Application Note

MCU Signal Sight 工具开发人员指南

Delaney Woodward, Peter Luong, Nima Eskandari

摘要

嵌入式应用的传统调试和评估方法通常需要昂贵的设备，并且只能对内部信号提供有限的可见性。当前的软件替代方案虽然成本更低，但无法提供准确评估实时控制应用所需的精度。这种分散的资源密集型调试环境对客户造成了障碍。MCU Signal Sight 工具提供软件驱动的高性能图形用户界面 (GUI)，为嵌入式工程师带来实时绘图和高级调试工具。MCU Signal Sight 可在熟悉的界面中实现精确、实时的信号可视化，具有数据导出和实时统计分析等强大功能。该工具由可轻松配置的轻量级软件驱动，可直接插入任何软件工程中。MCU Signal Sight 可无缝地集成到任何开发环境，作为独立工具集成在 Code Composer Studio™ (CCS) 内，也可通过 TI 的云工具在线提供。

内容

插图清单

商标

Code Composer Studio™ and LaunchPad™ are trademarks of Texas Instruments.
所有商标均为其各自所有者的财产。

1 简介

MCU Signal Sight 是一种图形用户界面 (GUI)，用于在虚拟示波器上绘制来自 C2000 微控制器(MCU) 应用程序的实时信号。SysConfig 工具用于自动生成在嵌入式应用程序内使用的自定义嵌入式软件库，以及具有相关绘图实用程序的自定义 GUI 应用程序。本应用手册探讨了 MCU Signal Sight GUI 的特性，详细介绍了如何在现有 CCS 工程中实现 MCU Signal Sight 支持，并概述了如何使用 C2000ware 中提供的 MCU Signal Sight 示例。

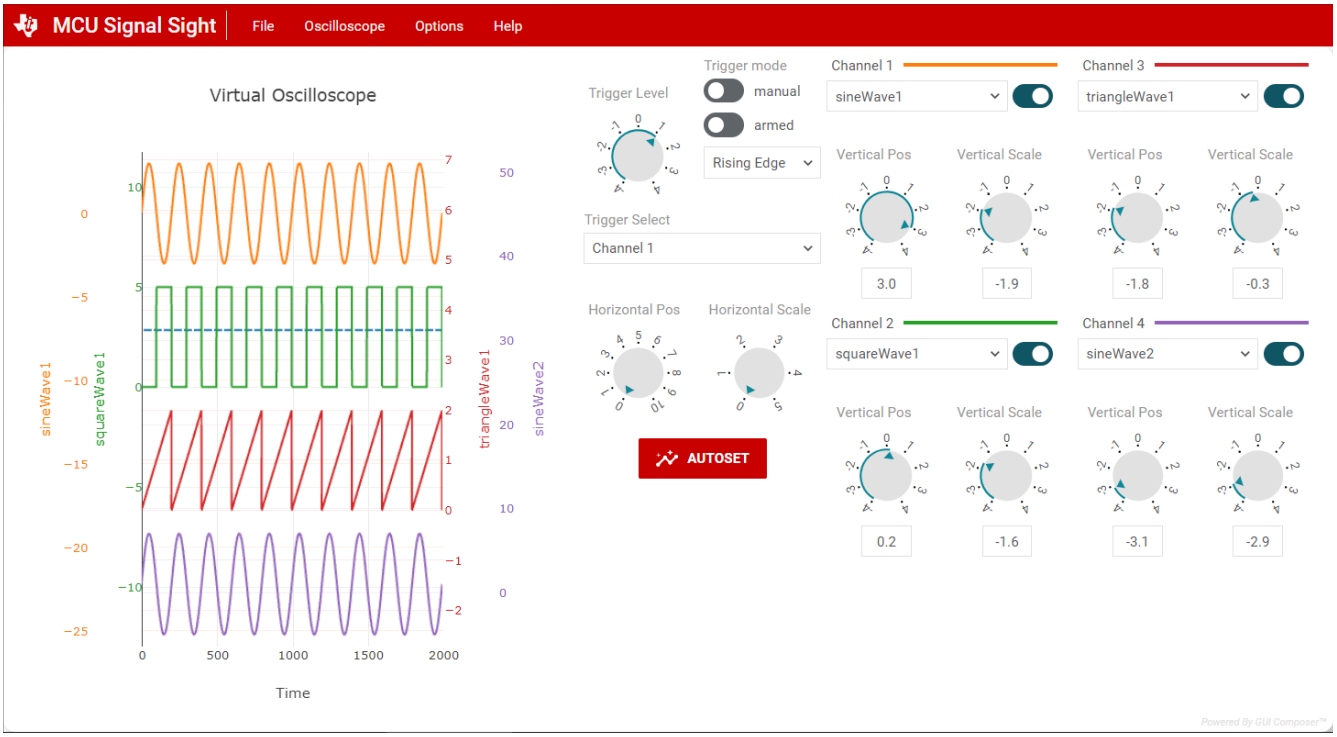

图 1-1. MCU Signal Sight GUI 工具图

2 特性

MCU Signal Sight 工具支持以下特性：

- 无需调试器连接即可实现 MCU 到 GUI 高速通信
  - 利用所有 C2000 LaunchPad™ 和 controlCARD 上的 UART 转 USB 桥接器
- 通过将 MCU 应用软件内部的信号和变量可视化来扩展示波器功能
- 多达四个同时绘制通道
- 根据 SysConfig 选择自动生成嵌入式和 GUI 代码
- 工具执行所需的 CPU 带宽极小
  - 启用应用程序非侵入式调试
- 增强了绘图信号分析
  - 手动和设防触发模式
  - 可自定义示波器视图、缓冲区大小和采样率设置
  - 波形分析仪
  - 自动设置绘图视图
- CSV 文件导出功能
  - 允许使用可解析格式的外部程序进行数据分析

2.1 软件设置

MCU Signal Sight 支持可以添加到任何具有 SysConfig 支持并使用 5.05.000 或更高版本 C2000ware 的现有 C2000 工程中。通过集成式 CCS 环境，可以完全生成 GUI 和必要的库文件并将其连接起来。此外，还可以下载独立版本的 MCU Signal Sight GUI，并独立于 CCS 安装运行。

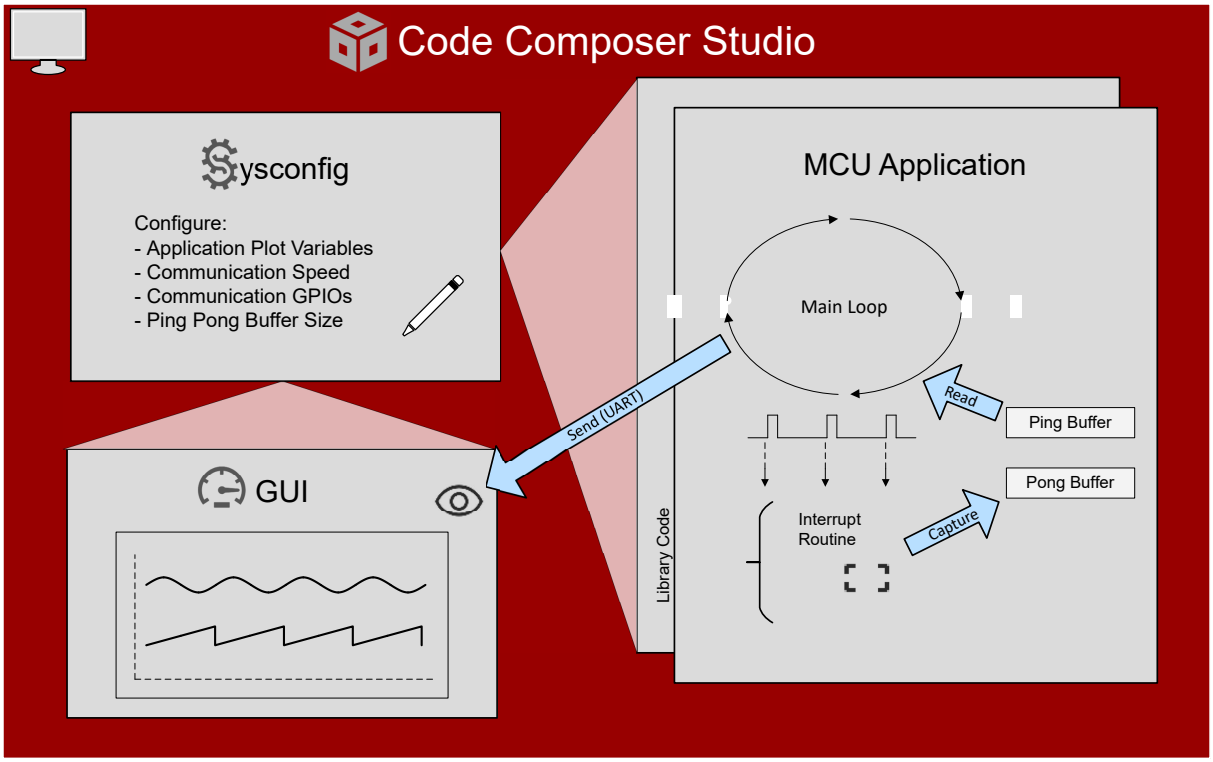

图 2-1. Signal Sight 软件图

在用户的 SysConfig (.syscfg) 文件中添加 MCU Signal Sight 模块后，SysConfig 工具会自动生成供目标应用程序使用的库文件，并通过虚拟示波器组件创建集成 GUI Composer 应用程序。

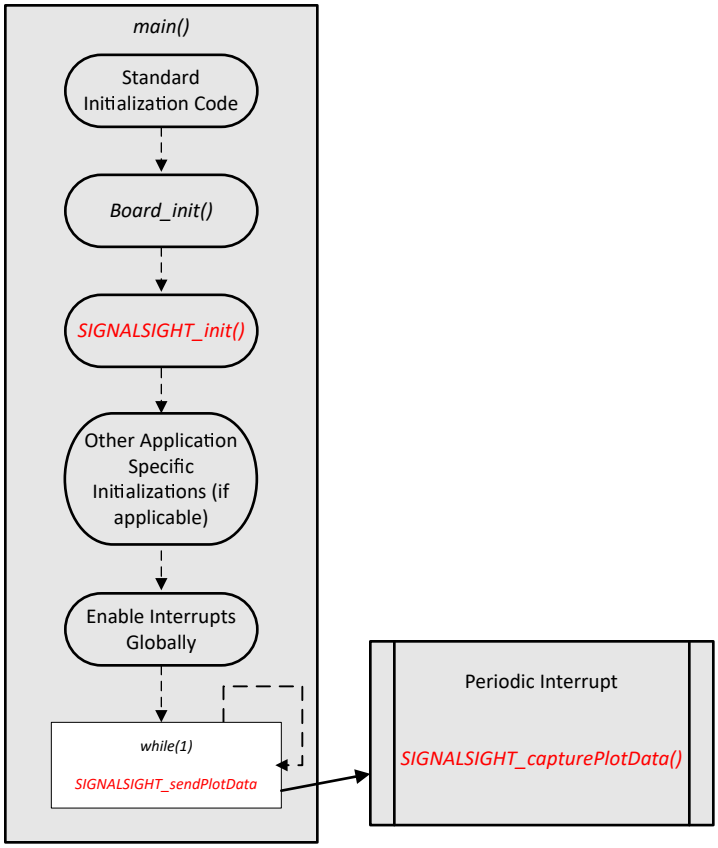

图 2-2. Signal Sight 软件流程图

库文件会在库文件内生成三个顶级函数供目标应用程序使用，如下文所述。要访问用户应用程序中的函数，请确保包含具有以下行的 signalsight.h 头文件： *#include <signalsight/signalsight.h>*

- *SIGNALSIGHT_INIT()* - Signal Sight 初始化函数
  - 在设备初始化、中断初始化和 SysConfig 初始化 (Board_init ()) 之后、但尚未全局启用中断之前，在初始化代码中调用此函数一次。
- *SIGNALSIGHT_sendPlotData()* - Signal Sight 捕获图数据函数
  - 在需要对数据进行采样的应用程序代码部分调用此函数。plot 变量的值保存在内存缓冲区中，以便在随后调用 SIGNALSIGHT_sendPlotData() 时传输。通常，这由周期性计时器或 ISR 中断调用。
- *SIGNALSIGHT_capturePlotData()* - Signal Sight 发送数据函数
  - 在应用程序中调用此函数，通过 SCI 外设将器件缓冲的数据发送至 GUI。建议将该函数放置在应用程序代码的低优先级位置（例如，代码主循环中），因为可以阻止 CPU 在此函数中等待，直至 SCI TX FIFO 中有足够的空间。

2.2 硬件设置

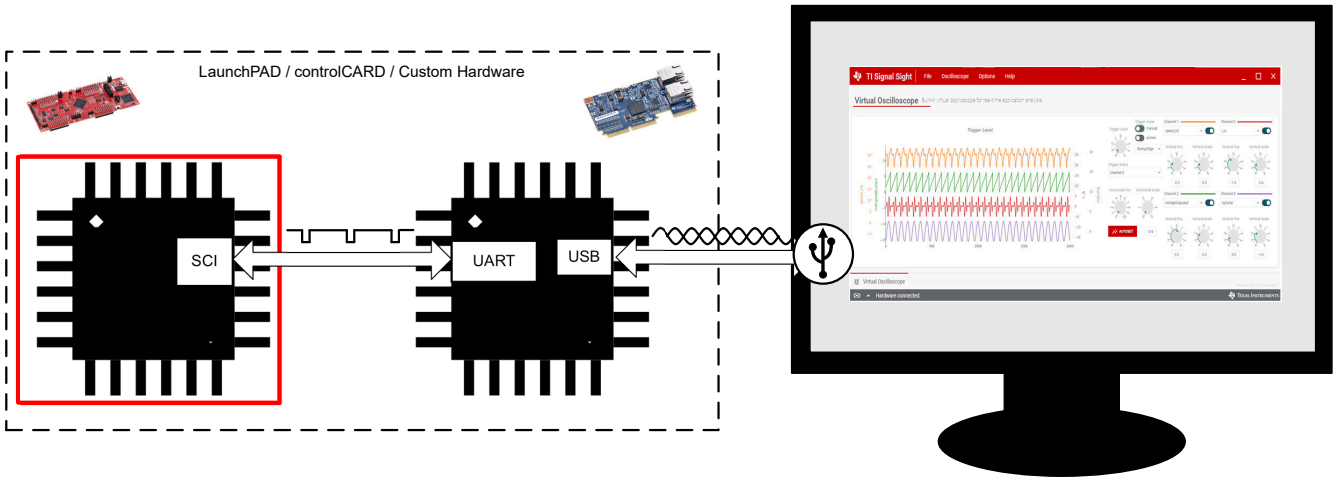

图 2-3. MCU Signal Sight 硬件图

使用 MCU Signal Sight 工具所需的唯一硬件是具有 SCI 外设的 C2000 器件，并使用某种形式的 UART 转 USB 桥接器与主机 PC 上的 GUI 应用程序进行通信。标准 C2000 LaunchPad 和 controlCARD 已包含板载桥接器件，在大多数情况下这些器件已更新了实现 UART 转 USB 桥接器的 XDS110 软件。在 MCU Signal Sight 硬件设置中，目标 MCU 利用板载 SCI（UART 协议）外设与 PC 上的 GUI 进行通信。

3 运行 C2000ware 示例

在 C2000ware SDK 中，提供了一个使用 MCU Signal Sight 工具的基本示例，名为 *transfer_ex8_signalsight_basic_demo*。运行此示例以验证硬件和软件设置有效，并了解该工具的功能。执行以下步骤：

1. 启动 CCS
2. 依次点击 *File → Import Projects → Browse*
3. 导航到以下 C2000WARE 安装路径中的示例：C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/transfer/ 文件夹。
4. 点击 *Finish*
5. 右键点击工程并点击 Properties
   a. 在 *General → Variables* 下，双击 *GUI_SUPPORT* 并将 *Value* 更改为 *1*
6. 右键点击工程，然后点击 *Debug Project*
7. 在 Debug 窗口中按下 Play
8. 导航至 *View → Plugins → myMCUSignalSight0*
9. 加载 GUI 后，即可点击 *Connect*
10. 在 GUI 窗口中，
    a. 点击 *Oscilloscope → Enable All Channels*
    b. 将 Trigger Mode 设置为 *manual*
11. 在 Signal Sight 界面窗口中查看正弦波、三角波和方波

请参阅 节 7，了解如何调试用户在执行这些步骤时可能遇到的任何问题。

4 向工程添加 Signal Sight

按照以下各节中的步骤，在 Signal Sight 虚拟示波器中绘制所需的应用程序变量。

4.1 SysConfig 步骤

MCU Signal Sight 工具内置在 Sysonfig GUI 中，因此在工程中必须有 SysConfig 文件才能使用该工具。如果用户在工程中已经有 SysConfig 文件 (*.syscfg)，请参阅 C2000 SysConfig 以了解如何将 SysConfig 文件添加到现有工程。

1. 双击 CCS 内的 SysConfig 文件以启动 SysConfig
2. 向下滚动至模块的 *MCU CONTROL CENTER AND TRANSFER* 部分，然后点击 *MCU Signal Sight* 模块旁边的 (+)

图 4-1. 添加 SysConfig 模块

3. （可选）在 *Name* 中自定义 GUI 模块实例的名称
4. 输入 *Target Buffer Size*。这将配置器件存储器中用于缓冲捕获数据的浮点数组的大小。选择较大的缓冲区大小可实现更快的数据采集速率。

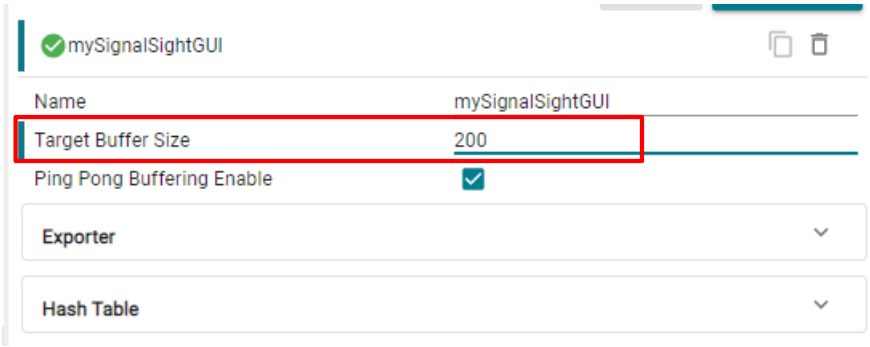

图 4-2. 输入缓冲区大小

备注

对于片上存储器受限的器件或存储器利用率较高的应用程序，较大的缓冲区大小可能会产生 *program does not fit into available memory...* 编译错误，表示数组无法容纳在可用器件存储器中。降低 *Target Buffer Size* 值，直到编译错误不再存在。

5. 打开 *MCU Signal Sight* SysConfig 模块内的 *Exporter* 模块下拉菜单
   a. （可选）在 *Name* 中修改封装层的名称
6. 打开 *Exporter* 模块内的 *SCI Transfer Communication Link* 下拉菜单
   a. （可选）在 *Name* 中修改通信层的名称
   b. （可选）在 *Baud Rate* 中修改应用程序使用的 SCI(UART) 波特率。选择较高的波特率可实现快速的数据采集。

备注

在有些情况下，高波特率可能会导致数据完整性丧失。此版本的工具已在 921600 位/秒的最大波特率下进行了全面测试。选择更高的波特率时都应小心。

7. 打开 *Exporter* → *SCI Transfer Communication Link* 子模块内的 PinMux 子模块下拉菜单

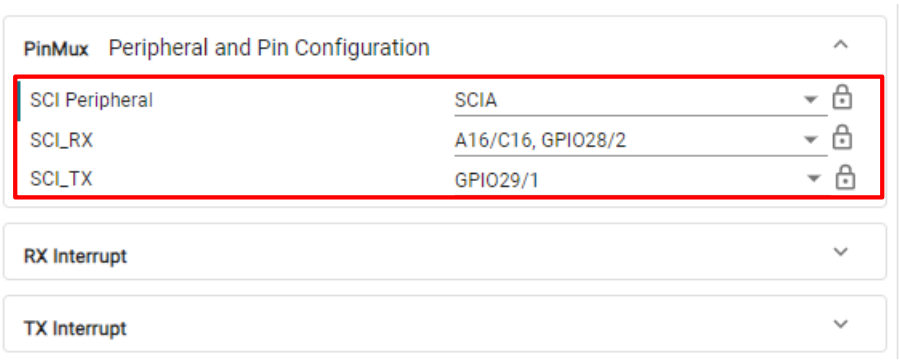

图 4-3. 配置 SCI GPIO

a. 选择要连接到 UART 转 USB 桥接器的所需 TX 和 RX GPIO 引脚。如果使用 LaunchPad 或 controlCARD，则向工程添加电路板支持（请参阅 C2000™ SysConfig：板支持），然后选择连接到 XDS 的引脚。

8. 打开 *MCU Signal Sight* SysConfig 模块内的 *Hash Table* 模块下拉菜单
   a. （可选）在 *Name* 中修改哈希表的名称
9. 为用户主应用程序中所启用的用于在虚拟示波器中绘图的每个变量添加 (+) 一个哈希表元素。修改每个元素的 *Name* 以匹配应用程序中的变量名称。

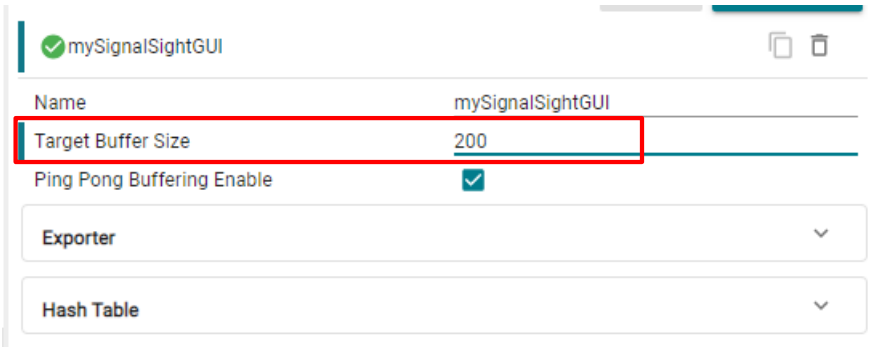

图 4-4. 添加哈希元素

备注

变量名称必须与应用程序变量名称完全匹配，并且变量必须是全局 32 位浮点。

4.2 目标应用程序步骤

现在已经完成了 SysConfig 选择，SysConfig 已自动生成包含库函数的文件，这些库函数可以在用户应用程序代码中调用以设置工具、捕获/缓冲变量数据并将数据传输到待绘制的 GUI。下面是在应用程序中实现这些函数的步骤。

1. 验证标准 SysConfig "board.h" 头文件包含在主 C 文件的顶部，并且在初始化例程的末尾调用 *Board_init()*。

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

```
Board_init();
```

2. 此外，还应包含 *signalsight/signalsight.h* 头文件

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

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

图 4-5. 目标代码包含

3. 调用 *Board_init()* 后，在应用程序初始化代码中添加对 *SIGNALSIGHT_init()* 的另一个函数调用。在全局启用中断之前，以此作为最后执行的代码。

```
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();
```

图 4-6. 目标代码初始化

4. 选择在程序中的哪个位置读取绘图变量的当前值并将这些值写入缓冲区。通常，这在 PWM 或 CPU 计时器 ISR 中定期完成。在应用程序的这个区域中添加对 *SIGNALSIGHT_capturePlotData()* 的函数调用。

```
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();
```

图 4-7. 捕捉和缓冲数据

5. 选择在程序中的哪个位置传输缓冲区的数据。通常，这以低优先级或在后台循环中完成，因为程序会被阻止，等待 SCI 模块传输数据。在应用程序的这个区域中添加对 *SIGNALSIGHT_sendPlotData()* 的函数调用。

```
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();
}
```

图 4-8. 传输缓冲数据

4.3 CCS 步骤

1. SysConfig 工具自动生成一个批处理文件（名为 gui_setup.bat），它可将自动生成的 GUI Composer 文件复制到 CCS 安装中的正确位置，以使 MCU Signal Sight GUI 可以作为插件从 CCS 菜单中启动。若要在每次编译工程时自动运行这个批处理文件，请按照以下步骤操作：
   a. 右键点击工程名称并选择 *Properties*。
   b. 导航至 *General* → *Variables*。
   c. 点击 (+)，添加一个名为 *GUI_SUPPORT* 的变量，然后将值设置为 1。*Type* 可以保留为 String。

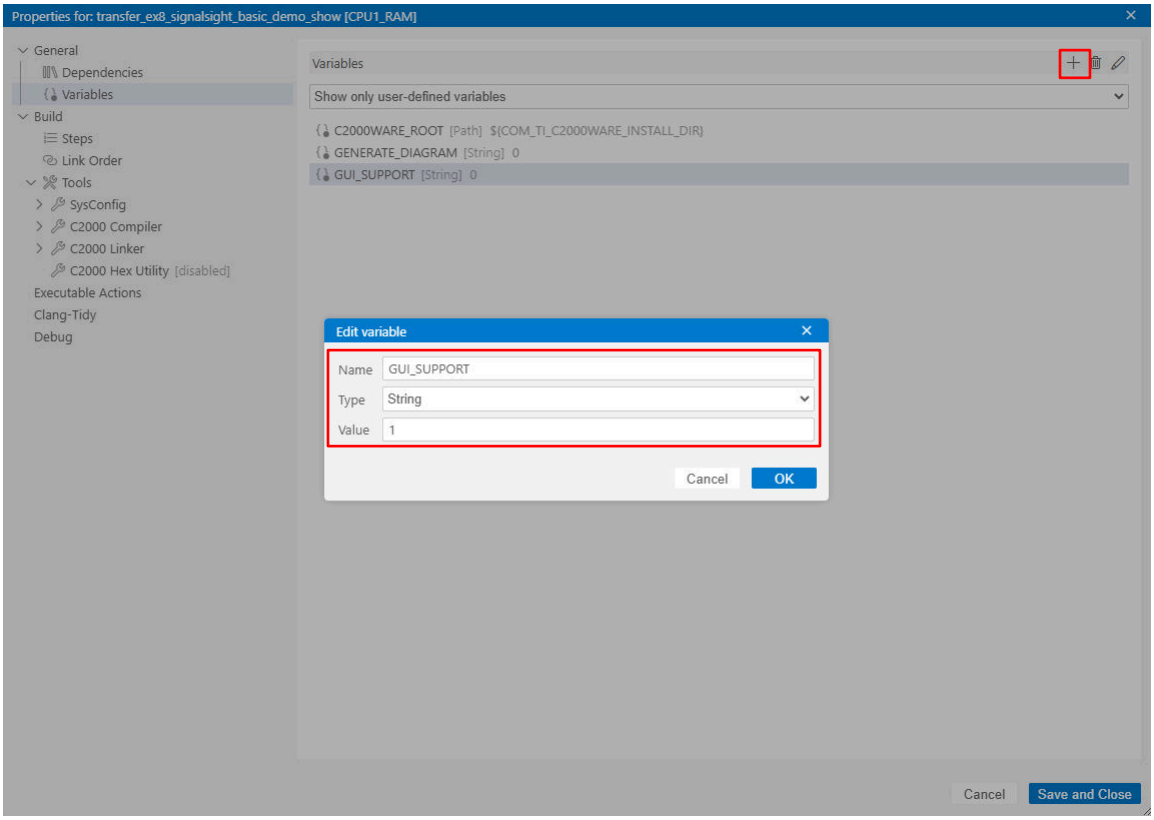

图 4-9. 添加 GUI_SUPPORT User_Defined 变量

   d. 在 Properties 菜单中导航至 *Build* → *Steps*。
   e. 添加包含以下行的条目：

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

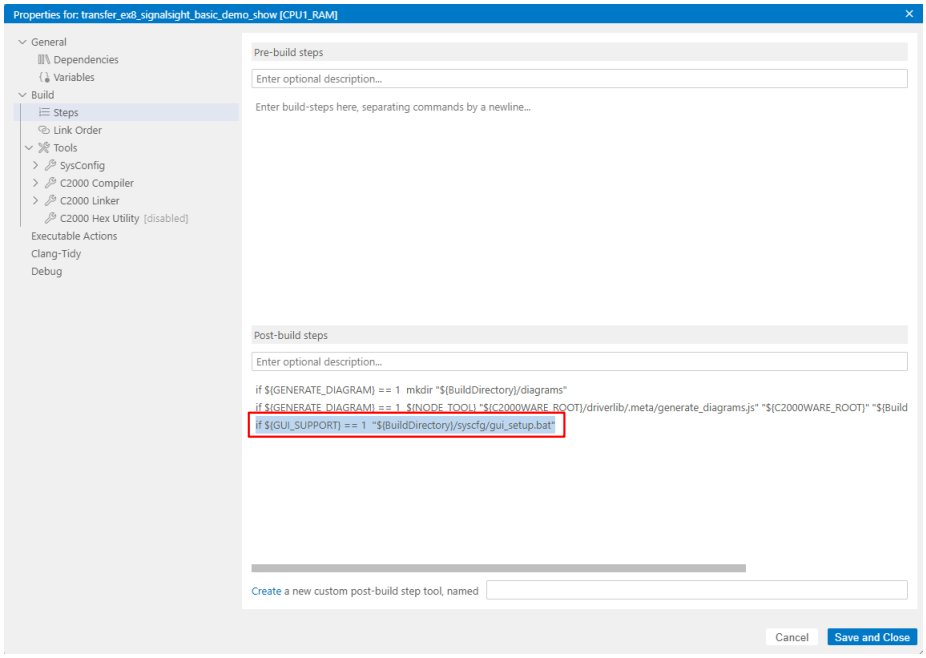

图 4-10. 添加编译后处理步骤

2. 此时，用户已准备好在添加了 MCU Signal Sight 支持的情况下编译工程。右键点击 CCS 中的工程名称，然后选择 *Build Projects*。

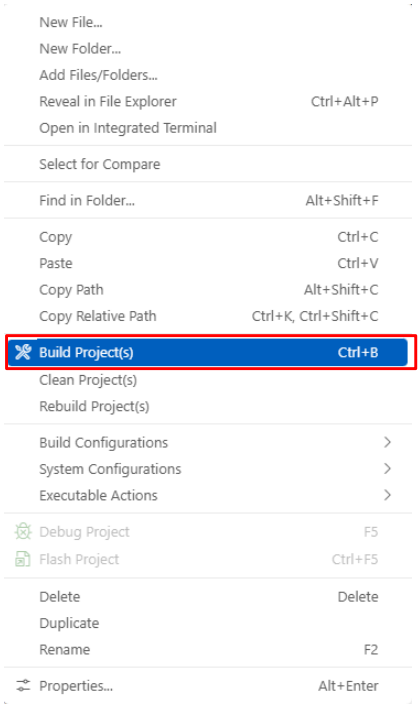

图 4-11. 编译 CCS 工程

3. 工程编译完成后，导航到 *Help → Reload Window* 以刷新 CCS。这会更新 CCS 的 *View → Plugins* 文件夹中填充的插件。

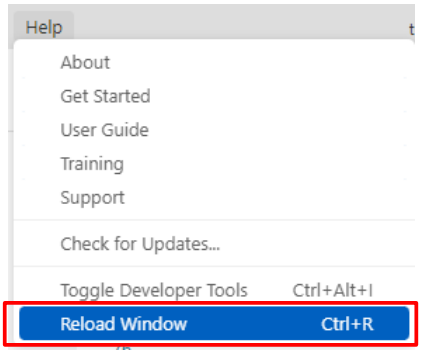

图 4-12. 重新加载 CCS 窗口

4. 通过右键点击 → *Debug Project*（启动 CCS 调试会话）或 *Run* → *Flash Project*（加载程序而不启动调试会话），将应用程序代码加载到器件。
5. 导航至 *View* → *Plugins* →{在 3 中设置的 GUI 名称} 以启动 MCU Signal Sight GUI。
6. 点击 *SELECT COM PORT* 按钮并按照以下步骤操作。
   a. 从 *Port* 下拉菜单中选择与 PC 设备管理器（在 PC 搜索栏中输入 *Device Manager*）中用户电路板使用的 COM 端口相匹配的 COM 端口。对于大多数 LaunchPad 和 controlCARD，COM 端口号位于名称 *(COM & LPT)* → *XDS110 Class Application/User UART (COM#)* 下。

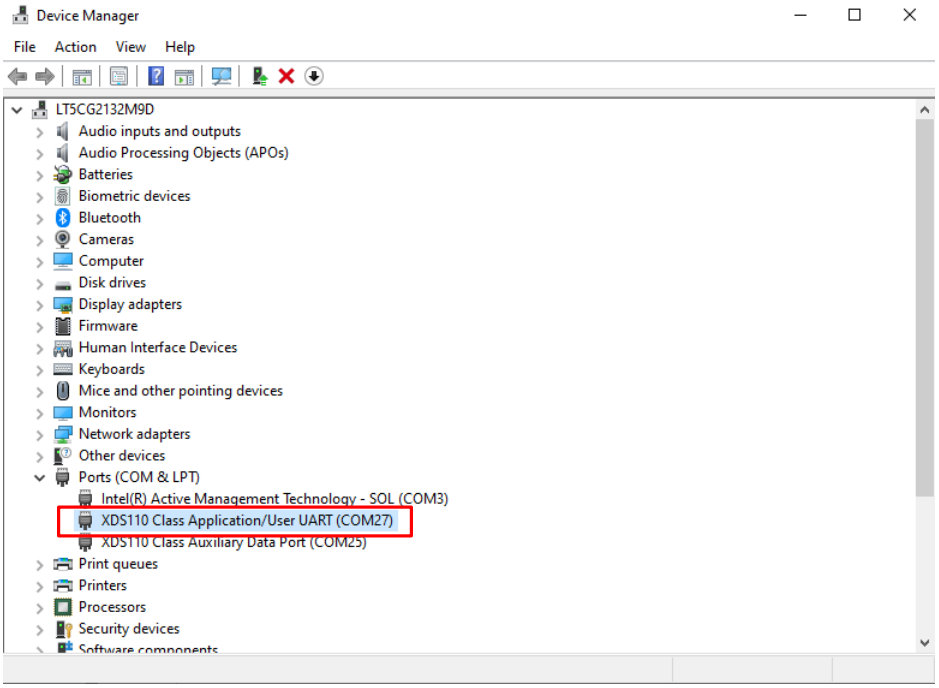

图 4-13. 来自设备管理器的 COM 端口号

   b. 选择或输入与 2 中所选输入相匹配的 *Baud Rate*。
   c. 点击 *OK*。
7. 在 GUI 中点击 *CONNECT*，然后等待几秒钟，以便进行初始通信握手。
8. 打开通道，在窗口中查看数据开始绘图。

备注

如果窗口中绘制了任何数据，则检查触发器设置并验证代码中的变量值是否正在主动变化。

如果这些步骤出现任何问题，请参阅节 7。

5 Signal Sight GUI 导航

Signal Sight 采用用户友好型 GUI，该界面与标准示波器的熟悉界面高度相似，同时提供多种增强功能和特性。下面的部分介绍了利用该工具最常用功能的步骤。

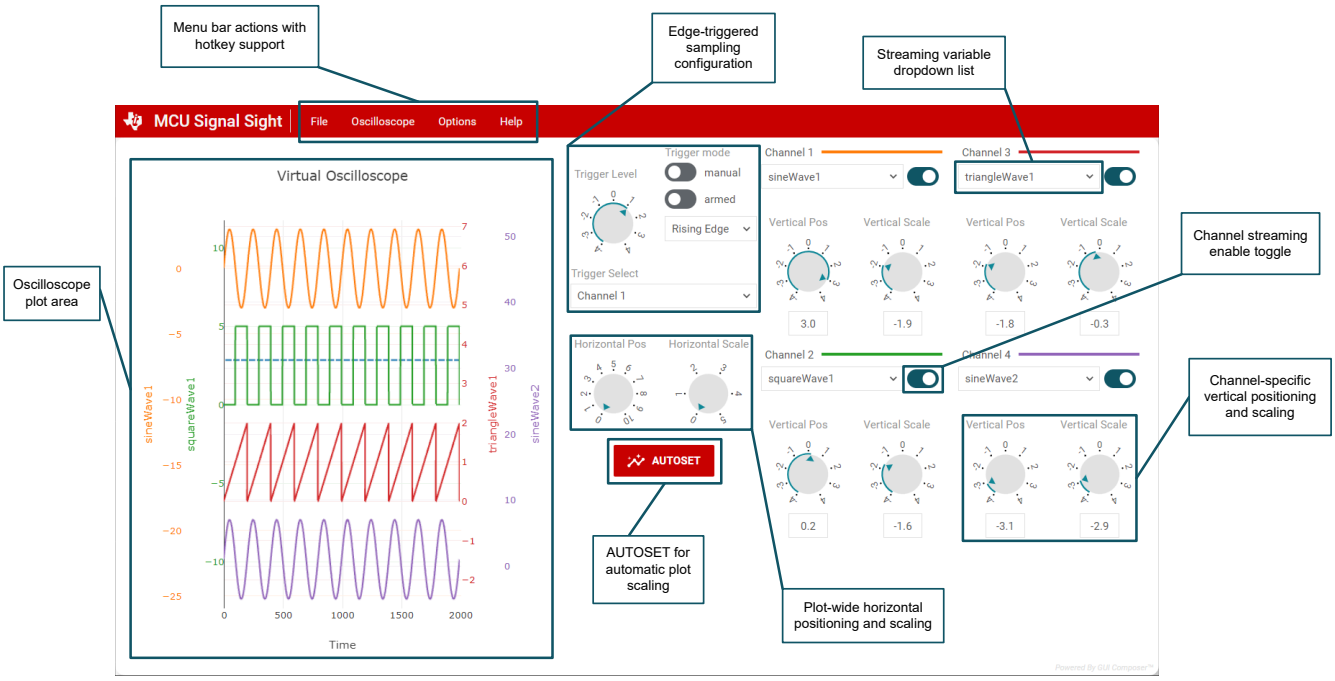

图 5-1. Signal Sight GUI 界面

5.1 验证目标连接

启动 Signal Sight GUI 时，将出现一个模式，提供关于连接到目标 MCU 器件的说明。根据调试环境以及之前是否使用过该工具，可以显示各种不同的消息。

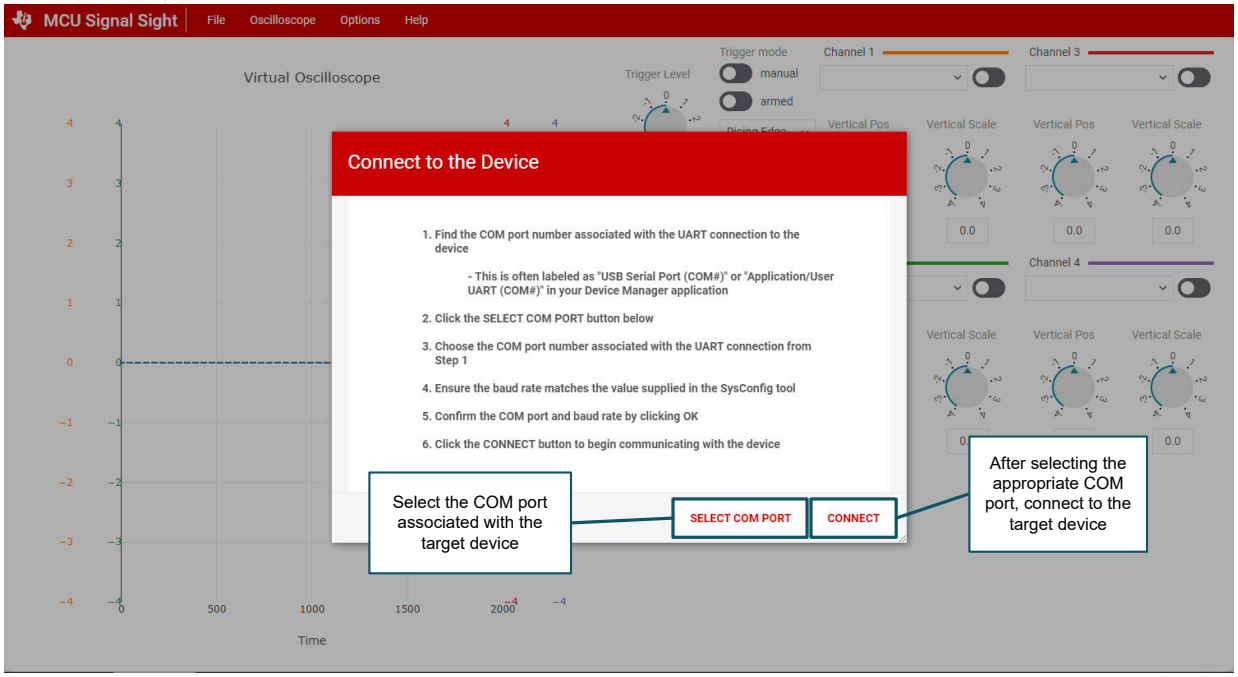

图 5-2. CCS 环境

在 CCS 环境中进行调试时，将会显示一个说明连接到目标的步骤的模式。按照这些步骤操作，并点击 *Connect* 以便与目标器件建立连接。

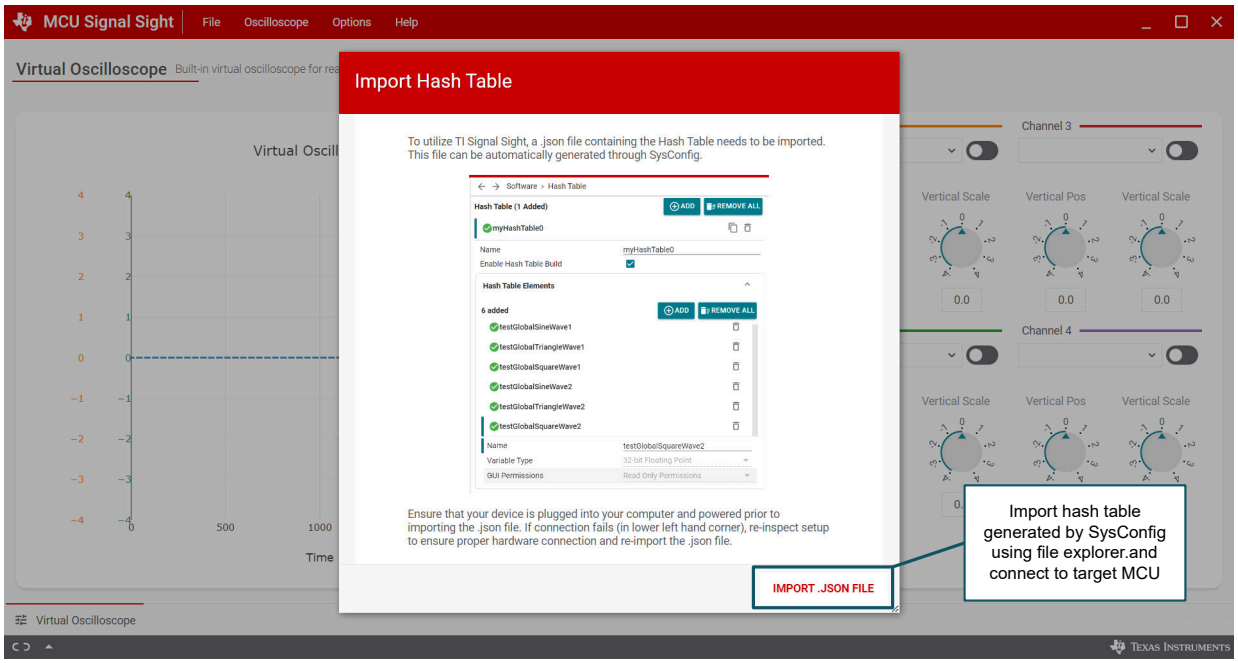

图 5-3. 独立 GUI 环境

在独立环境中使用 Signal Sight GUI 时，需要从外部提供 SysConfig 工具生成的哈希文件。为此，请选择 *IMPORT .JSON FILE* 按钮并浏览文件资源管理器。导入所需的 .json 文件。导入后，GUI 会向所连接的 MCU 器件发出连接请求。连接后，该模态将终止。

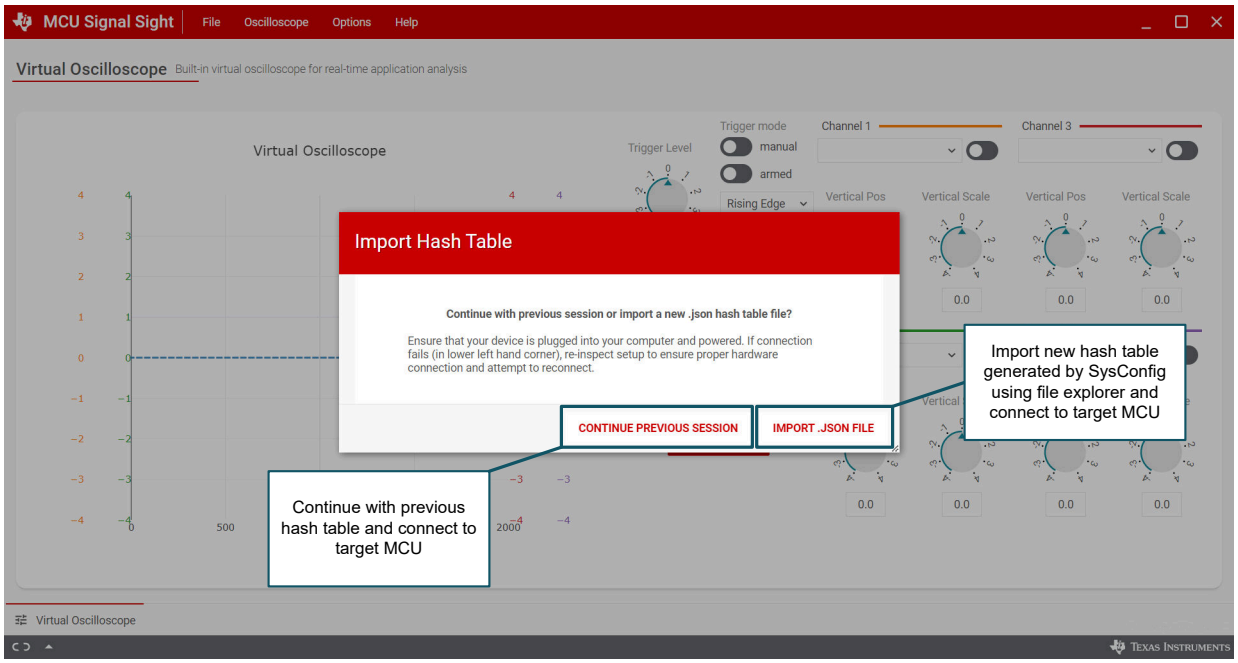

图 5-4. 恢复上一个会话

在随后使用该工具时，模态提示用户继续上一个 Signal Sight 会话。如果继续上一个会话，则点击 *CONTINUE PREVIOUS SESSION* 按钮加载之前使用的哈希表，并尝试连接到器件。如果要使用新工程进行调试，则点击 *IMPORT .JSON FILE*，提示用户导入哈希表文件并连接到目标 MCU。

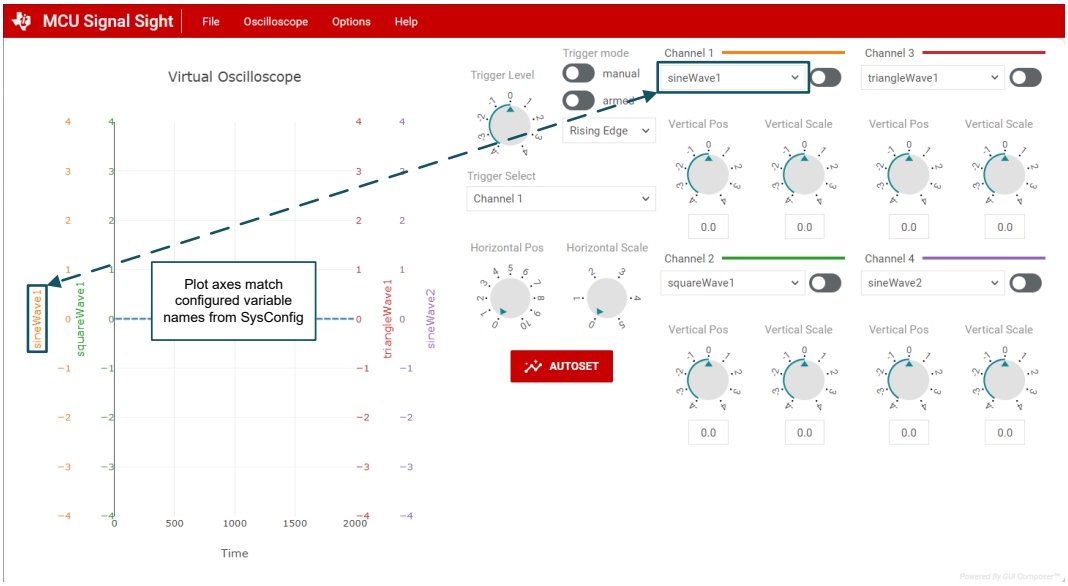

图 5-5. Signal Sight 界面

当 Signal Sight 工具成功连接到目标 MCU 时，将会出现示波器界面。哈希表中的前四个流式变量自动填充到通道 1-4 的下拉菜单中。流式变量也显示在图形的绘图轴上。

5.2 启用数据流

数据流请求直接从 PC 向目标 MCU 发起。要启动数据流式传输，请点击与要查看的输入通道对应的开关。要切换正在流化的特定变量，请点击下拉菜单并从列表中选择所需的流式变量。

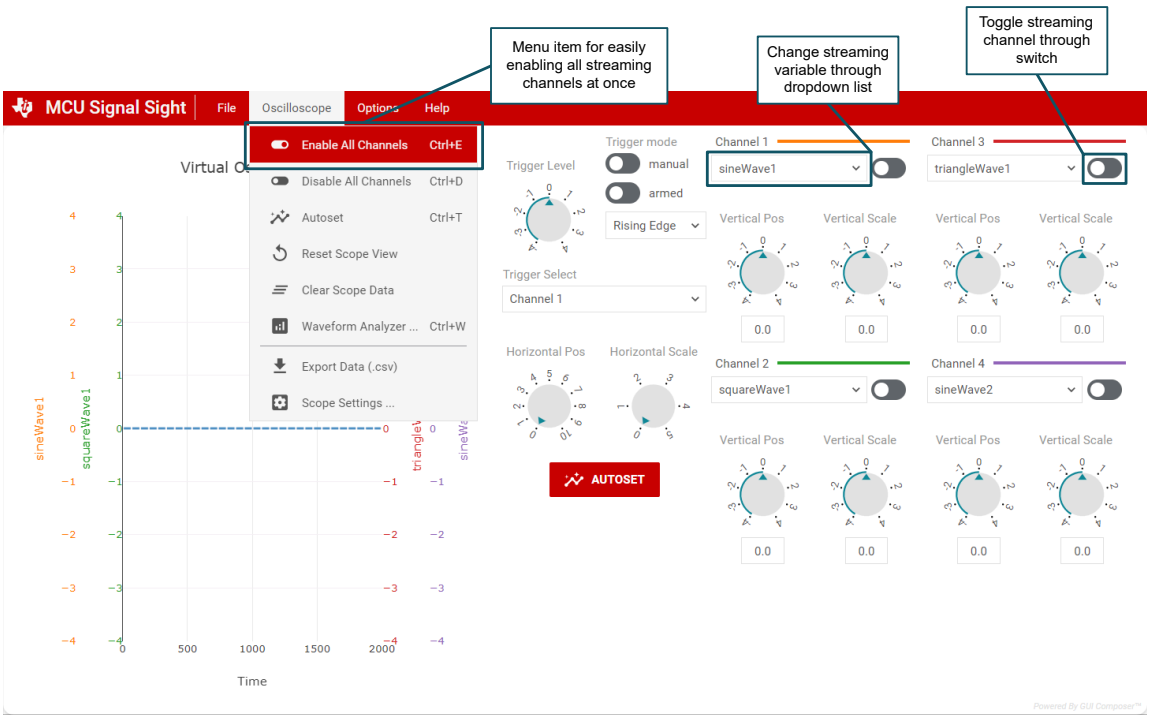

图 5-6. 启用数据流

启动和停止数据流的 Signal Sight 功能菜单操作。要快速启用图上的所有四个通道，请点击 *Oscilloscope → Enable All Channels*。这会自动将触发模式设置为自动，以启用连续流式传输。要禁用数据流，请点击 *Oscilloscope → Disable All Channels*。

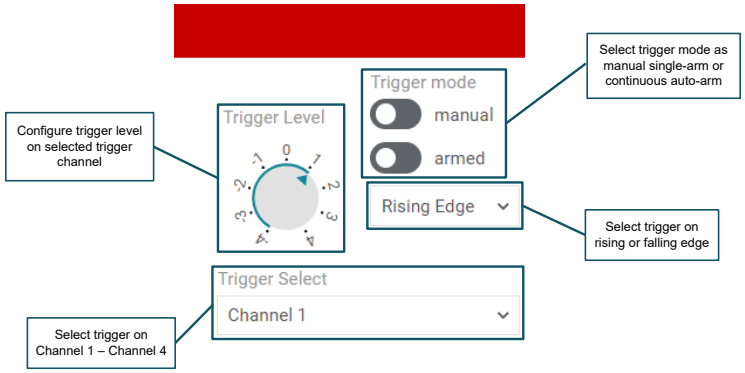

图 5-7. 触发条件设置

与物理示波器一样，虚拟示波器的图也是电平触发的。这意味着在传入数据点与示波器的触发电平相交之前，数据不会显示在图上。在触发器设置中，用户可以配置触发器设置，例如触发电平、要触发的通道和触发沿（上升沿或下降沿）。

还提供了各种触发模式。顶部开关控制触发器重复，底部开关控制触发器的布设。如果顶部开关处于关闭位置（左侧），则示波器触发器处于手动单臂模式。该模式用于捕获单个数据图。底部开关打开（右侧）后，示波器触发器会在传入数据流中查找边沿。发现边沿时，示波器显示屏开始绘制传入数据。当图中完全填充数据后，底部开关会自动关闭，触发器被禁用。

如果顶部开关处于打开位置（右侧），则示波器触发器处于连续臂模式。该模式用于连续运行示波器，并查看更长时期内的数据。示波器触发器持续对传入的数据流进行采样并绘制到图上，而无需手动重新布设触发器。要禁用此模式，只需关闭顶部开关即可。

5.3 调整绘图视图

当数据从 MCU 器件流式传输到 PC 时，在图表上填充数据点。

为方便起见，提供了 *Autoset* 按钮，它可自动确定所有示波器通道垂直缩放的最佳设置。此 AutoSet 按钮可以从主界面或菜单栏中的示波器菜单访问。

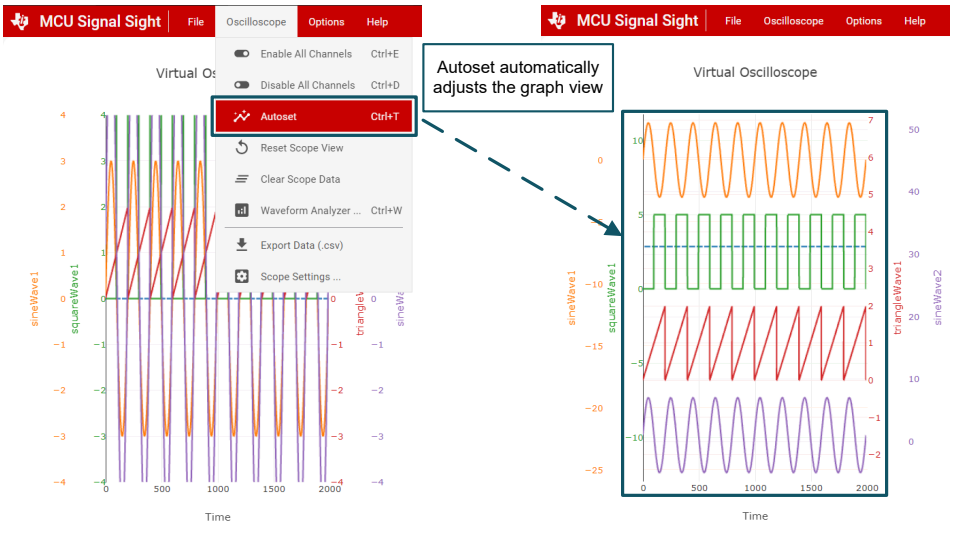

图 5-8. 自动设置绘图视图

Signal Sight 界面提供了用于手动调整绘图视图的旋钮。Horizontal Pos 旋钮用于控制 x 轴的位置。在 0 位置，绘图视图显示工具接收并勾画的传入数据。在 10 位置，绘图视图显示从图中清除的之前缓冲的数据。Horizontal

Scale 旋钮用于调整 x 轴的缩放比例，并可用于放大图形的某些部分。缩放比例采用对数。在 0 位置，完整缓冲区位于图的帧中。在 1 位置，图中只有一半的缓冲区。在 2 位置，仅显示四分之一的缓冲区。

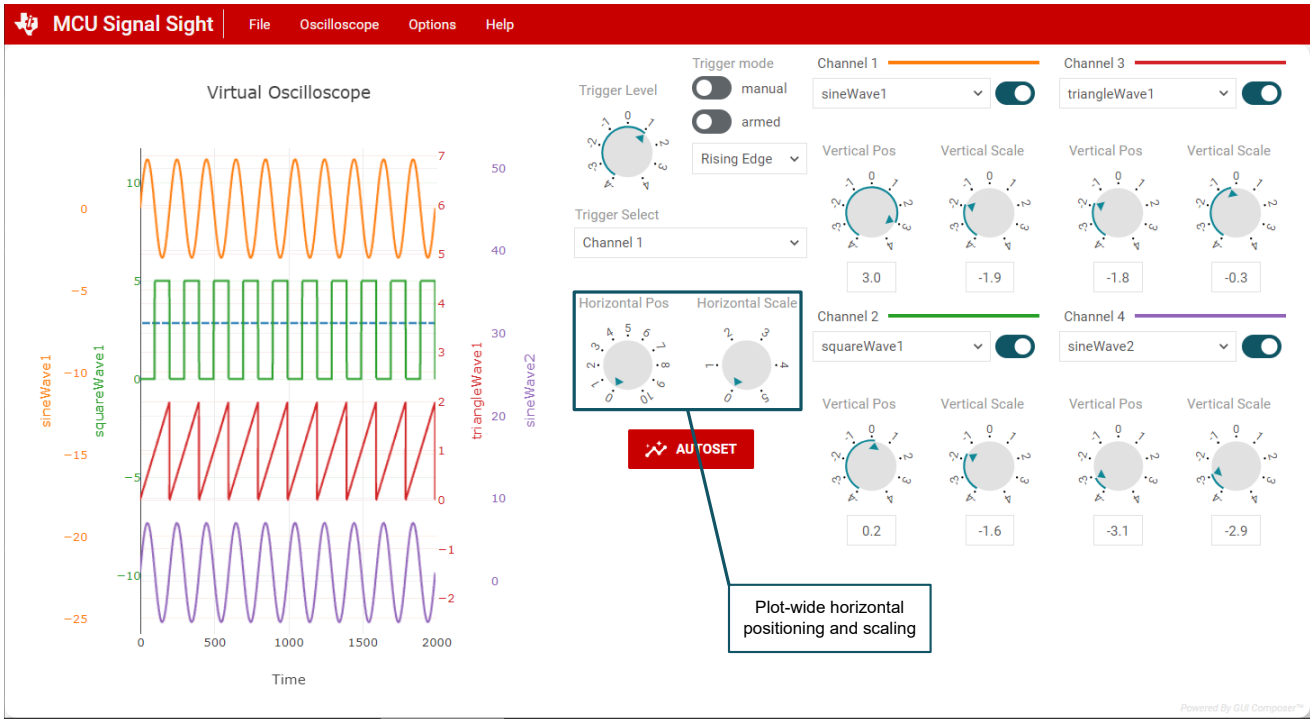

图 5-9. 水平旋钮

每个输入通道都有一个单独的垂直定位和垂直缩放旋钮。这些控件可供用户调整图形上特定图的视图。旋钮的功能与前面所述的水平旋钮相似。

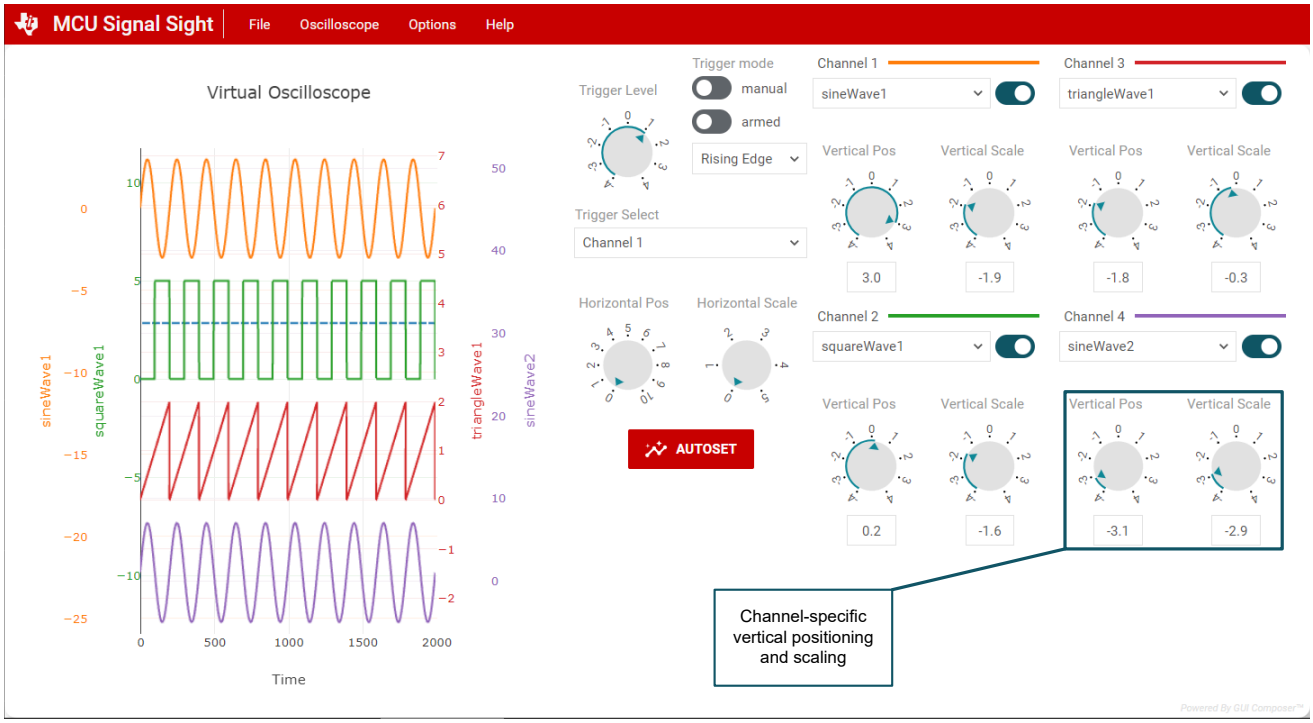

图 5-10. 垂直旋钮

要将示波器视图轻松地重置为默认视图配置，请点击 *Oscilloscope → Reset Scope View*。

5.4 菜单栏操作和热键

Signal Sight GUI 具有许多功能和快捷操作，可通过屏幕顶部的菜单栏调用。下表介绍了可用的菜单项和功能。

菜单	菜单项	菜单操作	热键
文件	导入哈希表 JSON	提示用户导入不同的哈希表以在不同工程中使用 GUI	
	重启	重启 GUI	Ctrl + R
	退出	关闭 GUI	Ctrl + X
示波器	启用所有通道	打开所有示波器通道并开始流式传输数据	Ctrl + E
	禁用所有通道	关闭所有示波器通道	Ctrl + D
	自动设置	自动确定所有示波器通道的垂直缩放的最佳设置	Ctrl + T
	重置示波器视图	将示波器图缩放比例恢复为默认值	
	清除过滤器数据	清除示波器图缓冲区中的所有数据	
	波形分析仪	打开波形分析仪菜单	Ctrl + W
	导出数据	将示波器图中显示的所有数据导出到 .csv 文件以进行外部分析	
	示波器设置	打开示波器设置菜单	
选项	串行端口设置	打开串行端口设置菜单	
	设置	打开 Signal Sight 设置菜单	
帮助	关于	显示关于应用程序的信息	

5.5 高级特性

Signal Sight GUI 在基本数据流功能之上还具有各种附加功能，可实现优化调试。

5.5.1 波形分析仪

Signal Sight GUI 采用专用的信号分析工具，称为波形分析仪。该工具可对示波器图上显示的传入数据进行快速统计分析。波形分析仪显示关键信息，例如最小数据点、最大数据点和均方根 (RMS)。要访问波形分析仪工具，请点击 *Oscilloscope → Waveform Analyzer*。

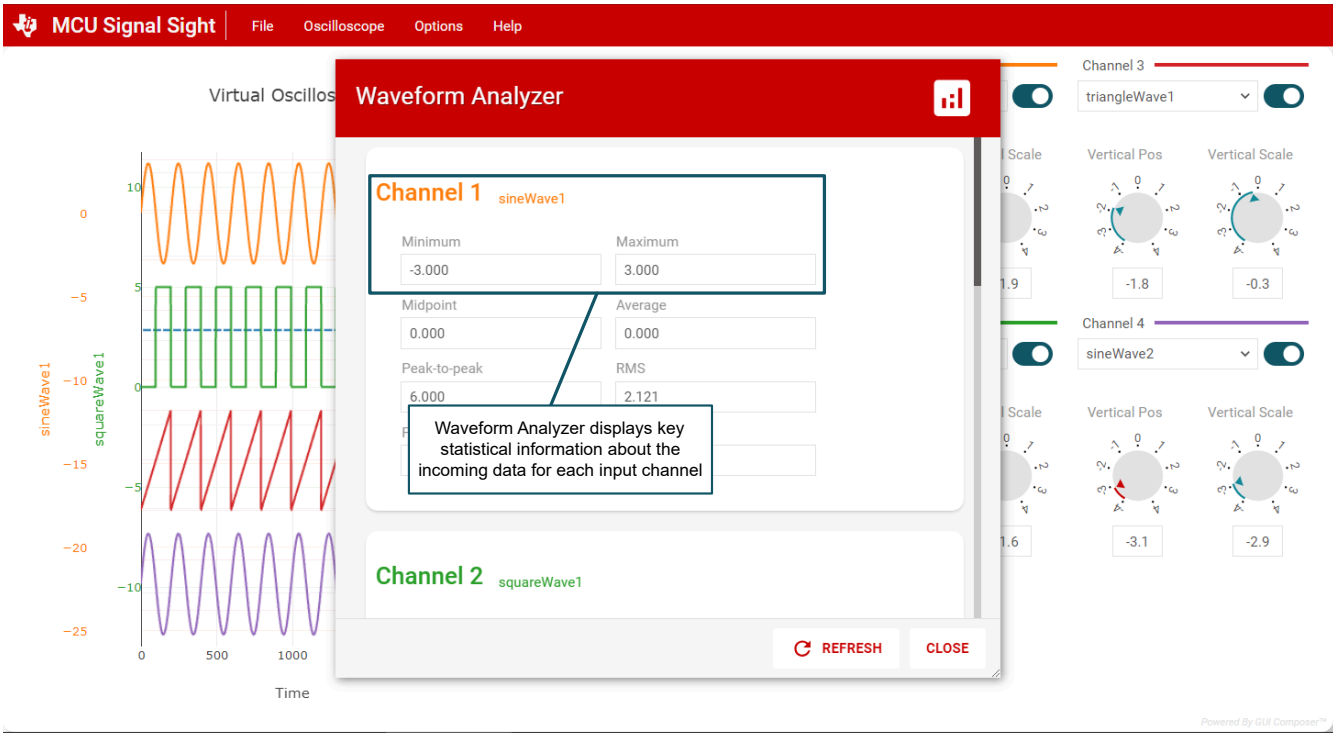

图 5-11. 波形分析仪工具

5.5.2 示波器设置

通过调整示波器设置菜单中的参数，可以从标准视图进一步自定义虚拟示波器图。要访问示波器设置菜单，请点击 *Oscilloscope → Scope Settings*。该菜单包含用于更改缓冲区大小和采样率的设置。它还允许用户编辑超出通道旋钮所允许设置的垂直缩放。如果传入数据波形太大而无法在默认设置下显示在图上，这会很有用。

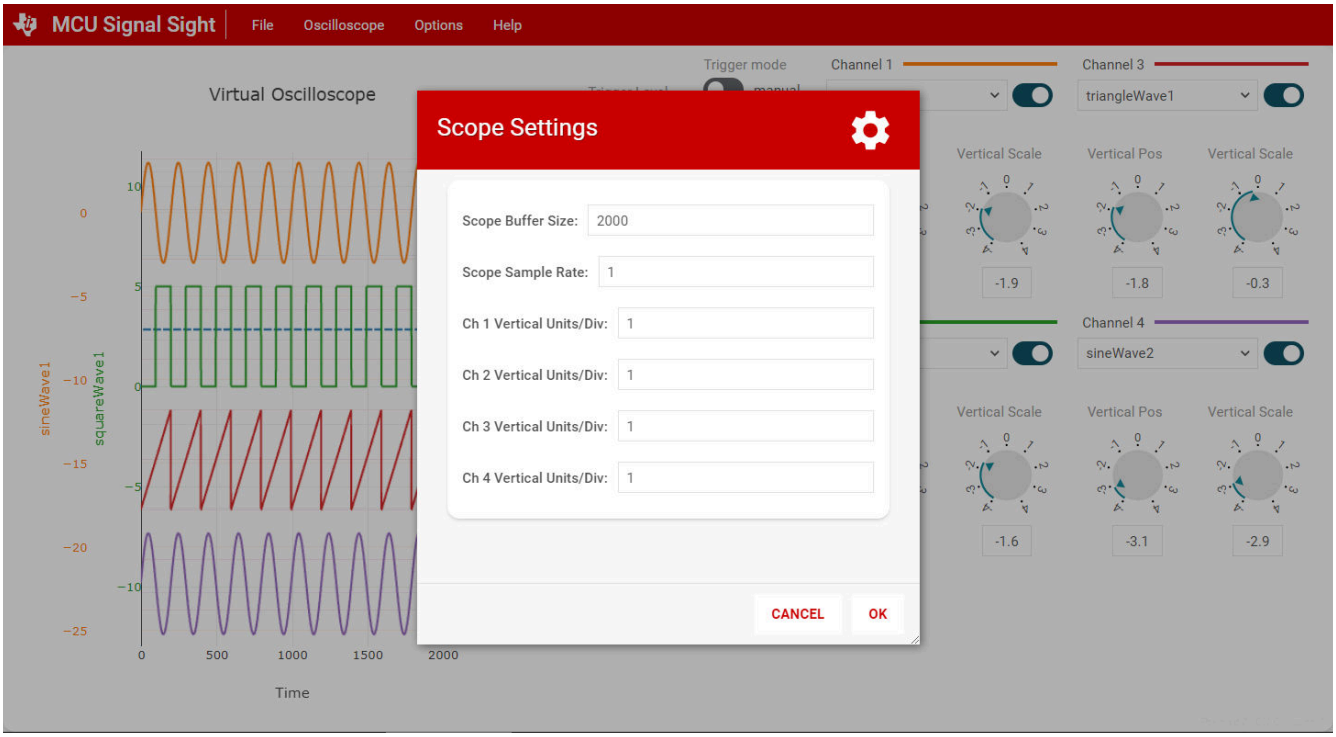

图 5-12. 示波器设置菜单

为了在图上捕获更多数据点，请编辑示波器设置菜单中的 Scope Buffer Size 参数。这允许用户在清除示波器视图之前增加数据点的数量。

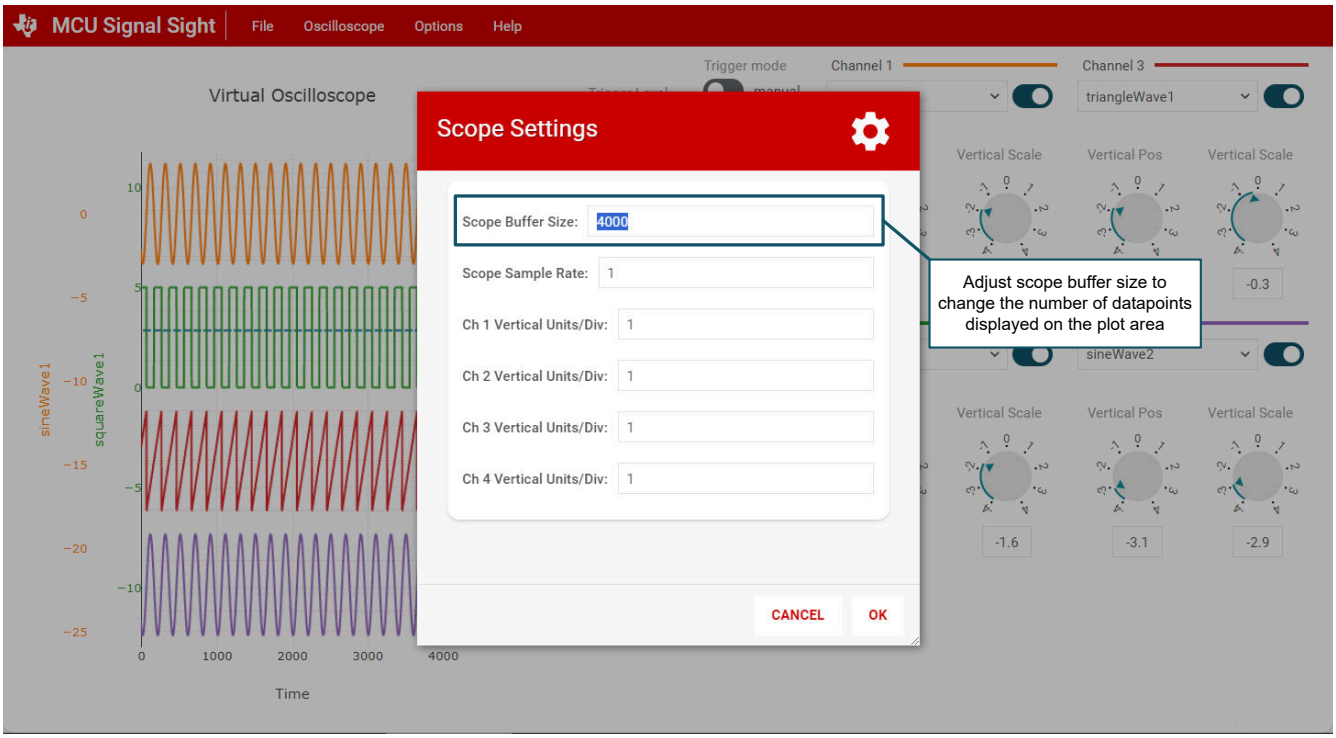

图 5-13. 更改示波器缓冲区大小

5.5.3 数据导出

为了轻松访问从目标 MCU 接收到的数据点，用户可以选择下载示波器图中所示的数据。要将数据导出到 .csv 文件，请点击 *Oscilloscope → Export Data (.csv)*。

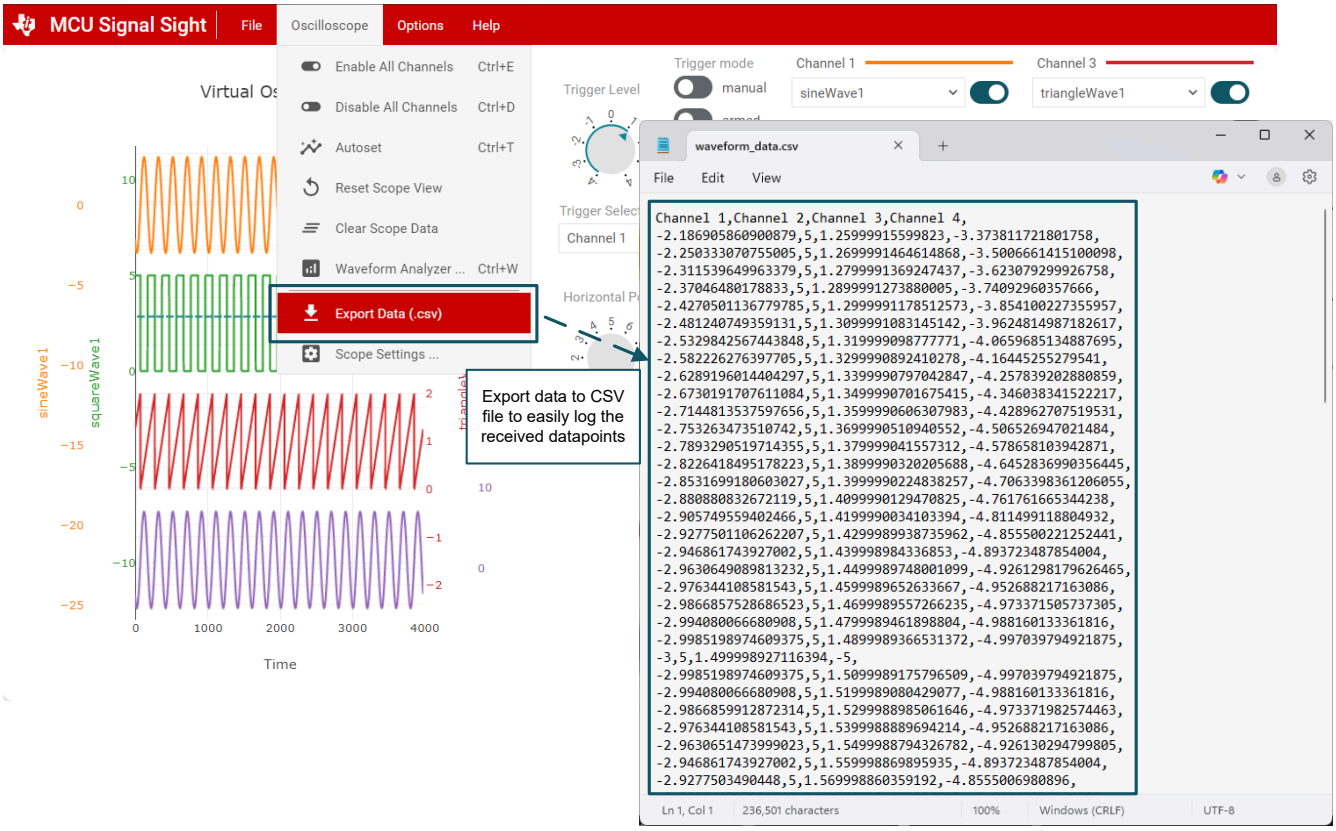

图 5-14. 将 Signal Sight 数据导出到 .csv 文件

6 关于工具

绘图刷新率

目前，该工具支持的唯一通信外设是 SCI (UART)，因此绘图速度取决于给定 C2000 器件上的最大 SCI 波特率。有关最大 SCI 波特率，请参阅相关器件数据表。

绘图变量

该工具的当前版本仅支持绘制 32 位浮点类型的全局变量。计划在将来的版本中支持有符号或无符号类型的变量以及结构内的变量。

7 疑难解答指南

有关不同故障排除场景的建议，请参阅以下常见问题解答。

- GUI 名称未显示在 *View* → *Plugins* 文件夹中
  - 检查 1：验证 SysConfig 和 C2000Ware 产品已在工程 *Properties* → *General* → *Dependencies* 中设置为预期版本。
  - 检查 2：导航至工程 *Properties* → *General* → *Variables* → *{下拉菜单}Show both user-defined and system variables* → *SYSCONFIG_TOOL* 变量。验证“sysconfig_cli.bat”文件的路径位于当前 CCS 安装内。该路径应类似于 C:\ti\ccsxxxx\ccs\utils\sysconfig_x.xx.x\sysconfig_cli.bat。
- 开启通道后数据未出现在图中
  - 检查 1：验证触发通道和电平已设置为通过（基于所绘制数据的性质）触发器的信号，并将触发模式设置为 *Manual*。
  - 检查 2：验证在打开通道时更新了 SS_currentStreamState 变量。
    - 通过 CCS 连接到目标代码。
    - 在观察窗口中，添加变量 SS_currentStreamState。
    - 如果此变量的值保持为 *NO_UPDATE*，则 UART 通信存在问题。
  - 检查 3：对于 LaunchPads/controlCARD - 验证 XDS、UART 开关处于正确的位置，并在 SysConfig Step 7a 的 SysConfig SCI PinMux 选择中配置了正确的 GPIO。
  - 检查 4：验证 GUI *Options* → *Serial Port Settings* 中的波特率设置与在 SysConf 步骤 6b 的 SysConfig 条目 *MCU Signal Sight* → *Exporter* → *SCI Transfer Communication Link* → *Baud Rate* 中选择的波特率一致。
- 该图似乎缺少/跳过了一个或多个通道的数据
  - GUI 具有当缓冲区已满时跳过采集数据样本的内置功能。
    - 通过 CCS 连接到目标代码（确保在 GUI 中点击 *Connect* 之前运行目标代码）。
    - 在观察窗口中，如果已启用乒乓缓冲，则添加变量“SS_streamDataSkips”和“SS_streamDataSkips2”。
    - 如果 skip 变量的值显著递增，则软件正在跳过采集数据样本。用户可以：
      - 从 SysConfig 步骤 4 中增加缓冲区大小。这样可以在缓冲区已满之前将更多数据放置在缓冲区中。
      - 降低应用程序代码中的数据采集频率（例如：从目标步骤 4 中降低 CPU 计时器 ISR 的频率）。这样可使 SCI 模块有更多的时间在添加更多数据之前清空缓冲区。
      - 在 SysConfig 步骤 6b 中升高波特率。这样可使 SCI 更快地从缓冲区发送数据，从而更快地清空缓冲区。
- GUI 图看起来太小，或者屏幕上缺少通道旋钮/自动设置按钮
  - 虚拟示波器功能设计为在 100% 比例的屏幕上查看。用户可以在 PC 设置中修改此设置。对于 Windows 中，该选项位于 *Settings* → *Display* → *Scale and Layout* 中。
- 数据绘图意外暂停
  - 通过点击 *Options* → *Serial Port Settings* 并选择 *Connect* 重新连接到目标。这会复位目标端的流状态。
- 如在工具中发现任何错误，请在 E2E 论坛上发帖。

8 总结

将 MCU Signal Sight GUI 无缝集成到现有的嵌入式 C2000 工程中，以了解应用中的信号并优化调试过程。借助独立的数据缓冲和传输，MCU Signal Sight 支持可以提供实时数据捕获，而不会影响 CPU 性能。与需要昂贵硬件但提供的信号可见性有限的传统调试方法、或者缺乏实时应用程序精度的现有软件实现不同，MCU Signal Sight 提供了全面的基于软件的实现。

9 参考资料

- 德州仪器 (TI)，MCU 控制中心工具开发人员指南应用手册
- 德州仪器 (TI)，C2000 实时微控制器外设用户指南
- 德州仪器 (TI)，C2000 SysConfig 应用手册
- 德州仪器 (TI)，DLT 开发人员指南（带工具）应用手册

重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。 严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 TI 的销售条款或 ti.com 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265