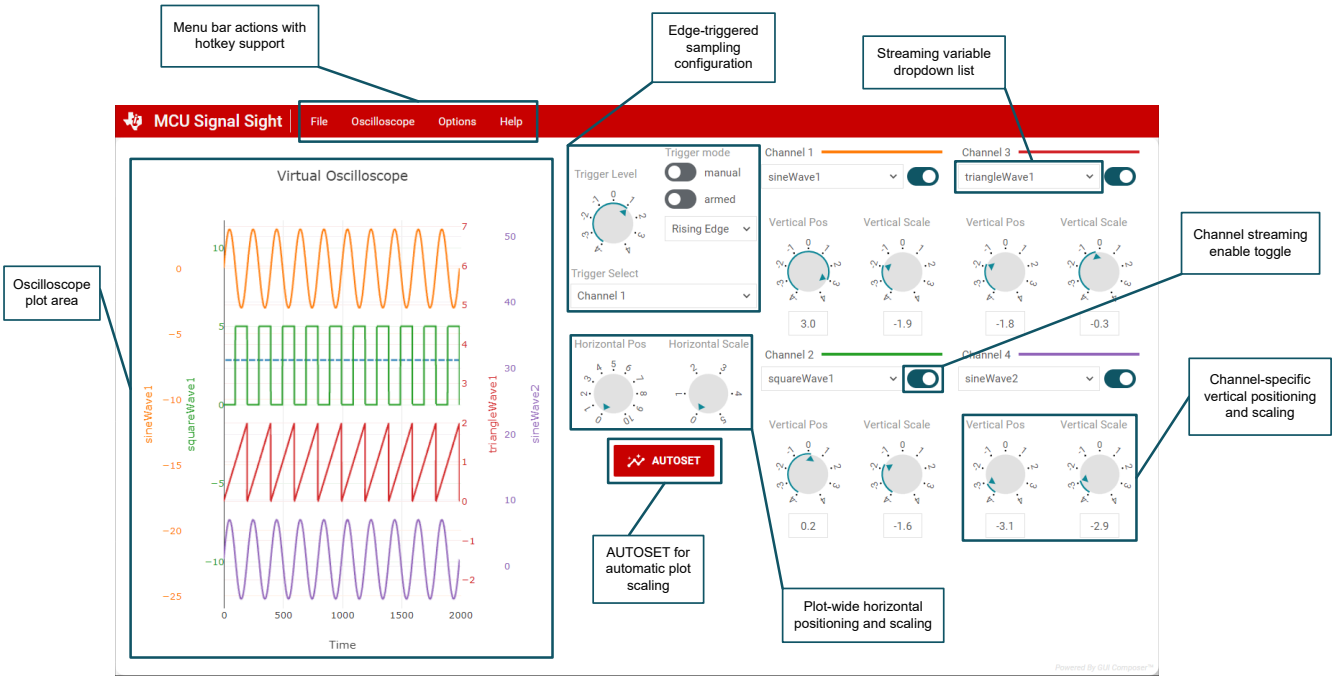

図 5-1. Signal Sight GUI インターフェイス

5.1 ターゲット接続を確認する

Signal Sight GUI を起動すると、ターゲット MCU デバイスへの接続方法を案内するモーダルが表示されます。デバッグ環境やツールの使用履歴によって、さまざまなメッセージが表示される場合があります。

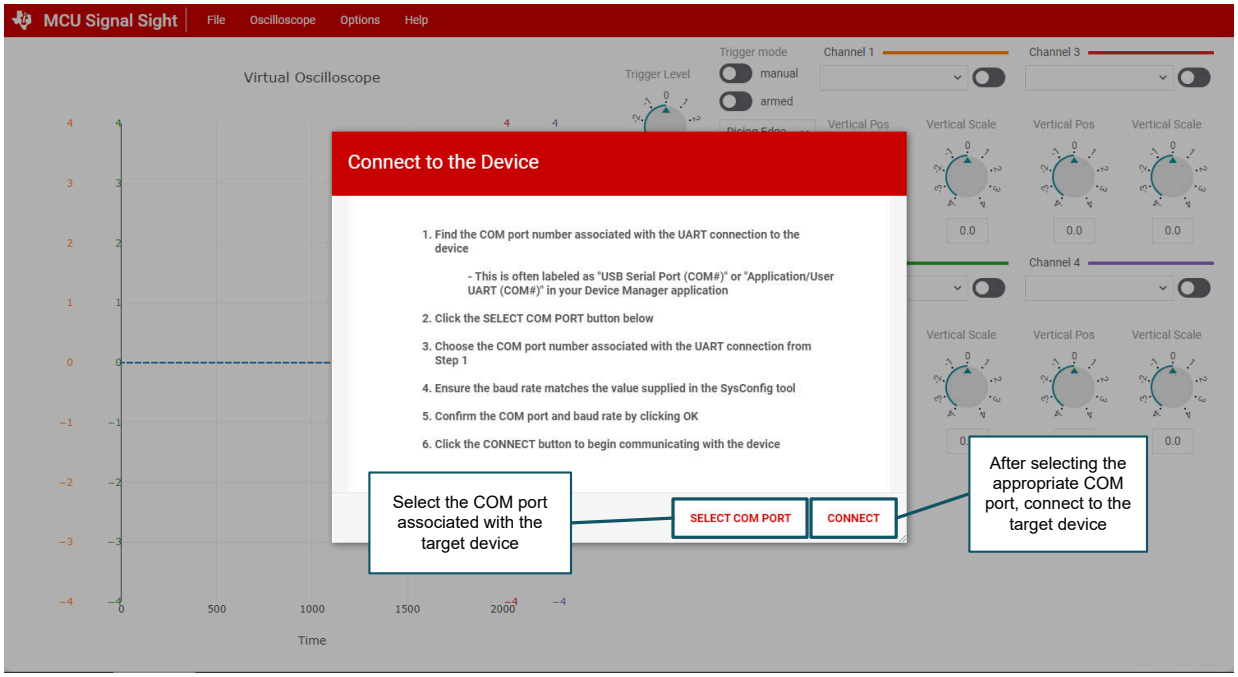

図 5-2. CCS 環境

CCS 環境内でデバッグする場合、ターゲットへの接続手順を説明するモーダルが表示されます。手順に従い、*[Connect] (接続)*をクリックしてターゲット デバイスとの接続を確立します。

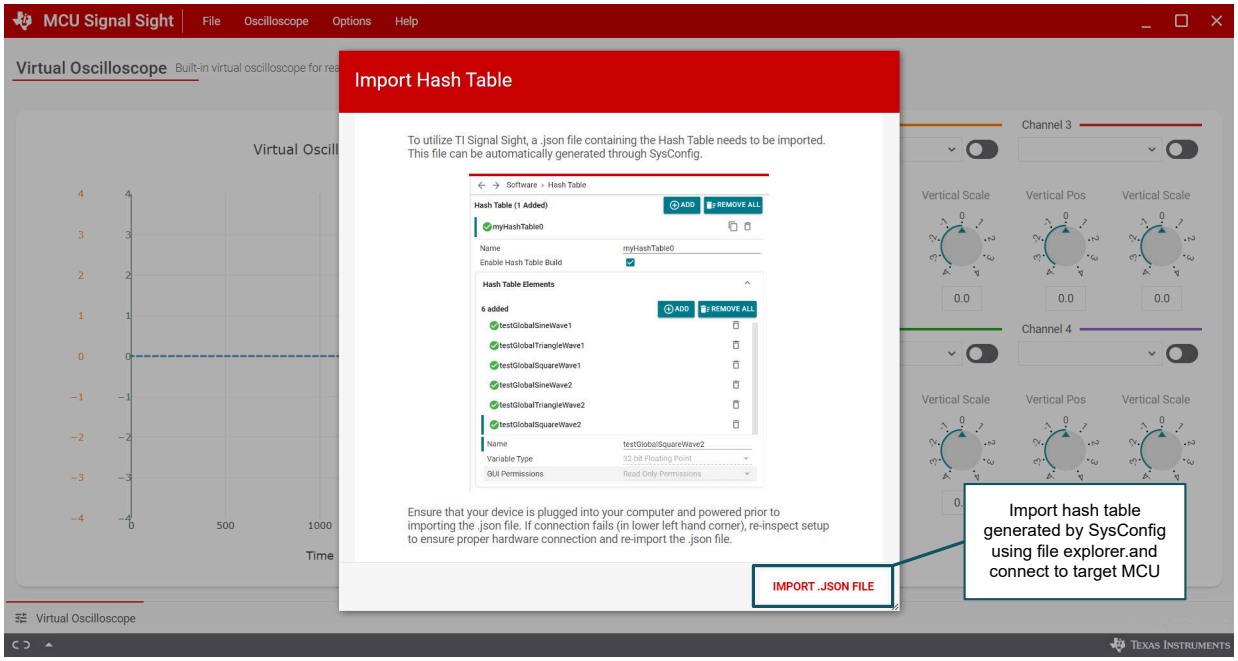

図 5-3. スタンドアロン GUI 環境

Signal Sight GUI をスタンドアロン環境で使用する場合は、SysConfig ツールによって生成されたハッシュ ファイルを外部から指定する必要があります。これを行うには、*IMPORT .JSON FILE* ボタンを選択し、ファイル エクスプローラ内を移動します。目的の .json ファイルをインポートします。インポートされると、GUI は接続されている MCU デバイスに接続要求を発行します。接続されると、モーダルは消えます。

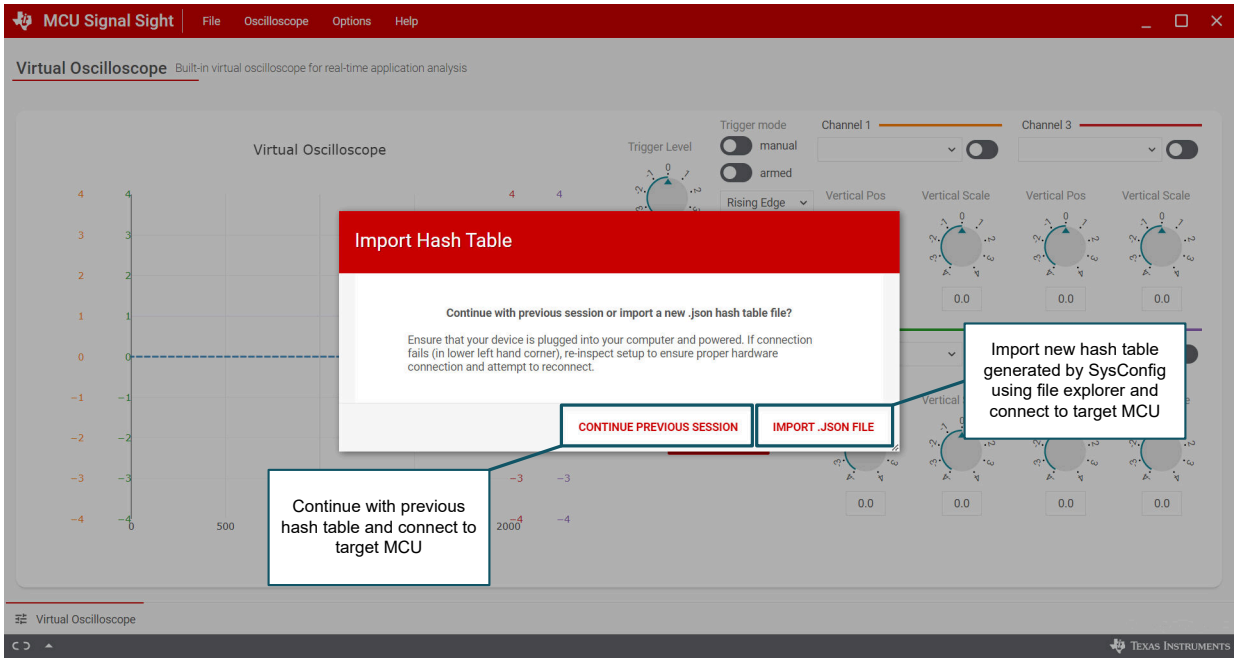

図 5-4. 前のセッションを復元する

ツールを再度使用する際、モーダルはユーザーに以前の Signal Sight セッションを続行するかどうかを促します。前のセッションを続行する場合は、*[CONTINUE PREVIOUS SESSION] (前のセッションを続行)*ボタンをクリックすると、以前に

使用したハッシュ テーブルがロードされ、デバイスへの接続が試行されます。新しいプロジェクトを使用してデバッグする場合は、*[IMPORT .JSON FILE] (.JSON ファイルをインポート)*をクリックすると、ハッシュテーブル ファイルをインポートして、ターゲットの MCU に接続するように求めるプロンプトが表示されます。

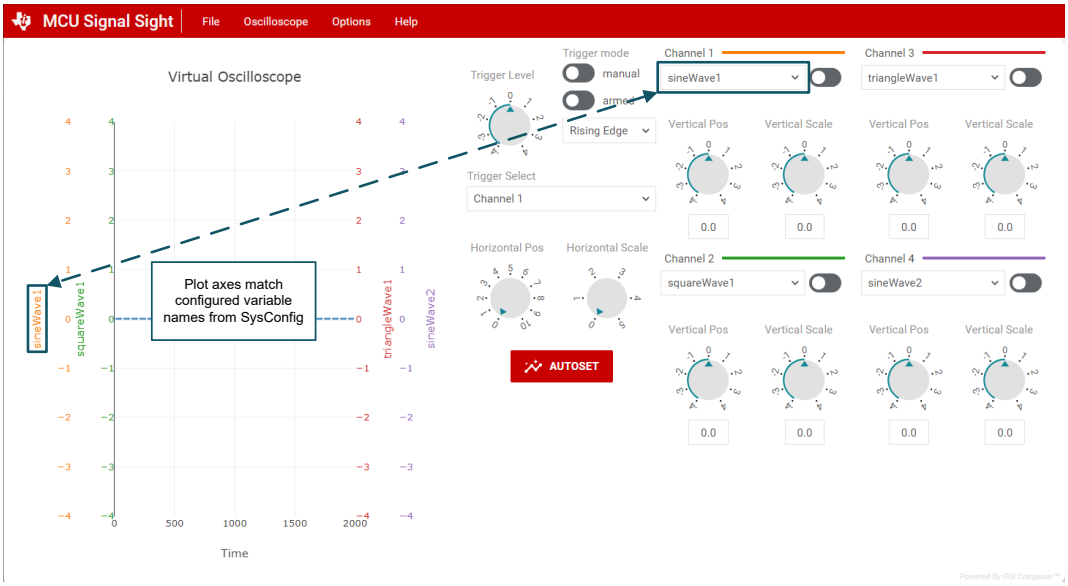

図 5-5. Signal Sight インターフェイス

Signal Sight ツールがターゲット MCU に正常に接続されると、オシロスコープ インターフェイスが表示されます。ハッシュ テーブルの最初の 4 つのストリーミング変数が、自動的にチャネル 1～4 のドロップダウン メニューに反映されます。ストリーミング変数は、グラフ上のプロット軸にも表示されます。

5.2 データ ストリーミングを有効にする

データ ストリーミング要求は、PC からターゲット MCU へ直接送信されます。データ ストリーミングを開始するには、表示したい入力チャネルに対応するスイッチをクリックしてください。ストリーミング中の特定の変数を切り替えるには、ドロップダウン メニューをクリックし、リストから目的のストリーミング変数を選択します。

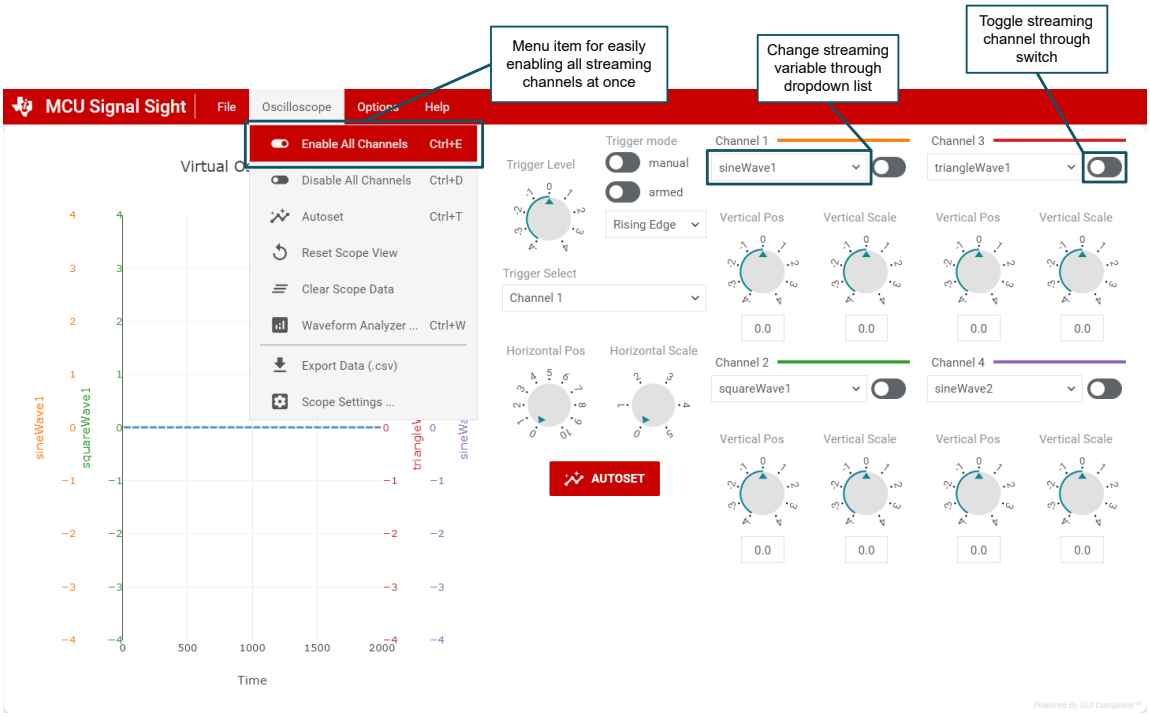

図 5-6. データ ストリーミングを有効にする

Signal Sight には、データ ストリームの開始と停止を行うメニュー アクションが用意されています。プロット上の 4 つのチャネルすべてをすばやくイネーブルにするには、*[Oscilloscope] (オシロスコープ) → [Enable All Channels] (すべてのチャネルをイネーブルにする)* をクリックします。これにより、トリガ モードが自動的に自動に設定され、連続ストリーミングがイネーブルになります。データ ストリーミングを無効にするには、*[Oscilloscope] (オシロスコープ) → [Disable All Channels] (すべてのチャネルを無効にする)*をクリックします。

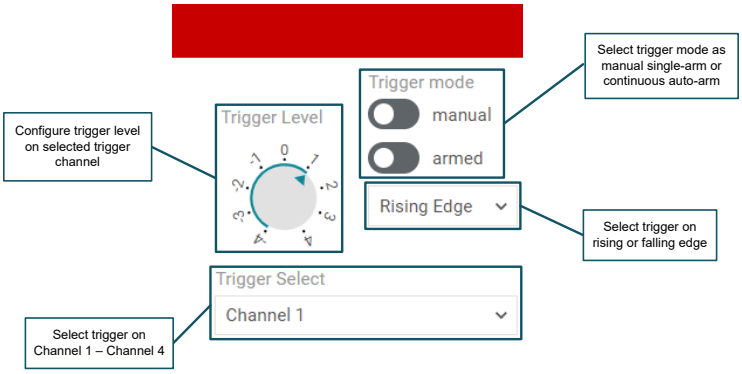

図 5-7. トリガ設定

物理的なオシロスコープと同様に、仮想オシロスコープのプロットはレベル トリガ方式です。つまり、入力データ ポイントがスコープのトリガ レベルと交差するまで、プロットにデータは表示されません。トリガ設定では、ユーザーはトリガ レベル、トリガ対象のチャネル、トリガ エッジ (立ち上がりまたは立ち下がり) などを構成できます。

さまざまなトリガ モードも使用できます。上のスイッチはトリガの繰り返しを制御し、下のスイッチはトリガのアーミングを制御します。上のスイッチがオフ位置 (左側) にある場合、スコープのトリガはマニュアル シングル アーム モードになります。このモードは、単一のデータ プロットをキャプチャするために使用されます。下のスイッチをオン (右側) にすると、スコープ トリガは入力データ ストリーム上のエッジを検出するようになります。エッジが検出されると、スコープ表示は入力データのプロットを開始します。プロットがデータで完全に満たされると、下部スイッチは自動的にオフになり、トリガがディスエーブルになります。

上のスイッチがオン位置 (右側) にある場合、スコープのトリガは連続アーム モードになります。このモードは、スコープを連続的に動作させ、長時間にわたってデータを表示するために使用されます。スコープ トリガは、受信したデータ ストリームを連続的にサンプリングし、トリガを手動で再アームすることなく、グラフにプロットします。このモードを無効にするには、上部スイッチをオフにします。

5.3 プロット表示の調整

MCU デバイスから PC にデータがストリーミングされると、データ ポイントがグラフに反映されます。

利便性のために *[Autoset] (自動セット)* ボタンが用意されており、すべてのスコープ チャネルの縦方向スケーリングに最適な設定を自動的に決定します。この [Autoset] (自動セット) ボタンは、メイン インターフェイスから、またはメニューバーの [Oscilloscope] (オシロスコープ) メニュー から利用できます。

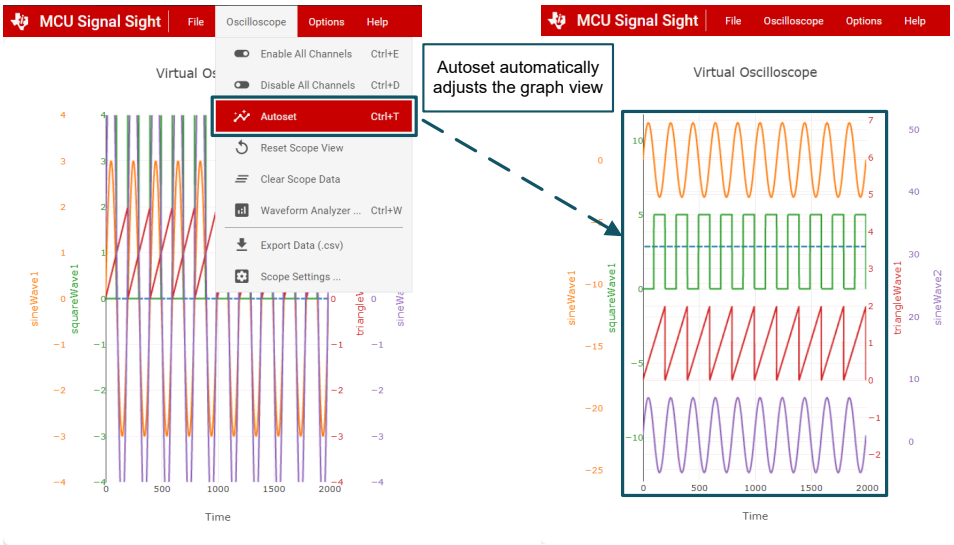

図 5-8. プロット表示の自動設定

Signal Sight インターフェイスには、プロットの表示を手動で調整するためのノブが備わっています。[Horizontal Pos] (水平位置ノブ) は、x 軸の位置を制御します。0 の場合、プロット表示には、ツールが受信したデータをトレースしたものが表示されます。10 の場合、プロットから消去されたバッファ データがプロット表示に表示されます。[Horizontal Scale] (水平スケール) ノブ は x 軸のスケーリングを調整し、グラフの一部を拡大表示する際に使用できます。スケーリングは対数です。0 では、フルバッファはプロットのフレーム内にあります。1 では、バッファの半分だけがプロットから見えます。2 では、バッファの 4 分の 1 のみが表示されます。

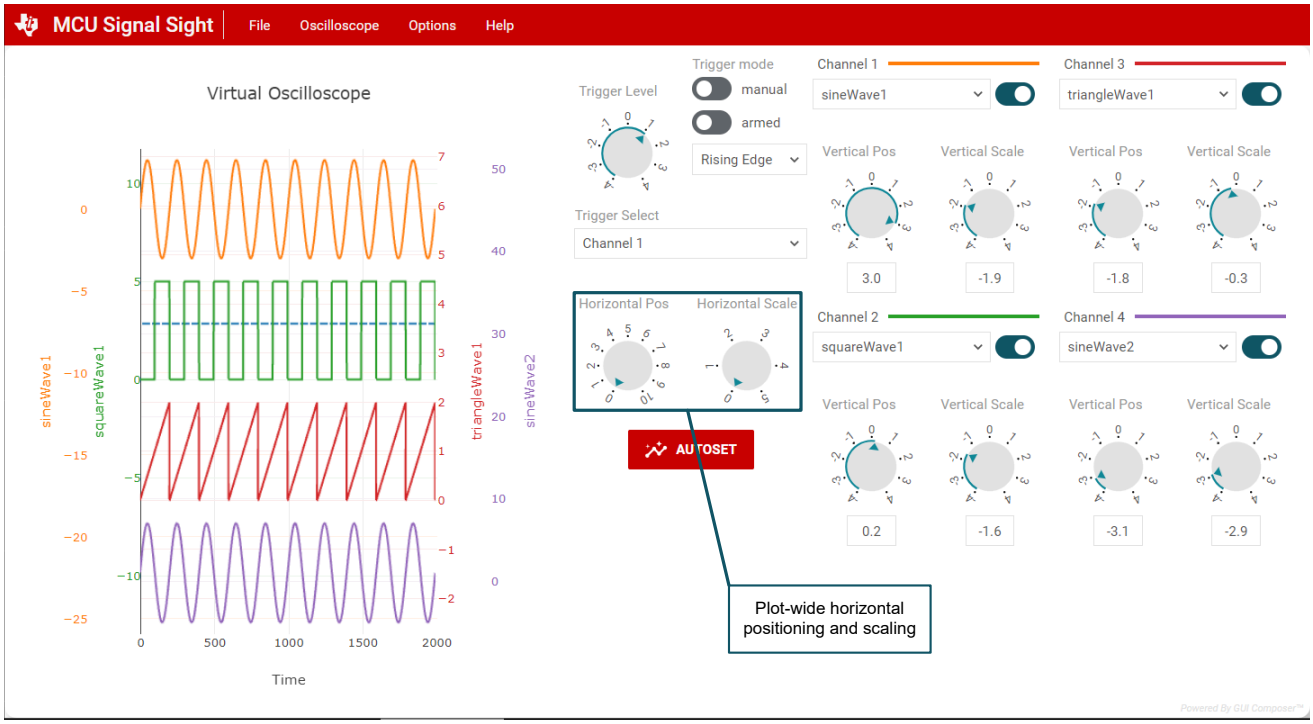

図 5-9. 水平ノブ

各入力チャネルには、個別の垂直位置決めノブと垂直スケーリング ノブがあります。これらのコントロールを使用すると、グラフ上の特定のプロット表示を調整できます。ノブは、前述の水平ノブと同様の機能を備えています。

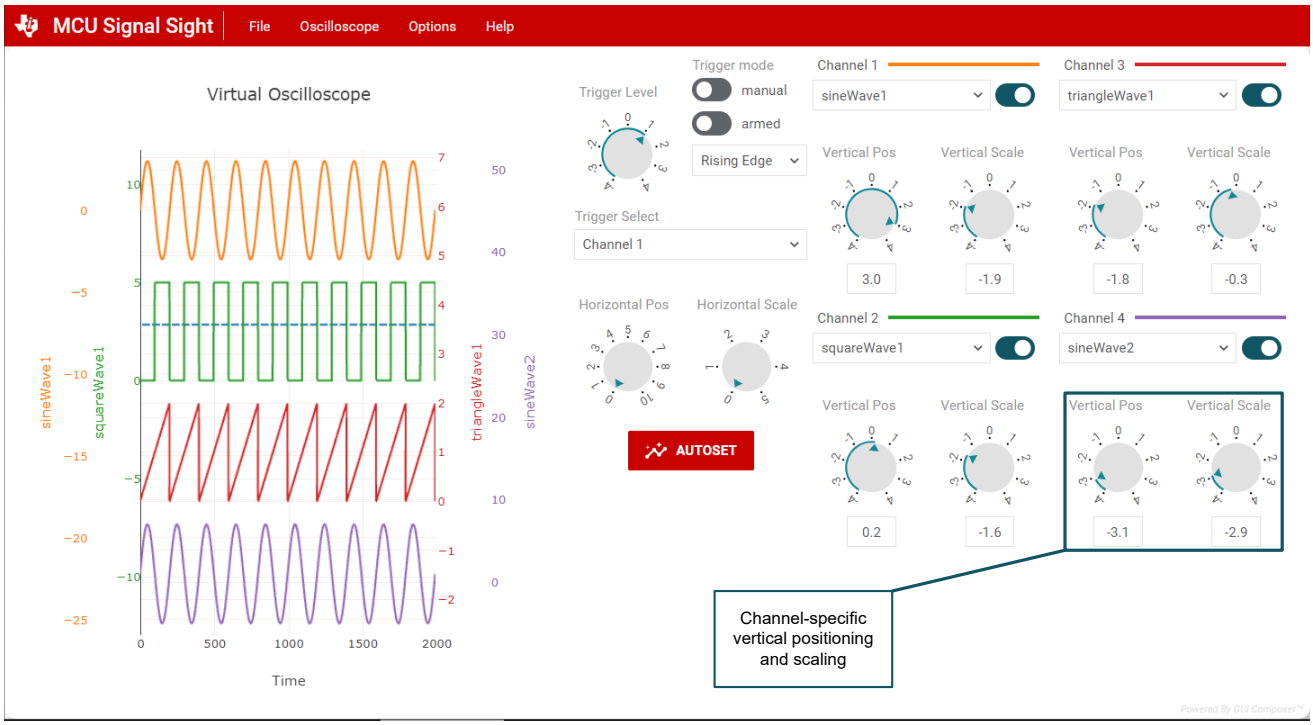

図 5-10. 垂直ノブ

スコープビューをデフォルトのビュー設定に簡単にリセットするには、*[Oscilloscope] (オシロスコープ) → [Reset Scope View] (スコープ表示のリセット)*をクリックします。

5.4 メニュー バー アクションとホットキー

Signal Sight GUI には、画面上部のメニュー バーからアクセスできる多数の機能やクイック アクションが備わっています。次の表に、使用可能なメニュー項目と機能を示します。

メニュー	メニュー項目	メニュー アクション	ホットキー
ファイル	ハッシュ テーブル JSON をインポート	ユーザーに別のハッシュ テーブルをインポートするよう促し、GUI を別のプロジェクトで使用できるようにします	
	最初からやり直します	GUI を再起動します	Ctrl + R
	終了	GUI を閉じます	Ctrl + X
オシロスコープ	すべてのチャネルをイネーブルにします	すべてのスコープ チャネルを切り替え、データのストリーミングを開始します	Ctrl + E
	すべてのチャネルを無効化します	すべてのスコープ チャネルをオフにします	Ctrl + D
	自動設定	すべてのスコープ チャネルの縦方向スケーリングに最適な設定を自動的に決定します	Ctrl + T
	スコープ表示をリセットします	スコープ プロットのスケーリングをデフォルト値に戻します	
	スコープ データをクリアします	スコープ プロット バッファ内のすべてのデータをクリアします	
	波形アナライザ	波形アナライザ メニューを開きます	Ctrl+W
	データのエクスポート	スコープ プロットに表示されているすべてのデータを .csv ファイルにエクスポートし、外部で解析できるようにします	
	スコープの設定	スコープ設定メニューを開きます	
オプション	シリアル ポート設定	シリアル ポート設定メニューを開きます	
	設定	Signal Sight 設定メニューを開きます	
ヘルプ	概要	アプリケーションに関する情報を表示します	

5.5 高度な機能

Signal Sight GUI には、基本的なデータ ストリーミング機能に加えて、デバッグを効率化するためのさまざまな追加機能が備わっています。

5.5.1 波形アナライザ

Signal Sight GUI には、波形アナライザと呼ばれる専用の信号分析ツールが搭載されています。このツールは、スコープ プロットに表示される入力データに対して、迅速な統計解析を提供します。波形アナライザには、最小データ ポイント、最大データ ポイント、実効値 (RMS) などの重要な情報が表示されます。Waveform Analyzer ツールにアクセスするには、*[Oscilloscope] (オシロスコープ) → [Waveform Analyzer] (波型アナライザ)*をクリックします。

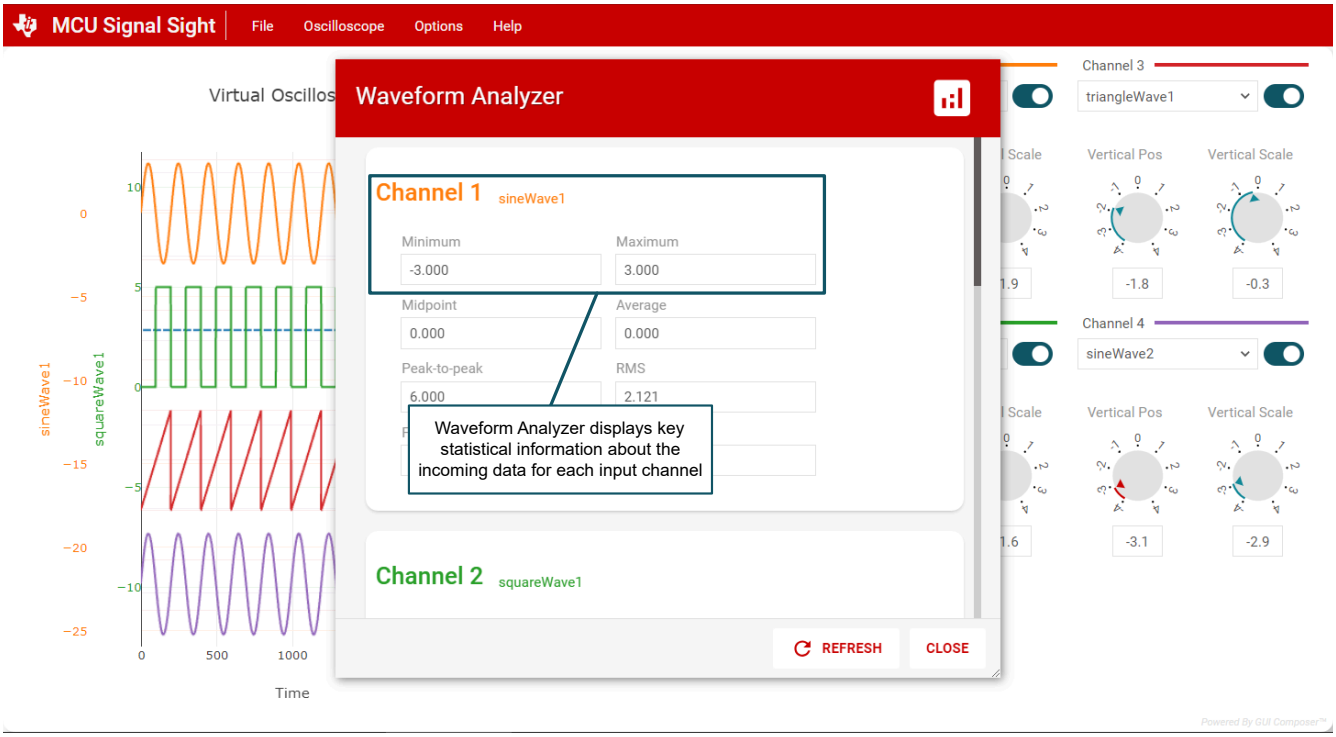

図 5-11. 波形アナライザ ツール

5.5.2 スコープの設定

仮想オシロスコープのプロットは、スコープ設定メニューで利用できるパラメータを調整することで、標準ビューからさらにカスタマイズできます。スコープ設定メニューにアクセスするには、*[Oscilloscope] (オシロスコープ)* → *[Scope Settings] (スコープ設定)*をクリックします。このメニューには、バッファ サイズとサンプリング レートを変更するための設定があります。これにより、ユーザーはチャネル ノブで許可されている設定を超えた垂直スケールを編集することもできます。これは、受信データの波形が大きすぎて、デフォルト設定でグラフに表示できない場合に便利です。

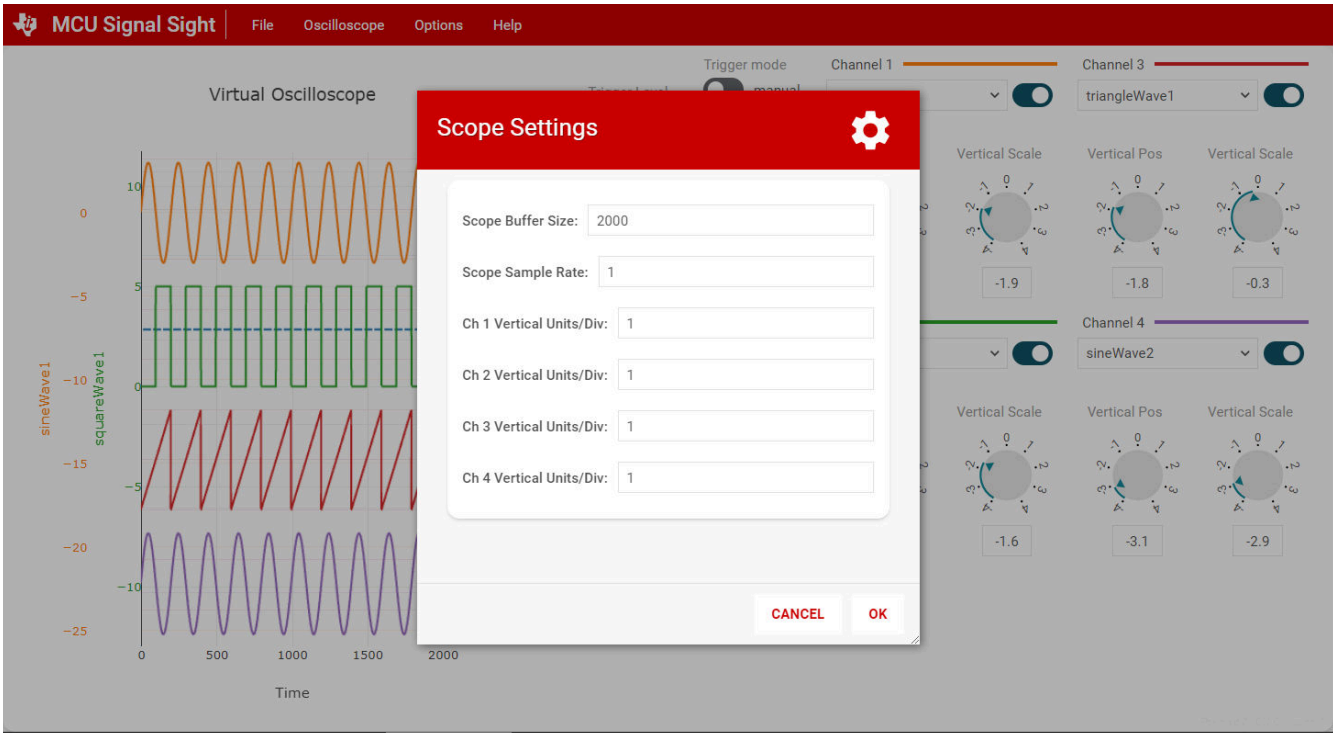

図 5-12. スコープ設定メニュー

プロット上でより多くのデータポイントを取得するには、スコープ設定メニューで、スコープ バッファ サイズ パラメータを編集します。これにより、スコープ表示がクリアされる前に取得できるデータポイント数を増やすことができます。

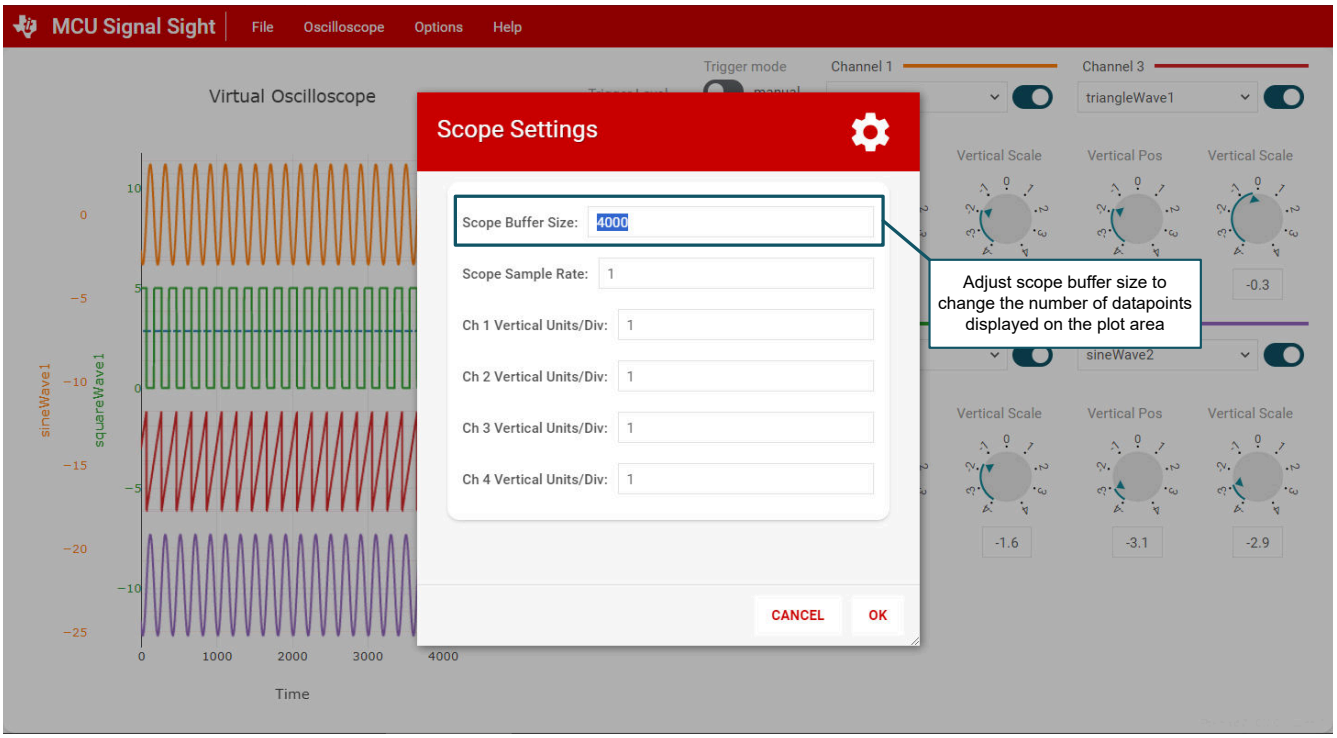

図 5-13. スコープ バッファ サイズの変更

5.5.3 データのエクスポート

ターゲット MCU から受信したデータポイントに簡単にアクセスできるよう、ユーザーはスコープ プロットに表示されているデータをダウンロードできます。データを.csv ファイルにエクスポートするには、*[Oscilloscope] (オシロスコープ) → [Export Data (.csv)] (データのエクスポート (.csv))*をクリックします。

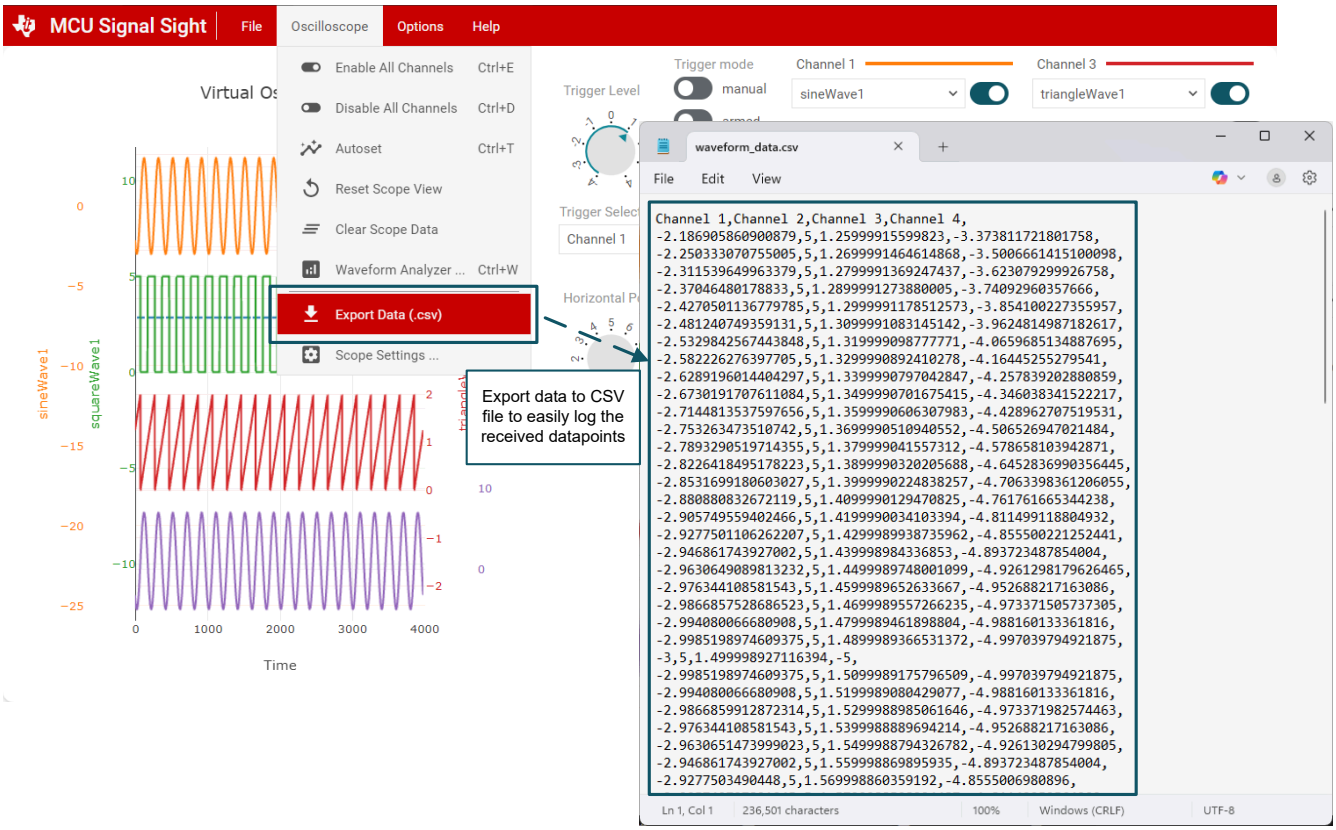

図 5-14. Signal Sight データの .csv ファイルへのエクスポート

6 ツールについて

プロット リフレッシュ レート

現在、このツールでサポートされている通信ペリフェラルは SCI (UART) のみであるため、プロット レートは使用する C2000 デバイスの最大 SCI ボーレートに依存します。最大 SCI ボーレートについては、関連するデバイスのデータシートを参照してください。

プロット変数

このツールの現行リビジョンでは、32 ビット浮動小数点型のグローバル変数のプロットのみがサポートされています。将来のリリースでは、符号付きまたは符号なしの型の変数や、構造体内の変数のサポートが予定されています。

7 トラブルシューティング ガイド

以下の FAQ を参照して、さまざまなトラブルシューティングのシナリオに対応するための提案をご覧ください。

- GUI 名は*[View] (表示)* → *[Plugins] (プラグイン)*フォルダに表示されません
  - チェック 1:プロジェクトの *[Properties] (プロパティ)* → *[General] (全般)* → *[Dependencies] (依存関係)* で、SysConfig と C2000Ware の製品が想定されたバージョンに設定されていることを確認してください。
  - チェック 2:プロジェクト*[Properties] (プロパティ)* → *[General] (一般)* → *[Variables] (変数)* →*{drop-down menu} (ユーザー定義変数とシステム変数の両方を表示)* → *SYSCONFIG_TOOL* 変数に移動します。現在の CCS インストール先 に、「sysconfig_cli.bat」ファイルへのパスが存在することを確認します。パスは、C:\ti\ccsxxxx\ccs\utils\sysconfig_x.xx.x\sysconfig_cli.bat のように表示する必要があります。
- あるチャネルをオンに切り替えた後、プロットにデータは表示されません
  - チェック 1:プロットされているデータの性質に基づき、トリガ チャネルとレベルが信号を通過するように設定されていることを確認し、トリガ モードを*手動*に設定します。
  - チェック 2:チャネルがオンに切り替えられたときに、SS_currentStreamState 変数が更新されていることを確認します。
    - CCS 経由でターゲット コードに接続します。
    - WATCH ウィンドウで、変数 SS_currentStreamState を追加します。
    - この変数の値が *NO_UPDATE* のままの場合、UART 通信に問題があります。
  - チェック 3:LaunchPad/controlCARD を使用する場合は、XDS/UART スイッチが正しい位置にあること、また SysConfig ステップ 7a の SysConfig SCI PinMux 選択で正しい GPIO が設定されていることを確認します。
  - チェック 4:GUI *[Options] (オプション)* → *[Serial Port Settings] (シリアル ポート設定)* の [Baud rate setting] (ボーレート設定) が、SysConfig ステップ 6b の *[MCU Signal Sight]* → *[Exporter] (エクスポータ)* → *[SCI Transfer Communication Link] (SCI 転送通信リンク)* → *[Baud Rate] (ボーレート)* で選択したボーレートと一致していることを確認します。
- プロットで、1 つ以上のチャネルのデータが欠落している、またはスキップされているように見えます
  - GUI には、バッファがいっぱいになったときにデータ サンプルのキャプチャをスキップする組み込み機能があります。
    - CCS 経由でターゲット コードに接続します (GUI で*「Connect」(接続)* をクリックする前に、必ずターゲット コードを実行してください)。
    - WATCH ウィンドウに、ping pong バッファリングが有効な場合は変数 SS_streamDataSkips と SS_streamDataSkips2 を追加します。
    - スキップ変数の値が大幅に増加している場合、ソフトウェアはデータ サンプルのキャプチャをスキップします。ユーザーは次のいずれかを実行します:
      - SysConfig ステップ 4 からバッファ サイズを増やします。これにより、バッファがいっぱいになる前に、より多くのデータを格納できるようになります。
      - アプリケーション コード内でデータ キャプチャの頻度を下げます (例: Step ターゲット ステップ 4 の CPU タイマ ISR の頻度を下げます)。これにより、SCI モジュールが新しいデータが追加される前にバッファを空にするための時間をより多く確保できます。
      - SysConfig ステップ 6b からボーレートを上げます。これにより、SCI がバッファからデータをより速く送信でき、その結果バッファをより早く空にできます。
- GUI プロットが小さすぎる、またはチャネル ノブや「Autoset」(自動設定) ボタンが画面に表示されていないように見えます
  - 仮想オシロスコープ機能は、画面のスケールを 100% に設定した状態で表示するように設計されています。ユーザーは PC の設定でこれを変更できます。Windows の場合、このオプションは*[Setting] (設定)* → *[Display] (ディスプレイ)* → *[Scale and Layout] (拡大縮小とレイアウト)* にあります。
- データ プロットが予期せず中断されました
  - *[Options] (オプション)* → *[Serial Port Settings] (シリアル ポート設定)* をクリックし、*[Connect] (接続)* を選択してターゲットに再接続します。これにより、ターゲット側のストリーミング状態がリセットされます。
- ツールに見つかったバグについては、E2E フォーラムに投稿します。

8 まとめ

MCU Signal Sight GUI を既存の C2000 組み込みプロジェクトにシームレスに統合することで、アプリケーション内の信号を可視化し、デバッグ プロセスを効率化できます。データのバッファリングと送信を分離することで、MCU Signal Sight のサポートは CPU パフォーマンスを損なうことなく、データのリアルタイム キャプチャを可能にします。高価なハードウェアを必要とし、信号の可視性が限られる従来のデバッグ手法や、リアルタイム アプリケーションには精度が不足する既存のソフトウェア実装とは異なり、MCU Signal Sight は包括的なソフトウェア ベースの実装を提供します。

9 参考資料

- テキサス インスツルメンツ、『MCU 制御センター ツール開発者ガイド』アプリケーション ノート
- テキサス インスツルメンツ、『C2000 リアルタイム マイコン ペリフェラル』データシート
- テキサス インスツルメンツ、『C2000 SysConfig』アプリケーション ノート
- テキサス インスツルメンツ、『DLT 開発者ガイド — ツール付き』アプリケーション ノート

重要なお知らせと免責事項

テキサス・インスツルメンツは、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、テキサス・インスツルメンツ製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した テキサス・インスツルメンツ製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとします。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている テキサス・インスツルメンツ製品を使用するアプリケーションの開発の目的でのみ、テキサス・インスツルメンツはその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。テキサス・インスツルメンツや第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、テキサス・インスツルメンツおよびその代理人を完全に補償するものとし、テキサス・インスツルメンツは一切の責任を拒否します。

テキサス・インスツルメンツの製品は、テキサス・インスツルメンツの販売条件、または ti.com やかかる テキサス・インスツルメンツ製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。テキサス・インスツルメンツがこれらのリソースを提供することは、適用されるテキサス・インスツルメンツの保証または他の保証の放棄の拡大や変更を意味するものではありません。

お客様がいかなる追加条項または代替条項を提案した場合でも、テキサス・インスツルメンツはそれらに異議を唱え、拒否します。

郵送先住所:Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

重要なお知らせと免責事項

テキサス・インスツルメンツは、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、 テキサス・インスツルメンツ製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した テキサス・インスツルメンツ製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとします。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている テキサス・インスツルメンツ製品を使用するアプリケーションの開発の目的でのみ、 テキサス・インスツルメンツはその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。 テキサス・インスツルメンツや第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、 テキサス・インスツルメンツおよびその代理人を完全に補償するものとし、 テキサス・インスツルメンツは一切の責任を拒否します。

テキサス・インスツルメンツの製品は、 テキサス・インスツルメンツの販売条件、または ti.com やかかる テキサス・インスツルメンツ製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。 テキサス・インスツルメンツがこれらのリソースを提供することは、適用される テキサス・インスツルメンツの保証または他の保証の放棄の拡大や変更を意味するものではありません。

お客様がいかなる追加条項または代替条項を提案した場合でも、 テキサス・インスツルメンツはそれらに異議を唱え、拒否します。

郵送先住所：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265