ModusToolbox™ LiveMonitor

User guide

ModusToolbox™ LiveMonitor version v1.0.0

About this document

Scope and purpose

ModusToolbox™ LiveMonitor is a tool provided as part of ModusToolbox™ ecosystem, that helps engineers to evaluate, run and optimize quickly embedded applications. This document is a user guide for LiveMonitor and helps developers understand how to use LiveMonitor in depth and learn all advanced features provided with it.

Intended audience

This guide targets two reader groups:

- End users — engineers or technicians who operate the GUI to monitor, measure, and analyze signal data from an embedded device.
- Firmware engineers — developers who use a ModusToolbox™ embedded application and who integrate the mtb-oscilloscope middleware library into a ModusToolbox™ embedded application

Table of contents

About this document **1**
Table of contents **2**
1 Introduction **4**
2 Prerequisites **5**
2.1 LiveMonitor general prerequisites 5
2.2 Firmware integration prerequisites for LiveMonitor Oscilloscope functionality 5
3 Getting started **6**
4 Creating a project **10**
5 LiveMonitor Workspace **11**
5.1 Editing Workspace 12
5.1.1 Components 14
5.1.2 Using formula attributes 19
5.2 Taskbar 25
5.3 Tabs 27
5.4 Run mode 27
6 LiveMonitor Oscilloscope **29**
6.1 Product overview 29
6.1.1 GUI feature summary 29
6.1.2 Firmware (mtb-oscilloscope) feature summary 30
6.2 Firmware integration: mtb-oscilloscope middleware 30
6.2.1 Architecture overview 30
6.2.2 Repository layout 30
6.2.3 Installation 31
6.2.4 Minimal integration example 32
6.2.5 Remote and runtime configuration 33
6.2.6 Memory Consumption 33
6.3 Interface Overview 33
6.4 Using the Oscilloscope 34
6.4.1 Scope Settings 35
6.4.2 Channel Settings 35
6.4.3 Scope Modes 37
6.4.4 Trigger configuration 37
6.4.5 Sampling clock and timebase 38
6.4.6 Cursor measurements 39
6.4.7 Chart panel 43
6.5 Chart navigation 44
6.5.1 How to zoom into a region of interest 45
6.5.2 Chart display tools 45
6.5.3 Auto Scale buttons 47
6.5.4 Scale Inputs 48
6.5.5 Horizontal Axis 48
6.5.6 Taskbar 49
6.5.7 Data Persistence 50
6.6 Troubleshooting 51
6.6.1 GUI troubleshooting 51
6.6.2 Firmware integration troubleshooting 51

Table of contents

Glossary **51**
Revision history **53**
Disclaimer **54**

Table of contents

1 Introduction

ModusToolbox™ LiveMonitor is a desktop tool part of ModusToolbox™ ecosystem which provides advanced GUI design features within a few mouse-clicks, for high-speed monitoring and commanding of embedded applications. LiveMonitor uses standard debug interface requiring no extra hardware and no firmware changes to the target application for the main functionality (exception for oscilloscope function).
It is a great tool to enhance the evaluation experience of any application including the wide range of evaluation boards and reference designs that Infineon offers.

ModusToolbox™ LiveMonitor is comprised of two main functionalities in the same tool:

- **LiveMonitor Workspace - GUI dashboard or GUI builder**:
  - Drag & drop style UI designer to build custom interface
  - Toolbox with several components (buttons, widgets, charts…etc.) to display and update parameters, variables
  - ELF parsing for the selected file
  - Multiple tab support in a single project
  - Protected save for sharing projects as executable, non-editable GUI
  - Design and run modes to create and test the workspaces
  - Project saving and loading
- **LiveMonitor Oscilloscope**: monitor selected signals in real time using the built-in oscilloscope to visualize and analyze signal behavior – **requires mt-oscilloscope middleware**
  - Multi-channel arbitrary selection within all variables of source code
  - Multiple triggering modes: continuous, single trigger, triggered, stop
  - Measurement cursors and statistics
  - Export option for further processing

2 Prerequisites

2.1 LiveMonitor general prerequisites

The following conditions must be met:

Requirement	Details
J-Link Driver	The driver must be installed. GUI contains the driver and provides an installation step if not detected.
Connected target device	The embedded hardware target must be powered, connected via J-Link supported SWD/JTAG probe, and running firmware (with ‘mtb-oscilloscope’ only if oscilloscope features are used).

2.2 Firmware integration prerequisites for LiveMonitor Oscilloscope functionality

The following tools and access rights are required before integrating the mtb-oscilloscope middleware library required to use the oscilloscope function of LiveMonitor:

Requirement	Details
ModusToolbox™ IDE or CLI	Version 3.x or later. Library Manager must be available (bundled with IDE or via "make library-manager").
Supported toolchain	GCC ARM, ARM Compiler 6, or IAR. See the library Release Notes for the current list of supported versions.
Target device	Any Infineon PSOC™ / XMC™ device supported by ModusToolbox™ with at least one periodic timer source.
Host–device connection	J-Link based SWD/JTAG debug probe for direct memory access. Often provided as on-board debugger in Infineon evaluation boards

3 Getting started

The GUI tool is delivered through the Infineon Developer Center, which allows you to install and update ModusToolbox™ LiveMonitor. If Infineon Developer Center is not already installed, download it: https://www.infineon.com/cms/en/design-support/tools/utilities/infineon-developer-center-idc-launcher
After opening Infineon Developer Center and signing in with your My Infineon account, locate ModusToolbox™ LiveMonitor and click Install (Figure 1).

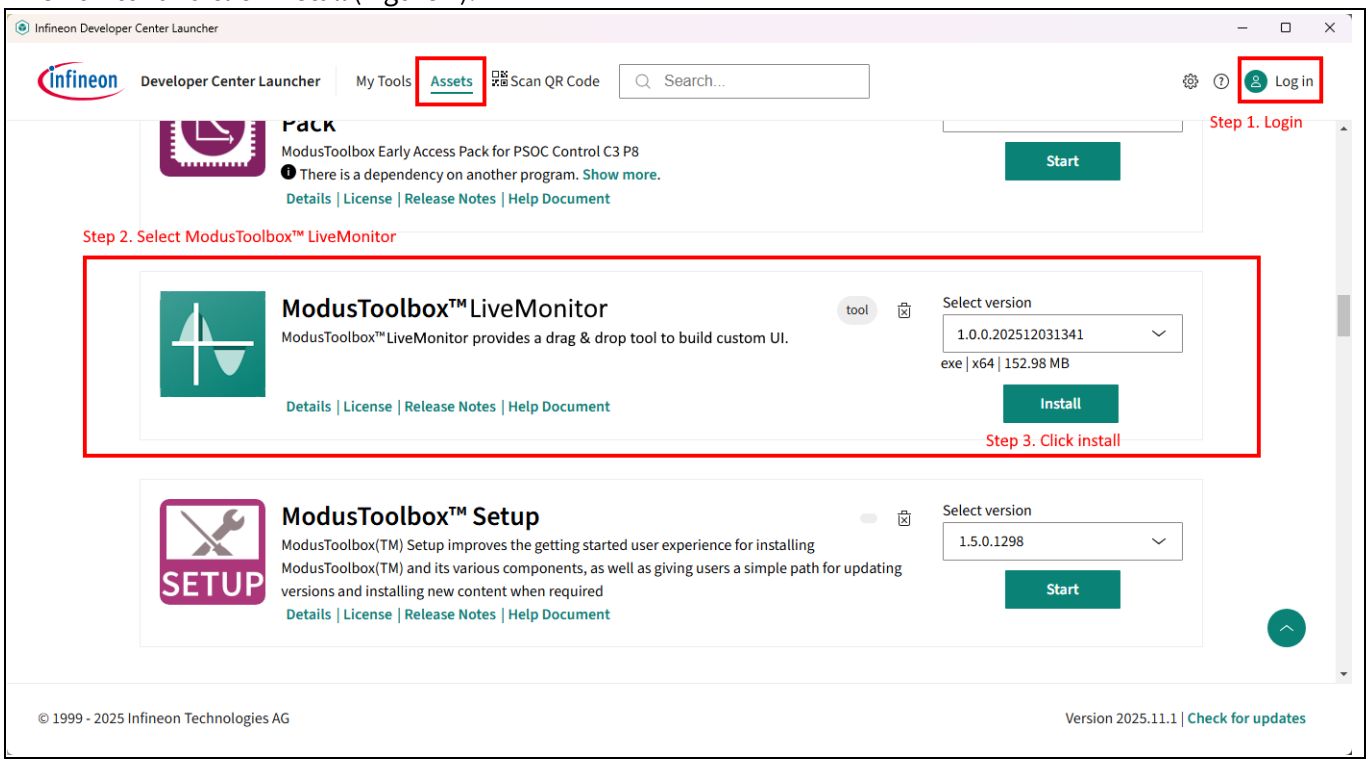


Figure 1 Infineon Developer Center

Once installation is complete, click Start to launch ModusToolbox™ LiveMonitor **(Figure 2 and Figure 3).**

Table of contents

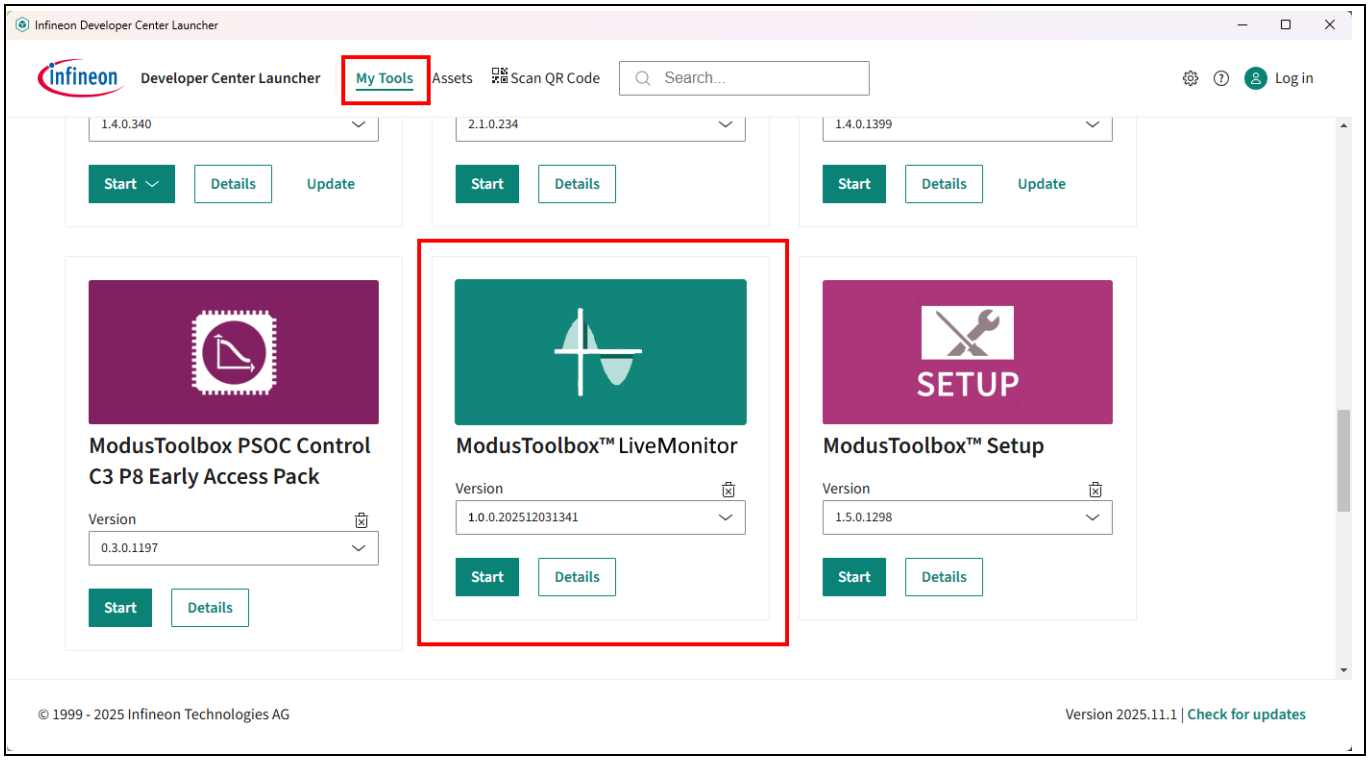


Figure 2 ModusToolbox™ LiveMonitor in the developer center after installation

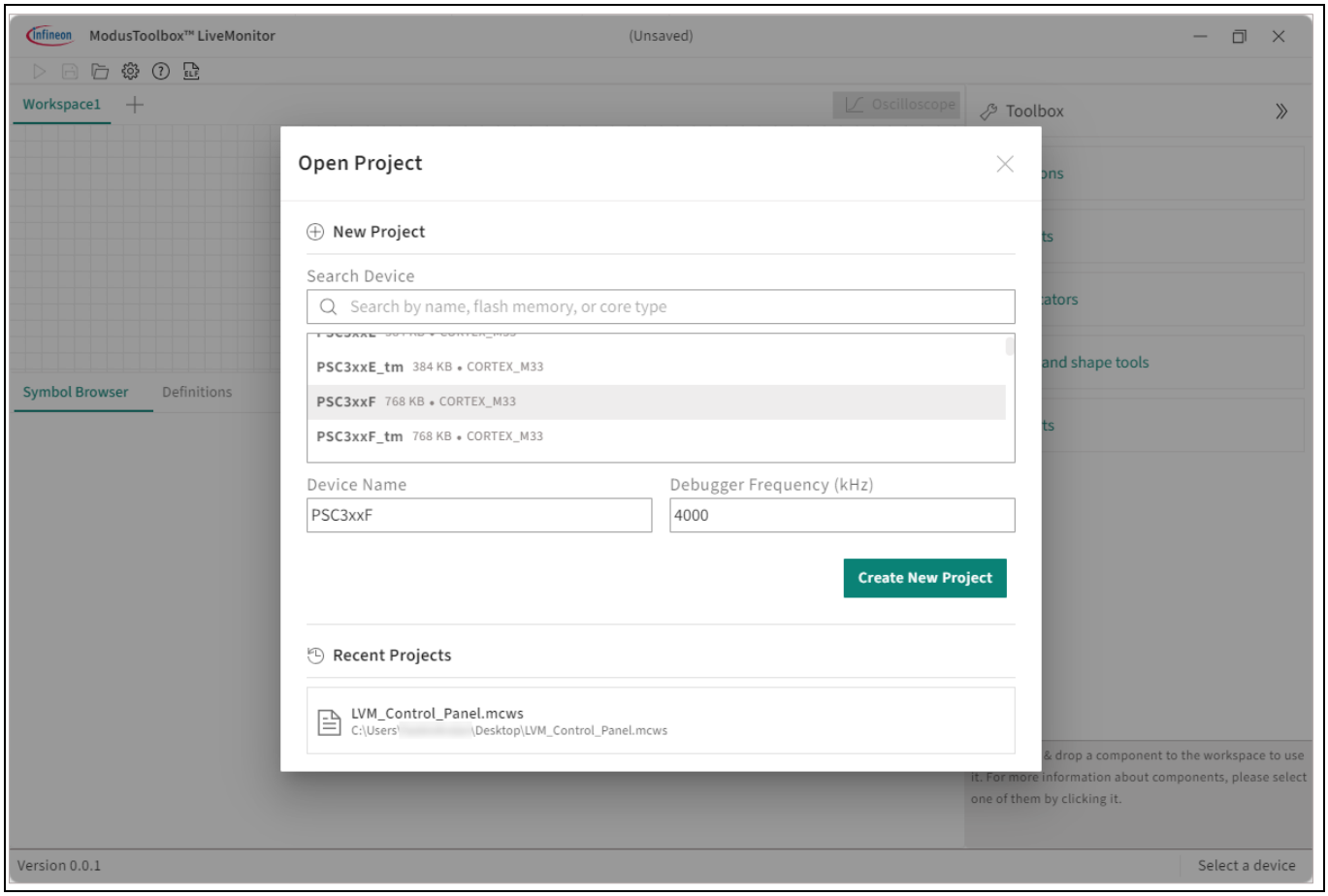


Figure 3 Getting started wizard of LiveMonitor

Another option is to install LiveMonitor using the ModusToolbox™ Setup Tool. In the Setup Tool, select LiveMonitor and click Next (see Figure 4). Follow the instructions to install and launch the GUI.

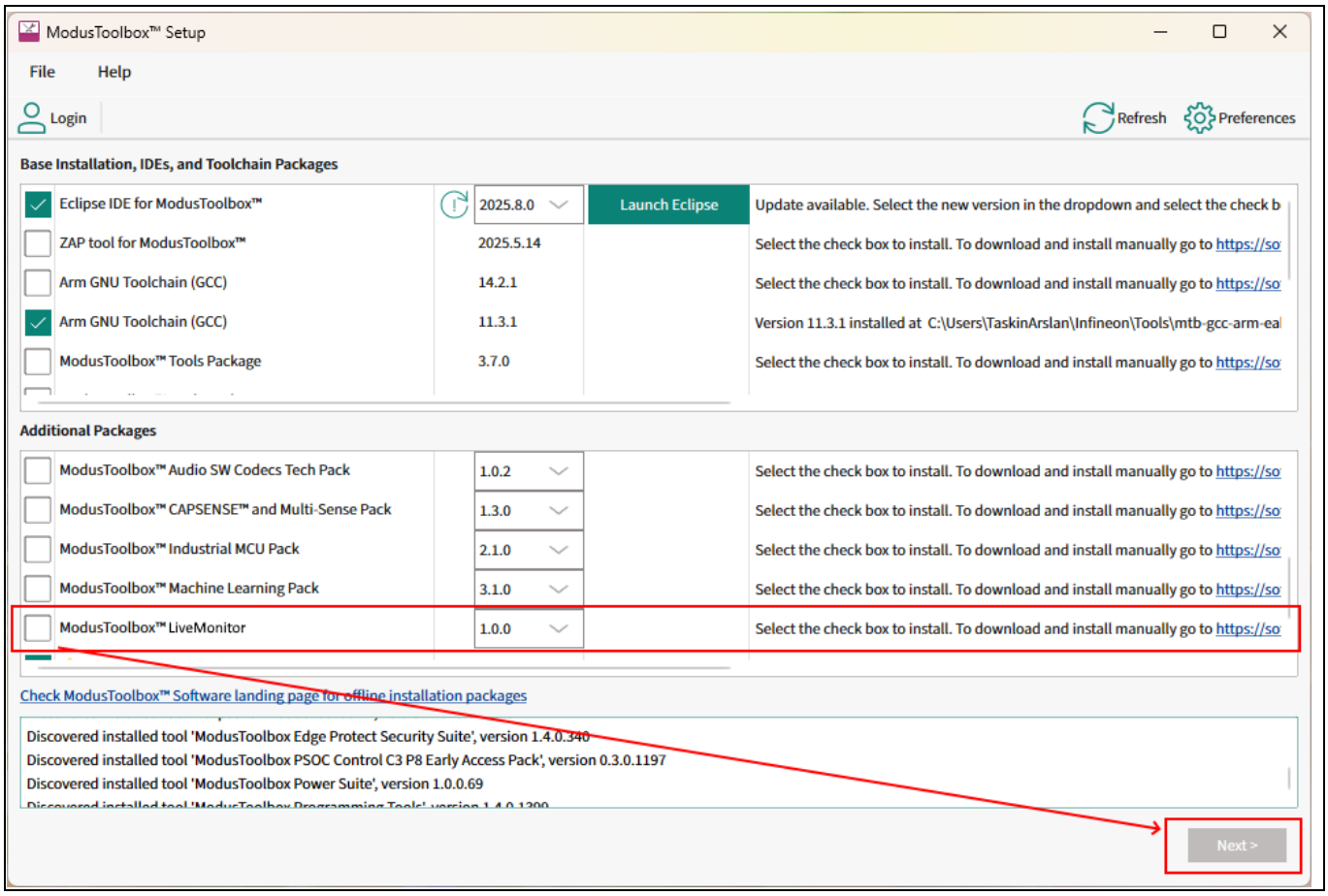


Figure 4 Install LiveMonitor using ModusToolbox™ Setup Tool

ModusToolbox™ LiveMonitor has no additional software dependencies. However, after the first installation, the GUI checks for the J-Link USB driver. If the driver is missing, the GUI displays a pop-up window requesting installation confirmation (Figure 5). You must install this driver to communicate with the board; without it, device communication will not function.

Table of contents

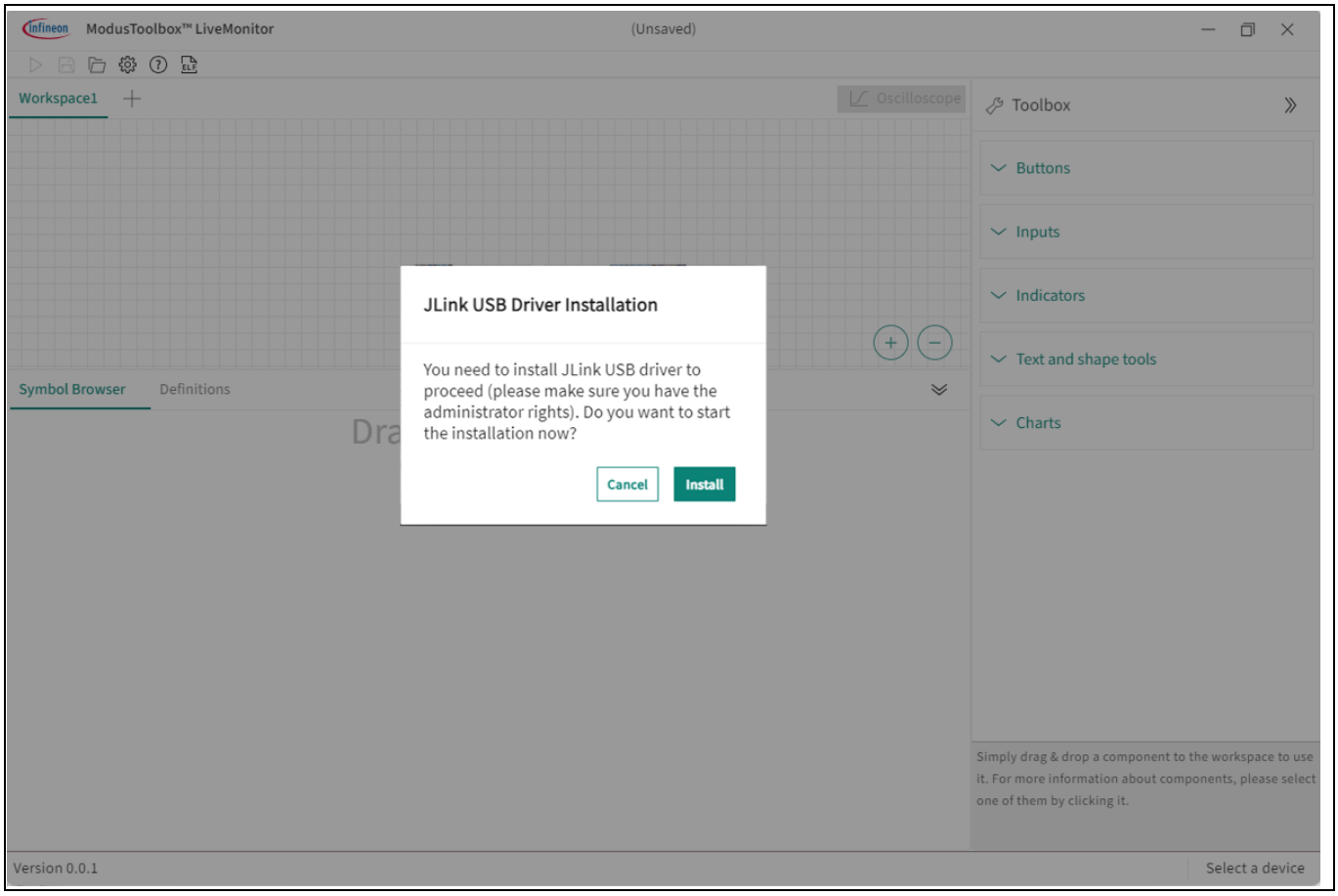


Figure 5 JLink driver must be installed to use the LiveMonitor

4 Creating a project

As the first step, connect the board to the PC using the USB cable. After you connect and power the board on, it will be displayed in the footer as “J-Link *SN*” (Figure 6), where SN is the serial number of the connected debugger. If you power the board on after opening the GUI, you need to select the serial number (SN) using the dropdown at the bottom-right corner of the GUI. To display the dropdown, simply click on the “Select a device” text.

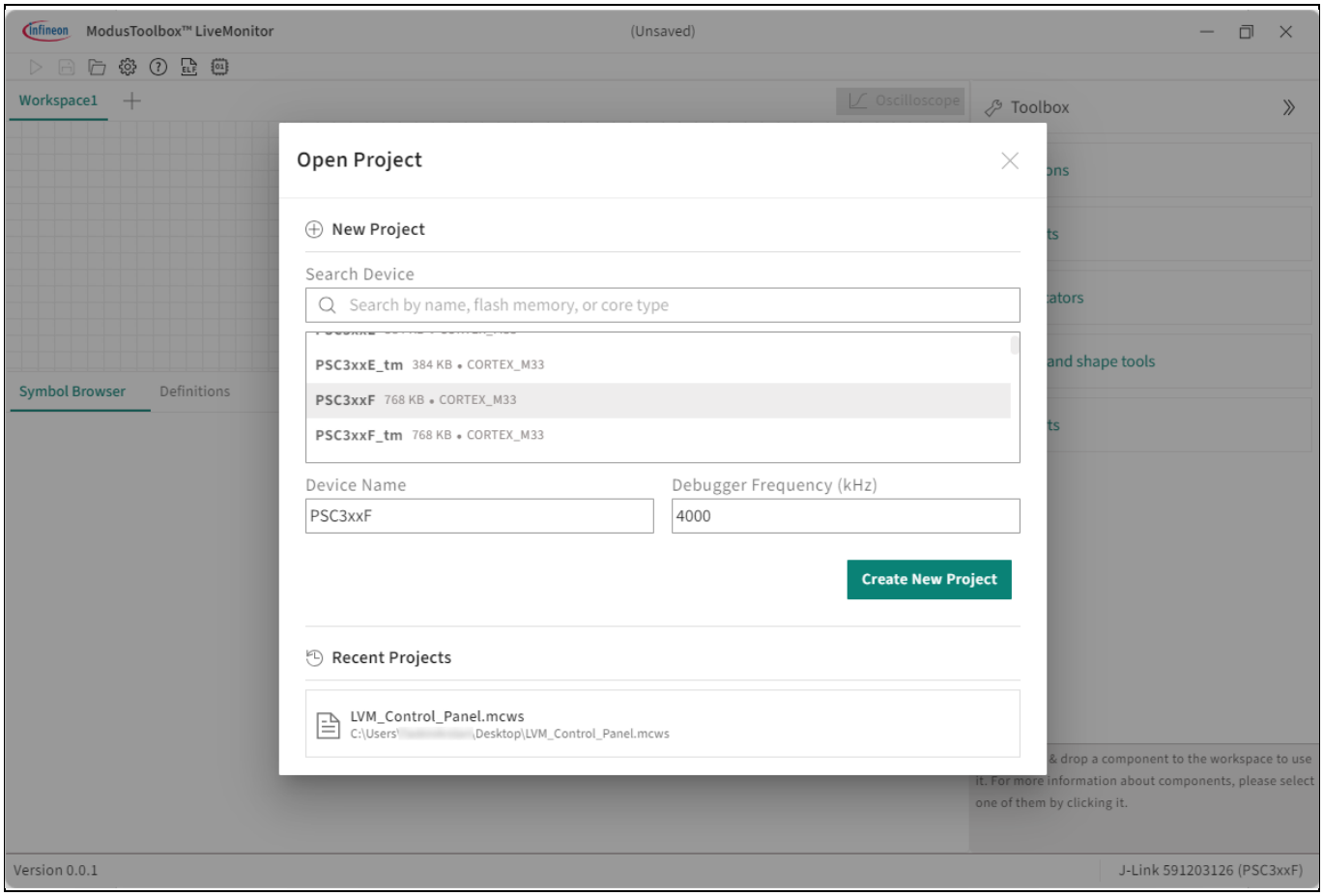


Figure 6 Board connection information

If there are multiple boards connected to the PC, the GUI connects to the first board it detects. If the selected board is not the board you want to work with, you need to select the correct board manually. If you need to clear the selection, select “None" option in the list.

You need to select the correct device name to open a connection to the device. Use search device box to search for a device in the startup wizard. You can change the device name later in the preferences popup whether by selecting one of the limited options, or by typing the device name. You can change the default debugger frequency, but if the debugger does not support it, it will fall back to the default frequency.

After configuring device specific information, you can either create a new project or open an existing project by selecting the project from the Recent Projects section.

5 LiveMonitor Workspace

ModusToolbox™ LiveMonitor is a graphical user interface (GUI) for users to create their own interfaces. It allows users to create custom interfaces using the components from its library.

After opening a new project, an empty workspace will be opened (Figure 7).

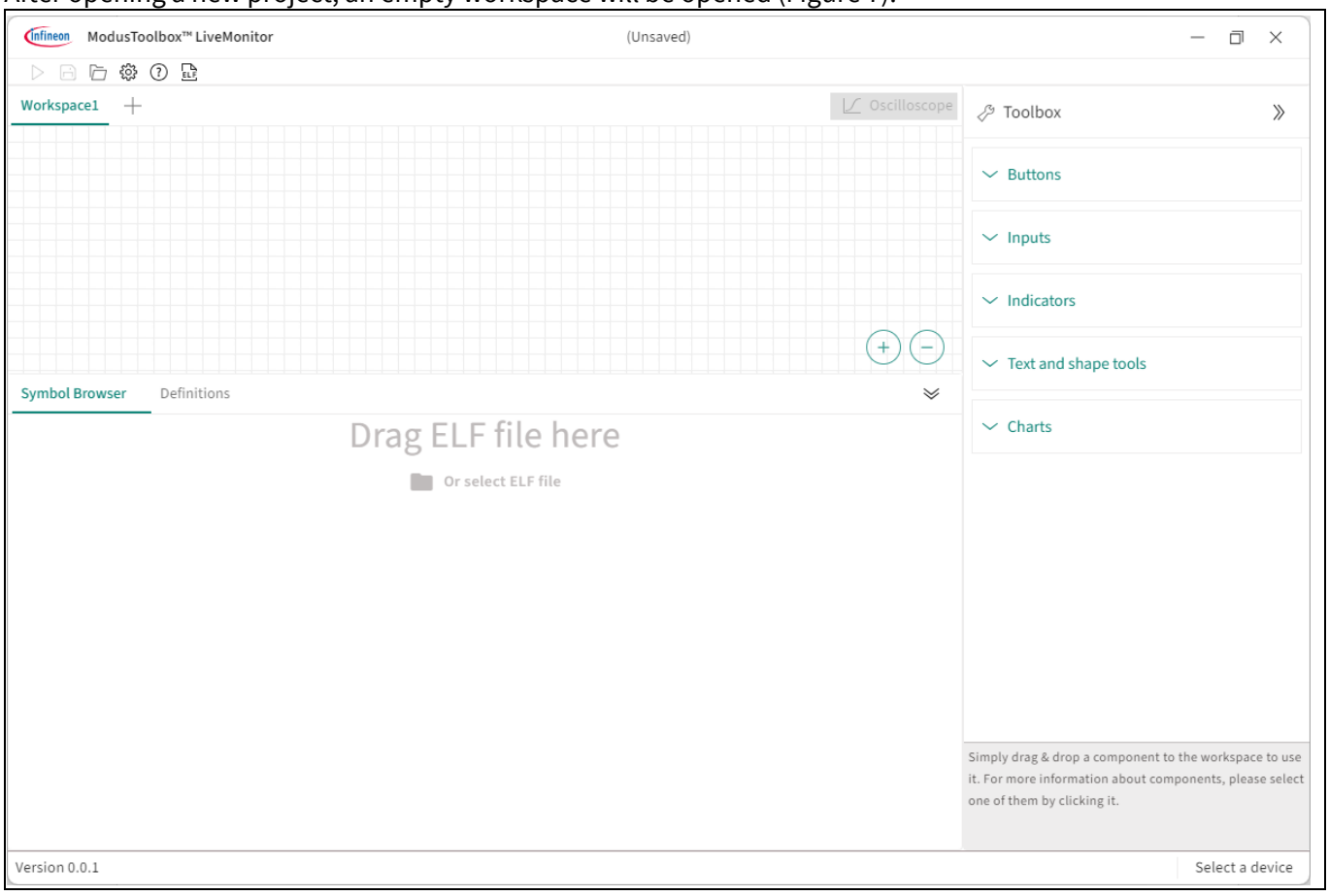


Figure 7 LiveMonitor

There are 3 main sections:

1. Workspace
2. Toolbox
3. Symbol browser

Workspace is the canvas where user can shape with the provided tools. Toolbox contains draggable components library and symbol browser is where a symbol table is displayed.
The are 2 modes:

1. Design mode
2. Run mode

In **Design Mode**, create the user interface by placing components in the workspace. In **Run Mode**, GUI removes the Toolbox and Symbol browser, with everything related to editing. It starts polling the data from the device and all components start doing their intended job.
Use the LiveMonitor without a connected device if you don't have a supported device yet. You can create projects, edit them and save them, but you cannot test the interface. Run mode can only be activated when a supported device is connected to the computer.

5.1 Editing Workspace

LiveMonitor starts directly in design mode for new projects. In design mode, add/remove/update components from the right-hand-side toolbox. Tools are grouped under different names. After one of the groups expanded, select a tool, drag and drop to the workspace area (Figure 8).

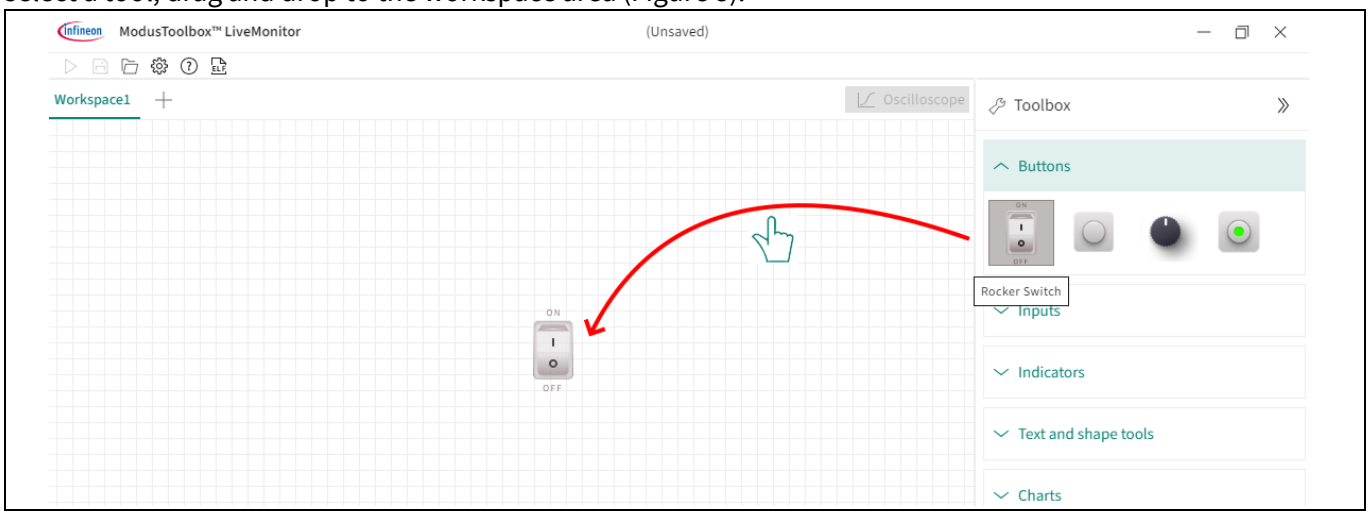


Figure 8 Drag & drop a component to the workspace

After dropping the component, set the attributes before using the widget. Open component configuration popup window by clicking the cog icon in the bottom right corner of the widget (Figure 9).

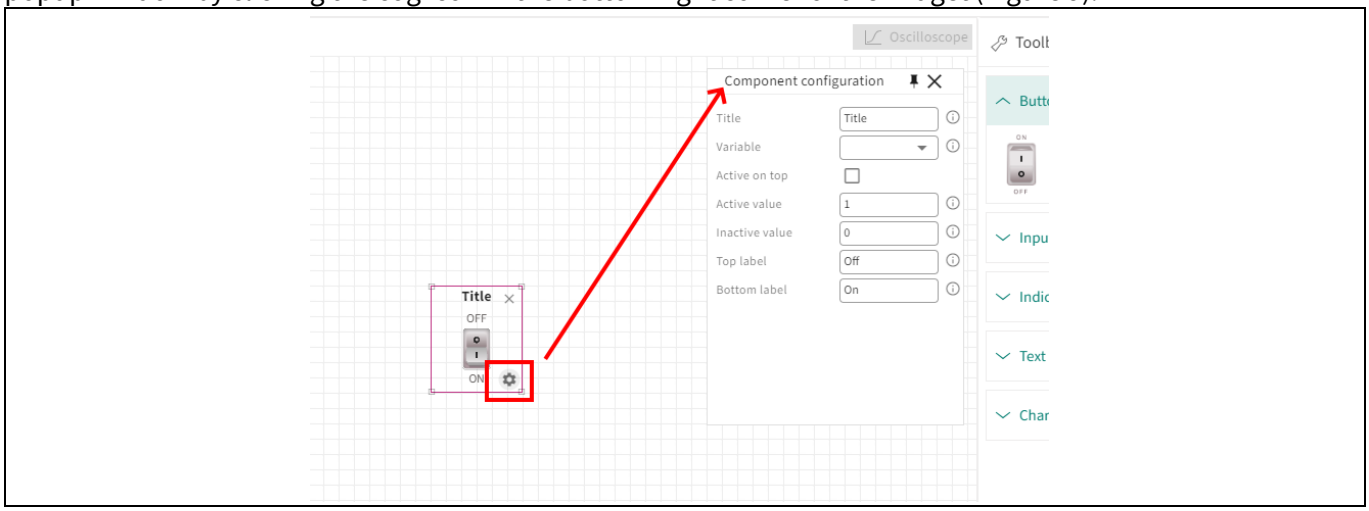


Figure 9 Opening component configuration window

The most important attribute to set is the variable. Select a variable from the list of the variables for the components to function. Select a variable by either selecting the variable from the attribute dropdown list or by searching and double clicking the variable row in the symbol browser (Figure 10).

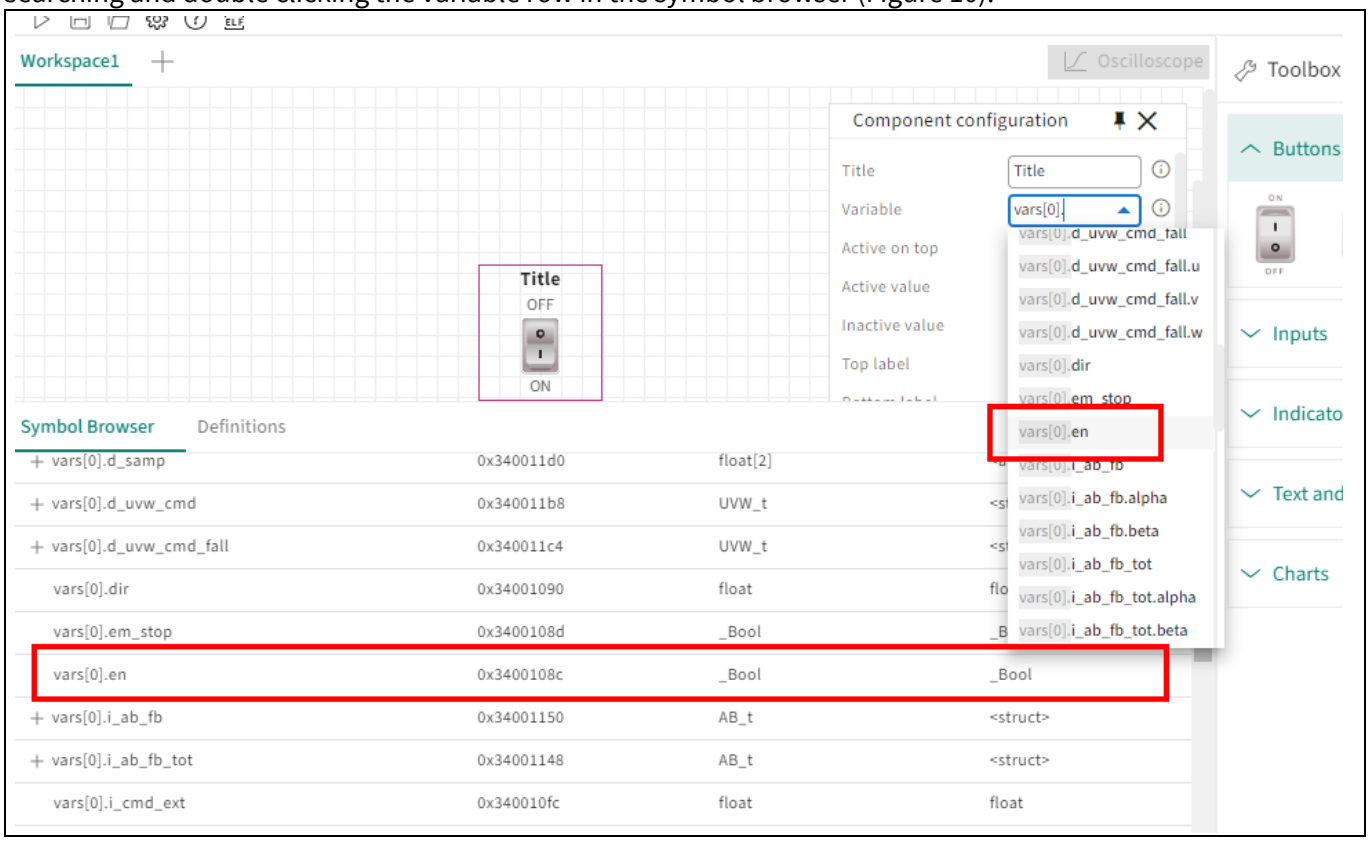


Figure 10 Assigning a variable to the component

There are other attributes for every component. Attribute list generally changes. To get more information on the attribute, see the information icon (ⓘ) right next to the attribute input. After hovering over this icon, a short information is displayed (Figure 11).

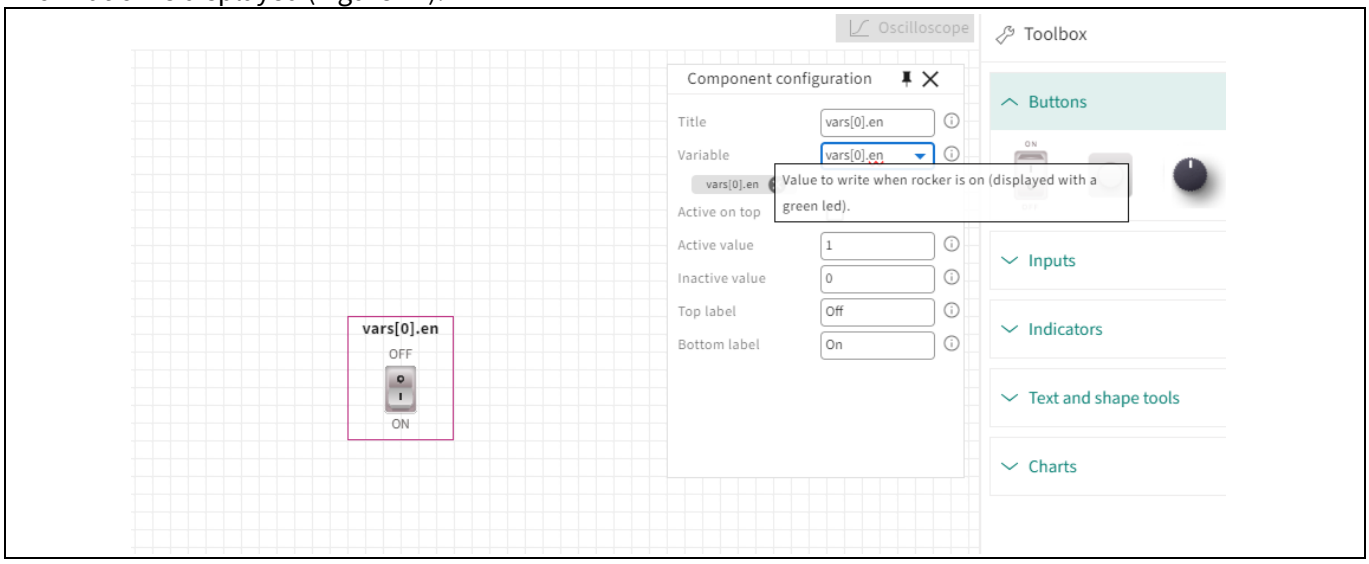


Figure 11 Displaying attribute information

There are other useful buttons on the component which can be used in the design mode. When hovering over the components, extra edit options are displayed. There are 4 resize handles at every corner, remove component icon on top right and sometimes edit icon on top left are displayed.
Use the resizer handles to resize the component (Figure 12). For some of the components resize handles are not present because these components have fixed sizes.

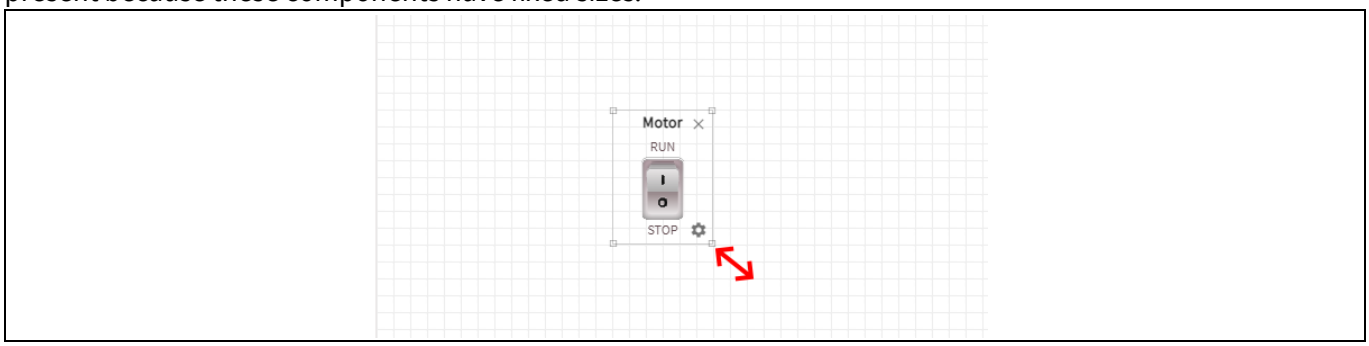


Figure 12 Resizing a component

Another icon is “remove component” icon. Use this icon to remove the component from the workspace (Figure 13).

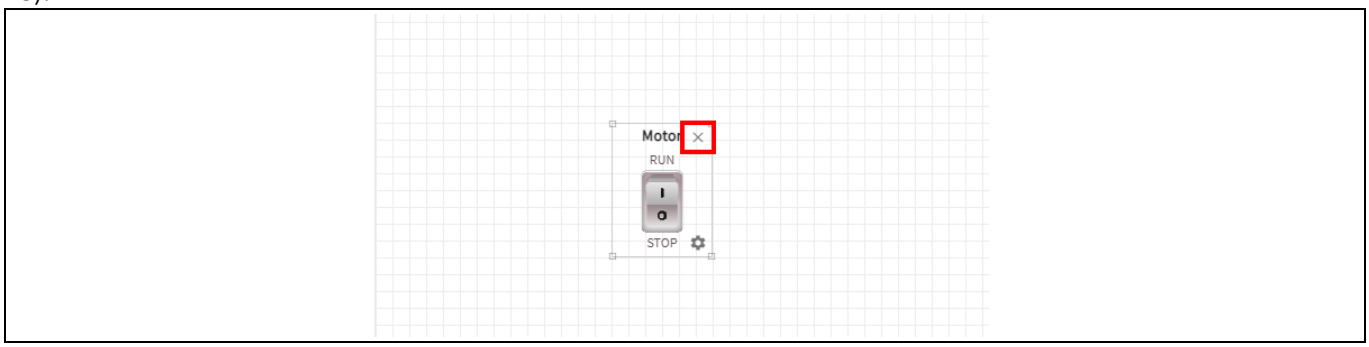


Figure 13 Remove component icon

After removing a component, bring it back by pressing Ctrl+Z. After saving the project or closing the window, Ctrl+Z will not bring previously removed components.

5.1.1 Components

Components are building blocks of the LiveMonitor application. Some of the most used components are Input, Select, LED, Radial Gauge, Line Chart, Logger and Container components.

5.1.1.1 Input component

Input component is used for displaying and writing data from, and to the board. Use it as a read-only component by selecting the “Read-only” attribute. When read-only, it only displays the value acquired from the board.

- “Is UI Parameter” can be selected to create an input with a “Parameter ID”, so that it can be used in other components. When writing a formula in a different component, “Parameter ID” can be used to refer to the value of this component. When defined as a UI parameter, you don’t need to select a variable. This component can be used for pure UI entries, GUI side definitions. If you select a variable for this component, and select UI parameter, the user entries will be written to the board, while updating the value of the parameter.

- If “Read-only” is not selected, you can enter the boundary values for this parameter using the min and max attributes.
- “Read formula” and “Write formula” attributes are used to calculate the value, a middleware transformation between the GUI and the board. Read formula is used to calculate what value will be displayed to the user, and write formula is used to calculate the value written to the board. Use the special keyword “value” in your formula to refer to the user input value (when writing) or assigned variable value (when reading). Use variable names in the formulas when needed, write them exactly as seen in the variable table. When providing both read and write formulas at the same time, you need to provide reverse formulas correctly to prevent an odd behavior when user enters a value but sees a different value after the entry (i.e. if write formula is “*value * x1 + b*”, then write formula needs to be “*(variable – b) / x1*”).
- Define “Disable formula” and “Show/hide formula” to dynamically disable the component or hide the component.

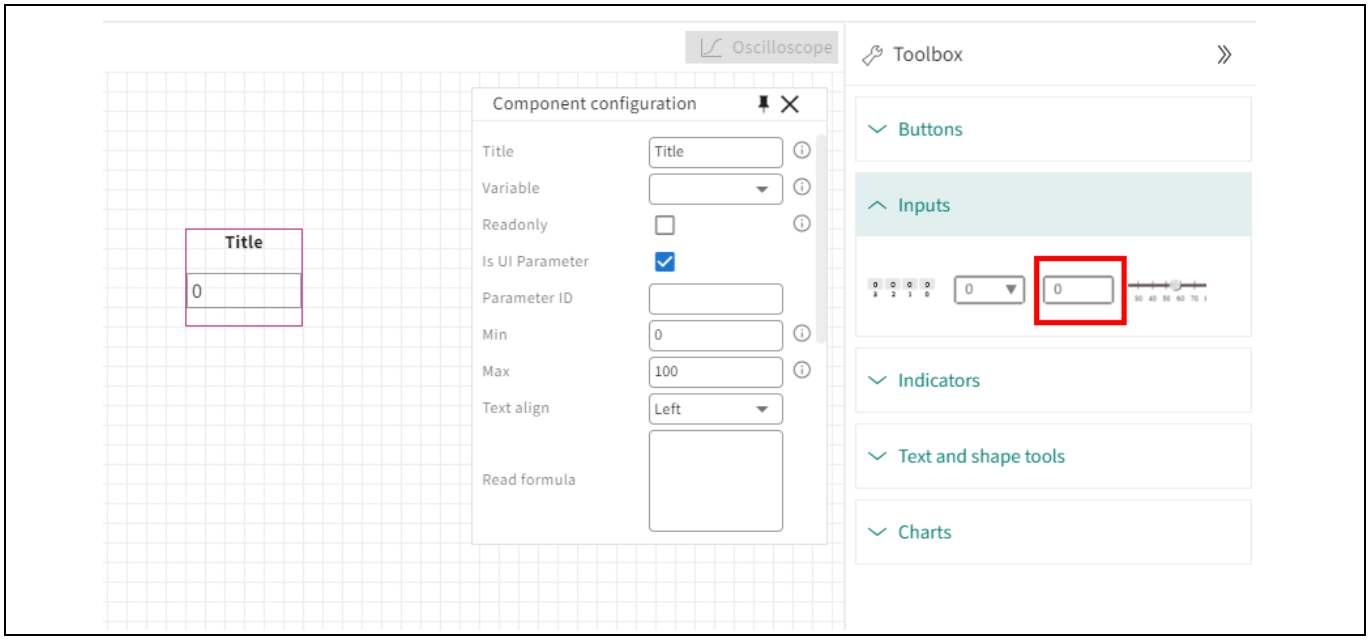


Figure 14 Input component

5.1.1.2 Select component

Select components are useful when mapping each variable value to a dictionary value. Generally, this component is helpful with Boolean-valued variables, Enum variables and any variables where the value can be explained better using a different representation.

- Option labels and Option values attributes need to be comma-separated values. Option values represent the values acquired from the board, while option label values are what users will see. The number of elements should be same for both attributes. The order of the labels and values are important, because each label will be associated with the value in the same order.

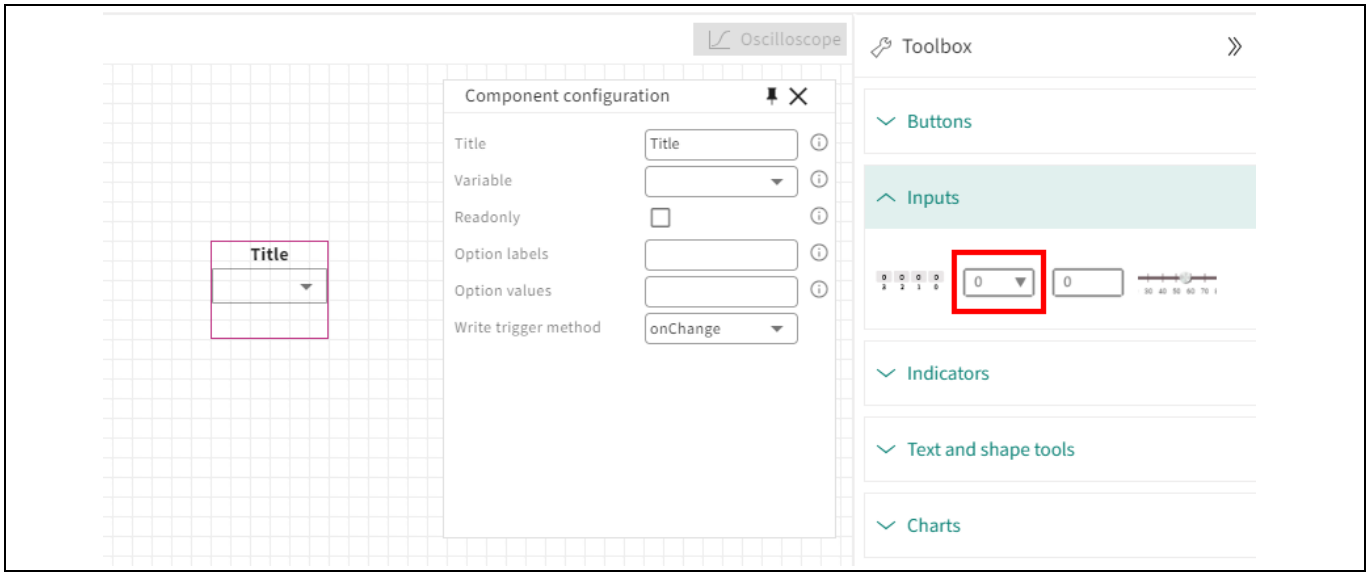


Figure 15 Select component

5.1.1.3 LED component

LED component is a simple indicator to display the state of binary variables.

- Formula is same as input components formula attributes. The only difference is the result of this formula will be interpreted as a Boolean value. If the result is falsy value (i.e. false, 0, “”, undefined, null, NaN…etc.) then the component will go into “inactive” state, otherwise “active” state.

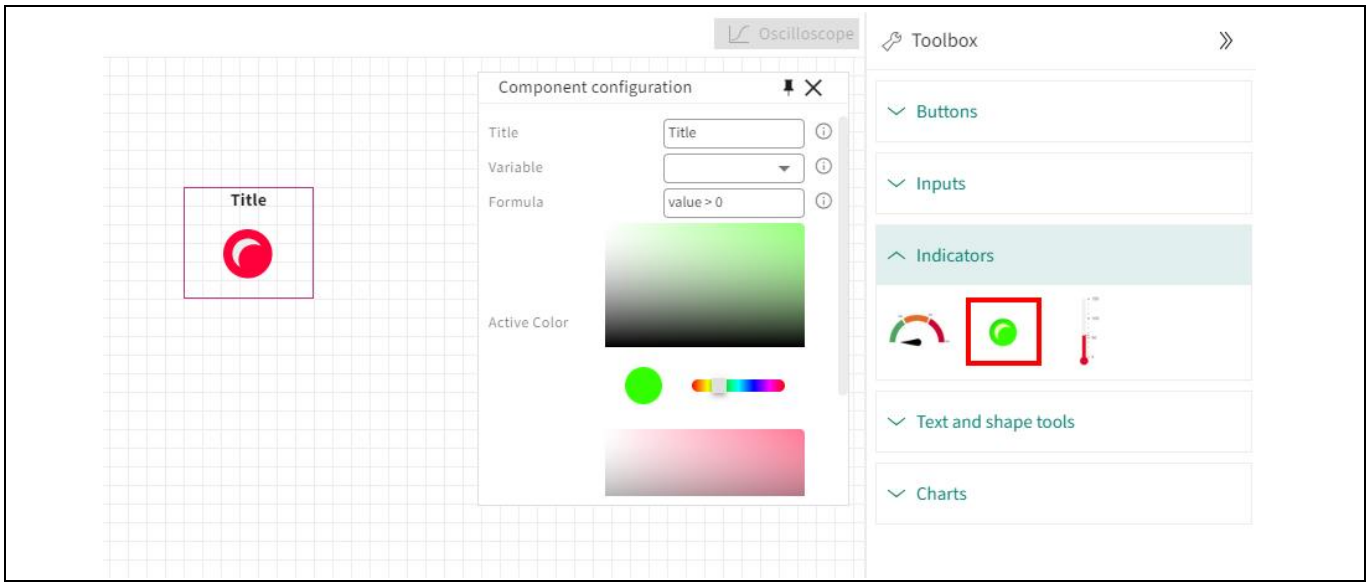


Figure 16 Select component

5.1.1.4 Radial gauge

Radial gauges are good for providing an immediate, intuitive readability for the selected variable value.

- Define as many sections as you want using the “Number of sections” attribute. Sections are each a subrange of a partition of the variable value range.

- Each section is configured using “Upper limit for section n” and “Color for section n” attributes.
- A scale formula can be used to map the variable value to a calculated value.

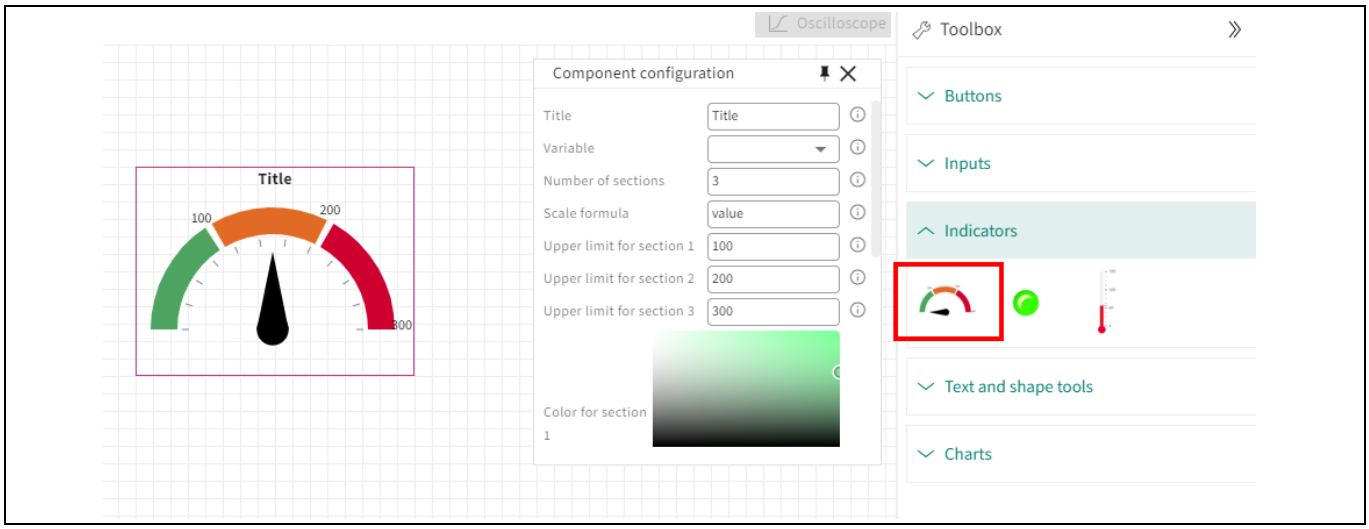


Figure 17 Radial gauge component

5.1.1.5 Line chart

Line charts are basic charts displaying the data versus time axis. Time axis here is configured using the “Polling Period” configuration in the top-level preferences window.

- “Maximum number of samples” can be used to configure the maximum number of visible sample points in the chart. This component will continue displaying new values by rolling the display after reaching the maximum number of samples.
- Up-to 20 variables can be selected. Each variable’s value will be displayed as a separate line in the chart. All variables share the same y-axis, so the range of the y-axis should be configured (using “y range minimum and maximum attributes) carefully considering the ranges of the selected variables.
- When in “Run” mode, double click anywhere within the chart to adjust y range to the best fit. Double clicking again will reset the y-axis back to default range.
- Drag the y-axis by clicking somewhere in the middle of the axis.
- Extend or shrink the range by clicking on the top or bottom of the axis and dragging up or down.
- Select a region in the chart to zoom into that region. Double click to zoom out.
- Use the Pause button at the bottom right corner to stop updating the chart. When clicked again, the chart will continue updating the display with the latest samples.
- Click the trace name in the legend to hide or show the line associated with that name.
- Select “Collect and draw” attribute to render samples at bulk. When enabled “Draw at this size” attribute will be displayed. If “Draw at this size” attribute is set to 10, then data will be collected up-to 10 samples, and the chart will be updated after collecting 10 samples. This mode is more efficient since it re-renders the chart less frequently.
- Use “Trace name(s)” attribute to provide custom names for the selected variables. The values should be comma-separated.

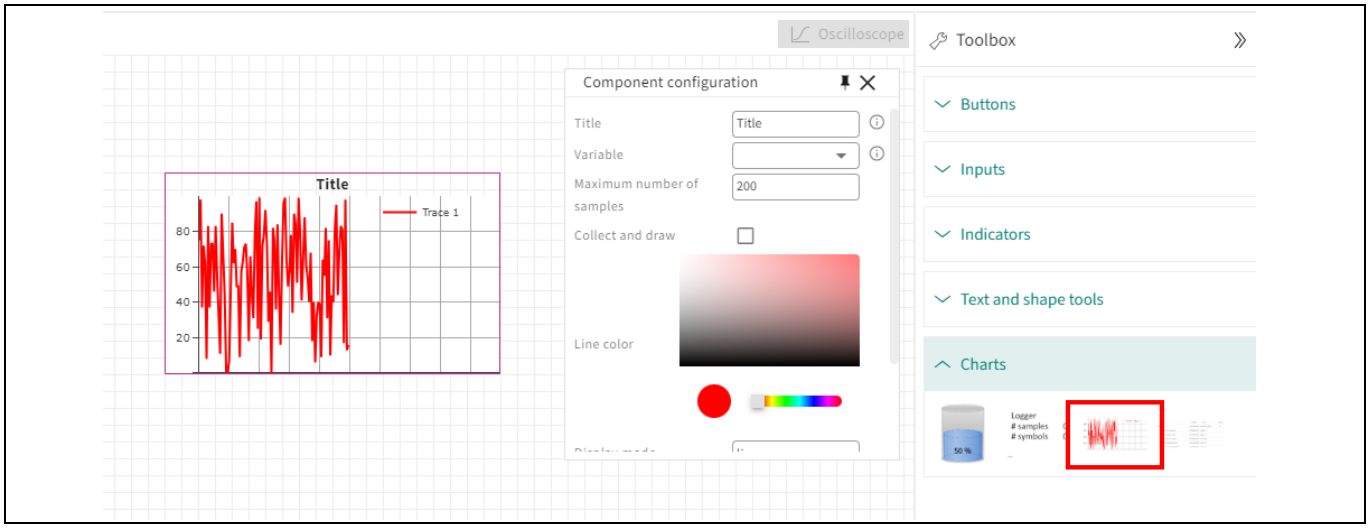


Figure 18 Line chart component

5.1.1.6 Logger component

Logger component is used for exporting acquired data as a csv file.

- Select a path to save the CSV file.
- Up-to 50 variables can be selected.
- The selected file will always be updated by adding new data. It's never overwritten.

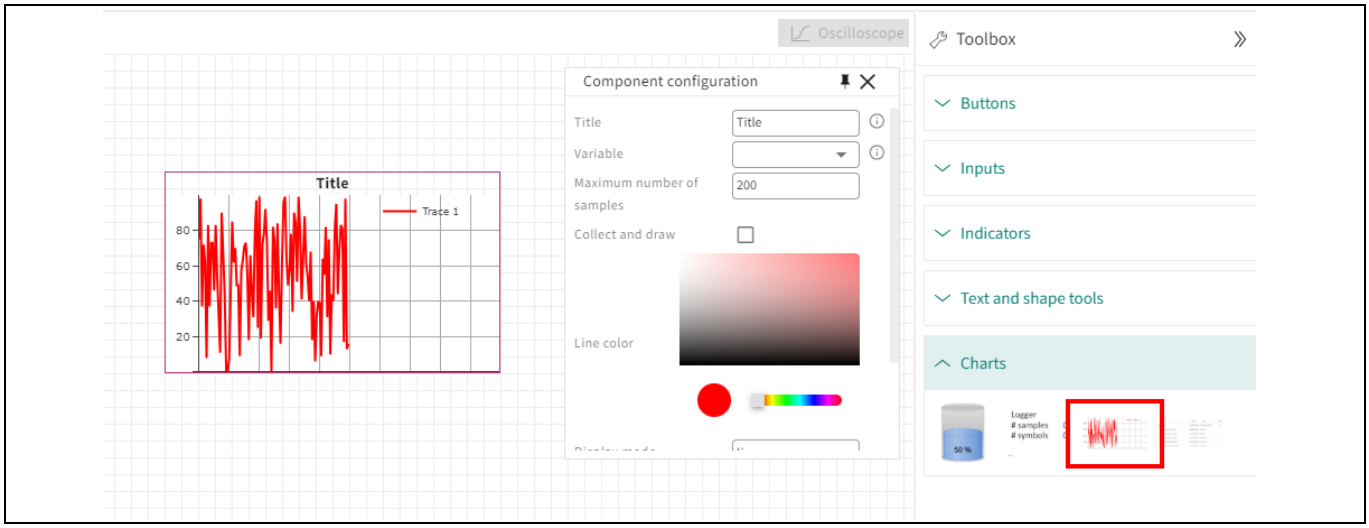


Figure 19 Logger component

5.1.1.7 Container component

Container components are used to group selected components. When a component is added to the container component, its position is locked within the container. When moving the container, all child components are moved with the parent container.

- When child components are moved-in the container component, they're locked to container automatically. To move the child component outside of the container, click the lock icon in the bottom-left of the child component.

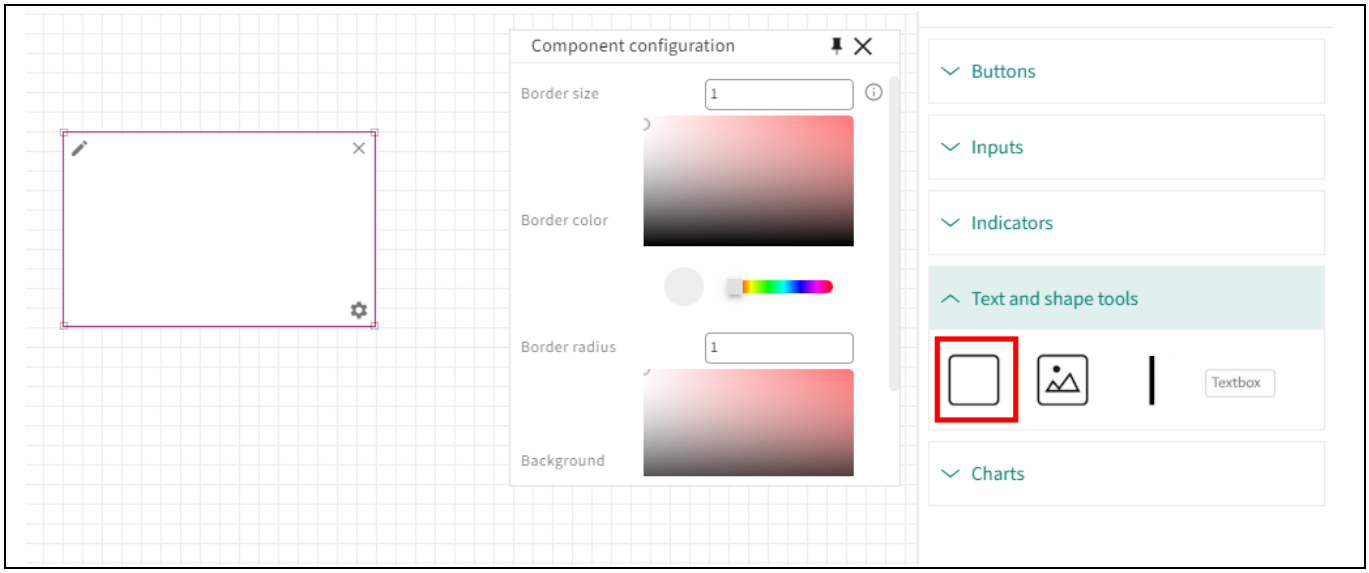


Figure 20 Container component

5.1.2 Using formula attributes

Formula attributes (see Figure 21) evaluate formula expressions. You can combine acquired values from the control board with numeric constants, operators, functions, and unit-aware calculations to produce derived values for display, logging, or further analysis.

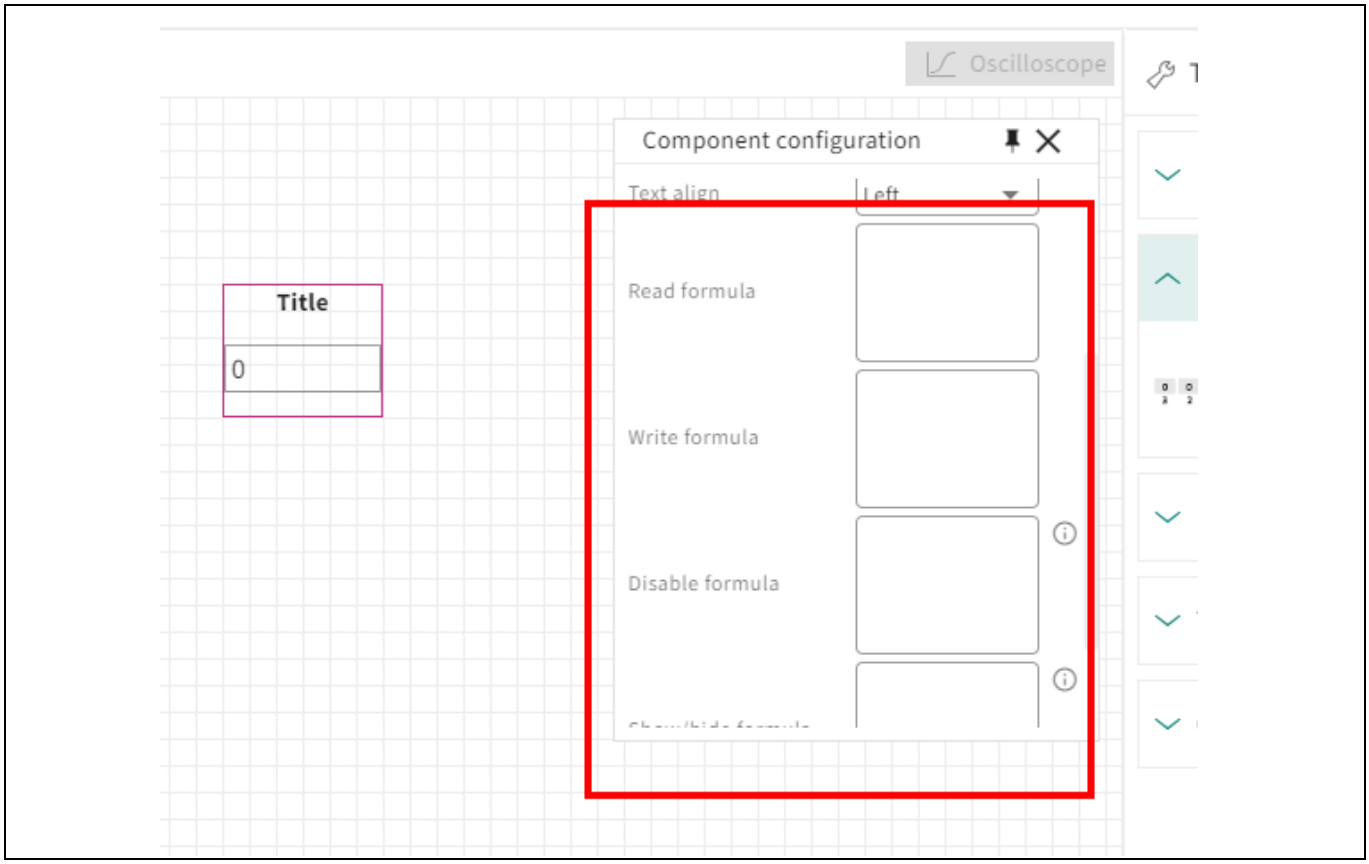


Figure 21 Formula attributes for formula expressions component

LiveMonitor uses MathJS expression parser for evaluating formulas. For more information check https://mathjs.org/docs/expressions/syntax.html.

5.1.2.1 Expression basics

When building expressions, the special name “value” can be used as a shorthand syntax for the selected variable. When applying formulas to the acquired data, “value” refers to the value of the selected variable. When transforming a value before writing it to the board, “value” refers to the input value provided by the user.

- An expression can be a single number, a unit-valued quantity, a Boolean, a string, a complex number, a matrix/array, or an object—depending on what you write.
- Use parentheses (...) to group operations and control evaluation order.
- Whitespace is ignored (except inside strings).
- Names are used for variables, functions, and constants.

Examples

- vbus * ibus
- (speed_rpm - 1500) / 1500
- sqrt(i_alpha^2 + i_beta^2)
- temp > 80 degC ? NaN : temp

5.1.2.2 Variables (acquired values)

The editor exposes acquired board values as variables. Use the exact signal/parameter names shown in the GUI ELF Table.

Examples

- speed_rpm
- i_phase_a
- vbus

5.1.2.3 Operators

Arithmetic

- Add/subtract: x + y, x - y
- Multiply/divide: x * y, x / y
- Power: x ^ y
- Unary: +x, -x
- Modulus: x % y

Examples

- speed_rpm * 2*pi / 60
- torque_Nm * speed_rad_s

Comparisons (Booleans)

- ==, !=, <, <=, >, >=

Examples

- speed_rpm > 0
- abs(i_phase_a) >= 10 A

Logical

- and, or, not

Examples

- (temp > 80 degC) and (speed_rpm > 1000)
- not (fault_active)

Bitwise (integers)

Useful for status words and flags:

- &, |, ^, ~, <<, >>

Example

- (status_word & 0x04) != 0

5.1.2.4 Built-in constants

- pi, e
- Infinity, NaN, true, false

Examples

- 2*pi*freq
- vbus < 0 ? NaN : vbus

5.1.2.5 Functions

A large function library is provided. Below are the most used categories for control and measurement data.

Numeric helpers

- abs(x), sign(x)
- round(x[, n]), floor(x), ceil(x)
- min(a, b, ...), max(a, b, ...)
- sqrt(x), nthRoot(x, n)
- hypot(a, b, ...)

Examples

- hypot(i_d, i_q)
- round(speed_rpm, 0)

Exponential and logarithmic

- exp(x)
- log(x) (natural log)
- log10(x), log2(x)

Example

- 20*log10(vout/vin) (gain in dB)

Trigonometry

- sin(x), cos(x), tan(x)
- asin(x), acos(x), atan(x), atan2(y, x)

If you use units, angles can be expressed as deg or rad.

Examples

- sin(30 deg)
- atan2(i_beta, i_alpha)

Conditionals

- condition ? valueIfTrue : valueIfFalse

Examples

- speed_rpm RPM < 0 ? 0 RPM : speed_rpm RPM
- temp > 100 ? "OVERHEAT" : "OK"

Complex numbers

- complex(re, im)
- re(x), im(x), conj(x), arg(x)

Arrays/matrices

- Create arrays: [a, b, c]
- Indexing: x[1]
- Aggregates: sum(x), mean(x), median(x), std(x), var(x), min(x), max(x)

5.1.2.6 Units

GUI supports a vast array of units across multiple physical dimensions within expression parser. You can attach and convert physical units directly into expressions. This helps prevent scaling mistakes and makes formulas self-documenting.

- **Length, Surface Area, & Volume:** Ranging from meters and liters to miles, acres, and gallons.
- **Time & Angle:** Units like seconds, days, years, radians, and degrees.
- **Mass & Temperature:** Grams, pounds, stones, Kelvin, and Fahrenheit.
- **Force, Pressure, & Energy:** Newton, psi, Joules, and Watts.
- **Electricity & Binary:** Ampere, Ohm, bits, and bytes.

Key Features:

- **Prefixes:** Standard metric prefixes (e.g., `k`, `m`, `M`) are supported.
- **Constants:** Includes built-in physical constants like `elementaryCharge`, and `coulomb`.

Creating unit values

Use either of the following forms:

- **Number + unit**: 12 V, 3.3 kOhm, 2500 rpm, 5 ms
- **Function form**: unit(12, "V")

Examples

- 3.3 V
- 50 us

Unit conversions

Convert between compatible units using:

- x to “targetUnit” (preferred)
- to(x, “targetUnit”)

Examples

- 25 degC to degF (or 25 celsius to fahrenheit)
- to(5000 rpm, "rad/s")
- to(1000 mA, "A")
- to(1 h, "s")

Unit-aware arithmetic

- You can add/subtract compatible quantities: 1 s + 500 ms
- Multiplication/division produces derived units: V * A = W, N*m * rad/s = W
- GUI will raise errors for incompatible operations (for example 1 V + 1 A).

Examples

- vbus * ibus (W)
- torque_Nm * to(speed_rpm, "rad/s")

Note: Please be aware that errors may occur when performing arithmetic operations with unit values. It is essential to convert these values to numbers prior to any arithmetic calculations (Figure 22).

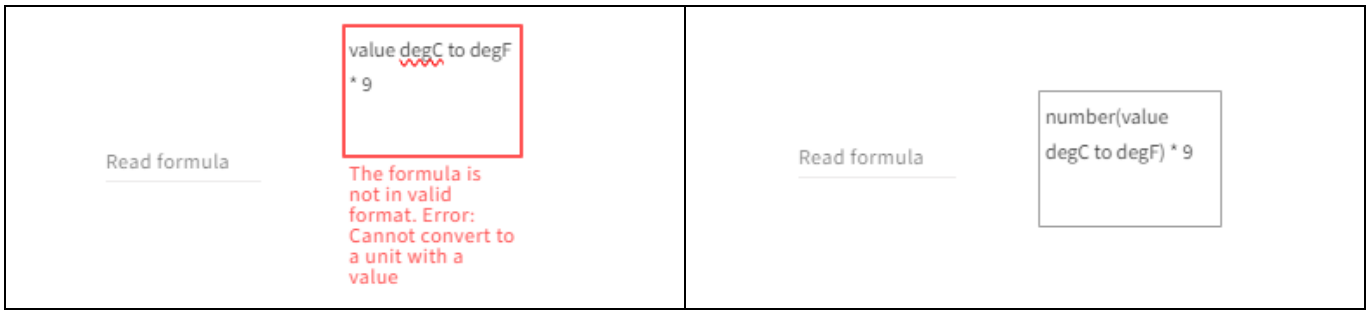


Figure 22 Convert units to number for arithmetic operations

Angle units

Angles can be specified with units:

- 30 deg, 0.5 rad

Example

- sin(90 deg) evaluates to 1

5.1.2.7 Assignment and reusable definitions

You can define variables and reuse them later in the same expression.

- Assignment: a = 10
- Multiple statements separated by “;” or new lines

Example

```
omega = speed_rpm;
p_mech = torque_Nm * omega;
eff = p_mech / p_elec;
eff
```

5.1.2.8 Strings and formatting

GUI supports strings in expressions:

- "text"
- String concatenation can be done via functions (for example concat(...))

Example

- temp > 80 degC ? "HOT", "OK"

5.1.2.9 Troubleshooting formulas

- **Unknown symbol/variable**: verify the signal name matches exactly.
- **Unit incompatibility**: convert with to(...) or ensure operands have compatible units.
- **Divide by zero / invalid math**: results may be Infinity or NaN; use (.?.:.) to guard.

5.2 Taskbar

There are 7 buttons in the taskbar:

1. Switch to design/run mode
2. Save project
3. Open project
4. Preferences
5. User guide
6. Select ELF file
7. Flash the device

Use “Save project” button to save the project with the “mtblm” extension (Figure 23). Open the saved projects later or share them with other people. If you don’t want other users to modify them, lock them for editing by saving them as “mtblmb” files. This is the binary format and opening this files starts them in run mode. Design mode is not allowed for these files.

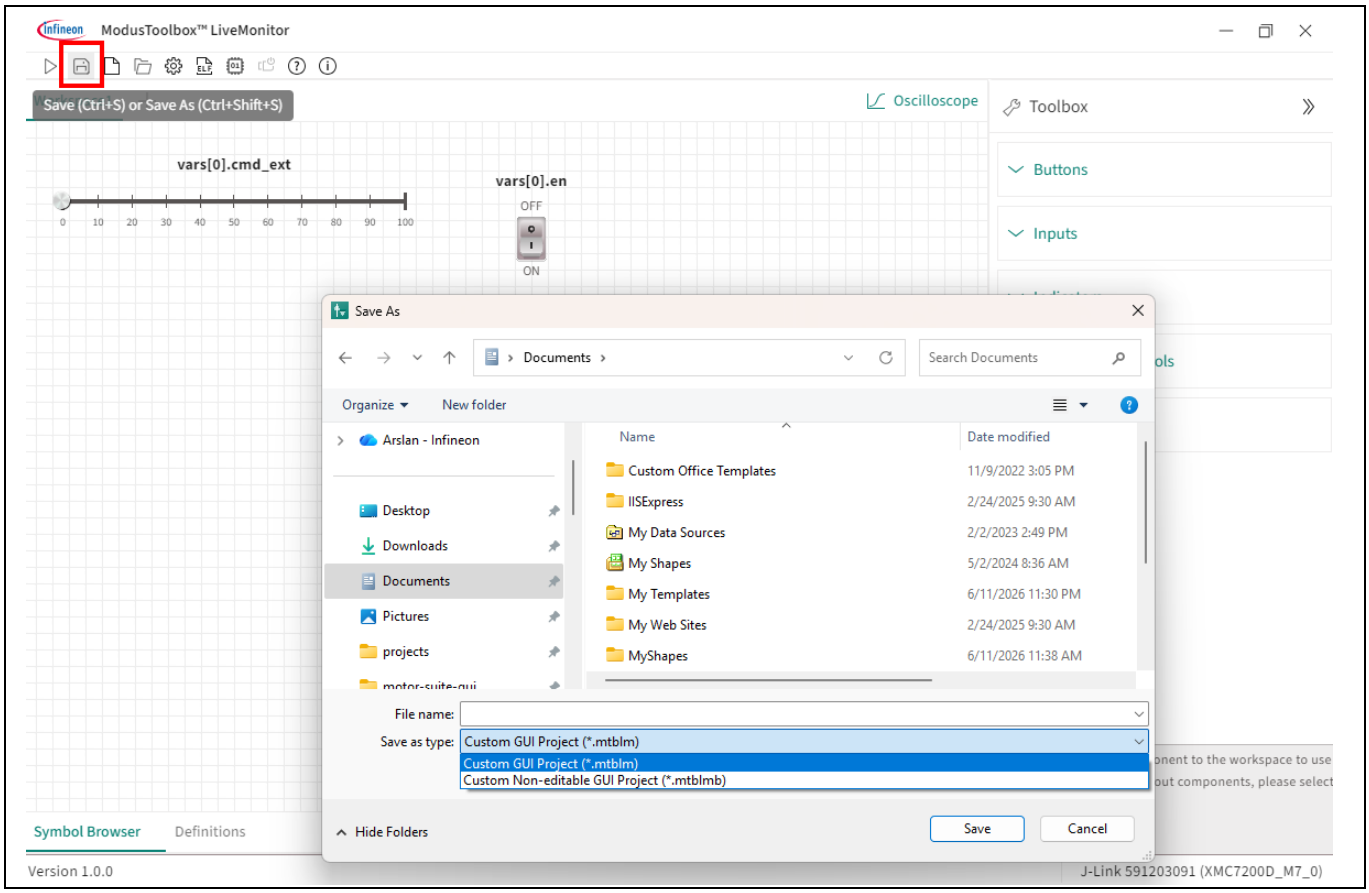


Figure 23 Saving the project

To open a previously saved project, use open project (Figure 24). After clicking this button, a dialog window will be opened to select the project.

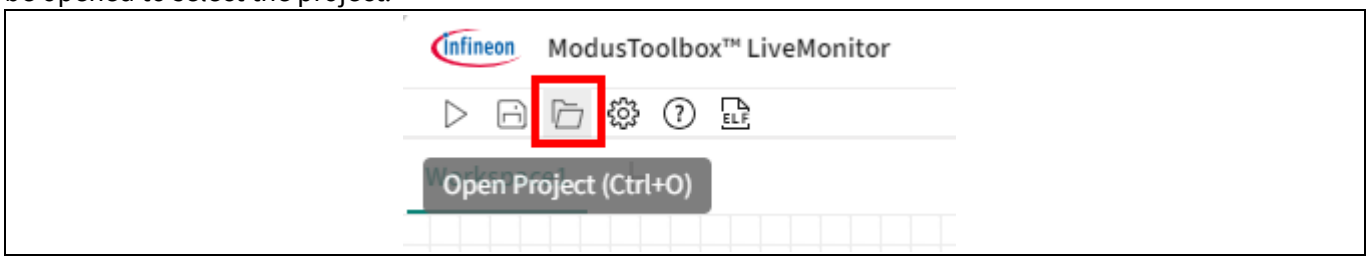


Figure 24 Opening a project

Open preferences window by clicking the preferences icon button. In the opened popup set the polling period. Polling period is a numeric value which determines the period of device reads. All assigned variables are read from the device with this period. The minimum value is 20ms and the default value is 100ms (Figure 25).

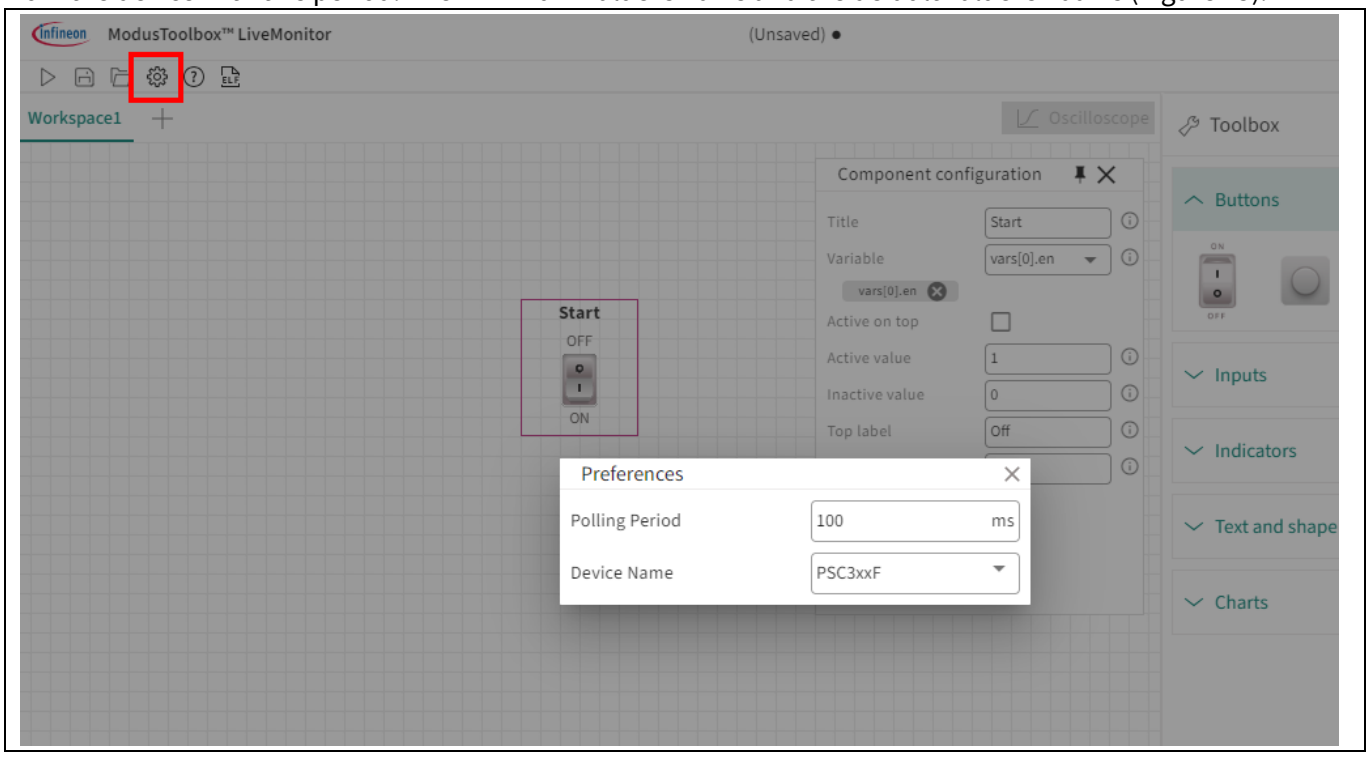


Figure 25 Preferences window

Set this number carefully. If reading all values and updating the interface takes more time, the number set here will be ineffective. For example, if user sets 20ms but whole process takes 30ms, then the actual period will be around 30ms and might change at every cycle depending on the time it takes in that specific cycle. Calculate the minimum required time by adding ~2ms for basic components like an input or button and ~20ms for complex components like charts or gauges, for each component.
Next button in the taskbar is user guide button (Figure 26). Clicking this button will open the user guide.

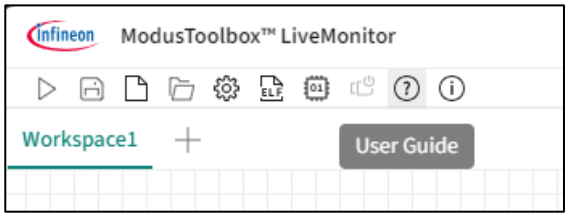


Figure 26 Opening the user guide

5.3 Tabs

On the left-hand side of the taskbar, see the tab settings. Use the icon button to create more tabs. Only one tab is active during operation. Every tab contains their own workspace. To add a new tab, click “Add a new tab” icon (Figure 27).

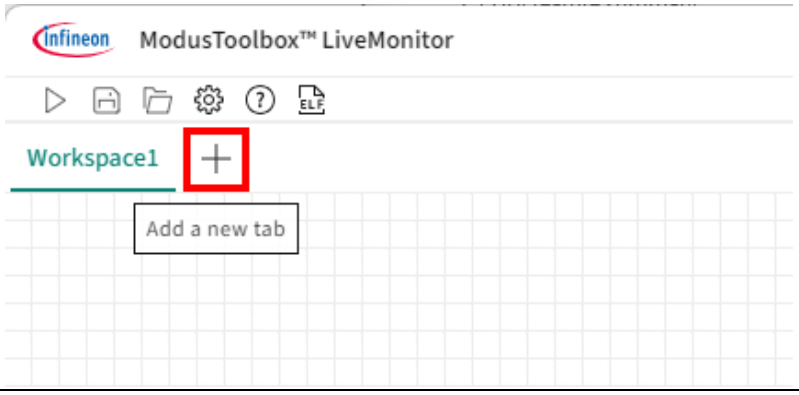


Figure 27 Adding a new tab

Rename the tabs by double clicking on them. If a tab name is double clicked, an input field will be displayed (Figure 28). After entering the new name for the tab, click the approval icon. New tab name will be used after saving the name.

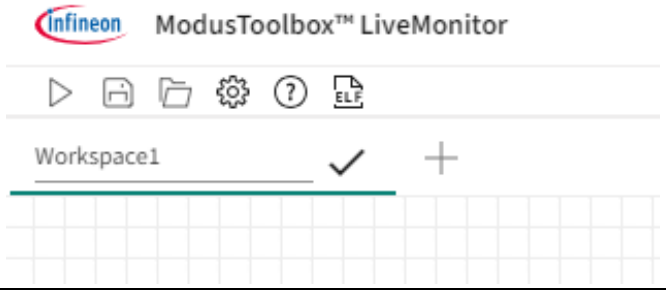


Figure 28 Renaming a tab

5.4 Run mode

After creating the interface, test it by switching to run mode. The first button in the taskbar switches between design and run modes (Figure 29).

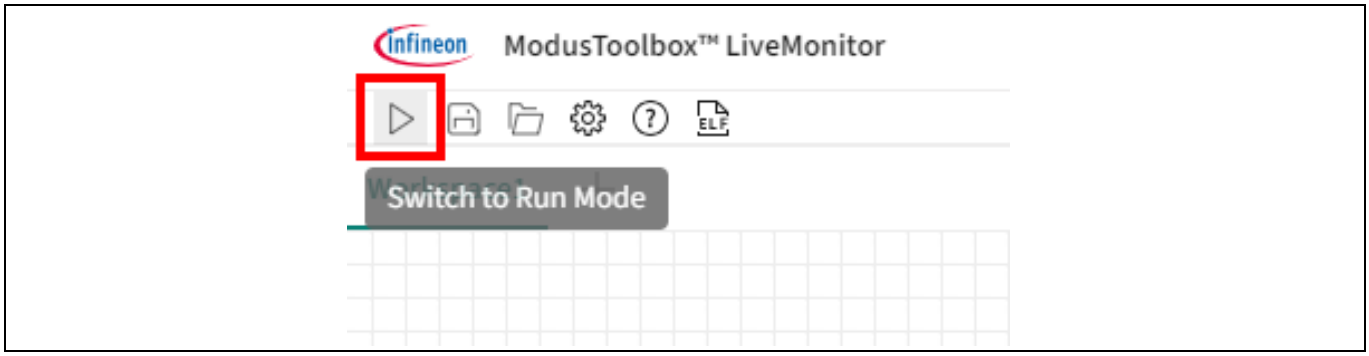


Figure 29 Switch to run mode

After clicking this button, the symbol browser, toolbox, edit icons and workspace will be removed. Device polling will start and all components will operate as expected (Figure 30). Switch to design mode again by clicking the same icon on the top left. Once it is clicked, the polling will stop, device connection will be closed, and all edit elements and windows will reappear in the window.

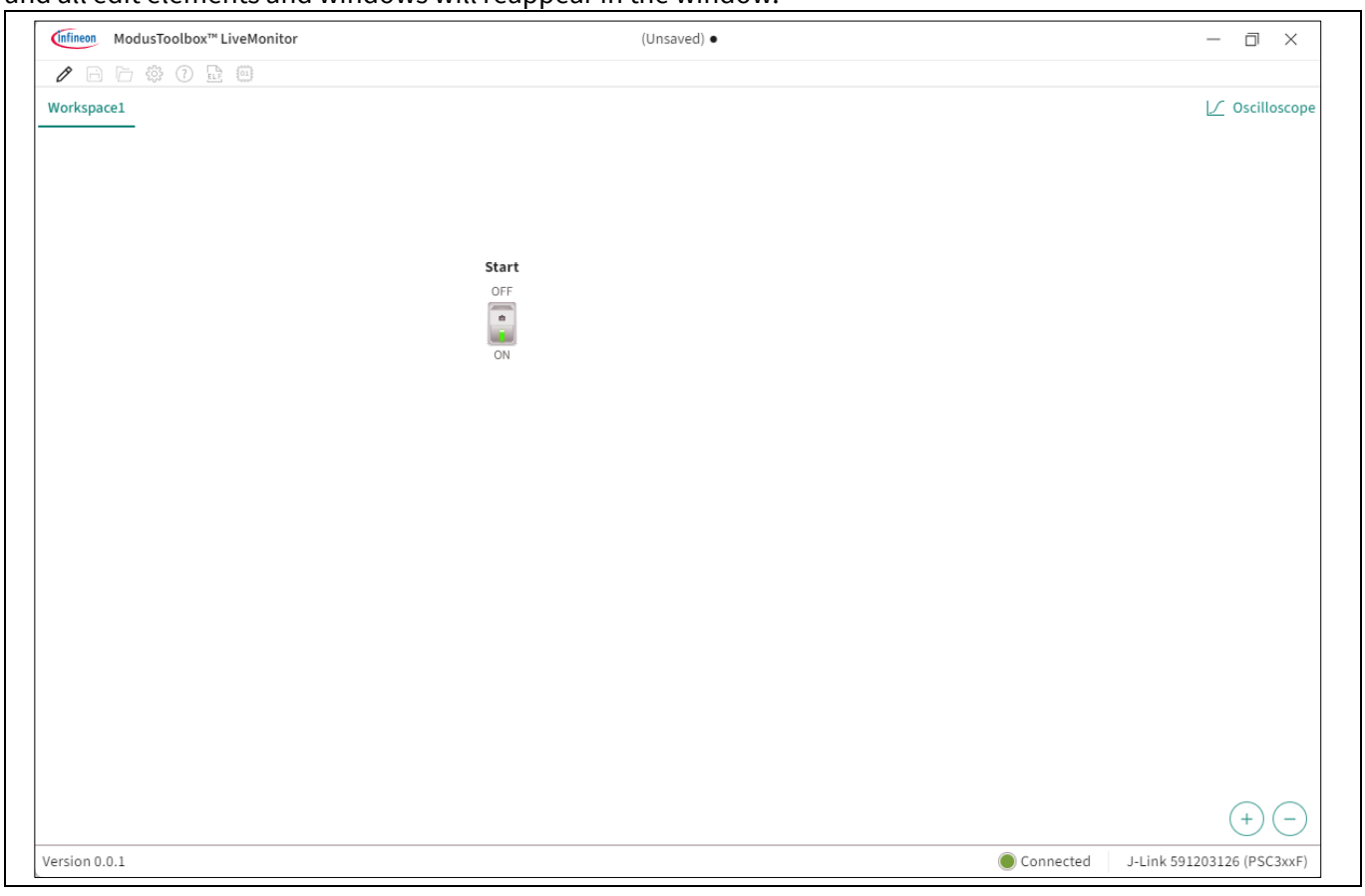


Figure 30 Running the project

6 LiveMonitor Oscilloscope

ModusToolbox™ LiveMonitor Oscilloscope is an interactive instrument panel that lets you visualize, measure, and analyze time-domain signal data captured from embedded hardware. It provides a real-time waveform chart together with a comprehensive settings panel, allowing you to configure channels, adjust trigger conditions, zoom into signal details, and export measurements — all from within your browser. In this document, the term **oscilloscope** refers specifically to the LiveMonitor Oscilloscope GUI application. The middleware library will be referred to as **mtb-oscilloscope middleware**.

Following is the list of oscilloscope features:

- Signal display for the selected traces through a line chart
- Controls and buttons to re-scale the axes for the channels
- Trace assignment up-to 8 different channels
- Channel enabling and disabling
- Real-time signal statistics display
- Frequency changing through divider manipulation
- Time unit display setting
- 4 different scope modes to stop or control sampling procedure
- Trigger slope, position, level and holdoff settings
- Measurements with cursors
- Exporting and importing data as CSV files
- Board connection control

The necessary information (device ID, device name, ELF file path, configuration options) to read the signals is sent from the host GUI. Device ID is displayed at the bottom right corner of the window.

Attention: The oscilloscope is integrated into multiple GUI tools (i.e. Motor Suite GUI, Power Suite GUI, LiveMonitor); however, only one can be active at any given time for a specific device. Please refrain from opening more than one oscilloscope window across different tools (when connected to the same device).

6.1 Product overview

The oscilloscope consists of a host-side GUI component and a firmware-side middleware library. Together they form a complete software oscilloscope solution.

6.1.1 GUI feature summary

Feature	Details
Channels	Up to 8 channels; individually enable/disable per capture.
Display modes	Auto Fit, Multiple Scales, Single Scale, Group Traces, Custom.
Trigger	Positive or negative slope; configurable level and position via drag handles directly on the chart.
Capture modes	Stop, Single Trig, Continuous, Triggered.
Cursors	Two independent cursors (Stop mode only); computes X1, X2, ΔX, ΔY, 1/ΔX (Hz), Y_AVG, Y_RMS, Y Min, Y Max.
Snapshot export	PNG download of the current chart view.
Timebase	Selectable display unit: µs, ms, or s.

File import	Supports loading previously captured recording files.

6.1.2 Firmware (mtb-oscilloscope) feature summary

Feature	Details
Channels	1–8, configurable via MTB_OSCILLOSCOPE_MAX_CH.
Buffer depth	Configurable via MTB_OSCILLOSCOPE_MAX_SAMPLES (default 1000).
Data types	C standard integer types up to 32 bits (signed/unsigned) plus float. Optional single-bit capture.
Trigger	Positive or negative slope; configurable level, position, hold-off.
Capture modes	Off / Single / Continuous / Triggered.
HAL/RTOS dependency	None. Sampling is application-driven via ISR call.
Memory footprint	Static, deterministic — no heap, no RTOS primitives.

6.2 Firmware integration: mtb-oscilloscope middleware

The mtb-oscilloscope middleware is the firmware-side counterpart of the Oscilloscope GUI. It periodically samples up to eight in-memory variables, handles trigger and capture-mode logic, and exposes collected buffers to the host.

Note: This chapter is for firmware engineers embedding the middleware in a ModusToolbox™ application. End users who only operate the GUI may skip this chapter.

Note: The code example, BSP or library (i.e. motor-ctrl-lib v3.2.0 and earlier, pfc-ctrl-lib v1.0.0) you are utilizing may have already imported the oscilloscope middleware by default. Please verify this before proceeding with the integration of the middleware.

6.2.1 Architecture overview

Module	Purpose
Data-capture core (mtb_oscilloscope.h / .c)	Sampling engine, channel configuration, trigger logic, and capture-mode state machine.

See https://github.com/infineon/mtb-oscilloscope for more information.

6.2.2 Repository layout

```
mtb-oscilloscope/
├── mtb_oscilloscope.h          Public API — sampling core
├── mtb_oscilloscope.c          Sampling / trigger state machine
├── mtb_oscilloscope_data.h     Public API — variable registry
├── mtb_oscilloscope_data.c     Registry implementation
└── imports/                    Application-owned configuration
    ├── mtb_oscilloscope_cfg.h
    └── mtb_oscilloscope_data_cfg.c
```

Attention: Never edit files inside imports/ in place. Copy them into your application source tree first (see Section 6.2.3.2, Step 2).

Note: mtb_oscilloscope_data.h, mtb_oscilloscope_data.c and mtb_oscilloscope_data_cfg.c files are not needed when using LiveMonitor Oscilloscope.

6.2.3 Installation

6.2.3.1 Step 1 — Add via Library Manager

1. **Open** your ModusToolbox™ application in the IDE or navigate to the application root.
2. **Launch** the Library Manager from the IDE menu, or run:

   ```
   make library-manager
   ```

3. **Locate** mtb-oscilloscope in the catalog, select the desired version, and click Apply.
4. **Wait** for the Library Manager to fetch sources and update the makefiles.

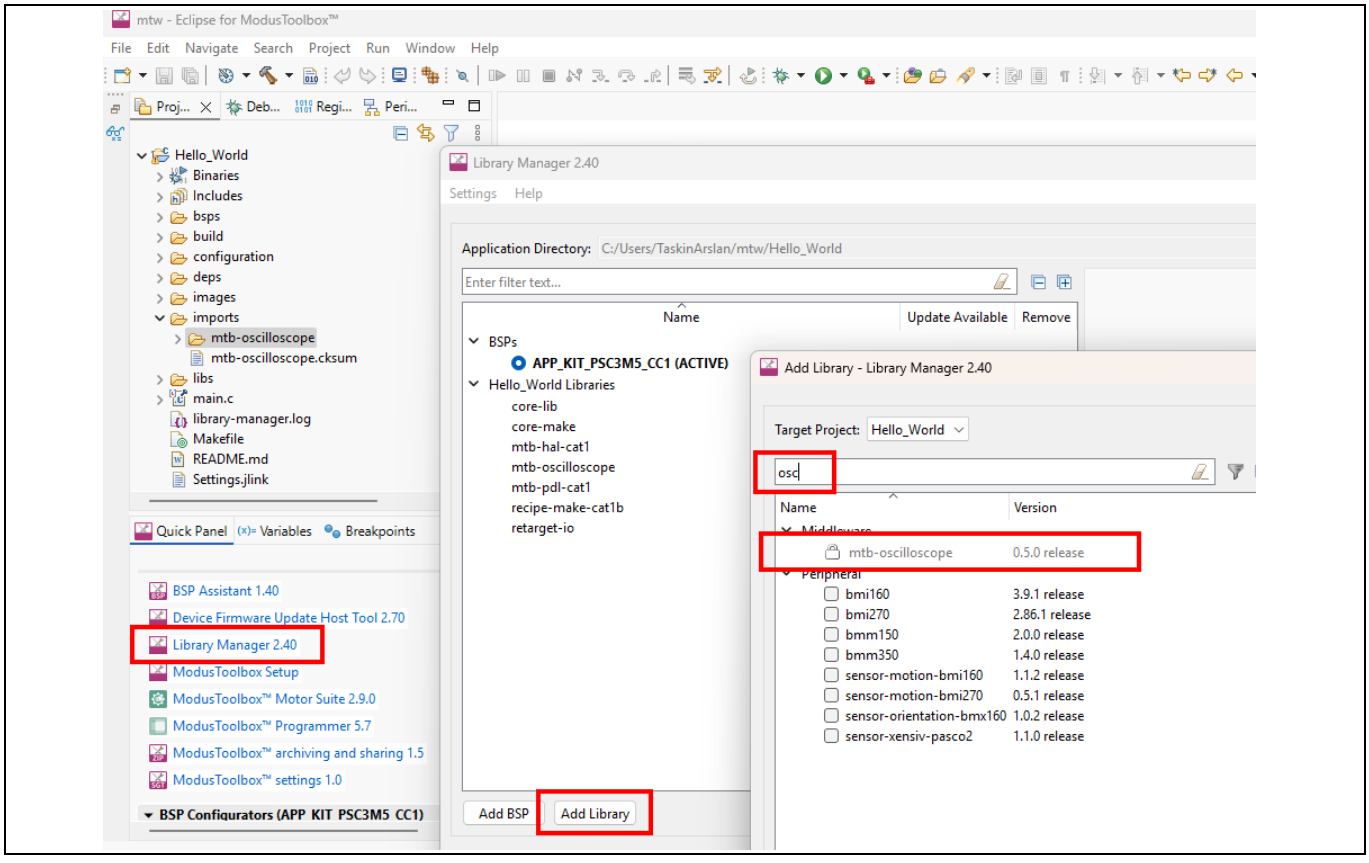


Figure 31 MTB Library Manager

6.2.3.2 Step 2 — Copy and customize configuration files

1. **Add** the asset imports/ folder to your .cyignore to prevent the template files from being compiled:

   ```
   imports/mtb-oscilloscope
   ```

Attention: You will not see .cyignore file in the MTB Project Explorer. You need to visit your workspace folder using File Explorer, and edit the file using a text editor explicitly.

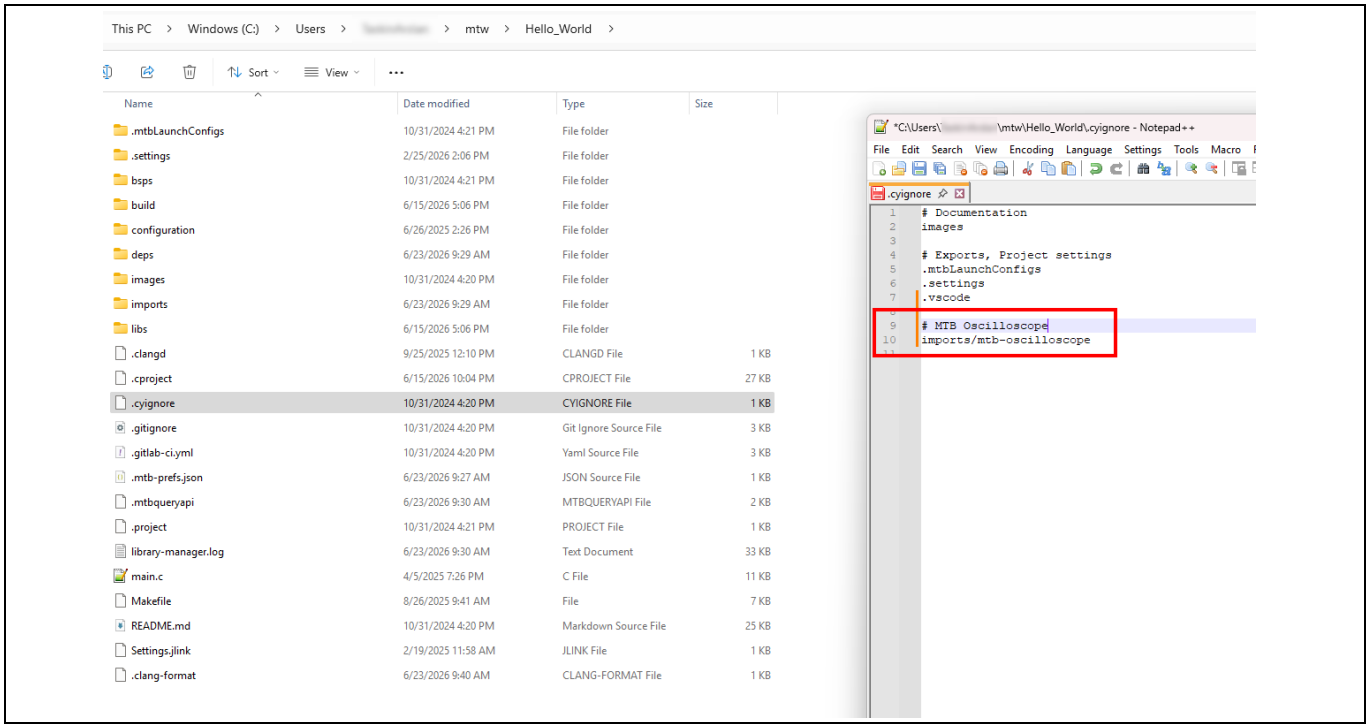


Figure 32 Configure .cyignore file

2. Copy mtb_oscilloscope_cfg.h from imports/mtb-oscilloscope/ into your application source root directory (or you can create another folder -i.e. oscilloscope/- and copy the file in this folder).

3. Open the copied mtb_oscilloscope_cfg.h and configure:

Macro	Default	Description
MTB_OSCILLOSCOPE_MAX_CH	8	Maximum number of channels (1–8).
MTB_OSCILLOSCOPE_MAX_SAMPLES	1000	Sample buffer depth per channel.
MTB_OSCILLOSCOPE_SAMPLING_CLK_HZ	—	Default hardware sampling clock in Hz. Must match the ISR period.
Data-type width enables	All on	Enable/disable 8-bit, 16-bit, 32-bit, and float channel type support. ***For LiveMonitor integration use the default configuration.***

6.2.4 Minimal integration example

The minimal integration requires two functions:

1. **Initialization of oscilloscope:** This function will configure the code for right sampling frequency, set parameters with initial values, establish the channels.
2. **Sampling data:** This function is normally used in the fast interrupt routine or at the precise code location where data capture is required.

```
#include "mtb_oscilloscope.h"


#define LOGGING_DELAY_MS   (1)
```

```
int main(void)
{
    mtb_oscilloscope_init(10000U);        /* Initialize library */
    for (;;) {
       mtb_oscilloscope_sampling();
       Cy_SysLib_Delay(LOGGING_DELAY_MS); /* Add delay between samplings,
it is suggested to use the timer interrupt instead */
       /* application work */
    }
}
```

Attention: mtb_oscilloscope_sampling() must be called at exactly the rate specified in mtb_oscilloscope_init(). An inaccurate ISR period causes incorrect time-axis values in GUI.

6.2.5 Remote and runtime configuration

All configuration calls operate on plain RAM variables. The host tool can configure the oscilloscope without firmware-side glue:

Method	How it works
Debug probe (SWD / JTAG)	Host probe reads/writes library RAM directly via the debug interface. No additional firmware code required. **LiveMonitor uses this method.**

6.2.6 Memory Consumption

Static memory is the only allocation model.

Estimated memory for default mtb_oscilloscope_cfg.h (MTB_OSCILLOSCOPE_MAX_CH=4, MTB_OSCILLOSCOPE_MAX_SAMPLES=1000, 32-bit samples) is:

Per channel:	sizeof(MTB_OSCILLOSCOPE _CH_SAMPLE) * MTB_OSCILLOSCOPE_MAX_SAMPLES = 4 bytes * 1000 = 4 KB per channel
4 channels:	16 KB sample buffer
mtb_oscilloscope.c code + raw data:	~4-6 KB
other code + dispatch table:	~1-2 KB
Total RAM:	~16 KB + ~512 B control state = ~17 KB
Total Flash:	~6-8 KB

MTB_OSCILLOSCOPE_MAX_CH and MTB_OSCILLOSCOPE_MAX_SAMPLES are the primary tuning parameters for memory-constrained targets.

6.3 Interface Overview

When the oscilloscope loads, the screen is divided into two main areas:

Table of contents

Area	What It Contains
Chart Panel	The large waveform display at the top. Shows all enabled channel traces over time, trigger markers, cursor lines, firmware information, and the Action Buttons toolbar.
Settings Panel	The control strip at the bottom. Houses the Channel Settings table, Scope Modes, Trigger configuration, Sampling Clock/Timebase controls, and the Cursors/Measure panel.

Note: The Settings Panel can collapse to give the Chart Panel more screen space. Click the chevron arrow in the 'Scope Settings' header to toggle visibility.



Figure 33 Oscilloscope window

6.4 Using the Oscilloscope

This chapter explains all tasks required to operate the oscilloscope GUI, presented in the order of a typical measurement workflow.

6.4.1 Scope Settings

Use scope settings to adjust the settings for channels, trigger options and measurements. There are 6 groups of settings in this section. Channel Settings, Timebase Configuration, Status Indicator, Scope Mode selector, Trigger slope and offset setter and Measure settings.

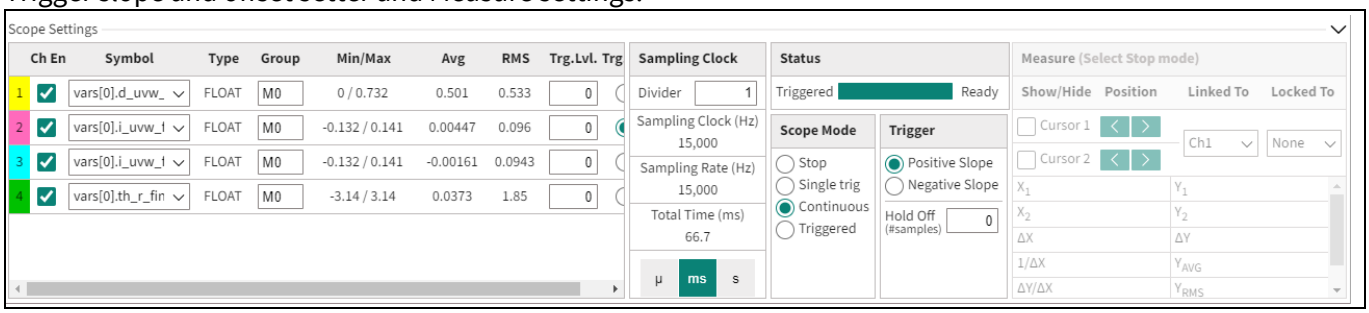


Figure 34 Scope settings

6.4.2 Channel Settings

The Channel Settings table occupies the left portion of the Settings Panel. Each row represents one input channel.

6.4.2.1 Channel table columns

Column	Description
Color badge	Colored square showing the channel number. The color matches the waveform trace.
Ch En (Channel Enable)	Tick to show the channel waveform. Un-tick to hide without losing configuration.
Symbol	Select which signal variable is mapped to this channel from the drop-down list.
Type	Read-only label showing the data type of the selected signal (e.g. float, int).
Group	Type a group name to cluster related channels in Group Traces display mode.
Min / Max	Running minimum and maximum values since the last statistics reset.
Avg	Running arithmetic average of the channel signal.
RMS	Running root-mean-square value of the channel signal.
Trg.Lvl. (Trigger Level)	The amplitude at which triggering fires. It can also be set by dragging the horizontal trigger line onto the chart.
Trg.Sel. (Trigger Select)	Radio button to select which channel acts as the trigger source.

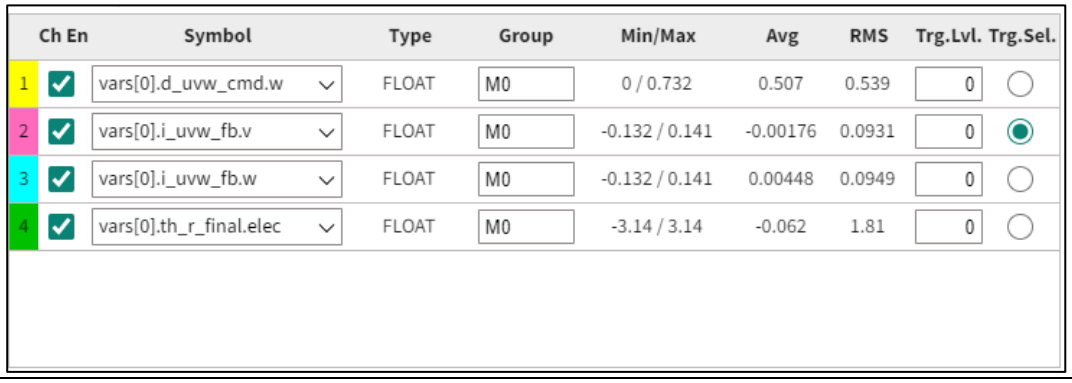


Figure 35 Channel settings

6.4.2.2 How to enable or disable a channel

1. **Locate** the row for the target channel on the Channel Settings table.
2. **Click** the checkbox in the Ch En column. A tick mark means the waveform is visible.

Disabling a channel hides its trace immediately and stops acquisition. Trigger level is preserved.



Figure 36 Disabling channel 3 removes the signal and the axis

6.4.2.3 How to assign a signal to a channel

1. **Click** the drop-down list in the Symbol column for the desired channel.
2. **Select** the signal name from the list.

3. **Observe** the waveform on the chart update to reflect the newly mapped signal.

6.4.3 Scope Modes

There are 4 different scope modes (Figure 37). Stop mode stops new data sampling while continuous mode samples new data continuously. In single trig mode, data is sampled only if the trigger conditions (trigger slope, trigger position…etc.) are met. Triggered mode works like single trig mode but samples new data continuously if trigger conditions are met. The trigger channel is channel 1 so single trig and triggered modes test the conditions only for channel 1.

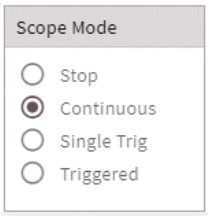


Figure 37 Scope Mode

6.4.3.1 Available capture modes

Mode	Behavior
Stop	Halts all data capture. The chart freezes at the last frame. Cursor measurements are only available in this mode.
Single Trig	Captures one complete frame after the trigger condition is met, then switches to Stop mode.
Continuous	Captures frames continuously without waiting for a trigger. Useful for always-active signals.
Triggered	Captures a new frame each time the trigger condition is met on the selected trigger channel.

6.4.3.2 How to change the scope mode

1. **Locate** the Scope Mode table on the Settings Panel.
2. **Click** the radio button next to the desired mode: Stop, Single Trig, Continuous, or Triggered.
3. **Observe** the status progress bar update to reflect the new acquisition state.

6.4.4 Trigger configuration

The Trigger panel is located to the right of the Scope Mode panel.

6.4.4.1 Trigger slope

Option	When it fires
Positive Slope	Signal crosses the trigger level while rising (low → high).
Negative Slope	Signal crosses the trigger level while falling (high → low).

1. **Locate** the Trigger panel inside the Scope Modes section.
2. **Click** the Positive Slope or Negative Slope radio button (Figure 38).

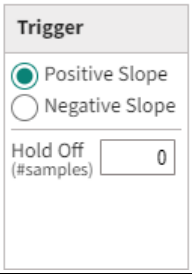


Figure 38 Trigger slope and holdoff

6.4.4.2 Trigger Hold Off

Hold Off defines the minimum number of samples that must be passed after one trigger event before the oscilloscope may trigger again.

1. **Locate** the Hold Off (#samples) numeric input field in the Trigger panel.
2. **Click** the field and type the desired number of samples.
3. **Press** Enter or click outside the field to confirm.

6.4.4.3 How to set the trigger level via the chart

1. **Locate** the horizontal dashed line on the chart matching the color of the trigger channel.
2. **Click and drag** the line up or down to the desired amplitude level.
3. **Release** the mouse button to confirm.

Note: Trigger level can also be typed into the Trg.Lvl. cell of the Channel Settings table.

6.4.4.4 How to set the trigger position via the chart

1. **Locate** the vertical dashed line on the chart (the Trigger Position marker).
2. **Click and drag** the line left or right to move the trigger point.
3. **Release** the mouse button to confirm.

6.4.5 Sampling clock and timebase

In time base configuration panel (Figure 39), update the sampling clock and rate by changing divider value.

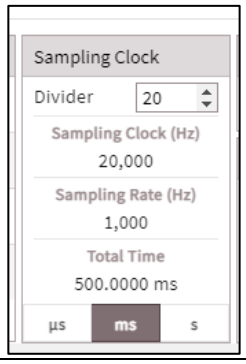


Figure 39 Time base configuration

6.4.5.1 Panel fields briefly

Field	Meaning
Divider	Integer divisor applied to the hardware clock. Range: 1–200. Increasing this value slows the effective sample rate.
Sampling Clock (Hz)	Raw hardware clock frequency. Read-only.
Sampling Rate (Hz)	Effective sample rate after applying the Divider (= Sampling Clock ÷ Divider). Updates automatically.
Total Time	Total time window of one full capture frame in the active timebase unit.

6.4.5.2 How to change the clock divider

1. **Locate** the Divider numeric field at the top of the Sampling Clock panel.
2. **Click** the field and type a new integer with a value between 1 and 200.
3. **Press** Enter or click outside the field to apply.

Note: A Divider of 1 means no division — sampling occurs at the full hardware clock rate.

6.4.5.3 How to switch the timebase unit

1. **Locate** the three toggle buttons at the bottom of the Sampling Clock panel: µ (microseconds), ms (milliseconds), and s (seconds).
2. **Click** the button for your preferred unit. The active button is highlighted.

After changing the value, the axis ticks in the horizontal axis and other time related data statistics (information boxes, measures…etc.) are updated (Figure 40).

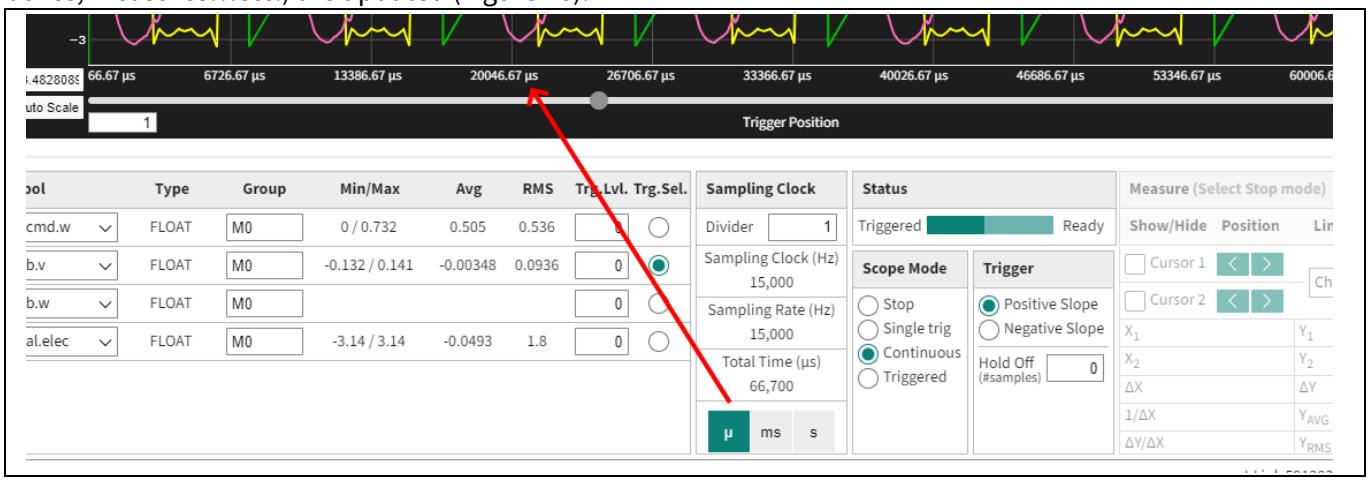


Figure 40 Changing time unit

6.4.6 Cursor measurements

Cursors allow precise time and amplitude measurement at specific points on a captured waveform. The Cursors panel is at the bottom of the Settings Panel.

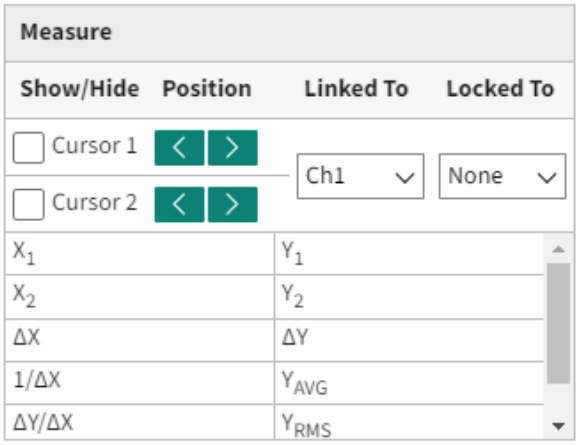


Figure 41 Measure

Attention: Cursor controls are disabled when the scope is in any mode other than Stop. Switch to Stop mode before enabling cursors.

6.4.6.1 Cursor panel controls

Column	Description
Show / Hide	Tick the Cursor 1 or Cursor 2 checkbox to make that cursor visible on the chart.
Position	Use the left and right chevron buttons to move the cursor one sample at a time.
Linked To	Select which channel Y-axis the cursor amplitude reading is measured against.
Locked To	Optionally lock Cursor 2 to move relative to Cursor 1, keeping the gap fixed.

6.4.6.2 How to take a cursor measurement

1. **Set** the Scope Mode to Stop.
2. **Tick** the Cursor 1 checkbox to display the first cursor line on the chart.
3. **Click** the right arrow button to position Cursor 1 at the desired sample.
4. **Tick** the Cursor 2 checkbox to display the second cursor line.
5. **Position** Cursor 2 using its arrow buttons.
6. **Read** the computed statistics on the table below the cursor controls.

Table of contents



Figure 42 Activating the first cursor

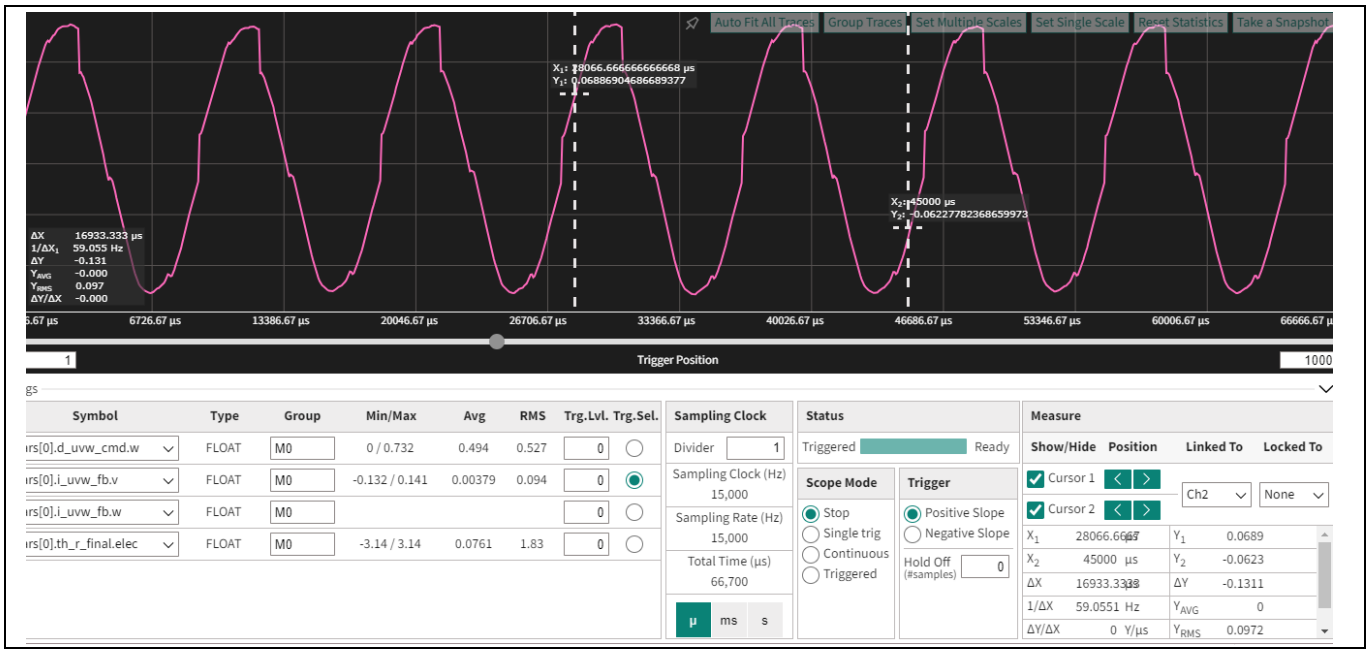


Figure 43 Activating both cursors

Table of contents

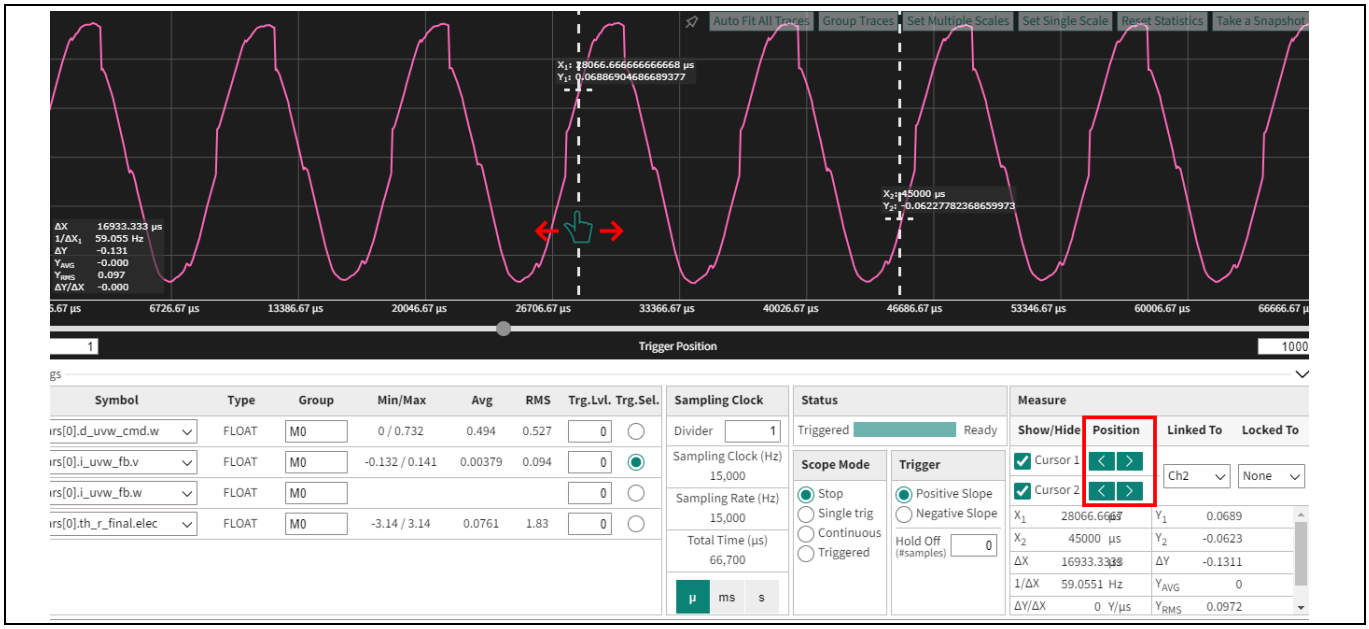


Figure 44 Repositioning cursors

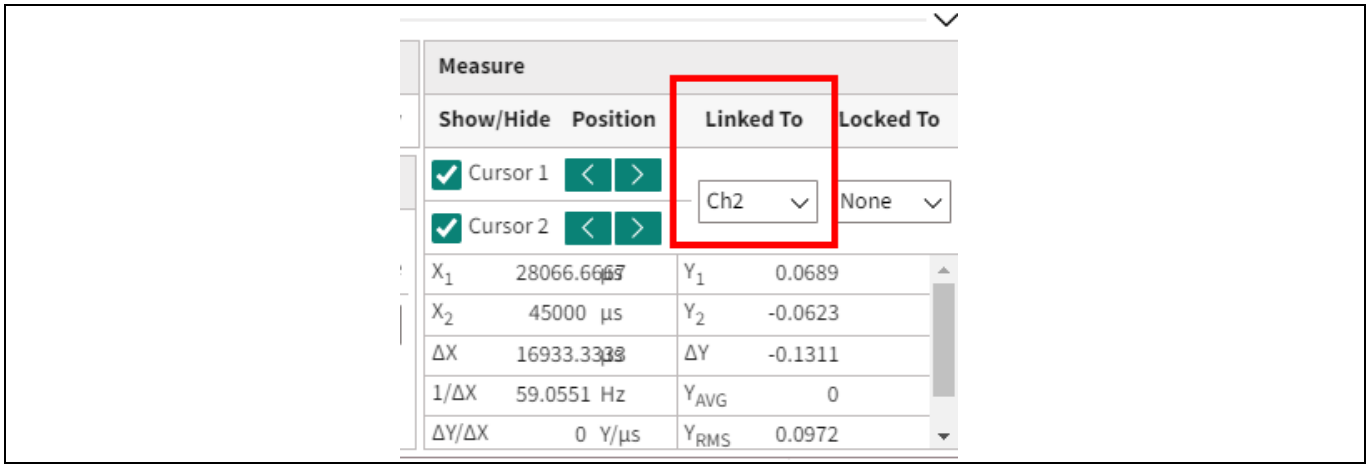


Figure 45 Linking a channel to cursors

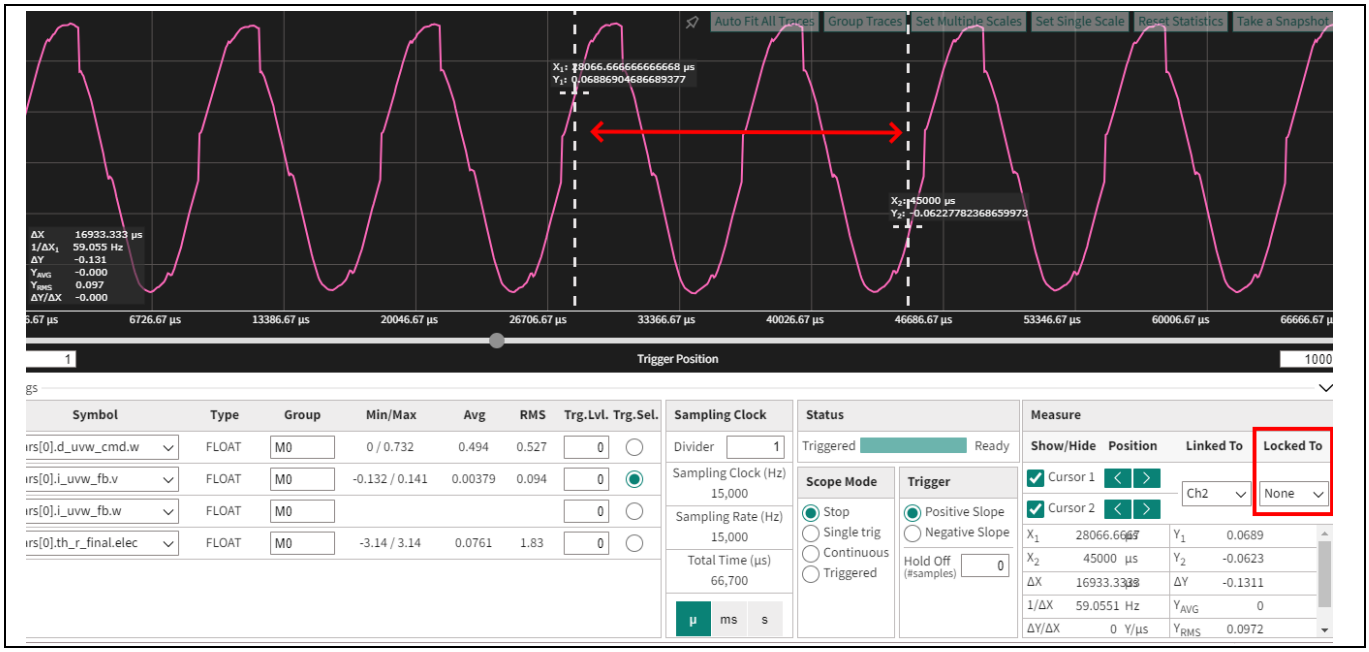


Figure 46 The distance between cursors remains same when positioning cursor 1 while it is locked to cursor 2

6.4.6.3 Cursor statistics reference

Measurement	Description
X1	Time position of Cursor 1 in the active timebase unit.
Y1	Signal amplitude at the Cursor 1 position.
X2	Time position of Cursor 2 in the active timebase unit.
Y2	Signal amplitude at the Cursor 2 position.
ΔX	Time difference: X2 − X1.
ΔY	Amplitude difference: Y2 − Y1.
1/ΔX	Frequency derived from ΔX, in Hz. Useful for period measurement.
Y_AVG	Average amplitude between the two cursors.
ΔY/ΔX	Slope (rate of change) between the two cursor positions.
Y_RMS	Root-mean-square amplitude between the two cursors.
Y Min	Minimum amplitude between the two cursors.
Y Max	Maximum amplitude between the two cursors.

6.4.7 Chart panel

After a successful connection, the connection indicator in the footer becomes green with the text “Running” and traces are sampled periodically. The chart displays the traces for all channels (Figure 47).

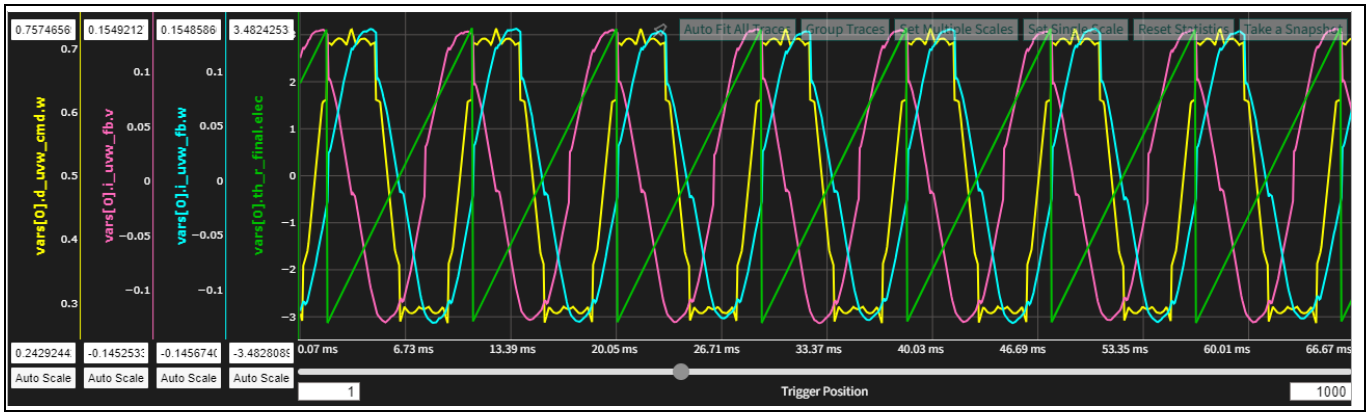


Figure 47 Chart section

The first 4 buttons on the upper right corner are used for fast rescaling of all axes, fifth button for resetting the statistics for all channels and the last one for saving the latest frame as an image file (Figure 49).

6.5 Chart navigation

Interaction	Effect
Click and select region on chart background	Zooms in to the selected region.
Click and drag on a Y-axis label	Pans that channel Y-axis range.
Double-click on the chart	Resets zoom and pan to the default level.
Horizontal range inputs (below chart)	Enter start/end sample values to set the visible time window precisely.

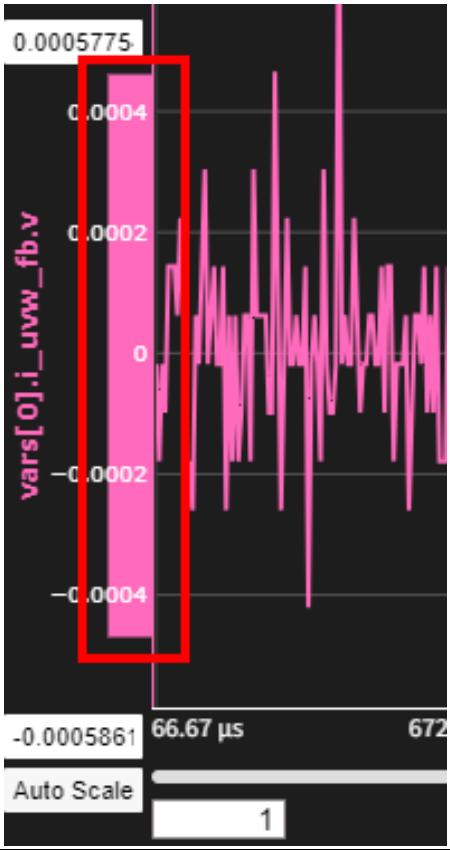


Figure 48 To pan the y-axis, click and drag this region (not whole axis is clickable)

6.5.1 How to zoom into a region of interest

1. **Locate** the horizontal range inputs below the chart, labelled with the active timebase unit.
2. **Enter** the start time in the left field and the end time in the right field.
3. **Observe** the chart to zoom into the specified range.
4. **Double-click** anywhere on the chart to return to the full view.

Note: Y-axis auto-scale can be toggled per channel by double-clicking directly on the Y-axis label on the left side of the chart.

6.5.2 Chart display tools

The Action Buttons toolbar provides quick display operations. By default, it appears as a floating overlay on the Chart Panel.

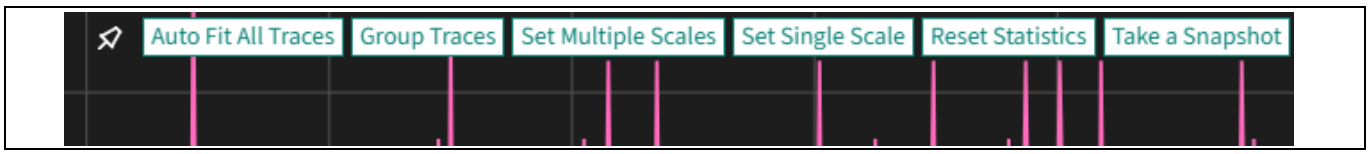


Figure 49 Quick action buttons

6.5.2.1 Available action buttons

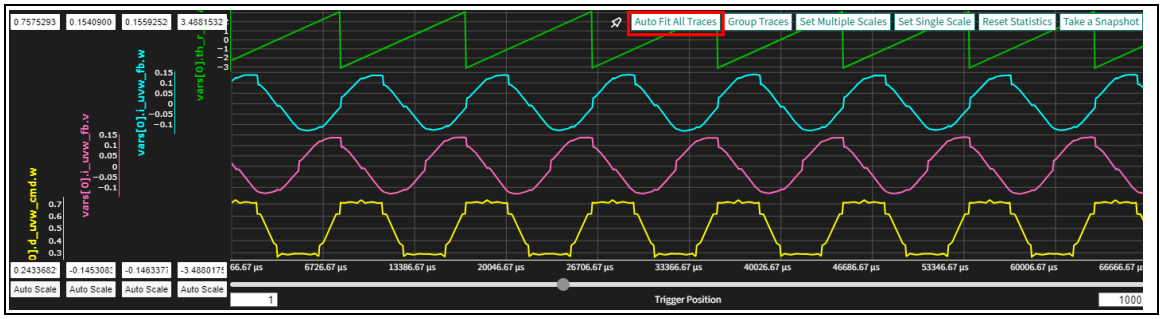
Button	What it does
Auto Fit All Traces	Automatically scales each channel Y-axis so its waveform fits neatly within the chart area.  **Figure 50 Auto fit all traces**
Group Traces	Arranges channels sharing the same Group name onto a common Y-axis region.



Figure 51 Group traces

Set Multiple Scales

Assigns each channel with its own Y-axis scaled to its global min/max.

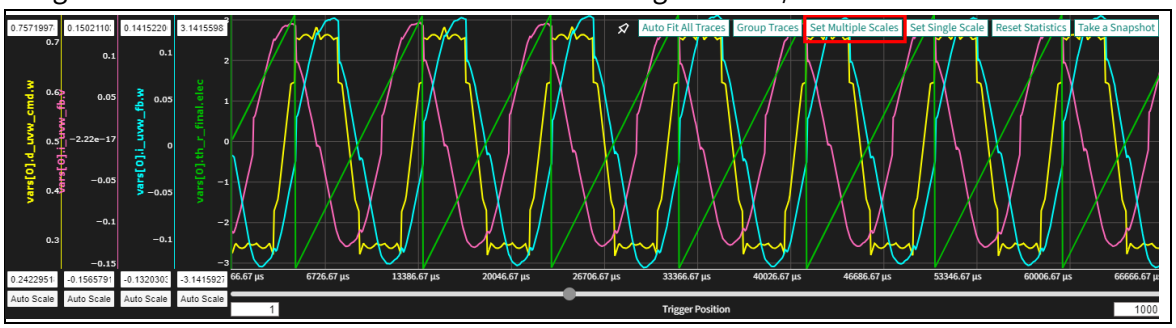


Figure 52 Set multiple scales

Set Single Scale

Forces all enabled channels onto a single shared Y-axis.

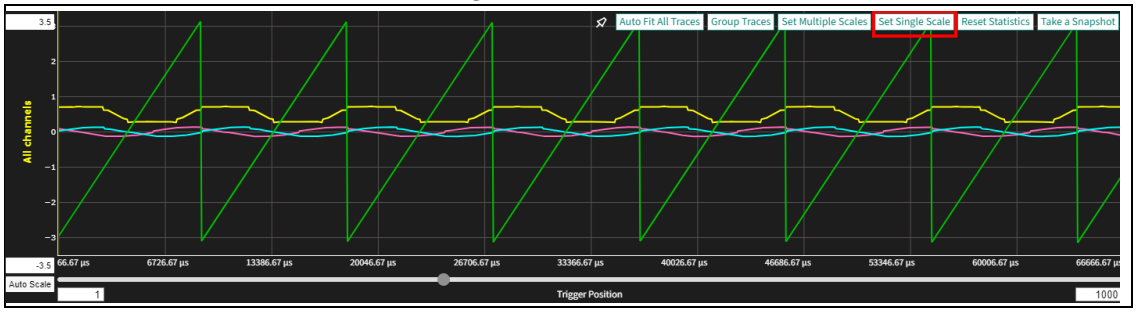


Figure 53 Set single scale

Reset Statistics

Clears running Min, Max, Avg, and RMS values and restarts accumulation.



Figure 54 Reset statistics

Take a Snapshot	Exports the current chart view as a PNG file to the browser download folder, named Oscilloscope_<date>.png.

6.5.2.2 How to dock the toolbar into the Settings Panel

1. **Click** the pin icon button in the floating toolbar on the chart.
2. **Observe** the toolbar move to the Tools panel inside the Settings Panel.

6.5.2.3 How to move the toolbar back to the chart

1. **Locate** the Tools panel in the Settings Panel.
2. **Click** the pin icon in the Tools panel header.

6.5.3 Auto Scale buttons

These buttons are used to auto scale the corresponding axis using the range of the signal. When enabled, it continuously updates the y-axis range at every new sampling frame.

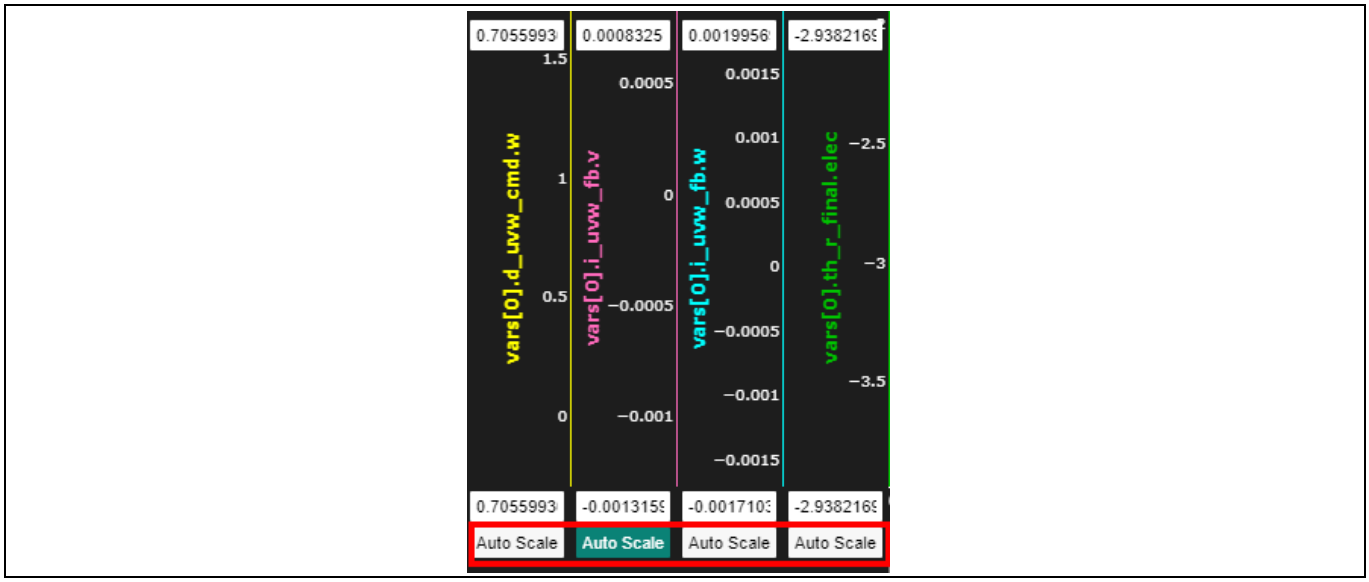


Figure 55 Vertical axes, auto scale buttons and scale inputs

6.5.4 Scale Inputs

Use these inputs to apply custom limits to the corresponding axis.

6.5.5 Horizontal Axis

The horizontal axis shows the time scale with the selected time units (Horizontal axis).

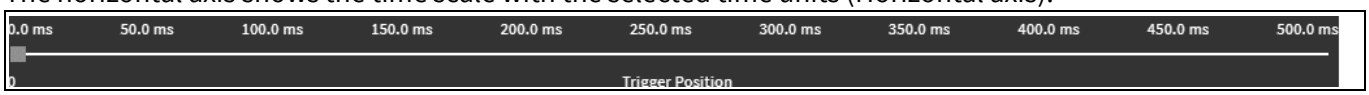


Figure 56 Horizontal axis

Under horizontal axis there is the trigger position slider. Use this slider to set the trigger position so that in single trig and triggered modes, this position can be referenced to sample data (Figure 57).

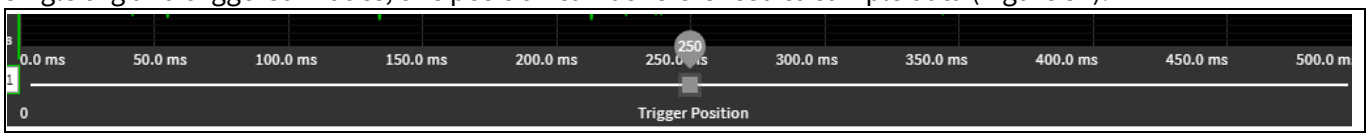


Figure 57 Trigger position slider

Another way to change the trigger position and the level is using the chart controls. When continuous mode is closed, if measurement cursors are not enabled, 2 dashed lines (one vertical and one horizontal) will be displayed on the chart. Use these lines to change the trigger position and trigger level.



Figure 58 Use dashed lines to change the trigger level and the position

6.5.6 Taskbar

There are 5 buttons in the taskbar. The first two buttons are used to export and import sampling data (as .csv files). The next button is used for opening or closing the device connection. Fourth button is used to open the user guide. Last button opens preferences window to set chart display preferences for the current session.

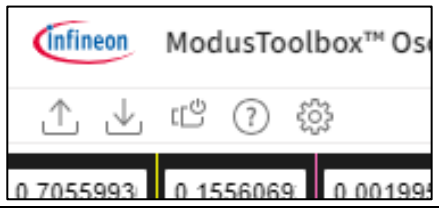


Figure 59 Taskbar buttons

These are the buttons with the display order:

6.5.6.1 Import button

Use import button to import a previously exported csv file. Only the csv files exported by the oscilloscope are supported. Importing a file preserves the state of the export conditions so that the scales and the channel settings are displayed at the same time as the export. As soon as a file is imported, scope mode changes to Stop mode. To continue sampling real time data, select the continuous mode again. The date of import and the firmware name are shown in the corner of the chart area as support information.

6.5.6.2 Export button

Use this button to export the current frame. Export button does not work in continuous scope mode, so select another mode before exporting the data.

6.5.6.3 Toggle connection button

Connect button opens a connection to the device in case the connection is lost. When there is a connection the connection indicator in the footer bar shows Running with a green led icon (Figure 60). Like the connect button, use the disconnect button to close the connection to the device. When the connection is closed the connection indicator in the footer bar shows Stopped with a red led icon (Figure 60).

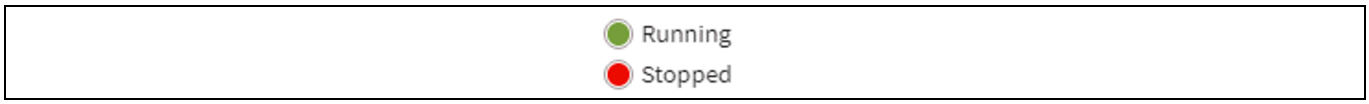


Figure 60 Connection status indicator

6.5.6.4 User guide button

This button opens a separate window with the user guide (this document).

6.5.6.5 Preferences button

Preferences button opens preferences window. In this window change the trace display mode, interpolation, line width, color and marker size. Apply these changes to one or more channels for some of the preferences. All these changes are applied during the session and are not saved for future use.

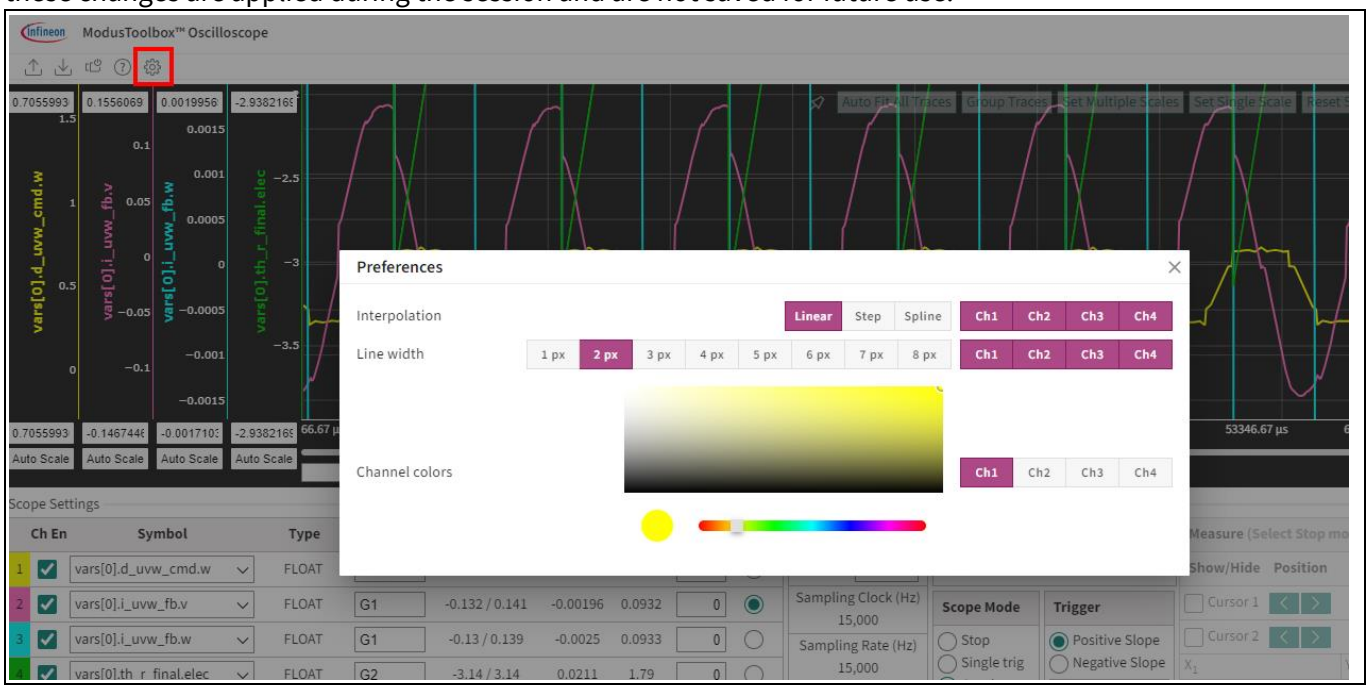


Figure 61 Preferences window

6.5.7 Data Persistence

After closing the oscilloscope, latest data and channel preferences are saved to temporary memory. During the session, channel selections, trigger settings and divider value will be reapplied, so that new oscilloscope session starts with the previous setup. When user saves a project, these oscilloscope data are also saved to the project file for future use.

6.6 Troubleshooting

6.6.1 GUI troubleshooting

If you see this	The likely cause is	Quick fix action
All controls are greyed out	Acquisition status is "error".	The host application must clear up the error. Check host application notifications to fix the error.
No waveform appears on the chart	All channels are disabled or no signal data is received.	Tick Ch En for at least one channel. Confirm the data source is active.
Waveform is a flat horizontal line	Signal amplitude is outside the visible Y-axis range.	Click Auto Fit All Traces.
Cursor measurements show no values	Scope is not in Stop mode.	Click the Stop radio button, then re-enable cursors.
Cursor fields empty in Stop mode	The cursor is not enabled.	Tick the Cursor 1 or Cursor 2 checkbox.
Waveform updates too slowly	Divider is too low, resulting in very high sample rate.	Increase the Divider value in the Sampling Clock panel.
Trigger never fires	Trigger level is outside the signal amplitude range or trigger channel disabled.	Enable the trigger channel; drag the trigger level line onto the waveform. Verify the correct slope is selected.
Chart does not respond to mouse	A dialog from the host application blocks the chart area.	Close any open dialogs and click once on the chart.

6.6.2 Firmware integration troubleshooting

If you see this	The likely cause is	Quick fix action
Build error: 'MTB_OSCILLOSCOPE_MAX_CH' undeclared	mtb_oscilloscope_cfg.h not on the compiler include path.	Verify the file exists in your source dir and the dir is in Makefile INCLUDES.
Time axis wrong (too fast or slow)	Clock frequency in mtb_oscilloscope_init() does not match ISR rate.	Measure the ISR period and pass the correct Hz value to mtb_oscilloscope_init().

Glossary

Acquisition
The process of sampling signal data from the hardware target and transferring it into the display buffer.

COM
Communications port (COM)

CSV
Comma Separated Values (CSV)

Glossary

Cursor
A movable vertical line on the chart for measuring time and amplitude at a specific sample position.

Divider
An integer divisor applied to the hardware sampling clock to reduce the effective sampling rate.

GUI
Graphical User Interface (GUI)

Hold Off
Minimum number of samples between consecutive trigger events.

ISR
Interrupt Service Routine — a firmware function called by a hardware timer at a fixed frequency.

MTB, ModusToolbox™
Infineon integrated development environment for PSOC™ and XMC™ MCUs.

PC
Personal Computer (PC)

RMS
Root Mean Square — statistical measure of signal magnitude.

SN
Serial Number (SN)

SWD / JTAG
Serial Wire Debug / Joint Test Action Group — debug interfaces for embedded device memory access.

Trigger
A configurable condition (level + edge) that initiates a new data capture frame.

USB
Universal Serial Bus (USB)

Revision history

Document revision	Date	Description of changes
**	2026-06-24	Initial release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2026-06-24

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2026 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email:

erratum@infineon.com

Document reference

002-43312 Rev. **

Important notice
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”).

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.