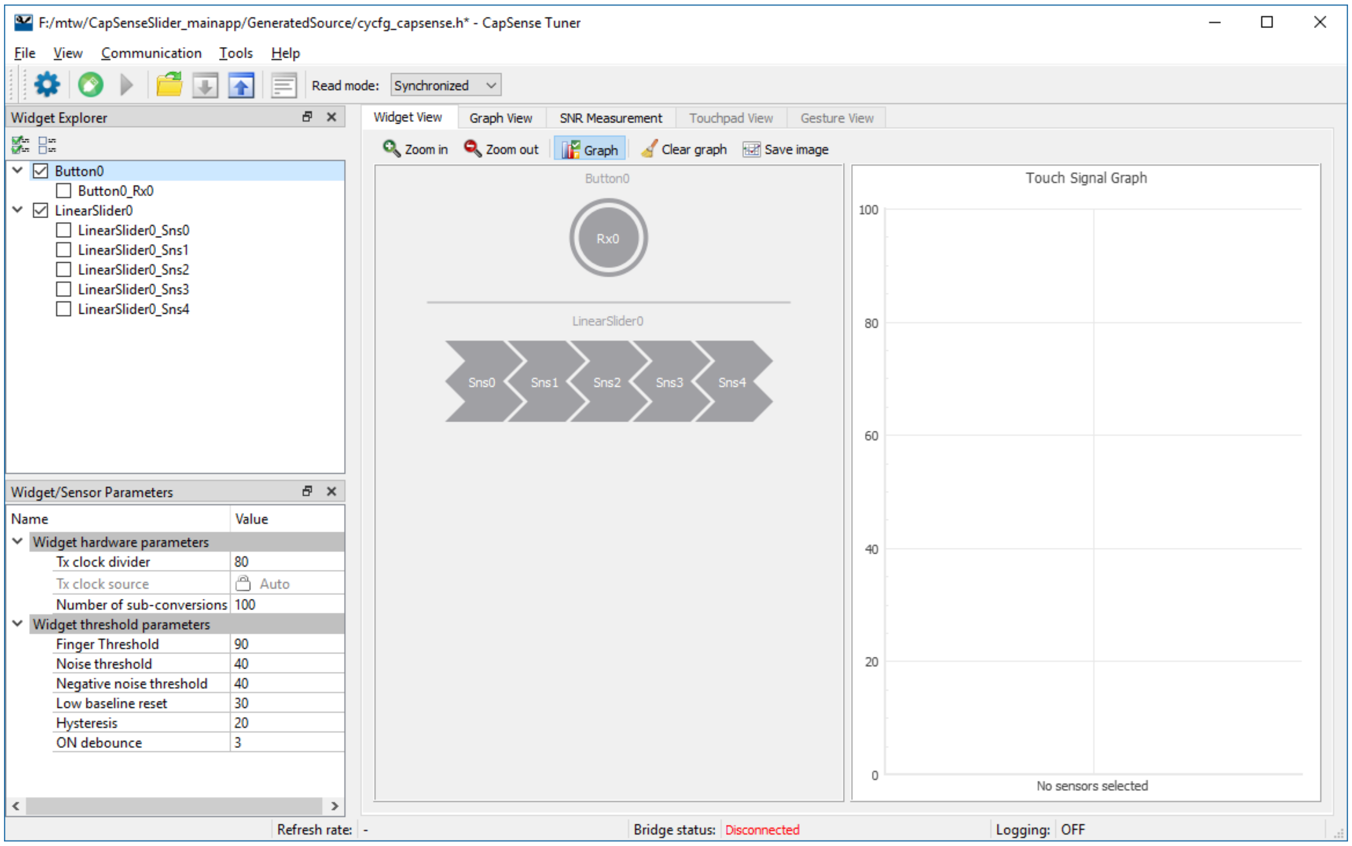
ModusToolbox™ CapSense® Tuner Guide

Version 1.1

Overview

The CapSense Tuner is part of a collection of tools included with the ModusToolbox software. Use it to tune your CapSense application.

Prior to using the CapSense Tuner, you must create a CapSense application and program it into the device. Refer to the *CapSense Configurator Guide* and the *CapSense Middleware Documentation* for information about creating and configuring such an application. The application must contain CapSense Middleware and a communication interface. The Tuner works with the I^{2}C and UART communication interfaces.



Name	Value
Widget hardware parameters	
Tx clock divider	80
Tx clock source	Auto
Number of sub-conversions	100
Widget threshold parameters	
Finger Threshold	90
Noise threshold	40
Negative noise threshold	40
Low baseline reset	30
Hysteresis	20
ON debounce	3

Launch the Tuner

The CapSense Tuner is a stand-alone tool that contains menus and several tabs used to tune a CapSense application. You can run the Tuner with a ModusToolbox IDE application. You can also run it independently of the ModusToolbox IDE. Then, you can either use the generated source with a ModusToolbox IDE application, or use it in any software environment you choose.

From a ModusToolbox IDE Application

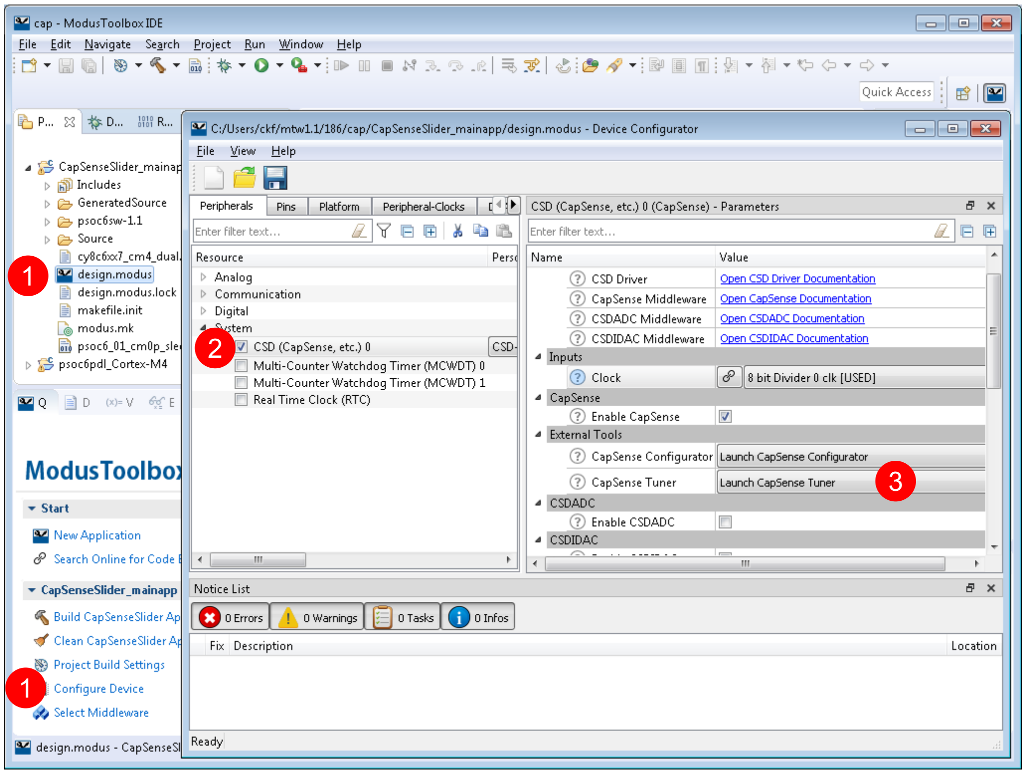
Note These steps are the minimum required to launch the CapSense Tuner. However, the Tuner will not work without a properly configured CapSense application programmed into the device.

1. Launch the Device Configurator from the ModusToolbox IDE.

2. On the **Peripherals** tab, enable the **CSD (CapSense, etc.)** resource.

   Note The CSD resource must be enabled and selected to to display the CSD parameters.

3. On the **Parameters** tab, click the **Launch CapSense Tuner** button.



Refer to the *Device Configurator Guide* for more information.

Independent of the ModusToolbox IDE

To run the CapSense Tuner independently, navigate to the install location and run the executable. The default install location on Windows is:

<install_dir>\tools\capsense-configurator-<version>

When run independently, the application opens without any information. You need to open the specific CapSense application header file that was created by the CapSense Configurator in order for the Tuner to work. See Menus for more information about opening a header file.

From the Command Line

You can run the configurator from the command line. However, there are only a few reasons to do this in practice. The primary use case would be to re-generate source code based on the latest configuration settings. This would often be part of an overall build script for the entire application.

For information about command line options, run the configurator using the `-h` option.

Quick Start

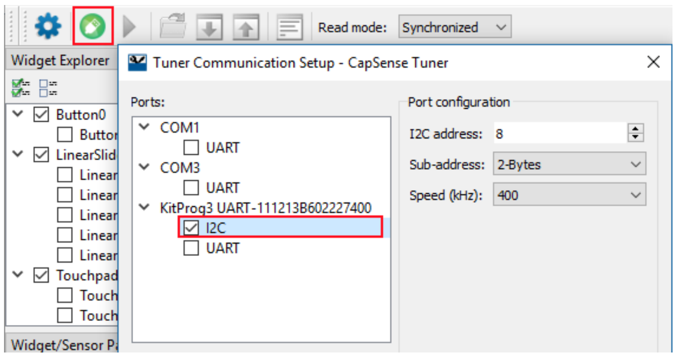
This section provides a simple workflow for how to use the CapSense Tuner.

1. Create a CapSense application with a tuner communication interface, and program the application onto the device. Refer to the *ModusToolbox IDE User Guide* for more details.

   Note The CapSenseSlider starter application, which is configured to work on PSoC 6 kits, can also be used.

2. Launch the CapSense Tuner.

3. Start Communication. Click **Connect** to establish connection and select the protocol. In this case "I2C."



Tuner Communication Setup - CapSense Tuner

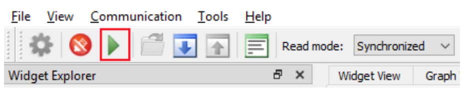
Ports:
- COM1
  - UART
- COM3
  - UART
- KitProg3 UART-111213B602227400
  - ☑ I2C
  - UART

Port configuration

I2C address: 8
Sub-address: 2-Bytes
Speed (kHz): 400

Note Refer to the *KitProg User Guide* for supported configurations and modes.

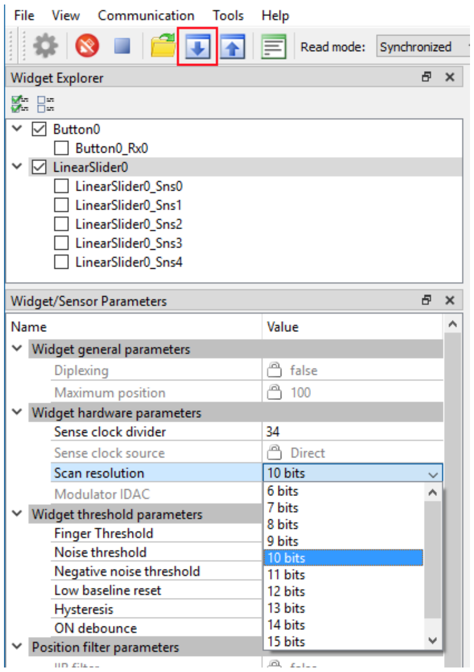
4. Click **Start** to extract data.



5. Touch the sensors on the hardware and notice the change in sensor/widget status on the **Widget View** tab. See Widget View.

6. Open the **Graph View** tab. Check the sensors in the Widget Explorer Pane to observe sensor signals in the graph. Touch the sensors and notice the signal change on the graph. See Graph View.

7. Change values of widget/sensor parameters. Then, apply the new settings to the device using the **Apply to Device** button.



Read mode: Synchronized

- Button0
  - Button0_Rx0
- LinearSlider0
  - LinearSlider0_Sns0
  - LinearSlider0_Sns1
  - LinearSlider0_Sns2
  - LinearSlider0_Sns3
  - LinearSlider0_Sns4

Widget/Sensor Parameters

Name	Value
Widget general parameters	
Diplexing	false
Maximum position	100
Widget hardware parameters	
Sense clock divider	34
Sense clock source	Direct
Scan resolution	10 bits
Modulator IDAC	
Widget threshold parameters	
Finger Threshold	
Noise threshold	
Negative noise threshold	
Low baseline reset	
Hysteresis	
ON debounce	
Position filter parameters	

6 bits
7 bits
8 bits
9 bits
10 bits
11 bits
12 bits
13 bits
14 bits
15 bits

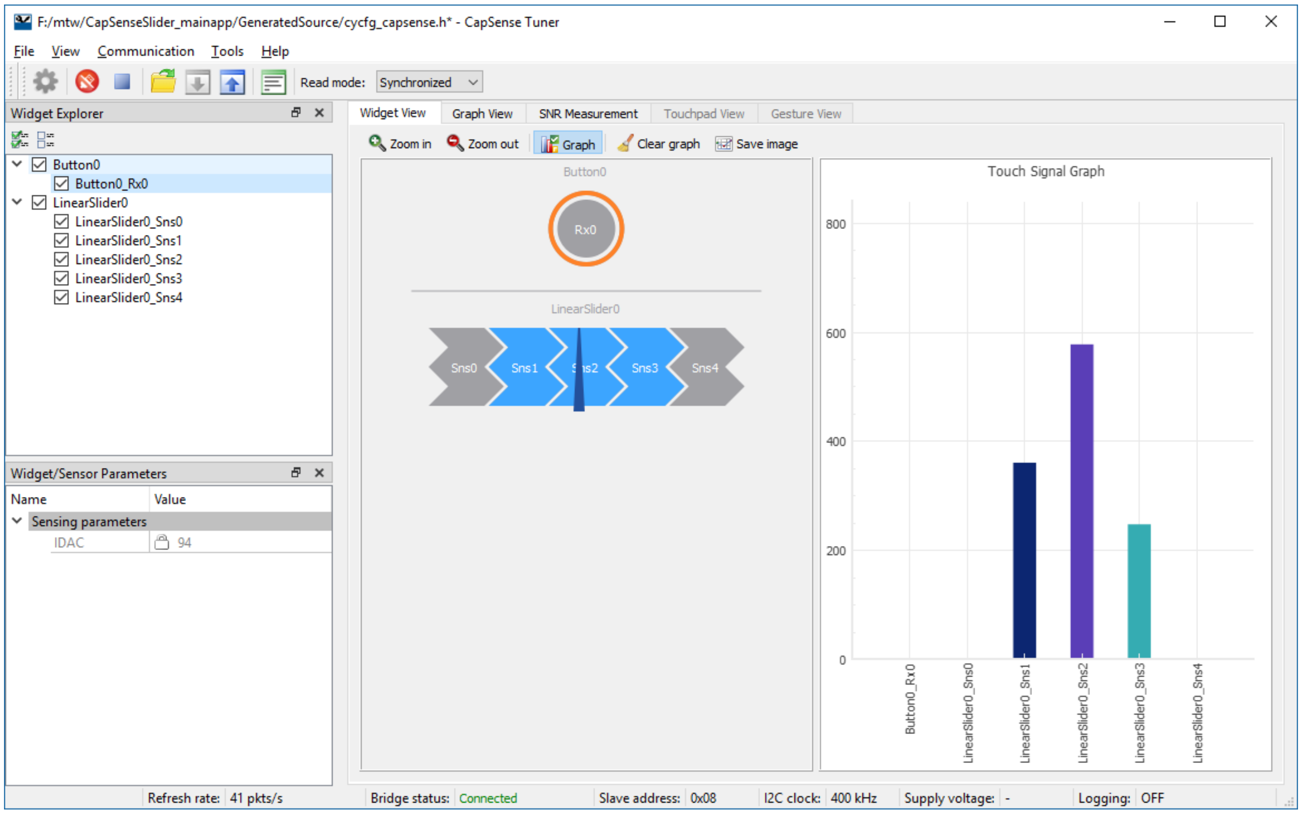
You can do this when using Manual or SmartSense (Hardware parameters only) modes for tuning.

- To edit the threshold parameters, use SmartSense (Hardware parameters only) mode.
- To edit all the parameters, use Manual mode.
- When SmartSense (Full Auto-Tune) is selected for CSD tuning mode, parameters are read only (except the Finger Capacitance parameter).

8. Save the Tuner parameters. Click **Apply to Project** button.
9. Exit the Tuner application.

General Interface

The Tuner application contains menus, a toolbar, panes, tabs, and a status bar, which are all used to tune a CapSense application. This section describes each of these elements.



Menus

The main menu commands to control and navigate the Tuner:

File Menu

- **Open** – Opens a header file. This command is visible only when running independently from ModusToolbox IDE.
- **Apply to Device** – Commits the current values of the widget/sensor parameters to the device. This item becomes active if a value of any configuration parameter from the Tuner application is changed (i.e. if the parameter values in the Tuner and the device are different). This is an indication that the changed parameter values need to be applied to the device.
- **Apply to Project** – Commits the current values of widget / sensor parameters to the CapSense project.
- **Exit** – Asks to save changes if there are any and closes the Tuner. Changes are saved to the configuration file.

View Menu

- Contains options to hide or show the dock windows and the toolbar.

Communication Menu

- **Connect** – Connects to the device via a communication channel selected in the Tuner Communication Setup dialog. When the channel was not previously selected, the Tuner Communication dialog is shown.
- **Disconnect** – Closes the communication channel with the connected device.
- **Start** – Starts reading data from the device.
- **Stop** – Stops reading data from the device.

Tools Menu

- **Tuner Communication Setup…** – Opens the configuration dialog to set up a communication channel with the device.
- **Options…** – Opens the configuration dialog to set up different tuner preferences.

Help Menu

- **View Help** – Opens the CapSense Tuner User Guide (this document).
- **About CapSense Tuner** – Open the About box to display version information.

Toolbar

Contains frequently used buttons that duplicate the main menu items:

- – Opens the configuration dialog to set up a communication channel with the device. Same as the **Tools > Tuner Communication Setup** menu item
- – Connects to the device via a communication channel selected in the Tuner Communication Setup dialog. Same as the **Communication > Connect** menu item
- – Closes the communication channel with the connected device. Same as the **Communication > Disconnect** menu item
- – Starts reading data from the device. Same as the **Communication > Start** menu item
- – Stops reading data from the device. Same as the **Communication > Stop** menu item
- – Opens a header file. Same as the **File > Open** menu item
- – Commits the current values of the widget/sensor parameters to the device. Same as the **File > Apply to Device** menu item
- – Commits the current values of widget / sensor parameters to the CapSense project. Same as the **File > Apply to Project** menu item
- – Starts or stops data logging into a specified file
- **Read mode** – Selects the Tuner communication mode with a device. The following options are available (when I^{2}C communication interface is used):
  - **Asynchronized** – When selected, the Tuner reads data asynchronous to sensor scanning and data processing. Due to asynchronous nature, data coherency is not guaranteed. Because of this, Tuner may read the partially updated sensor data. For example, the device completed scanning of only the first sensor and remaining sensors are yet to be scanned, at this moment, the Tuner shall read data of latest scan from first sensor and data of previous scans from remaining sensor. A similar situation, where tuner reads data before prior to the completion of

data processing tasks such as baseline update and filter execution all sensors can happen. The SNR and noise measurements are less accurate due to non-coherent data reading and only limited set of tuning parameters can be edited in real time in this mode.

- **Synchronized** – This communication mode is available when Application firmware periodically calls a corresponding Tuner function: Cy_CapSense_RunTuner(). After selecting Synchronized Communication mode, the CapSense Tuner synchronizes data reading and firmware execution to guarantee data coherency. In this mode, CapSense middleware waits for tuner read/write operation to be completed prior to execution sensor scan and processing tasks and tuner does not read/write data until CapSense middleware completes the scanning and data processing tasks for all widgets in the application. The SNR and noise measurements are most accurate and most tuning parameter can be edited in real time updating in this mode.

Widget Explorer Pane

The Widget explorer pane contains a tree of widgets and sensors used in the CapSense application. The Widget nodes can be expanded/collapsed to show/hide widget's sensor nodes. It is possible to check/uncheck individual widgets and sensors in Widget explorer pane. The Widget checked status affects its visibility in the *Widget View*, while the sensor checked status is controlling the visibility of the sensor raw count / baseline / signal / status graph series in the Graph View and signals in the *Touch Signal Graph* on the *Widget View*.

Selecting a widget or sensor in the Widget Explorer Pane with a click updates the selection in the *Widget/Sensor Parameters Pane.*

Note For CSX widgets, the sensor tree displays individual nodes (Rx0_Tx0, Rx0_Tx1 …) contrary to the configurator where the CSX electrodes are displayed (Rx0, Rx1 … Tx0, Tx1 ...).

Widget/Sensor Parameters Pane

The Widget/Sensor parameters pane displays the parameters of the widget or sensor selected in the Widget Explorer tree. The grid is similar to the grid on the Widget Details tab in the CapSense configurator. The main difference is that some parameters are available for modification in the configurator, but not in the Tuner. This pane includes the following parameters:

- **Widget General Parameters** – Cannot be modified from the Tuner because corresponding parameter values reside in the Flash widget structures that cannot be modified at runtime.
- **Widget Hardware Parameters** – Cannot be modified for the CSD widgets when CSD tuning mode is set to SmartSense (Full Auto-Tune) or SmartSense (Hardware parameters only) in the CapSense configurator. In Manual tuning mode (for both CSD and CSX widgets), any change to Widget Hardware Parameters requires hardware re-initialization which can be performed only if the Tuner communicates with the device in Synchronized mode.
- **Widget Threshold Parameters** – Cannot be modified for the CSD widgets when CSD tuning mode is set to SmartSense (Full Auto-Tune) in the configurator. In Manual tuning mode (for both CSD and CSX widgets), the threshold parameters are always writable (Synchronized mode is not required). The exception is the ON debounce parameter that also requires hardware re-initialization (in the same way as the hardware parameters).
- **Sensor Parameters** – Sensors-specific parameters. The Tuner application displays only IDAC values or/and compensation IDAC value. The parameter is not present for the CSD widget when Enable Compensation IDAC is disabled on the configurator CSD Settings tab. When CSD Enable IDAC auto-calibration or/and CSX Enable IDAC auto-calibration is enabled, the parameter is Read only and displays the IDAC value as calibrated by the firmware. When auto-calibration is disabled, the IDAC value entered in the CapSense Configurator appears, and the parameter is writable in Synchronized mode.

- **Filter Parameters** and **Centroid Parameters** – Cannot be modified at run-time from the Tuner because, unlike the other parameters, these parameter values reside in the Flash widget structures that cannot be modified at runtime.
- **Gesture Parameters** – Synchronized communication mode must be selected to update the Gesture parameters during run-time from the Tuner application.

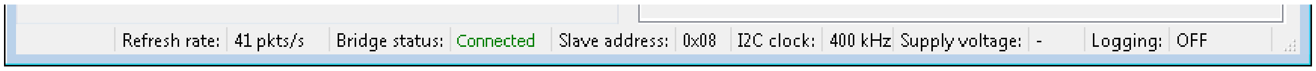
Tabs

The application consists of the following tabs:

- Widget View – Displays the widgets, their touch status, and the touch signal bar graph.
- Graph View – Displays the sensor data charts.
- SNR Measurement – Provides the SNR measurement functionality
- Touchpad View – Displays the touchpad heatmap.
- Gesture View – Displays the Gesture operation.

Status Bar

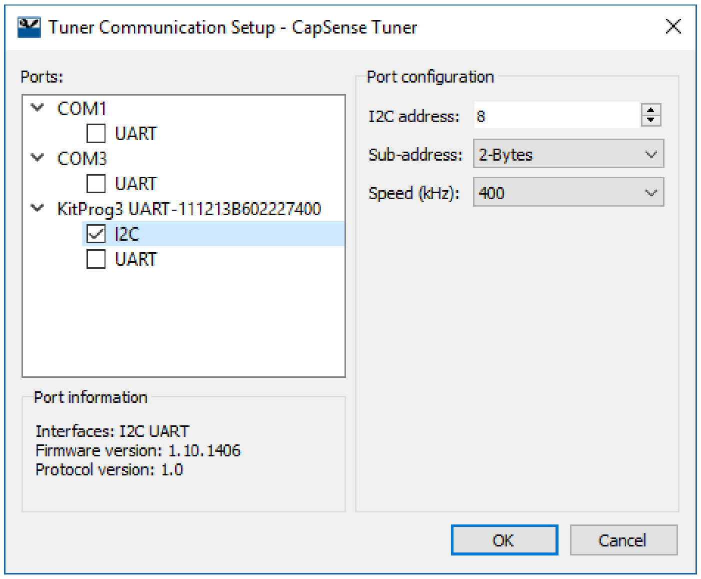
The status bar is located at the bottom of the GUI. It displays information related to the communication state between the Tuner and the device.


Refresh rate: 41 pkts/s | Bridge status: Connected | Slave address: 0x08 | I2C clock: 400 kHz | Supply voltage: - | Logging: OFF

- **Refresh rate** – A count of read samples performed per second. The count depends on multiple factors: the selected communication channel, communication speed, and amount of time needed to perform a single scan.
- **Bridge status** – Either **Connected**, when the communication channel is active, or **Disconnected** otherwise.
- **Slave address** [I^{2}C specific] – The address of the I^{2}C slave configured for the current communication channel.
- **I2C clock** [I^{2}C specific] – The data rate used by the I^{2}C communication channel.
- **Supply voltage** – The supply voltage.
- **Logging** – Either **ON** (when the data logging to a file in progress) or **OFF** otherwise.

Tuner Communication Setup

The Tuner Communication Setup dialog is used to establish communication between the Tuner and target device.


Tuner Communication Setup - CapSense Tuner
Ports:
COM1
UART
COM3
UART
KitProg3 UART-111213B602227400
I2C
UART
Port configuration
OK
Cancel

Port configuration: I2C address: 8; Sub-address: 2-Bytes; Speed (kHz): 400

Port information

Interfaces: I2C UART
Firmware version: 1.10.1406
Protocol version: 1.0

The dialog can be opened by selecting **Tools > Tuner Communication Setup…** in the menu or by clicking the **Tuner Communication Setup** button. The dialog opens automatically after clicking the **Connect** button if no device was previously selected.

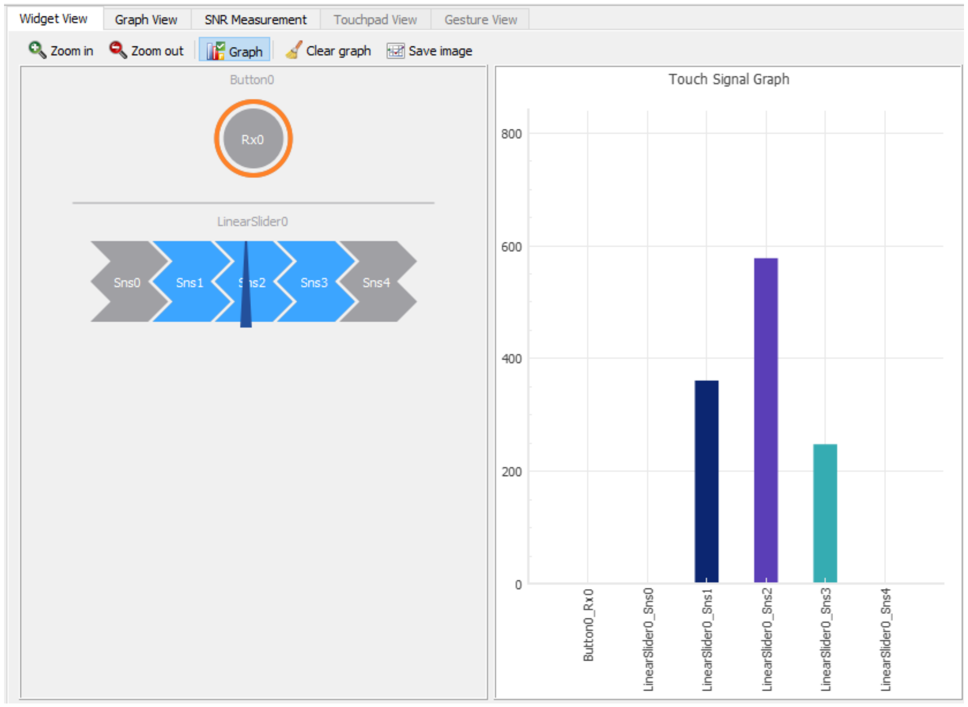
Select the device and communication protocol by checking the item with the protocol name. Change the protocol configuration parameters to match the configuration of the communication peripheral in the application.

The Tuner supports I^{2}C and UART communication interfaces.

Note The parameters in the Tuner Communication Setup dialog must be identical to the parameters of the I^{2}C or UART driver in the application.

Widget View Tab

The **Widget View** provides a visual representation of all widgets selected in the Widget Explorer Pane. If a widget consists of more than one sensor, individual sensors may be selected to be highlighted in the Widget Explorer Pane and Widget/Sensor Parameters Pane.



The Widget and/or sensors are highlighted when the device reports their touch status as active.

Some additional features are available depending on the widget type.

Touch Signal Graph

The Widget view also displays a Touch Signal Graph. This graph contains a touch signal level for each sensor selected in the Widget Explorer Pane.

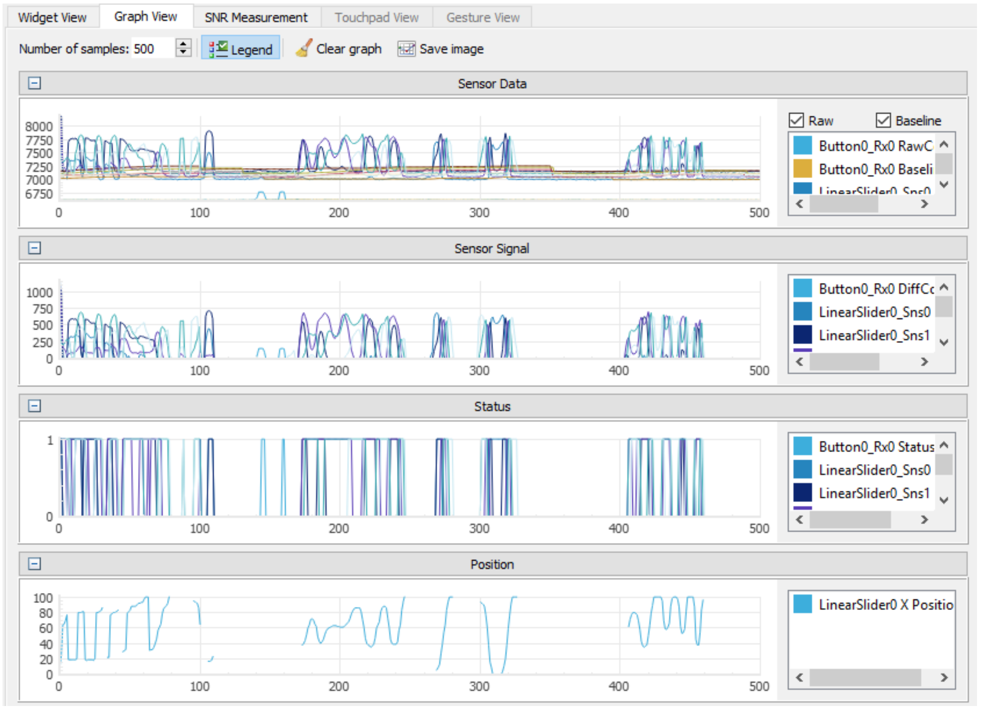
Toolbar

The toolbar provides quick access to different Tuner configuration options that affect the Widget View display.

- **Zoom in** – Zooms in the widgets.
- **Zoom out** – Zooms out the widgets.
- **Graph** – Shows or hides the Touch Signal Graph.
- **Clear graph** – Clears the Touch Signal Graph.
- **Save image** – Opens the dialog to save the Widget View tab as an image. Supported formats: .PNG, .JPG, .BMP.

Graph View Tab

The Graph View displays graphs for selected sensors in the Widget Explorer Pane.



The following charts are available:

- **Sensor Data graph** – Displays raw counts and baseline.
- **Sensor Signal graph** – Displays a signal difference.
- **Status graph** – Displays the sensor status (Touch/No Touch).
- **Position graph** – Displays touch positions for the Linear Slider, Radial Slider, and Touchpad widgets.

Toolbar

The toolbar provides quick access to different Tuner configuration options that affect the Tuner graphs display.

- **Number of samples** – Defines the total amount of data samples shown on a single graph.
- **Show / Hide legend** – Shows or hides the sensor series descriptions (with names and colors) in graphs.
- **Clear graph** – Clears all graphs.
- **Save image** – Opens a dialog to save the Graph View tab as an image. Supported formats: .PNG, .JPG, .BMP.

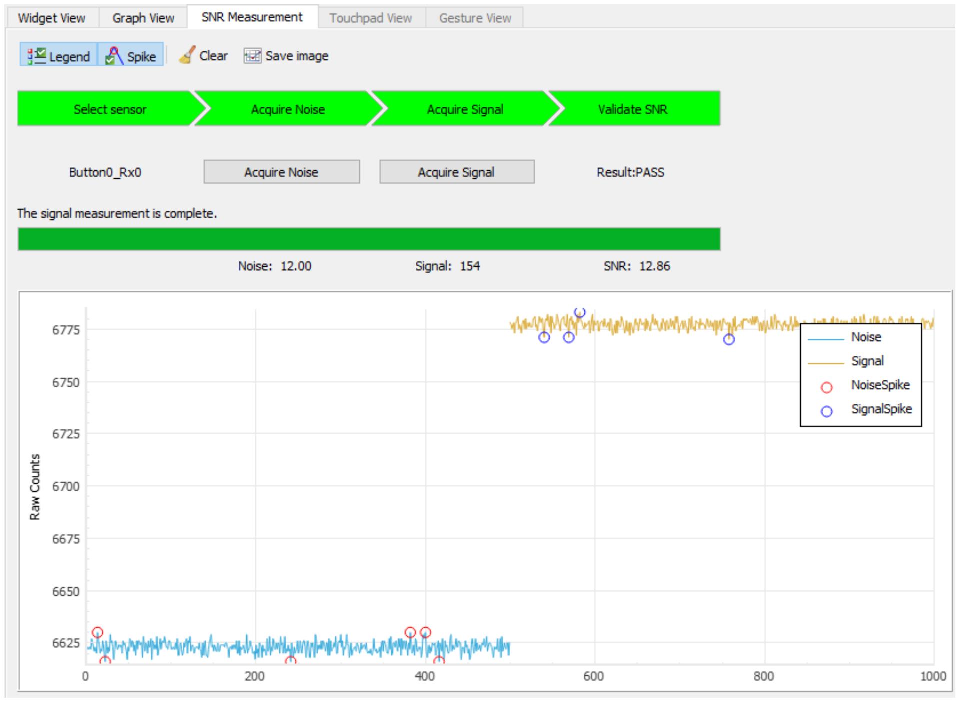
Graph Functionality

The Pan and Zoom features allow you to examine graphs in more detail. Use a mouse drag for Pan. Use the mouse wheel for Zoom. The [**Ctrl**] key + mouse wheel zooms all graphs simultaneously. Press the [**Esc**] key to undo zoom and pan.

Click on a line series to highlight the corresponding series in the legend.

SNR Measurement Tab

The **SNR Measurement** tab allows measuring a SNR (Signal-to-Noise Ratio) for individual sensors. It provides the user interface to acquire noise and signal separately and then calculates a SNR based on the captured data. The obtained value is then validated by a comparison with the required minimum (5 by default, can be configured in the Options dialog).

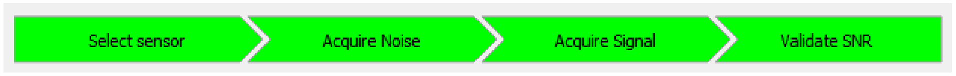


The signal measurement is complete.

Noise: 12.00 Signal: 154 SNR: 12.86

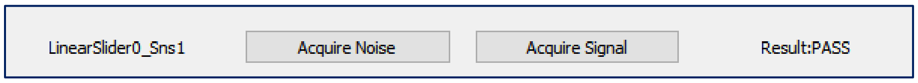
The SNR Measurement tab contains several areas, as follows.

At the top of the **SNR measurement** tab, there is a bar with the status labels. Each label status is defined by its background color.



- **Select sensor** – Green when there is a sensor selected; gray otherwise.
- **Acquire noise** – Green when noise samples are already collected for the selected sensor; gray otherwise.
- **Acquire signal** – Green when signal samples are already collected for the selected sensor; gray otherwise.
- **Validate SNR** – Green when both noise and signal samples are collected, and the SNR is above the valid limit; red when the SNR is below the valid limit, and gray when either noise or signal are not yet collected.

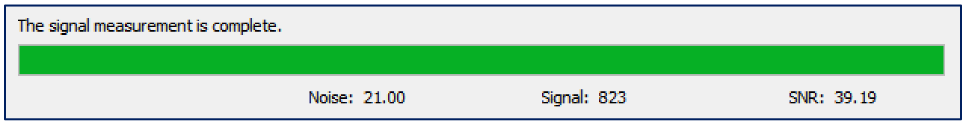
Below the top status labels bar, there are the following controls.



- **Sensor name** – The sensor selected in the Widget Explorer Pane or None (if no sensor selected).
- **Acquire Noise** – This button is disabled when the sensor is not selected or communication is not started. When acquiring noise is in progress, the button can be used to abort the operation.

- **Acquire Signal** – This button is disabled when the sensor is not selected, communication is not started, or noise samples are not yet collected for the selected sensor. When acquiring signal is in progress, the button can be used to abort the operation.
- **Result** – This label shows either N/A (when the SNR cannot be calculated due to noise/signal samples not collected yet), PASS (when the SNR is above the required limit), or FAIL (when the SNR is below the required limit).

Below the controls bar, there is the current status message and the progress of the current operation. Below the progress bar, there are three labels:

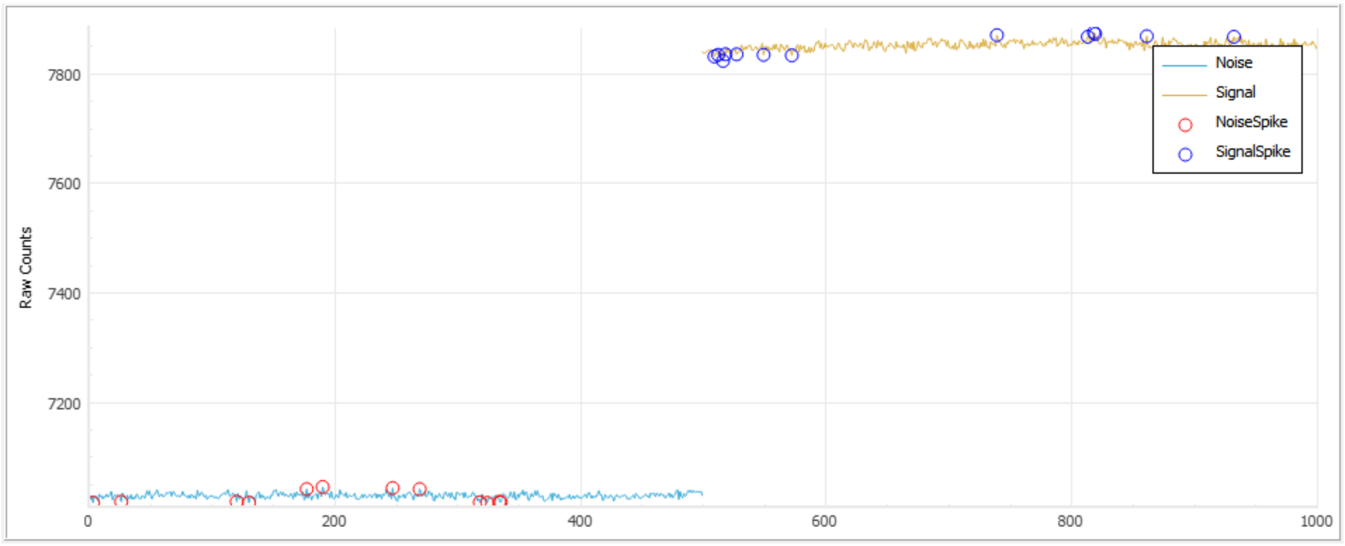


The signal measurement is complete.

Noise: 21.00 Signal: 823 SNR: 39.19

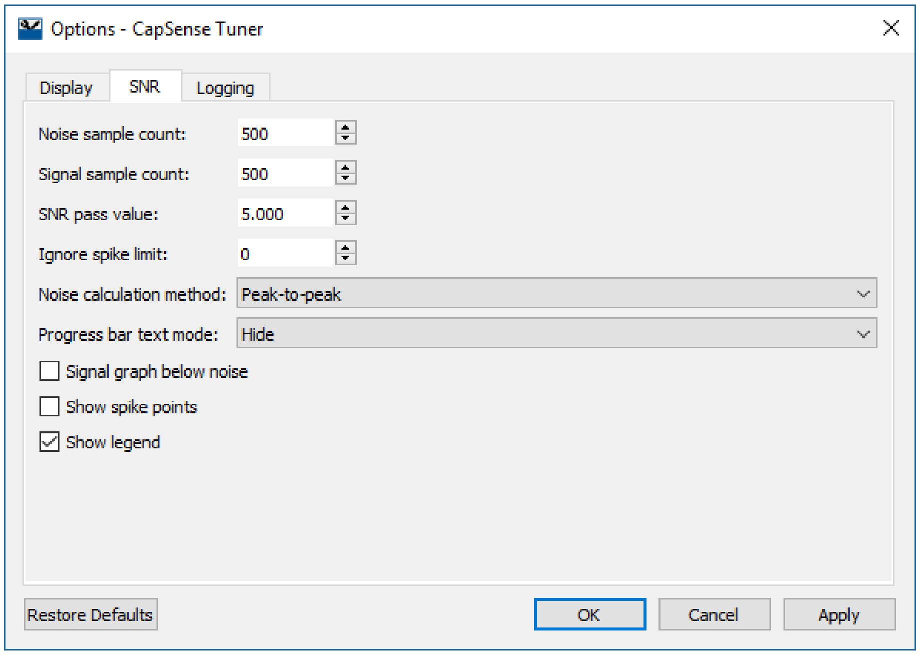
- **Noise** – The label that shows the noise average value calculated during the last noise measurement for the selected sensor, or N/A if no noise measurement is performed yet.
- **Signal** – The label that shows the signal average value calculated during the last signal measurement for the selected sensor, or N/A if no signal measurement is performed yet.
- **SNR** – The label that shows the calculated SNR value. This is the result of the Signal/Noise division rounded up to 2 decimal points. When a SNR cannot be calculated, N/A is displayed instead.

At the bottom there is a chart that displays the measured sensor noise and signal. Noise and signal spikes displayed on the chart are points which are ignored during SNR calculation.



SNR Options

SNR parameters can be modified in the Options dialog. See Tuner Configuration Options for descriptions of other tabs on this dialog.


Options - CapSense Tuner

Noise sample count: 500
Signal sample count: 500
SNR pass value: 5.000
Ignore spike limit: 0
Noise calculation method: Peak-to-peak
Progress bar text mode: Hide
☐ Signal graph below noise
☐ Show spike points
☑ Show legend

- **Noise sample count** – The count of samples to acquire during the noise measurement operation.
- **Signal sample count** – The count of samples to acquire during the signal measurement operation.
- **SNR pass value** – The minimal acceptable value of the SNR.
- **Ignore spike limit** – Ignores a specified number of the highest and the lowest spikes at noise / signal calculation. That is, if you specify number 3, then three upper and three lower raw counts are ignored separately for the noise calculation and for the signal calculation.
- **Noise calculation method** – Allows selecting the method to calculate the noise average. The following methods are available for selection:
  - **Peak-to-peak** (by default) – Calculates noise as a difference between the maximum and minimum value collected during the noise measurement.
  - **RMS** – Calculates noise as a root mean-square of all samples collected during the noise measurement.
- **Progress bar text mode** – The label shown with the progress bar:
  - **Hide** (by default) – No label.
  - **Percent** – The number of samples in percent acquired during the signal measurement operation.
  - **Value** – The number of samples acquired during the signal measurement operation.
- **Signal graph below noise** – Displays the signal series below the noise threshold on the graph.
- **Show spike points** – Highlights the spike points on the graph.
- **Show legend** – Shows the graph legend.

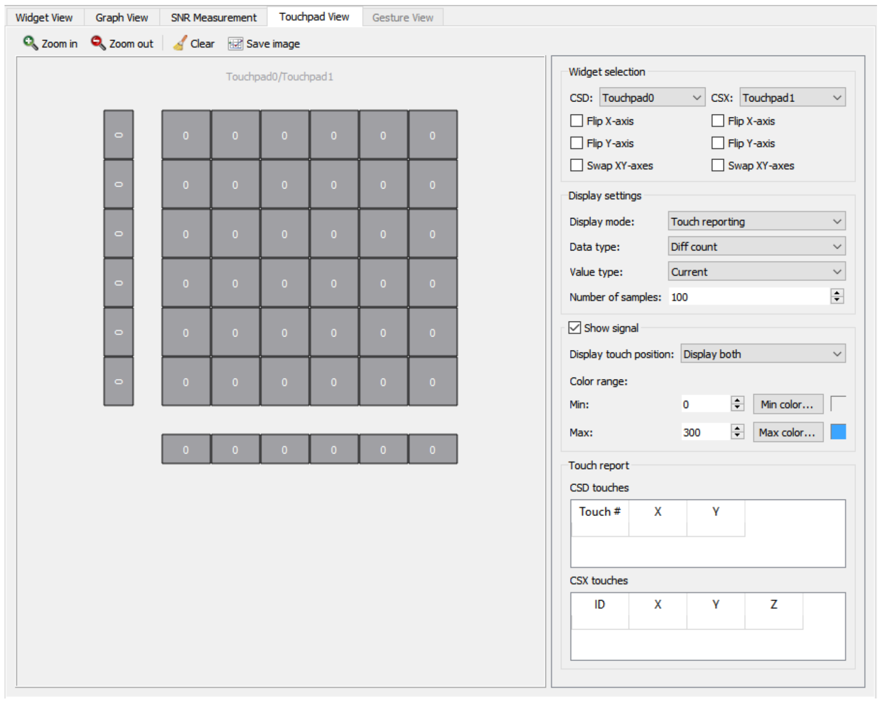
SNR Measurement Procedure

1. Connect to the device and start communication (by pressing **Connect**, then **Start** on the toolbar).
2. Switch to the **SNR Measurement** tab.
3. Select a sensor in the Widget Explorer Pane with a click.
4. Make sure no touch is present on the selected sensor.
5. Press **Acquire Noise** and wait for the required count of samples to be collected.
6. Observe the Noise label is updated with the calculated noise average value.
7. Touch the selected sensor to produce signal on it.
8. Press **Acquire Signal** and wait for the required count of samples to be collected.
9. Observe the Signal label is updated with the calculated signal average value
10. Observe the SNR label is updated with the SNR (signal-to-noise ratio).

Touchpad View Tab

This tab provides a visual representation of signals and positions of a selected touchpad widget in the heatmap form. Only one CSD and one CSX touchpad can be displayed at a time.

Note The **Touchpad View** tab is not visible when there are no touchpad widgets in the configuration.



Widget selection

CSD: Touchpad0 CSX: Touchpad1

☐ Flip X-axis ☐ Flip X-axis
☐ Flip Y-axis ☐ Flip Y-axis
☐ Swap XY-axes ☐ Swap XY-axes

Display settings

Display mode: Touch reporting
Data type: Diff count
Value type: Current
Number of samples: 100

☑ Show signal
Display touch position: Display both
Color range:
Min: 0 Min color...
Max: 300 Max color... 🟦

Touch report

CSD touches

Touch #	X	Y

CSX touches

ID	X	Y	Z

Toolbar

- **Zoom in** – Zooms in the Touchpad widget.
- **Zoom out** – Zooms out the Touchpad widget.
- **Clear** – Clears the touchpad.

- **Save image** – Opens a dialog to save the Touchpad View tab as an image. Supported formats: .PNG, .JPG, .BMP.

Widget Selection

Consists of the configuration options for mapping the physical touchpad orientation to the identical representation in the heatmap:

- **CSD combo box** – Selects any CSD touchpad to be displayed in the heatmap.
- **CSX combo box** – Selects any CSX touchpad to be displayed in the heatmap.
- **Flip X-axis** – Flips the displayed X-axis to match orientation of physical touchpad.
- **Flip Y-axis** – Flips the displayed Y-axis to match orientation of physical touchpad.
- **Swap XY-axes** – Swaps the X- and Y-axes for the touchpad.

Display settings

Manages heatmap data that to be displayed. These options are applicable for a CSX touchpad only.

- **Display mode** – The drop-down menu with 3 options for the display format:
  - **Touch reporting** – Shows the current detected touches only.
  - **Line drawing** – Joins the previous and current touches in a continuous line.
  - **Touch Traces** – Plots all the reported touches as dots.
- **Data type** – The drop-down menu to select the signal type to be displayed: Diff count, Raw count, Baseline.
- **Value type** – The drop-down menu to select the type of a value to be displayed:
  - **Current** – the last received value.
  - **Max hold** – the maximum value out of latest ‘Number of samples’ received.
  - **Min hold** – the minimum value out of latest ‘Number of samples’ received.
  - **Max-Min** – difference between maximum and minimum values.
  - **Average** – average value of latest ‘Number of samples’ received.
- **Number of samples** – Defines a length of history of data for the **Line Drawing**, **Touch Traces**, **Max hold**, **Min hold**, **Max-Min** and **Average** options.

Show signal

Enables displaying data for each sensor if checked, otherwise displays only touches. This option is applicable for the CSX touchpad only.

- **Display touch position** – Selects the touchpad from which data is displayed in heatmap. The three options:
  - Display only CSX
  - Display only CSD
  - Display both
- **Color range** – Defines a range of sensor signals within which the color gradient is applied. If a sensor signal is outside of the range, then a sensor color is either minimum or maximum out of the available color palette.

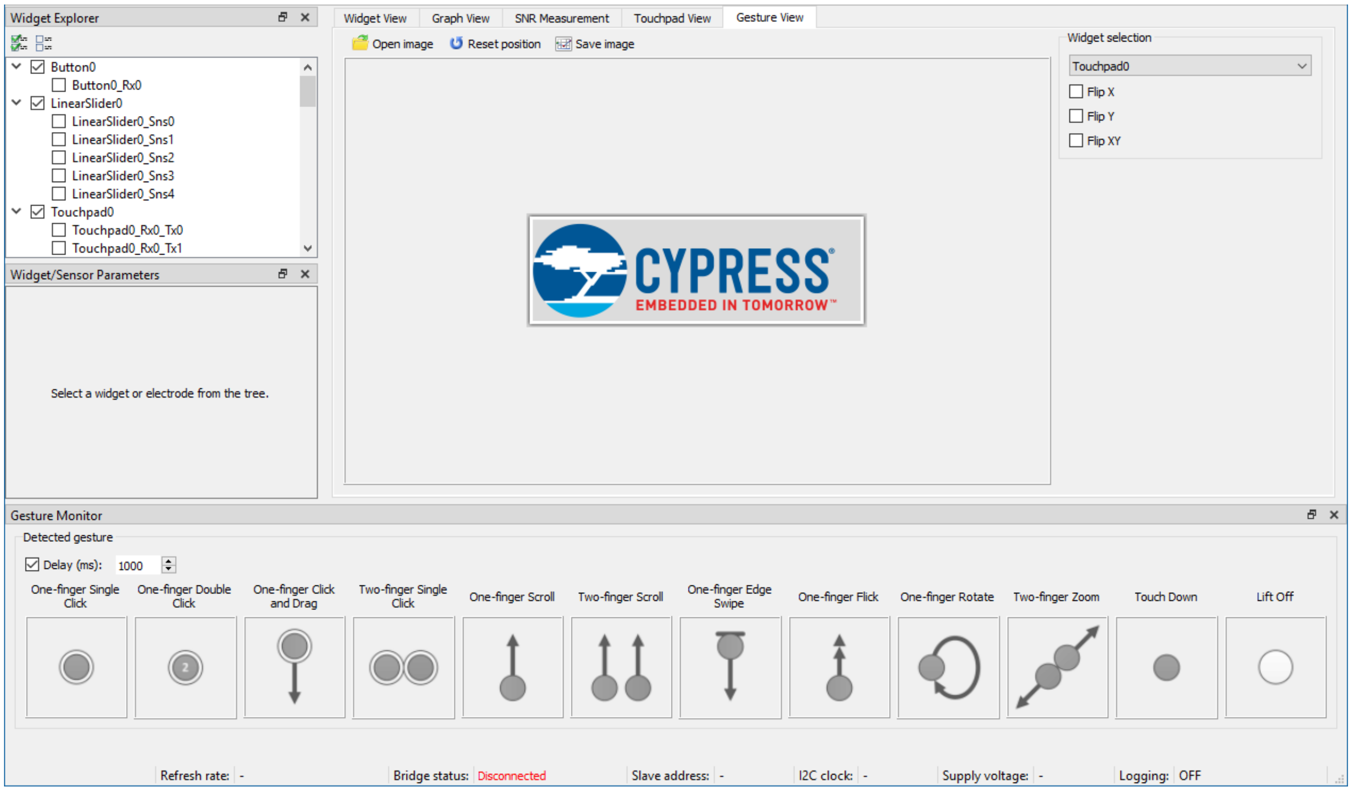
Touch report

- **CSD touches table** – Displays the current X and Y touch position (Z value is always 0) of the detected touches on the CSD touchpad. If the CSD touchpad is neither configured nor touch is detected, the touch table is empty. When two-finger detection is enabled for a CSD touchpad, then two touch positions are reported.
- **CSX touches table** – Displays the current X and Y touch position and Z values (amplitude) of the detected touches on the CSX touchpad. If the CSX touchpad is neither configured nor touches is detected, the touch table is empty. The middleware supports simultaneous detection up to three touches for a CSX touchpad touch, so the touch table displays all the detected touches.

Gesture View Tab

This tab provides a visual representation for evaluation and tuning of gestures. This tab can display gestures from one widget at a time.

Note The **Gesture View** tab is not visible when there are no gesture widgets in the configuration.



Note Use of Synchronized communication mode or UART communication is recommended for Gesture validation, to make sure no gesture events such as a touchdown or lift off is missed during communication.

Toolbar

- **Open image** – Opens a custom image instead of the Cypress logo.
- **Reset position** – Resets the image position and zoom. The image is moved to the center of the panel.
- **Save image** – Opens a dialog to save the Gesture View tab as an image. Supported formats: .PNG, .JPG, .BMP.

Widget Selection

Allows selecting a widget and controls that the display in the Tuner matches the orientation physical widget on hardware.

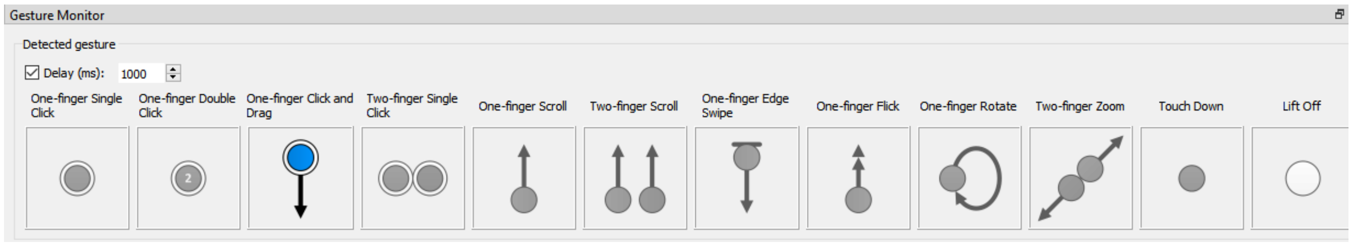
- **Combo box** – Selects the widget with Gesture enabled to display the Gesture from it on this pane.
- **Flip X** – Flips the direction of the X-axis to match orientation of physical widget.
- **Flip Y** – Flips the direction of the Y-axis to match orientation of physical widget.
- **Flip XY** – Swaps the X- and Y-axes to match orientation of physical widget.

Image Pane

The image with Cypress logo reacts on the detected gestures (scroll and zoom) and moves around the pane. The image can be changed using the **Open image** tool button.

Detected Gesture

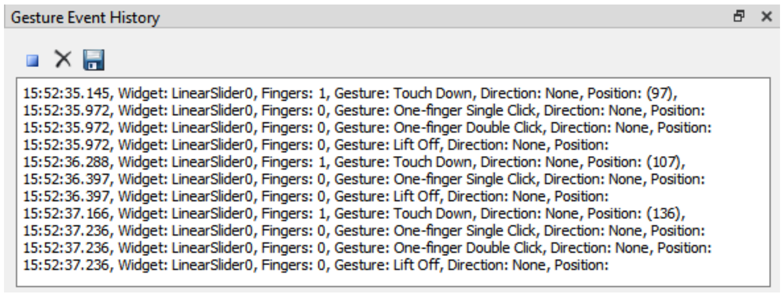
Provides visual indication for a detected Gesture.



If the delay checkbox is enabled, a Gesture picture is displayed only for the specified time-interval. If disabled, the last reported gesture picture is displayed until a new Gesture is reported.

Gesture Event History

Logs the detected gestures information.

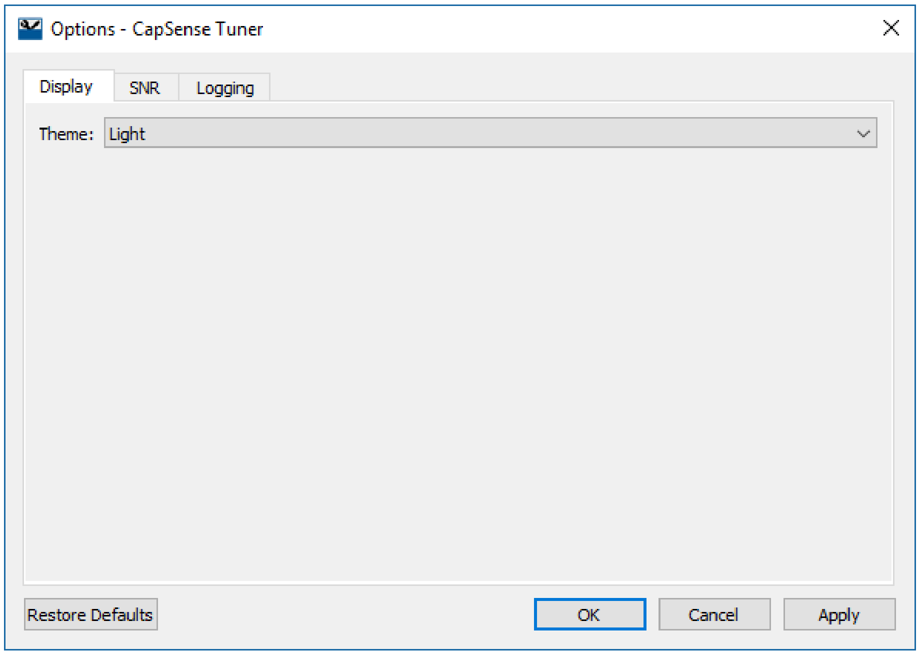


```
15:52:35.145, Widget: LinearSlider0, Fingers: 1, Gesture: Touch Down, Direction: None, Position: (97),
15:52:35.972, Widget: LinearSlider0, Fingers: 0, Gesture: One-finger Single Click, Direction: None, Position:
15:52:35.972, Widget: LinearSlider0, Fingers: 0, Gesture: One-finger Double Click, Direction: None, Position:
15:52:35.972, Widget: LinearSlider0, Fingers: 0, Gesture: Lift Off, Direction: None, Position:
15:52:36.288, Widget: LinearSlider0, Fingers: 1, Gesture: Touch Down, Direction: None, Position: (107),
15:52:36.397, Widget: LinearSlider0, Fingers: 0, Gesture: One-finger Single Click, Direction: None, Position:
15:52:36.397, Widget: LinearSlider0, Fingers: 0, Gesture: Lift Off, Direction: None, Position:
15:52:37.166, Widget: LinearSlider0, Fingers: 1, Gesture: Touch Down, Direction: None, Position: (136),
15:52:37.236, Widget: LinearSlider0, Fingers: 0, Gesture: One-finger Single Click, Direction: None, Position:
15:52:37.236, Widget: LinearSlider0, Fingers: 0, Gesture: One-finger Double Click, Direction: None, Position:
15:52:37.236, Widget: LinearSlider0, Fingers: 0, Gesture: Lift Off, Direction: None, Position:
```

Tuner Configuration Options

The Tuner application allows setting different configuration options with the Options dialog. Settings are divided into groups:

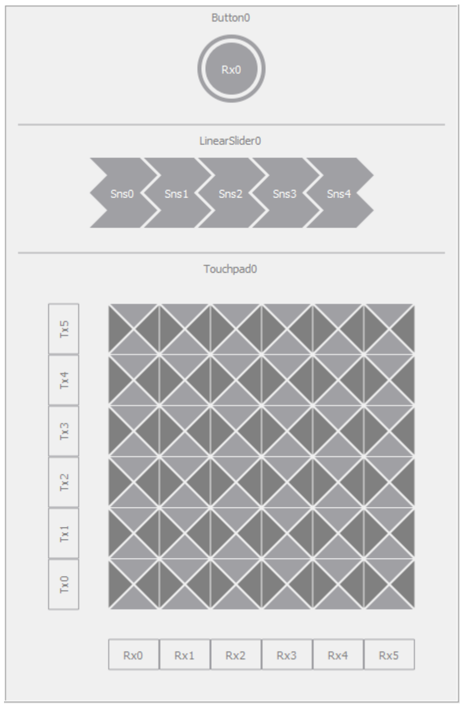
Display Options



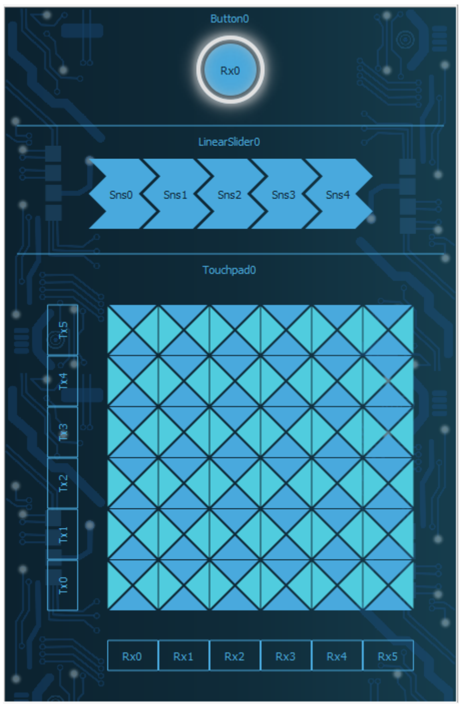
Theme: Light

- **Theme** – Defines the visual style of the widgets.

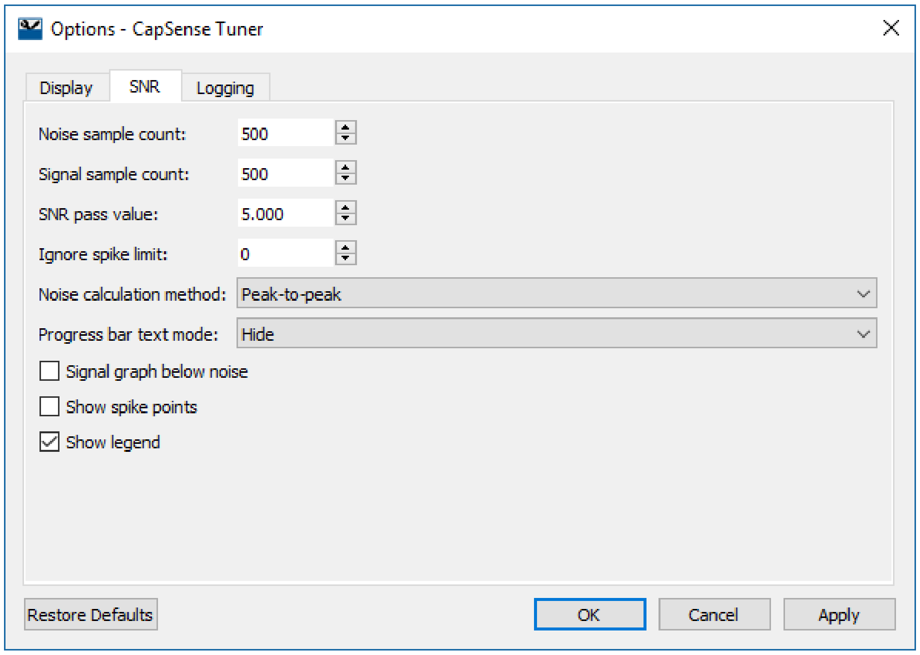
Light theme



Dark theme



SNR Options



Noise sample count: 500

Signal sample count: 500

SNR pass value: 5.000

Ignore spike limit: 0

Noise calculation method: Peak-to-peak

Progress bar text mode: Hide

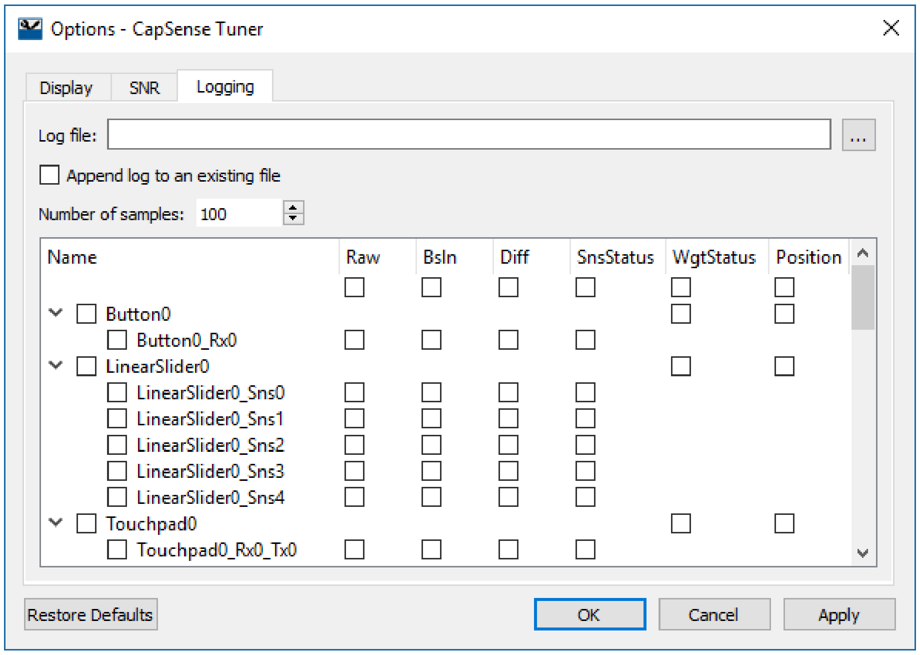
☐ Signal graph below noise

☐ Show spike points

☑ Show legend

See SNR Options section for a description of this tab.

Logging Options



Log file:

☐ Append log to an existing file

Number of samples: 100

Name	Raw	Bsln	Diff	SnsStatus	WgtStatus	Position
	☐	☐	☐	☐	☐	☐
☐ Button0					☐	☐
☐ Button0_Rx0	☐	☐	☐	☐		
☐ LinearSlider0					☐	☐
☐ LinearSlider0_Sns0	☐	☐	☐	☐		
☐ LinearSlider0_Sns1	☐	☐	☐	☐		
☐ LinearSlider0_Sns2	☐	☐	☐	☐		
☐ LinearSlider0_Sns3	☐	☐	☐	☐		
☐ LinearSlider0_Sns4	☐	☐	☐	☐		
☐ Touchpad0					☐	☐
☐ Touchpad0_Rx0_Tx0	☐	☐	☐	☐		

- **Log File** – Selects the file for information to be stored and its location. The log file extension is *csv*.
- **Append log to an existing file** – When checked, the selected file is never over-written and the file is expanded with new data. When this option is not selected, the file is overwritten.
- **Number of samples** – Defines a log session duration in samples.
- **Data configuration check box table** – Selects data to be collected into a log file.

References

Refer to the following documents for more information, as needed. All of these documents are available from the IDE **Help** menu and **Quick Panel**.

- CapSense Configurator Guide
- Device Configurator Guide
- ModusToolbox IDE User Guide
- CapSense Middleware Documentation
- PDL API Reference Guide
- Device Datasheets
- Device Technical Reference Manuals

Troubleshooting

Problem	Workaround
On common Linux distributions, the serial UART ports (usually /dev/ttySx or /dev/ttyUSBx devices) belongs to the root user and to the dialout group. Standard users are not allowed to access these devices.	An easy way to allow the current user access to the Linux machine's serial ports is by adding the user to the dialout group. This can be done using the following command: `$sudo usermod -a -G dialout $USER` **Note** For this command to take effect, you must log out and then log back in.
On Linux, attempts to set up or start the CapSense Tuner communication leads to the Tuner closing without any messages.	To enable Tuner communication under Linux, install the udev rules for KitProg3: • Disconnect the KitProg device. • Execute in the terminal (root access required): `sh $CYSDK/tools/fw-loader/udev_rules/install_rules.sh` • Reconnect the KitProg device.
On macOS, the CapSense Tuner can't be launched from the Launchpad.	Launch the Tuner from within the Device Configurator or use the command line option.
On common Linux distributions, the Tuner forbids communication protocol selection after re-plugging KitProg during communication.	Refer to the "Installation Procedure on Ubuntu Linux (x64)" section in the *Cypress Programmer 2.1 CLI User Guide*.

Note The Tuner provides information about the root cause of connection to KitProg in an operation log. The operation log is located in:

Windows: *<user_home>\AppData\Local\Cypress Semiconductor Corporation\capsense-tuner\capsense-tuner.log*

Linux: */home/<user_home>/.config/Cypress Semiconductor Corporation/capsense-tuner/capsense-tuner.log*

macOS: */Users/<user_home>/Library/Preferences/Cypress Semiconductor Corporation/capsense-tuner/capsense-tuner.log*

Version Changes

This section lists and describes the changes for each version of this tool.

Version	Change Descriptions
1.0	New tool.
1.1	Added UART interface.
	Added possibility to save different views as images.
	Added zoom and pan functionality for graphs.
	Fixed minor issues.