Moku Time & Frequency Analyzer

User Manual

LIQUID
INSTRUMENTS

Table of Contents

Introduction

The Time & Frequency Analyzer records the timestamps of user-defined events in measured signals, with optional computation and analysis of the intervals between events. The instrument (shown in Figure 1) can be used to characterize the number and rate of occurrence of transient events in time-domain signals and offers integrated data logging, statistical analysis, histogram view and analog output. Instrument specifications can be found in your Moku's specification sheet.

This manual is intended to help users understand the user interface and underlying architecture of the instrument. It also includes a general example in the quick start guide and a small number of in-depth examples to provide a foundation for new users.

These user manuals are tailored to the graphical interfaces available on macOS, Windows, iPadOS, and visionOS. If you'd prefer to automate your application, you can use Moku API; available for Python, MATLAB, LabVIEW, and more. Refer to the API Reference to get started.

AI-powered help is available to aid both workflows. AI help is built into the Moku application, and provides fast, intelligent answers to your questions, whether you're configuring instruments or troubleshooting setups. It draws from Moku manuals, the Liquid Instruments Knowledge Base, and more, so you can skip the datasheets and get straight to the solution.

Access AI help from the main menu ☰.

For more information on the specifications for each Moku hardware, please refer to our product documentation, where you can find the specifications and the Time & Frequency Analyzer datasheets.

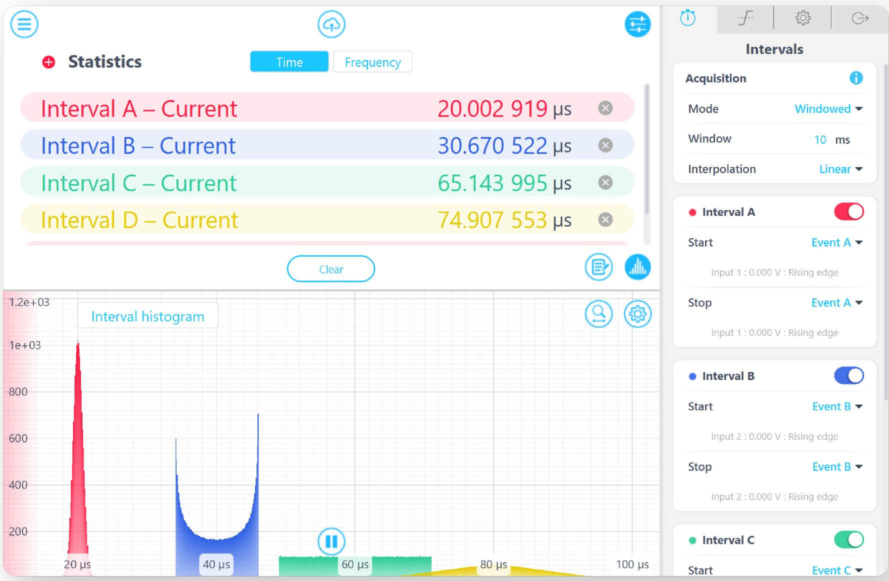


Figure 1. Time & Frequency Analyzer user interface showing the statistics panel (top), histogram panel (bottom) and the settings panels (right).

Quick start guide

Here we outline how to make a rise time measurement in order to highlight a typical use case of the Time & Frequency Analyzer. Further, more detailed, examples may be found in the examples section.

In this example we measure the mean rise time of a 1 Vpp, 1 MHz, zero-mean, square wave signal. The signal is connected to Input 1 of the Moku front panel.

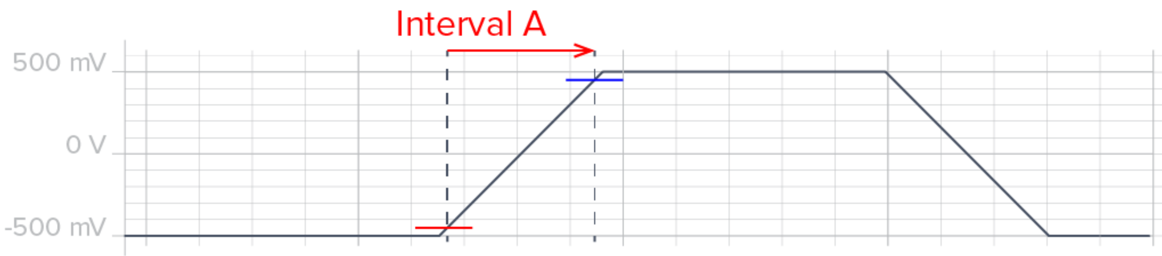


Figure 2. The rise time illustrated by measuring the time interval between event A (red) and event B (blue).

- **Step 1: Configure acquisition settings**
  - Choose between windowed, gated and continuous
    Here we select a windowed acquisition with duration 10 ms to give fast updates. This will allow us to collect a large number (~10, 000) of samples in each window period and reduce variance with averaging.
  - Decide whether linear interpolation is appropriate
    In this example we expect a smoothly varying, low-noise signal with several points on the edge of interest. We thus enable interpolation.
- **Step 2: Configure event detectors**
  - Select the appropriate source and input settings
    Here our signal of interest is connected to Input 3. We set the input range to 4 Vpp to avoid any clipping of our 1 Vpp signal. We also choose DC coupling, 300 MHz range, and 1 MΩ impedance. In order to measure rise time we consider two events: Event A, where the signal crosses the 5% threshold of the edge amplitude and Event B, where the signal crosses the 95% threshold of the edge amplitude. Both events share the same source and thus have the same input settings.
  - Configure the threshold, edge and holdoff settings
    We configure event A to use a threshold of -450 mV and event B to use +450 mV. Both edges are set to rising. As we are not concerned with spurious threshold crossings, we leave the holdoff at 0 s.
- **Step 3: Configure intervals**
  - Define an interval via its start and stop events
    We are interested in the time elapsed between events A and B and configure our interval to start at event A and stop at event B, shown in Figure 3.
- **Step 4: Configure live statistics**
  - Select interval statistics of interest from current, minimum, maximum, mean and count. Configure the histogram to show intervals.
    We decide to display the duration of the current and mean interval statistics over our measurement window. We use the "Autoset start/stop" to optimize the histogram to display all measurements captured within the acquisition window.

- **Step 5: Configure outputs**
  - Assign analog outputs to interval duration or count and scale as needed
    We do not require this feature in this example.
- **Step 6: Log or export data**
  - Log raw timestamps using the embedded Data Logger or save live statistics and histograms. Configure which events to include in logging from the events settings panel.
  - We save the histogram as a CSV for offline analysis by clicking the icon.

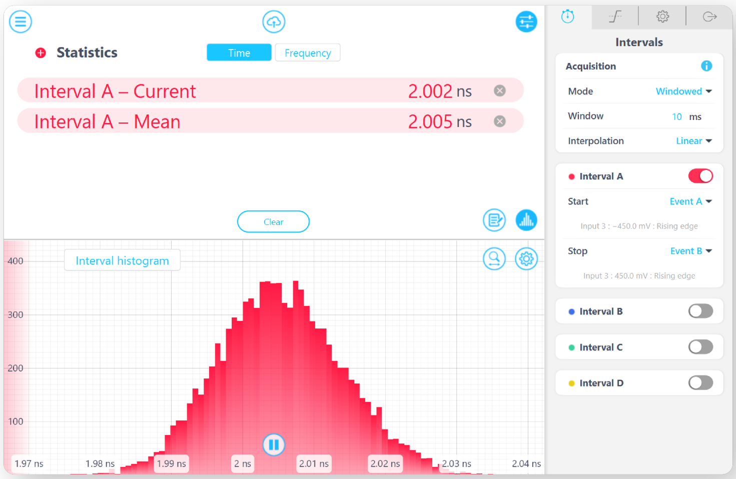


Figure 3. Rise time measurement, indicating a mean rise time of 2.005 ns.

Acquisition

- Type: Windowed
- Period: 10 ms
- Interpolation: Linear

Event A

- Source: Input 1
- Threshold: -450 mV
- Edge: Rising

Event B

- Source: Input 1
- Threshold: 450 mV
- Edge: Rising

Interval A

- Start: Event A
- Stop: Event B

Figure 3 shows the Moku application configured as outlined above, showing an average rise time of 2.005 ns with an approximately normal distribution.

Principles of operation

Architecture

The principal objectives of the Time & Frequency Analyzer are to detect events and to measure the intervals between them. The instrument can show live statistics and histograms, log to memory or generate analog voltage output. Understanding the instrument's fundamental architecture will enable you to make more precise and accurate measurements on your Moku device.

The core to concepts of the Time & Frequency Analyzer are events and intervals, defined as follows:

Events are threshold crossings identified according to user-defined parameters (threshold level, edge type and holdoff).

Intervals are the time elapsed between two selected events.

Figure 4 shows the signal flow in the Time & Frequency Analyzer. Input signals can be routed to one or more event detectors. Detected events can then be sent to any interval analyzer or logged in the Data Logger. Each interval analyzer can measure a combination of different event detectors. The interval analyzer outputs are used to create histograms, generate statistics, or drive an analog output.

This flexible layout enables many different measurement paradigms, as shown in Figure 4.

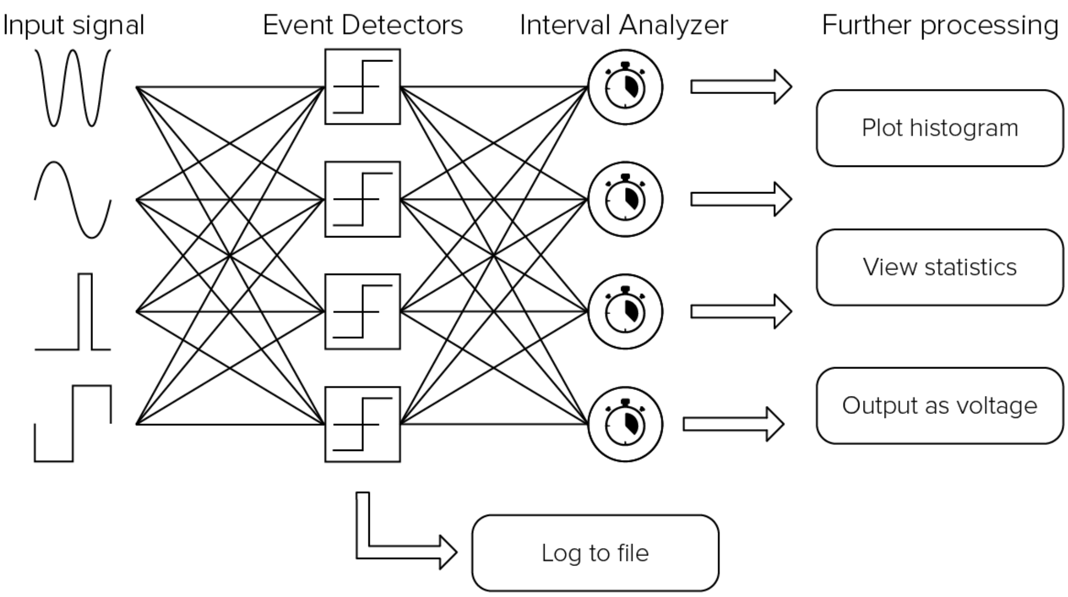


Figure 4. Architecture of the Time & Frequency Analyzer.

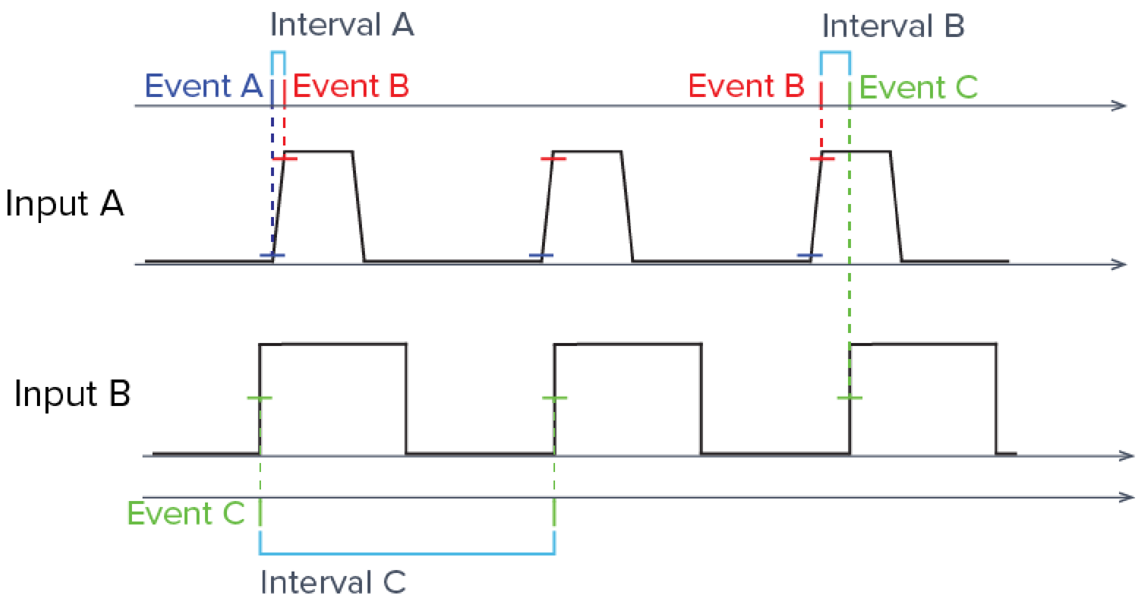


Figure 5. Intervals may be formed by several different combinations of events. The Time & Frequency Analyzer can measure the interval between successive occurrences of a single event or between two different events from one or more inputs.

Figure 5 outlines several of the ways the Time & Frequency Analyzer can be configured to measure different intervals. **Interval A** measures the rise time of Input A by starting on Event A and ending on Event B. **Interval B** begins on Event B and ends on Event C, effectively measuring the phase delay between Input A and Input B. **Interval C** counts the period of Input B by starting and ending on Event C.

Using the instrument

Acquisition mode

The acquisition mode determines how the statistics, histogram and output signals are computed. Windowed, gated and continuous acquisition are available and can be used with `Linear` or `None` interpolation (Figure 6).

Changes to the acquisition settings will reset the statistics and histogram, which can also be done by pressing 

.

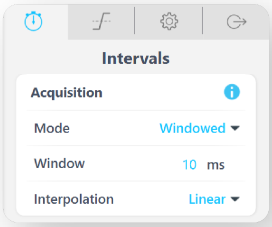


Figure 6. Acquisition settings panel.

Windowed

Statistics, histograms and output signals are computed over each window period and updated when the window ends (see Figure 7). These settings can be configured as shown in Figure 8.

By choosing the window length, you can balance averaging and update rate.

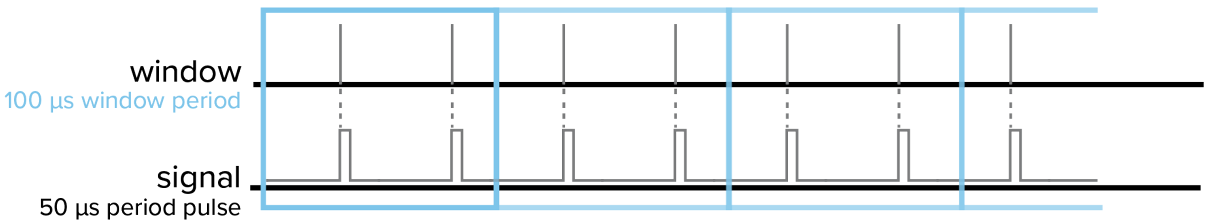


Figure 7. Windowed acquisition detects two events (grey) in each of the configured window periods (blue) from the underlying signal (black). As events reset at the end of each acquisition period, two events are detected in each window.

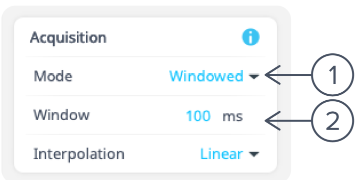


Figure 8. Windowed acquisition settings to capture events in 100 ms windows

(1) Select windowed acquisition mode.

(2) Select the window period in which to capture and read statistics, histograms and output signals.

Gated

In gated acquisition mode, data is considered only when an external gating signal is above a user-defined threshold, as shown in Figure 9. The gate input source can be any of the Moku device's input channels or an external trigger (if available). Similarly to windowed acquisition, statistics and histogram data are reset at the end of each gate period.

Gated acquisition is useful when the input signal is only meaningful at certain times, such as after a stimulus. Figure 10 shows how to configure gated acquisition settings.

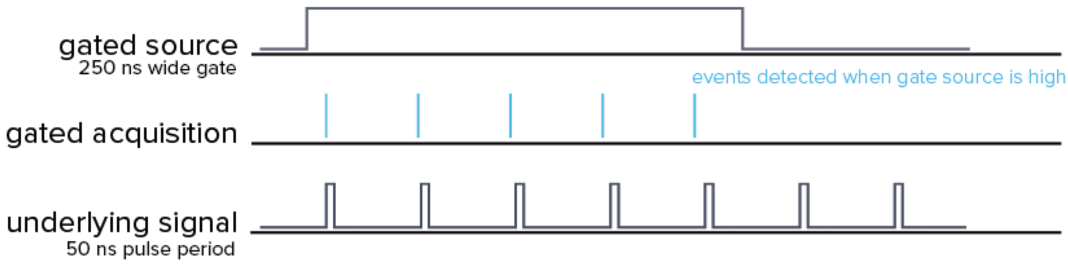
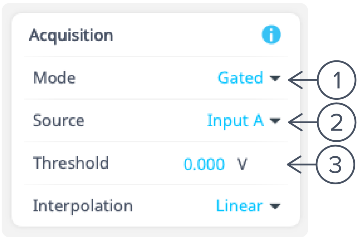


Figure 9. The gate is active and events are registered while the gate signal is above threshold.



① Select gated acquisition mode.

② Select the gate source where the gate period will be calculated.

③ Choose the threshold voltage for the gated signal to determine the gate period.

Figure 10. Gated acquisition settings.

Continuous

In continuous mode, statistics, histograms and output signals are computed continuously (see Figure 11). Displayed values are updated every time the screen is refreshed (approximately every 50 ms). Statistics and histograms can be cleared manually by clicking Clear.

Figure 12 illustrates how one may select this mode.

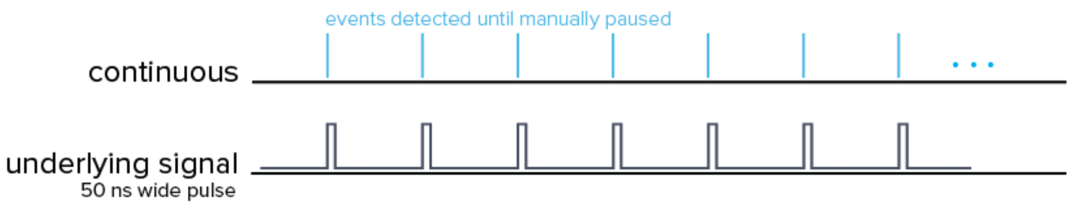
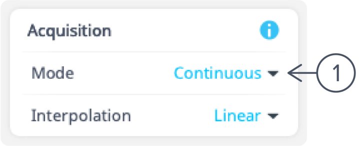


Figure 11. Continuous acquisition.



① Select continuous acquisition mode.

Figure 12. Continuous acquisition settings.

Interpolation

The option to apply `None` or `Linear` interpolation is available for all acquisition modes.

When `None` interpolation is enabled, the first sample above the event detection threshold defines the event's timestamp (Figure 13, left). This limits time resolution to the device's maximum sample rate; for example, a 1.25 GSa/s rate yields an 800 ps resolution.

`Linear` interpolation can improve temporal resolution if the input signal is roughly linear around the threshold crossing (Figure 13, right). However, it may introduce a slight bias towards multiples of the minimum time resolution for non-linear signals. Read more on how interpolation affects a signal measurement.

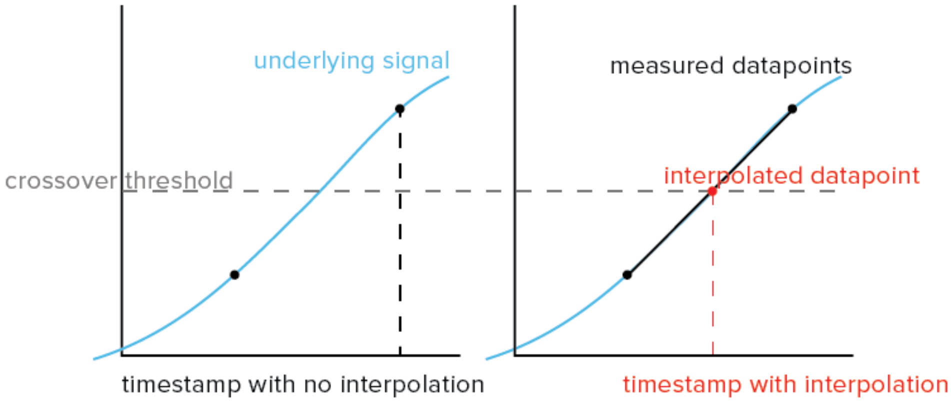


Figure 13. Timestamp differences between None (left) and Linear (right) interpolation.

When to use linear interpolation

Linear interpolation is useful for waveforms that are approximately linear around the event detection threshold, or change slowly relative to the sample rate of your Moku. It reduces the effects of time and amplitude quantization (Figure 14).

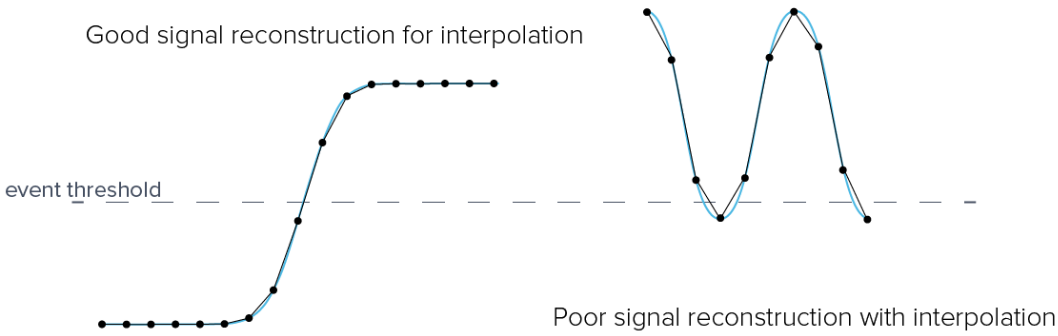


Figure 14. Examples of good and poor signal reconstruction with Interpolation.

For signals that are not linear in the region of the threshold, interpolation can bias event timestamps towards multiples of the sampling frequency, especially in signals with high-frequency components. This can be mitigated using a low-pass filter or selecting `None` interpolation.

Events

The acquisition and event controls are set in the events setting panel shown in Figure 15.

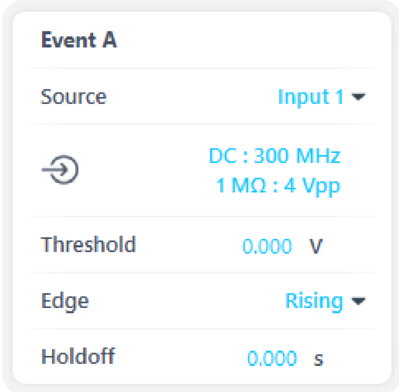


Figure 15. Events settings panel.

Event source Select the event source. This can be from an internal (used in Multi-Instrument Mode) or external input (Input 1, 2, etc., or external trigger inputs).

Event frontend settings If available, set the input impedance, coupling and range of your input signal.

Threshold An event is detected when the selected edge of the input signal crosses a threshold voltage. Therefore, the trigger level must be within the signal's bounds to detect events.

Edge The edge chosen for event detection can be a rising, falling or both rising and falling edges. The device looks for threshold crossings with the correct edge properties.

Holdoff Holdoff begins when an event is detected.

Logging Events

Toggle each event detector ON or OFF to include or exclude each one from logging (Figure 16). Read more about logging events.

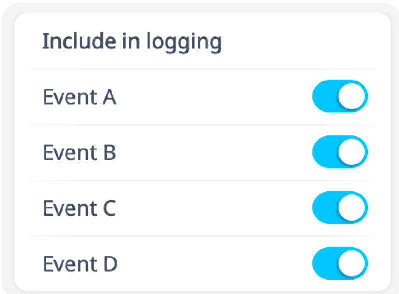


Figure 16. Event logging panel.

Event detection

Events are detected when a signal crosses a set threshold in the defined direction (rising, falling or both).

An optional **holdoff** parameter prevents unwanted event detection (Figure 17). After an event is detected, the detector ignores additional events until the holdoff period expires. Without holdoff, all events meeting the edge and threshold criteria are logged, which can be undesirable for packetized signals or noisy asymmetric pulses like those from avalanche photodiodes (APDs) or superconducting nanowire single-photon detectors (SNSPDs).

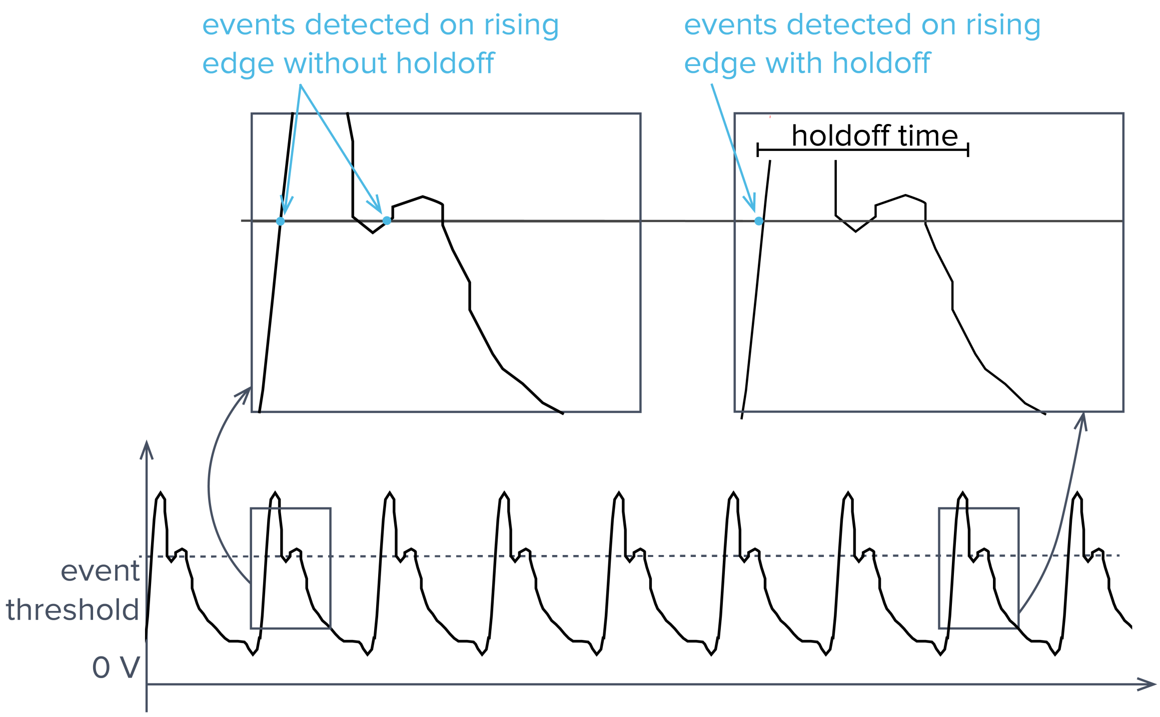
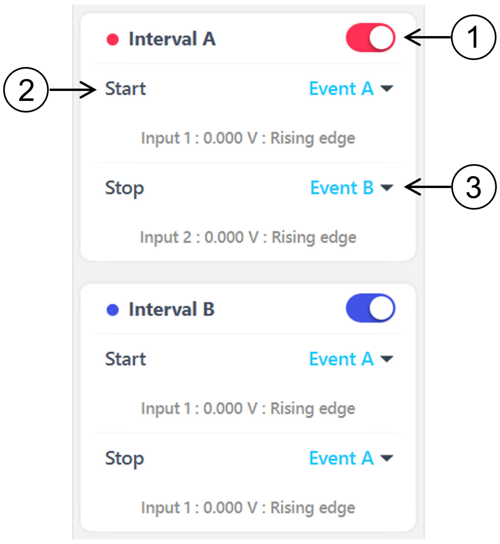


Figure 17. The holdoff feature can be enabled to avoid detection of unwanted events.

Intervals

Intervals are computed based on the settings in the intervals panel (Figure 18) and the advanced settings panel (Figure 20). These intervals generate the Time & Frequency Analyzer's statistics, histograms, and outputs.

To count individual events or estimate frequencies, start and stop the interval on the same event (see Frequency Counter Example).



① **Interval toggle** Turn ON and OFF the statistics and histogram channel corresponding to the Interval.

② **Start** Selects the event on which to start the interval.

③ **Stop** Selects the event on which to stop the interval.

Figure 18. Intervals settings panel.

In the case where more than one start event occurs before a stop event, the "multiple starts" option can determine which interval to use. Figure 19 shows the "Use first" of multiple starts (dashed line) will have a longer interval than when "Use last" of multiple starts (solid line).

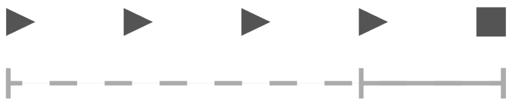


Figure 19. Multiple start event options; first (dashed line) and last (solid line).

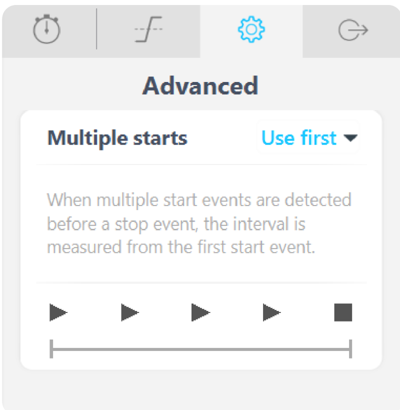


Figure 20. Advanced settings panel, where you can select which event to begin an interval on when multiple are detected.

Output

The Time & Frequency Analyzer can output the interval duration or count as an analog voltage, using a configurable scaling factor. The output values may optionally be offset or inverted.

Outputs are matched to their corresponding intervals. For example, Output 1 is always matched to Interval A, Output 2 to Interval B, and so on for all channels.

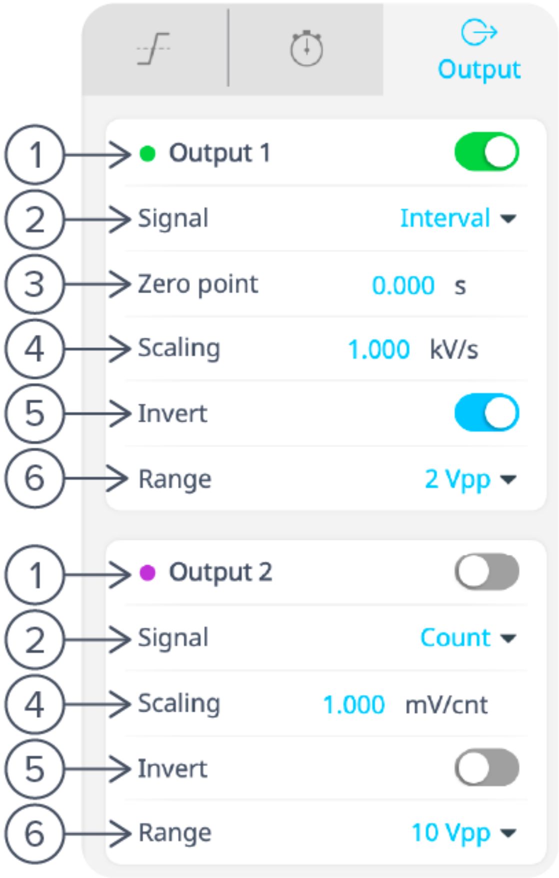


Figure 21. Output panel settings for Interval (Output 1) and Count (Output 2).

Output controls

Interval outputs are updated continuously regardless of the acquisition mode. Count outputs are updated at the end of each acquisition window or gate period. In continuous mode, the count is updated continually.

To output your interval duration or count:

- ① Toggle the Output ON.
- ② Choose the count or interval output signal.
- ③ Set the zero point to your desired value, for interval outputs.
- ④ Select the desired scaling ratio of the interval duration (in seconds) to volts.
- ⑤ Optionally invert the output signal.
- ⑥ If available, select the output range of your signal.

Interval output

Interval output outputs the interval duration as a voltage. To center the signal around 0 V, match the **zero point** to the interval period. The variance in the interval duration will then appear as voltage fluctuations around zero.

The interval **scaling** is the conversion from the interval (in seconds) into volts.

```
Output voltage = (interval duration - zero point) * scaling factor
```

For example, if the interval is 1 ms and the zero point is set to 1 ms, the output will be 0 V. With an interval of 2 ms and a scaling factor of 1 kV/s, the output will be (2 ms - 1 ms) * 1000 V/s = 1 V.

Count output

Count output outputs the interval count as a voltage.

For example, if the **count scaling** to is set to 1 mV/count, the output of an interval with 1000 counts per window period will be 1000 counts * 1.0 mV/count = 1 V.

Statistics

All statistics (Figure 22) are based on the intervals that are set in the intervals settings tab. See intervals for how to set the interval measurement.

Event statistics show the statistics of intervals and interval count within the acquisition period. Statistics update at the end of each acquisition period, either at the end of each window or gate period or continually when in continuous acquisition mode.

Tap on the statistic to change the interval it measures.

Clear all acquired data using the Clear button. This will reset the statistic, histogram and signal outputs.

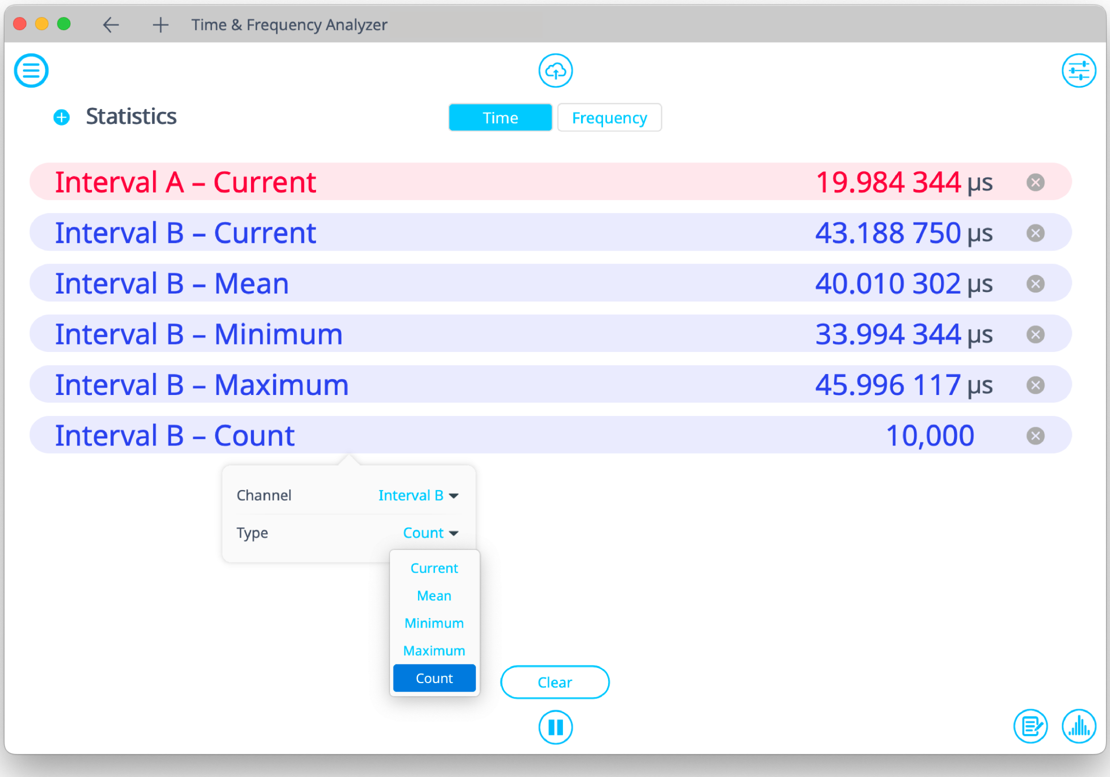


Figure 22. Statistics panel user interface.

Add or edit a statistic

To add a new statistic tap the plus button ⊕. This will add another statistic of the next type in the list.

To remove a statistic click the ⊗ button on the right side of the statistic.

To edit a statistic tap or click on it. Here you can change the interval it measures and the type of statistic measurement.

To change units select the units, time or frequency, that you prefer. This will only affect the units of the statistics.

To clear data press the clear button Clear. This will clear data for all functions, including statistics, histograms and output signals.

To export data press the export button at the top of the screen, see sharing and exporting data.

Computation of statistics

Interval

Interval statistics can be viewed in **time or frequency units**. When set to frequency units, interval statistics function as a frequency counter (see the in depth examples). Available statistics include:

- **Current** The most recent measured interval. If within a gated or windowed acquisition mode, it is the last measured interval within this period.
- **Mean** The average of all intervals that occurred during the acquisition period (gated or windowed acquisition mode) or since the last reset (continuous acquisition mode).
- **Minimum** The shortest interval that occurred during the acquisition period (gated or windowed acquisition mode) or since the last reset (continuous acquisition mode).
- **Maximum** The longest interval that occurred during the acquisition period (gated or windowed acquisition mode) or since the last reset (continuous acquisition mode).
- **Count** The interval count is increased each time the interval is completed. This keeps increasing until manually cleared (in continuous mode), or counts the number of intervals completed within the window or gate period, updating at the end of each period (gated or windowed acquisition mode).

Partial or incomplete intervals are discarded at the end of a window or gate period. This means that the count of intervals is often one less than the number of events that have occurred in that window or gate.

Histogram

The histogram can be configured to display either the interval histogram, or a $g^{(2)}$ correlation function.

Interval histogram

The Interval histogram displays a collection of time measurements by dividing the time axis (x-axis) into fixed-width bins, with bars indicating the number of intervals within each bin (y-axis). **Bin width** adjusts automatically with the span, with a minimum of 1 sample period. View and configure the live histogram of each acquisition period in the histogram panel (Figure 23).

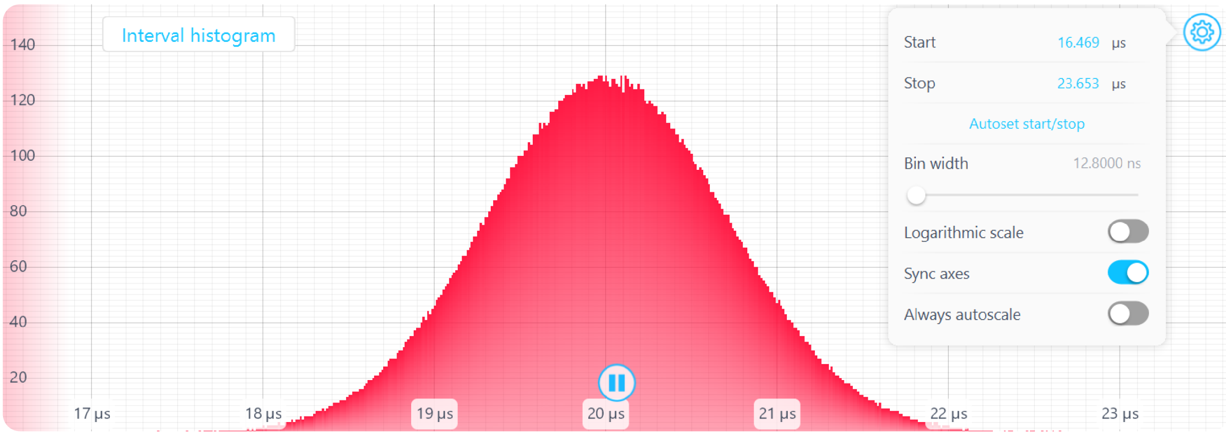


Figure 23. Histogram interval panel in the Time & Frequency Analyzer.

$g^{(2)}$ correlation function histogram

For quantum applications, it is often essential to characterize lights' statistical emission properties. One way to determine these statistical properties of light emission is through the $g^{(2)}(\tau)$ correlation function, a widely used metric in classical and quantum optics. The second-order correlation function, $g^{(2)}(\tau)$, measures the fraction of events occurring in a time interval τ relative to the average number of detected events over a span T.

In this mode, the x-axis of the plot represents τ (the time delay between events), and the y-axis represents the level of correlation.

- If $g^{(2)}(\tau) > 1$, then the events are corrrelated at that time delay. For example, $g^{(2)}(\tau) = 2$ means that there are twice as many intervals of that length compared to the average number of intervals separated by any time delay τ .
- If $g^{(2)}(\tau) < 1$, then the events are anti-correlated at this time delay. This means that intervals are less likely to occur with this time delay than compared to the average number.
- If $g^{(2)}(\tau) = 1$, then there is no correlation at this time delay, meaning that there are approximately as many intervals of that length are there are average number of intervals across all time delays.

Notably, the $g^{(2)}$ calculator of Time & Frequency Analyzer accounts for n-th photon cascading events, whereas the histogram function only plots first-order events. For more details on the calculation method, how to log these events, and its use in quantum optics, see the application note on the topic.

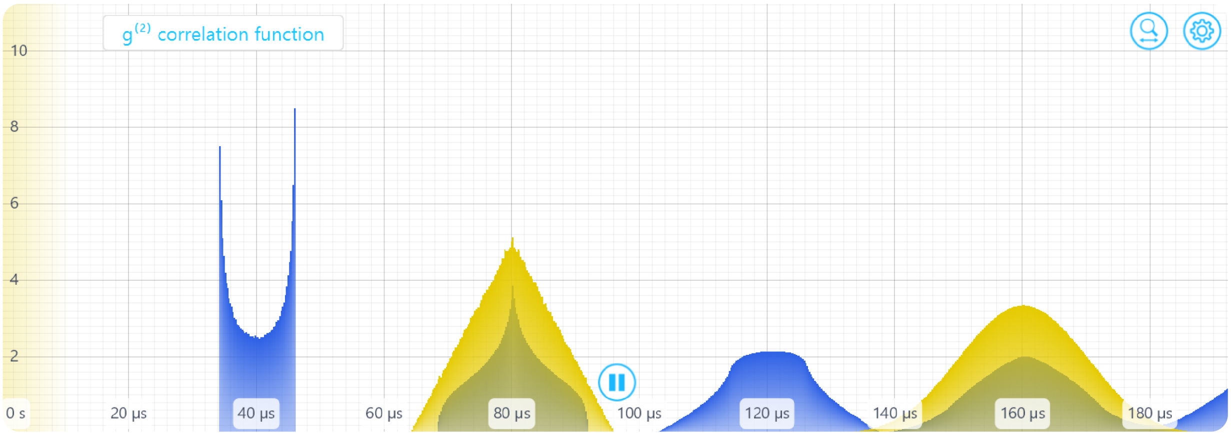


Figure 24. Histogram $g^{(2)}$ correlation function histogram.

Navigation

Navigating all histogram types as follows:

- **Zoom** Scroll to zoom along the major axis; hold Command/Ctrl while scrolling to zoom along the minor axis. Select the primary zoom-axis with the axis-zoom button.
  On iPad, use pinch gestures.
- **Pan** Click and drag to move the interval data or graph area.
- **Axes** Double click on either axis to autoscale it and center the graph.
- **Play/Pause** Pauses or resume data capture; zoom in on features while paused. Resume capturing after a trigger event or data export.
- **Right Click** Right-click on intervals, the graph area, or axes for more options.

Histogram data can be exported as a CSV file, MATLAB file or screenshot, see saving and sharing data. Resume streaming after exporting live data by pressing the play button.

The configuration panel settings can be set from the configuration button, or from the histogram context menu, which can be accessed by right clicking in the histogram area.

- **Start** Set the start of the histogram measurement.
- **Stop** Set the end of the histogram measurement.
- **Autoset start/stop** Sets histogram start and stop values once, to display all current data.
- **Bin Width** Select the bin width used in the histogram. The minimum bin width is determined by the device specifications and the histogram zoom level. From the minimum, data points are combined into bins to increase the bin width.
- **Logarithmic scale** Toggle ON/OFF a logarithmic scale for the histogram y-axis.
- **Sync Axes** scale of all visible Interval channels to match the selected Interval scale.
- **Always Autoscale** Continuously adjusts the y-axis of the selected Interval channel to fit graph.

Context menu

Right click (desktop) or long press (iPad) on the histogram graph area or Interval traces will open the corresponding context menu (Figure 25).

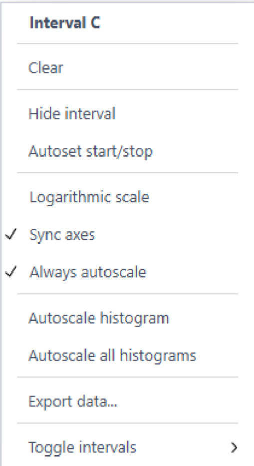


Figure 25. Context menu for the Interval trace.

- **Clear** all acquired data, resetting statistics, histogram and signal outputs.
- **Hide interval** Toggle the interval that is currently selected OFF.
- **Autoset start/stop** Determines the bounds of the current interval and sets the histogram start and stop values to show all data, once.
- **Logarithmic scale** Toggle ON/OFF a logarithmic scale for the histogram y-axis.
- **Sync axes** scale of all visible Interval channels to the same as the selected Interval scale, once.
- **Always autoscale** Continually track and scale the vertical axis of the selected Interval channel to remain on the screen.
- **Autoscale histogram** Scale the vertical axis of the selected Interval channel to remain on the screen, once.
- **Autoscale all histograms** Continually track and scale the vertical axis of all visible Interval channels to remain on the screen.
- **Export data...** Pauses acquisition and opens the live data export dialog.
- **Toggle intervals** Select the Interval channel to toggle ON/OFF, turning off the Interval channel in statistics and histogram.

Data logging

The Time & Frequency Analyzer features an embedded Data Logger that records raw event timestamps relative to the start of data acquisition (Figure 26). The data logs are saved on the Moku device and can be exported to a computer, iPad, or cloud service via the Moku application.

Logging stops either after the specified duration or when manually ended by the user. It can be configured to start immediately or after a user-defined **delay**, which helps reduce post-processing for signals with an initial deadtime. You must close the histogram to open the Data Logger and begin logging data, histograms will be disabled for the duration of logging.

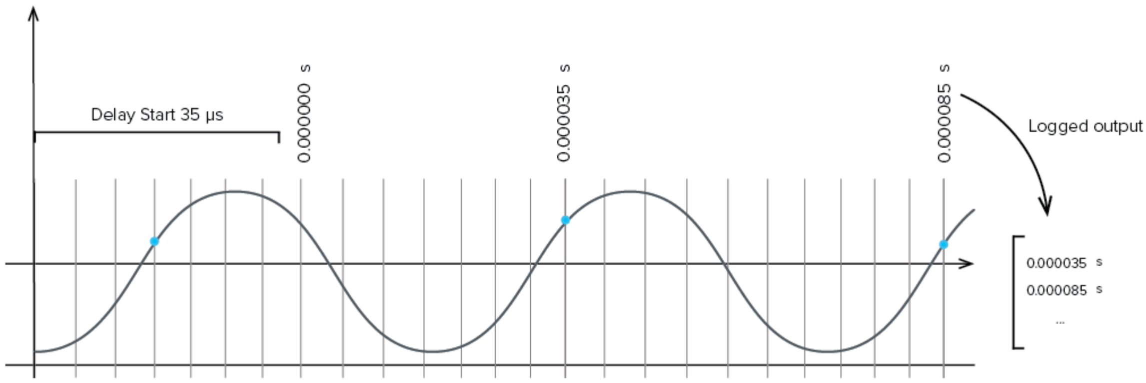


Figure 26. Event timestamps logged from the Time & Frequency Analyzer.

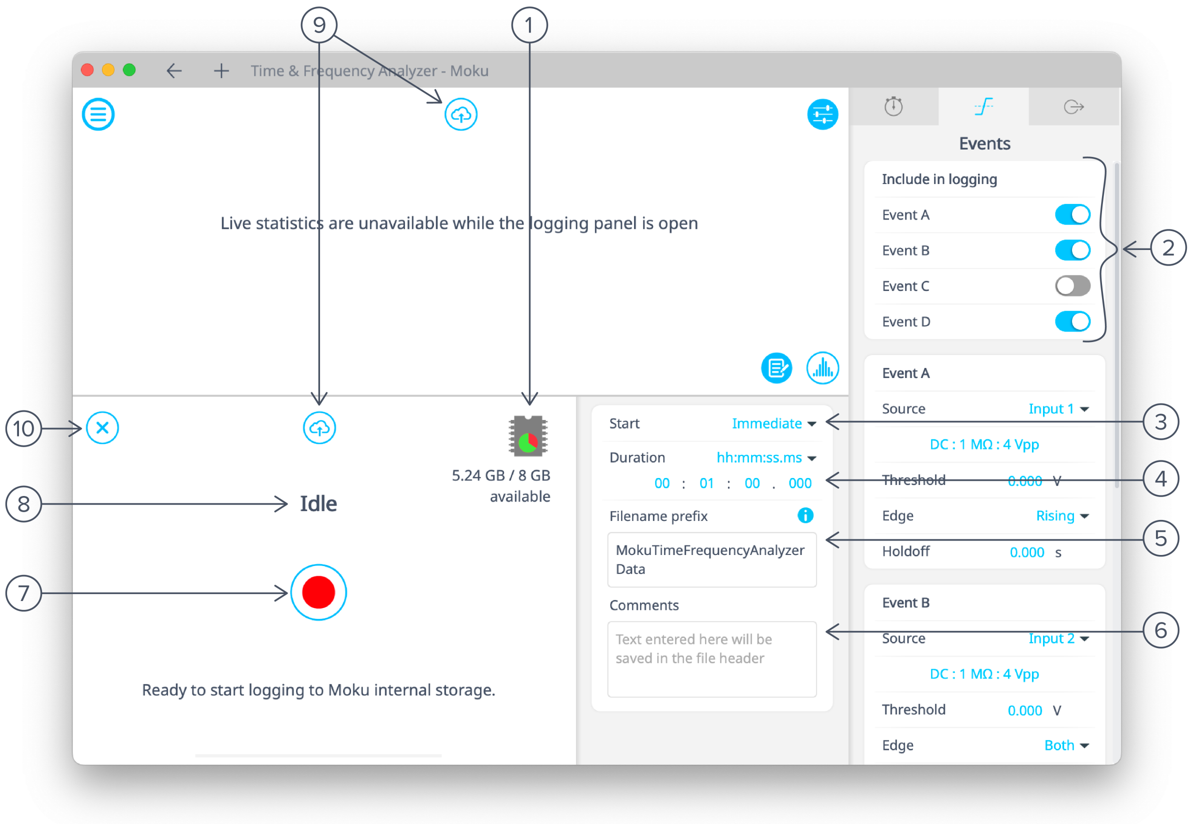


Figure 27. Data Logging user interface and relevant settings.

Figure 27 demonstrates the steps to configure the Data Logger. To set up a Data Logging session:

- ① Check there is enough **memory** remaining on your Moku device available for logging.
- ② **Event** Toggle the events to be logged ON or OFF
- ③ Configure the **start delay**, up to 240 hrs.
- ④ **Duration** Set the log duration, up to 10,000 hours.
- ⑤ Configure the **filename** prefix to be used on the data filename.
- ⑥ Add a **comment** to be saved in the file header.
- ⑦ Press the **start/stop** button to start your data logging (or press it to stop).

- ⑧ Check the data logger **status** changes from Idle to Logging (or "waiting" if using a delay).
- ⑨ **Share** or save your resulting log file, see saving and sharing data for more details on how to export data files.
- ⑩ Close the Data Logger panel to continue viewing the Interval statistics or histogram.

Exporting from the Data Logger

File types: Moku saves and exports logged data as LI files (.li), which can be converted to CSV, MATLAB, HDF5, or Numpy formats.

Data access: Export data logs to files, Dropbox, Mail, or iCloud (see sharing and saving data).

To log signal amplitude values along with event timestamps, refer to this example. For optimal data throughput, events may not be saved in sequential order.

The API can also be used to log, plot and stream data. Find examples of how to import your data in our API examples.

CSV files record events as event-timestamp pairs in rows, beginning with a header that is % commented and data in a single list. Each row represents an event, with the first column indicating the source channel (1 for Event A, 2 for Event B) and the second column providing the timestamp in seconds. The events are guaranteed to be ordered within each channel but not necessarily between channels

```
% Moku:Pro Time & Frequency Analyzer
% Windowed acquisition, window length 100 ms
% Linear interpolation
% Event A - Input A, 0.000 V, Rising edge, 0.000 s holdoff
% Event B - Input B, 0.000 V, Rising edge, 0.000 s holdoff
% Interval A - Start: Event A, Stop: Event A
% Interval B - Start: Event B, Stop: Event B
% Histogram - Start time 0.000 000 00 s, stop time 100.000 000 us
% Output A - Interval A, Zero point: 0.000 s, 1.000 0 kV/s, Invert off, 2 Vpp
% Output B - Interval B, Zero point: 0.000 s, 1.000 0 kV/s, Invert off, 2 Vpp
% Internal 10 MHz clock
% Acquired 2024-06-05 T 11:38:52 +1000
% Source (1 = Event A; 2 = Event B), Timestamp (s)
1, 1.3000109375000001e-05
2, 1.3199265625000001e-05
1, 1.3200140625000001e-05
1, 1.3399242187500001e-05
2, 1.3399984375000000e-05
2, 1.3599265625000001e-05
1, 1.3600078125000001e-05
2, 1.3799273437500001e-05
2, 1.3999289062500001e-05
2, 1.4199289062500001e-05
```

This CSV file can be imported into Python with the following script:

```python
import csv

def load_event_data(path):
    """ Load event and timestamp data from a CSV file.
        Args: path (str): The file path to the CSV file.
        Returns: list: list of tuples of (channel, timestamp) """
    with open(path, 'r') as f:
        load = csv.reader(f)
        data = [(int(row[0]), float(row[1])) for row in load if not row[0].startswith('%')]
        data.sort(key=lambda x: x[1]) # sort by timestamp
    return data

def events_by_channel(data):
    """ Convert event data into a dictionary of events by channel.
        Args: data (list): list of tuples of (channel, timestamp).
        Returns: dict: maps channel number to list of timestamps for that channel """
    channels = {d[0] for d in data}
    return {c: [d[1] for d in data if d[0] == c] for c in channels}

# Example usage:
# event_list = load_event_data('filename.csv')
# channel_events = events_by_channel(event_list)
```

Share data

Export data by clicking the share icon. One can export Live data and Logged data in several formats.

Live data

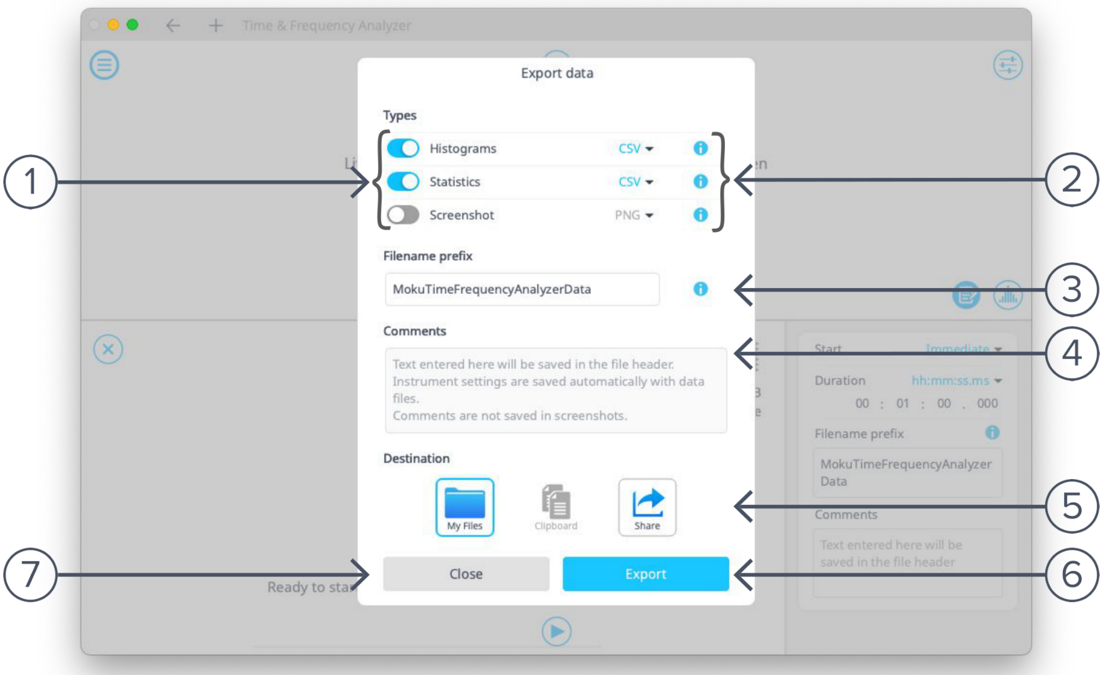


Figure 28. Data exporting user interface and settings.

To save live data:

① Select the type of data to export:

- **Traces** Saves the trace data for all visible signal traces, in either a CSV or MATLAB format.
- **Screenshots** Save the app window as an image, in either a PNG or JPG format.
- **Settings** Saves the current instrument settings to a TXT file.
- **Measurements** Saves the active measurement values, in either a CSV or MATLAB format.
- **High-res data** Saves the full memory depth of statistic values for all visible channels, in LI, CSV, HDF5, MAT or NPY format.

② Select the **export format**.

③ Select the **filename prefix** for your export. This is defaulted to “TimeandFrequencyAnalyzerMokuData” and can be changed to any filename of alphanumeric characters and underscores. A timestamp and the data format will be appended to the prefix to ensure the filename is unique.

For example: “MokuTimeandFrequencyAnalyzerData_YYYYMMDD_HHMMSS_Traces.csv”

④ Enter additional **comments** to be saved in any text-based file header.

⑤ Select the export **destination** on your local computer. If "My Files" or "Share" is selected, the exact location is selected when the "Export" button is clicked. Multiple export types can be exported simultaneously using "My Files" and "Share", but only one export type can be exported to the clipboard at a time.

⑥ **Export** the data, or

⑦ **Close** the export data window, without exporting.

Logged data

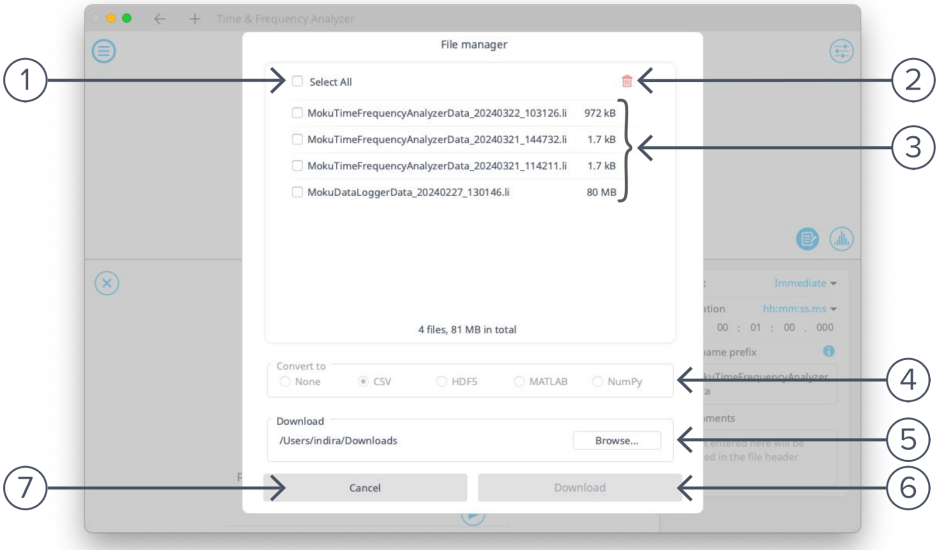


Figure 29. File exporting user interface and settings.

To save logged data:

① **Select all** files logged to the device's memory, to download or convert.

② **Delete** the selected file/s.

③ Browse and **select file/s** to download or convert.

④ Select an optional **file conversion format**.

⑤ Select a **location** to export your selected files to.

⑥ **Export** the data.

⑦ **Close** the export data window, without exporting.

Example configurations

This section details example settings and configurations to:

- Measure the time between events
- Set up the Time & Frequency Analyzer as a frequency counter
- Log Time & Frequency event timestamps and data log amplitude values
- Decode PWM

Measure time between events

To measure the time between events we need to set the interval of interest to begin on one event and end on another event (Figure 30). In this example a 10 kHz zero-mean pulse wave is connected to Input 1 and a 20 kHz zero-mean sine wave is connected to Input 2.

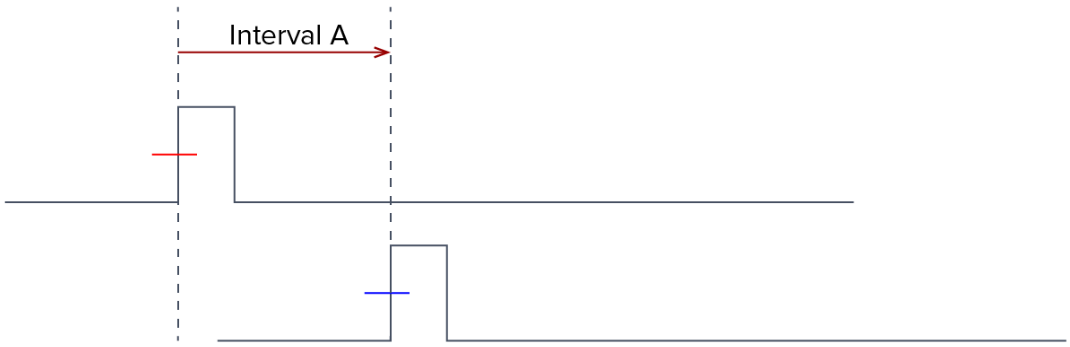


Figure 30. Interval measurement between two different pulse signals.

Setup the Time & Frequency Analyzer as shown in Figure 31:

- **Step 1:** Setup your Moku device
  - Connect signals to Input 1 (first event) and Input 2 (second event) and open the Time & Frequency Analyzer.
- **Step 2:** Configure event detectors
  - Detect events at the mean threshold, on the rising edge:
    - **Event A:** source: Input 1, threshold: 0 V, edge: rising, holdoff: 0.0 s
    - **Event B:** source: Input 2, threshold: 0 V, edge: rising, holdoff: 0.0 s
  - Set the analog front end settings
    - For both events, select DC coupling, 50 Ohm impedance, 4 Vpp range, and 300 MHz bandwidth. This will ensure the Time & Frequency Analyzer will accurately measure the events of the 10 kHz and 20 kHz, 1 Vpp, zero-mean signals, with impedances matched to the source.
- **Step 3:** Configure acquisition settings
  - Use windowed acquisition with a 100 ms window period and linear interpolation. This will ensure the intervals will repeat multiple times over the window to give an accurate mean interval measurement.
- **Step 4:** Configure the interval
  - Set Interval A to start on Event A and stop on Event B.
- **Step 5:** Configure live statistics and histogram

- Select current and mean interval statistics for Interval A.
- Select the histogram to display intervals ("Interval histogram"), then configure it to show the full range of acquired data by pressing `Autoset start/stop` in the histogram configuration or context menu.

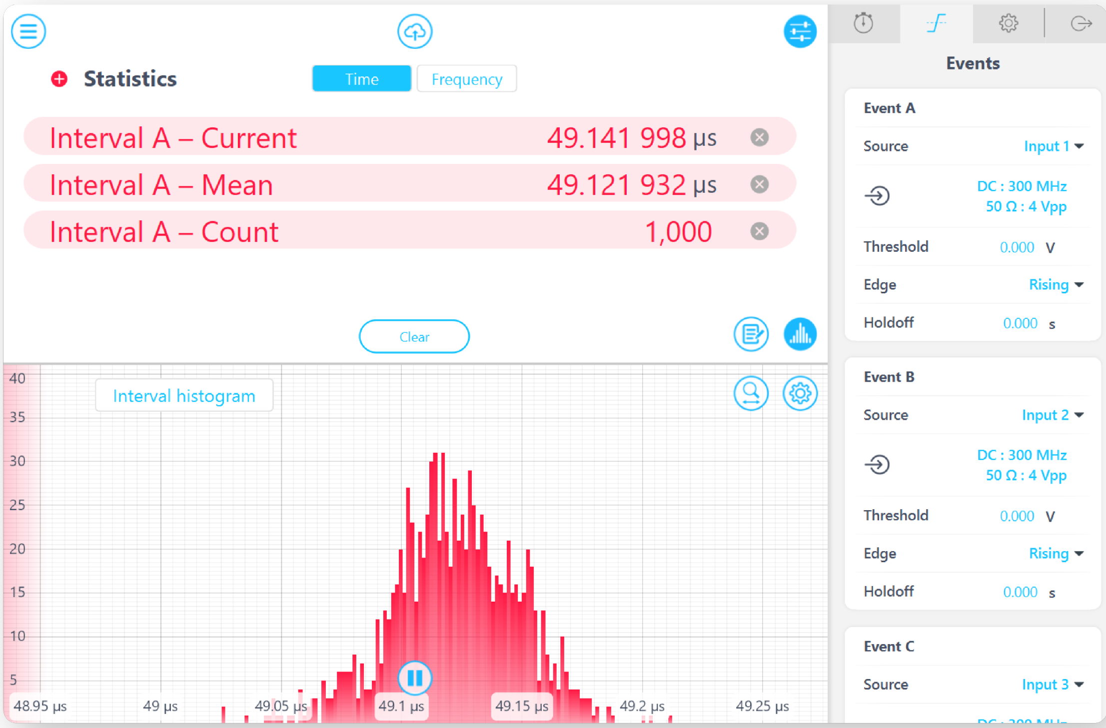


Figure 31. Screenshot of Time & Frequency Analyzer settings to measure the time between two events.

Frequency counter

To use the Time & Frequency Analyzer as a frequency counter set an Interval to start and finish on the same event (Figure 32). The statistics can be displayed in units of time or frequency.

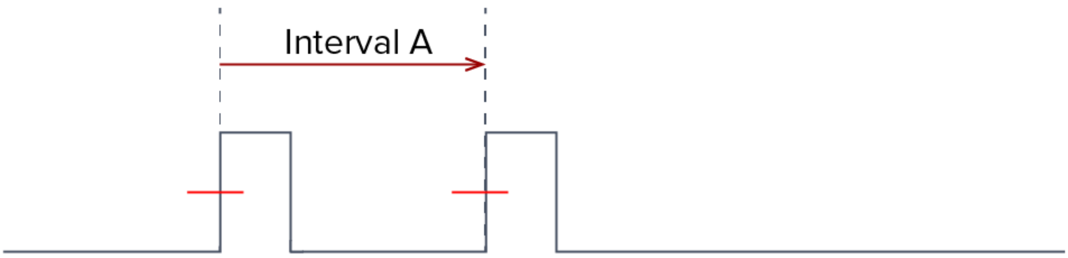


Figure 32. Interval measurement of a single event to count the signal's frequency.

Setup the Time & Frequency Analyzer as shown in Figure 33:

- **Step 1:** Setup you Moku device
  - Connect your signals to Input 1 and open the Time & Frequency Analyzer.

- **Step 2:** Configure event detectors
  - Detect events at the mean threshold, on the rising edge:
    - **Event A:** source: Input 1, threshold: 0 V, edge: rising, holdoff: 0.0 s
  - Set the analog front end settings
    - For both events, select DC coupling, 50 Ohm impedance, 4 Vpp range, and 300 MHz bandwidth. This will ensure the Time & Frequency Analyzer will accurately measure the signal's events, with impedances matched to the source.
- **Step 3:** Configure acquisition settings
  - Use windowed acquisition with a 100 ms window period and linear interpolation.
- **Step 4:** Configure intervals
  - Set Interval A to start and stop on Event A.
- **Step 5:** Configure live statistics and histogram
  - Select current and mean interval statistics for Interval A. Change the units to "Frequency", if preferred, at the top of the Statistics screen.
  - Select the histogram to display intervals ("Interval histogram"), then configure it to show the full range of acquired data by pressing `Autoset start/stop` in the histogram configuration or context ⚙ menu.

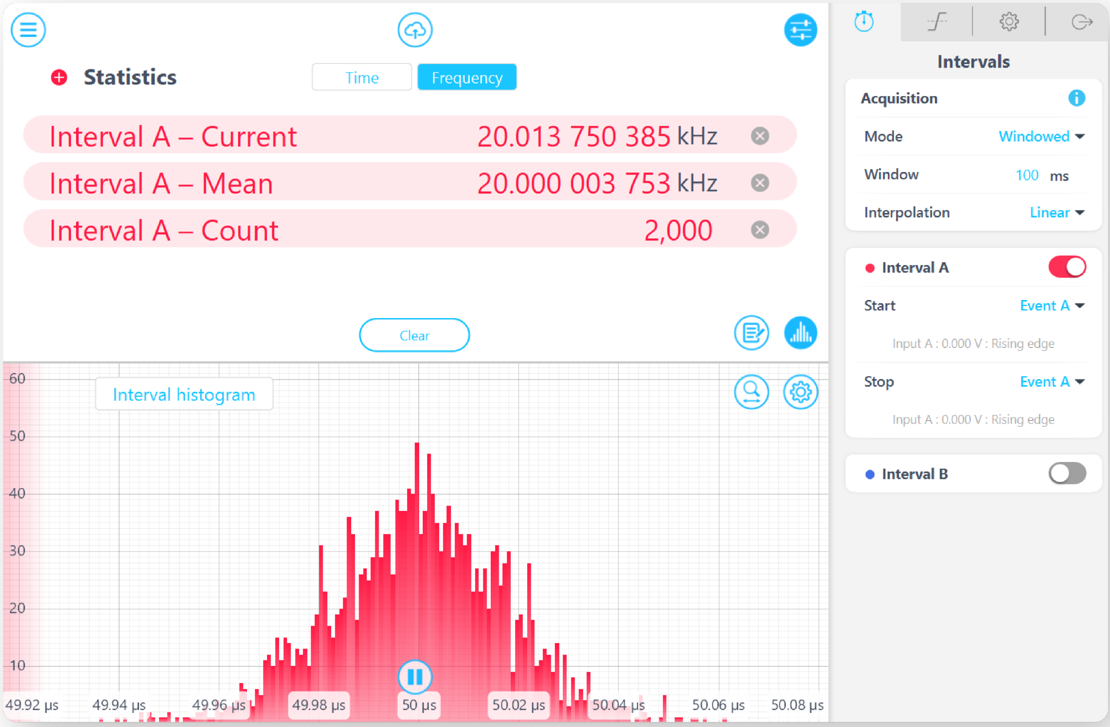


Figure 33. Screenshot of Time & Frequency Analyzer settings to count the signal's frequency.

Log Time & Frequency voltage and event signals

You may like to log the raw signal itself, alongside the event timestamps in the Time & Frequency Analyzer. To do this you can use Multi-Instrument Mode and route the signal to both the Data Logger and Time & Frequency Analyzer.

Figure 34. Multi-Instrument Mode, with Data Logger in Slot 1 and Time & Frequency Analyzer in Slot 2.

To do this:

- **Step 1:** Setup you Moku device
  - Connect to your Moku device and open Multi-Instrument Mode. Load the Data Logger into Slot 1 and the Time & Frequency Analyzer into Slot 2. Connect the signal of interest to Input 1, routing this to Input A of both Slot 1 and Slot 2 (Figure 34).
  - Sync the instruments, press to sync the instrument slots, found in the main menu .
- **Step 2:** Configure instrument settings as needed for your signal
  - Configure these as needed for your application.
  - Ensure the inputs and events to "Include in logging" are toggled ON in the Data Logger and Time & Frequency Analyzer respectively, so that it will be captured when logging. Figure 35 shows a completed log.

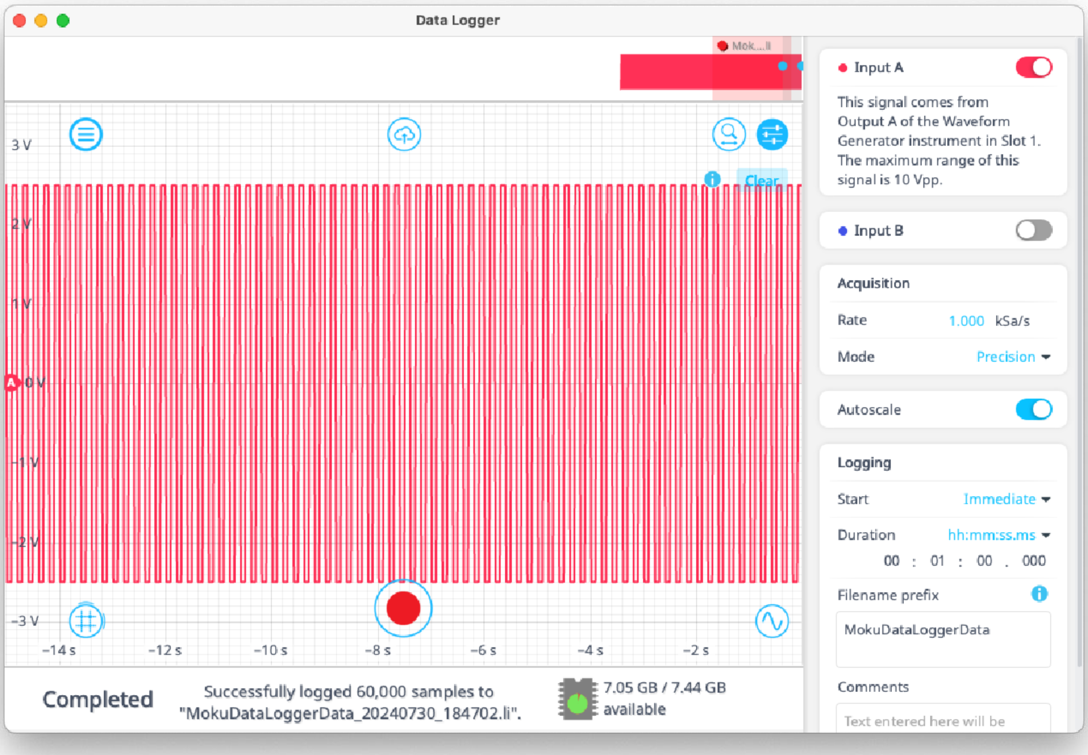


Figure 35. Successfully completed log in the Data Logger.

Decode PWM

To decode Pulse Width Modulation (PWM) we need to look at the interval width of each pulse, so we set one event detector to detect the rising edge and another to detect the falling edge (Figure 36).

The Time & Frequency Analyzer will output the interval as a scaled analog voltage. The variation due to PWM can be viewed in the Moku Oscilloscope when used in Multi-Instrument Mode. Alternatively, this signal can be recorded with the Data Logger or sent to the PID Controller as part of a closed-loop control system.

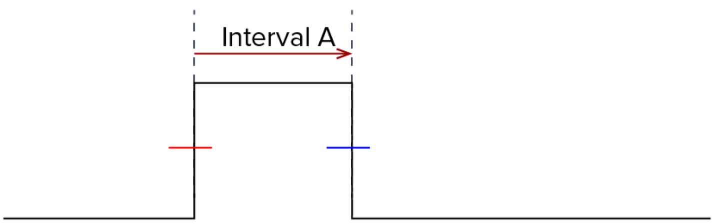


Figure 36. Interval measurement between two events set on the signal's rising and falling edges of a signal to measure the width of the signal's pulses.

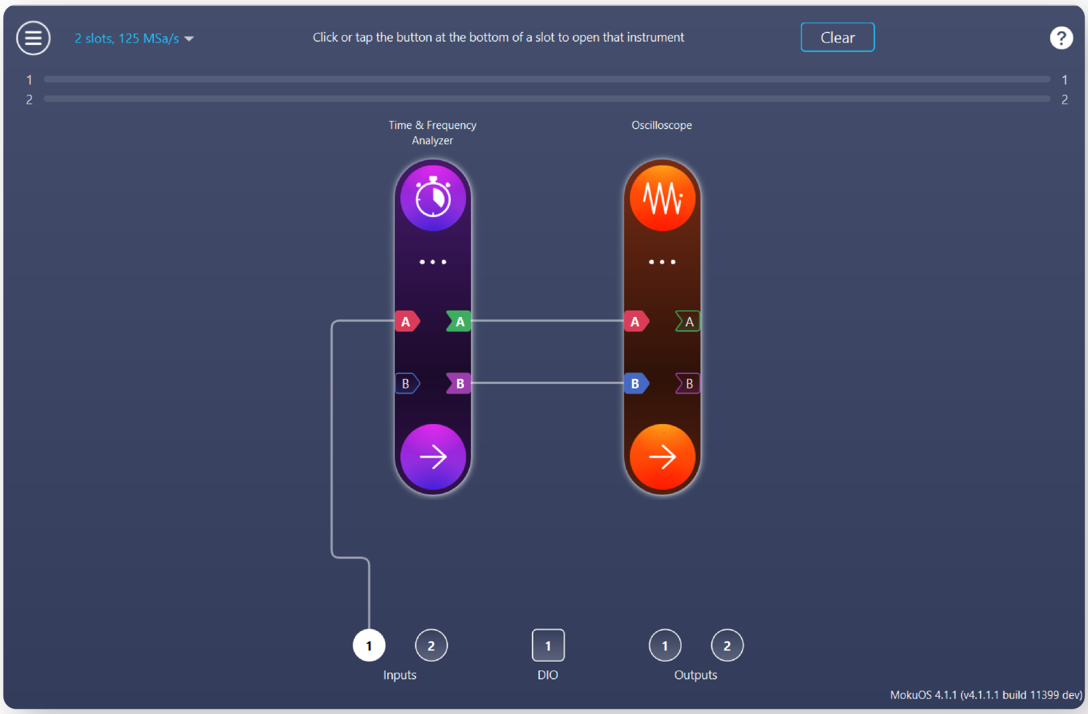


Figure 37. Multi-Instrument Mode setup of the Time & Frequency Analyzer and Oscilloscope used to decode PWM.

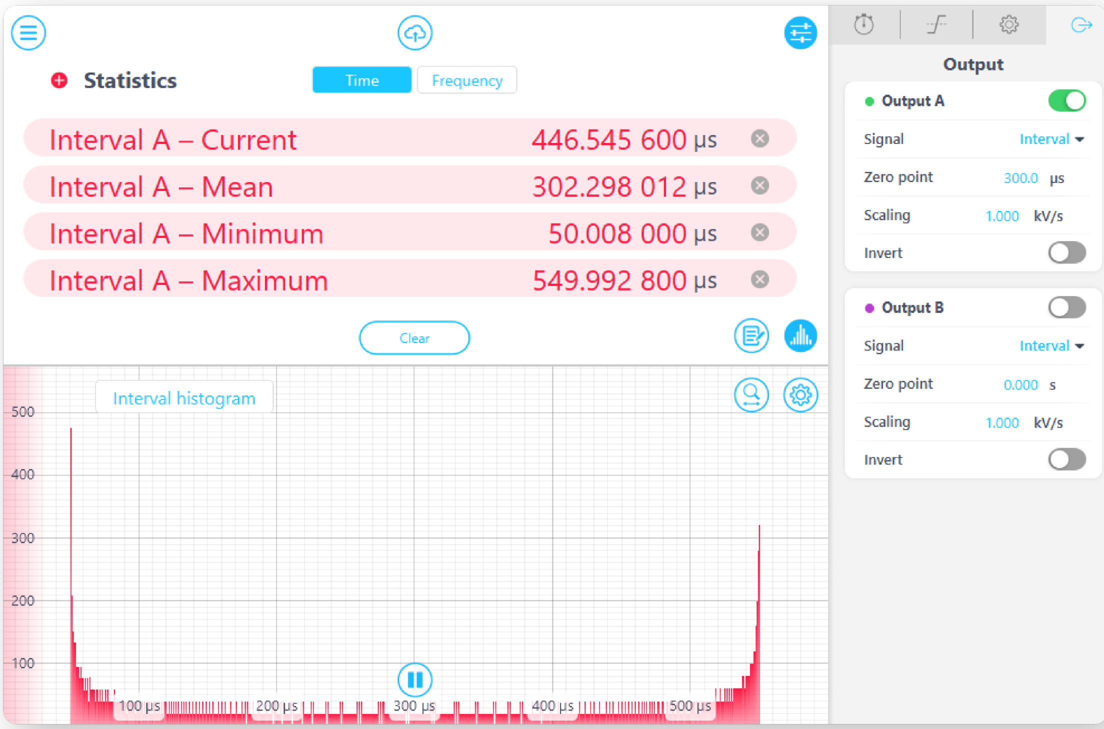


Figure 38. Time & Frequency Analyzer settings to decode PWM signals and output their interval length as a voltage.

- **Step 1:** Setup you Moku device
  - Connect your signal to Input 1 and open Multi-Instrument Mode.
    In this example we are using the Time & Frequency Analyzer in Slot 1 and the Oscilloscope in Slot 2.
  - Ensure the Slot 1 signal Output A is routed to the Input A of Slot 2, and press to sync the instrument slots, found in the main menu (Figure 37).
- **Step 2:** Configure event detectors
  - To measure pulse width, set one event to detect the rising edge and another to detect the mean of the falling edge. For a 1 Vpp pulse with a zero mean, use 0 V as the event threshold.
    - **Event A:** source: Input 1, threshold: 0 V, edge: rising, holdoff: 0.0 s
    - **Event B:** source: Input 1, threshold: 0 V, edge: falling, holdoff: 0.0 s
  - Set the analog front end settings
    - For both events, select DC coupling, 50 Ohm impedance, 4 Vpp range, and 300 MHz bandwidth. This will ensure the Time & Frequency Analyzer will accurately measure the signal's events, with impedances matched to the source.
- **Step 3:** Configure acquisition settings
  - Use windowed acquisition with a 100 ms window period and linear interpolation. This will ensure the intervals will repeat multiple times over the window to give an accurate mean interval measurement.
- **Step 4:** Configure intervals
  - Set Interval A to start on Event A and stop on Event B.
- **Step 5:** Configure outputs
  - Set Output A to interval mode, to output Interval A as a voltage.
  - Set the zero point to the mean value of the interval, measured to be 300 µs in this example, to show the variance around 0 V in the Oscilloscope.
  - With pulse widths ranging from 50 - 550 µs, scale by 1.0 kV/s (1 mV per µs) for visibility on the Oscilloscope.
  - Turn ON Output A to send the signal to the Oscilloscope.
- **Step 6:** Analyze signal in Oscilloscope
  - Adjust the Oscilloscope scale to display the modulation signal. Figure 39 shows the modulation signal is a 1.002 Hz sine wave. The minimum, maximum, and mean interval values in the Time & Frequency Analyzer show the pulse width deviation is 250 µs (± 250µs around the mean interval of 300 µs).

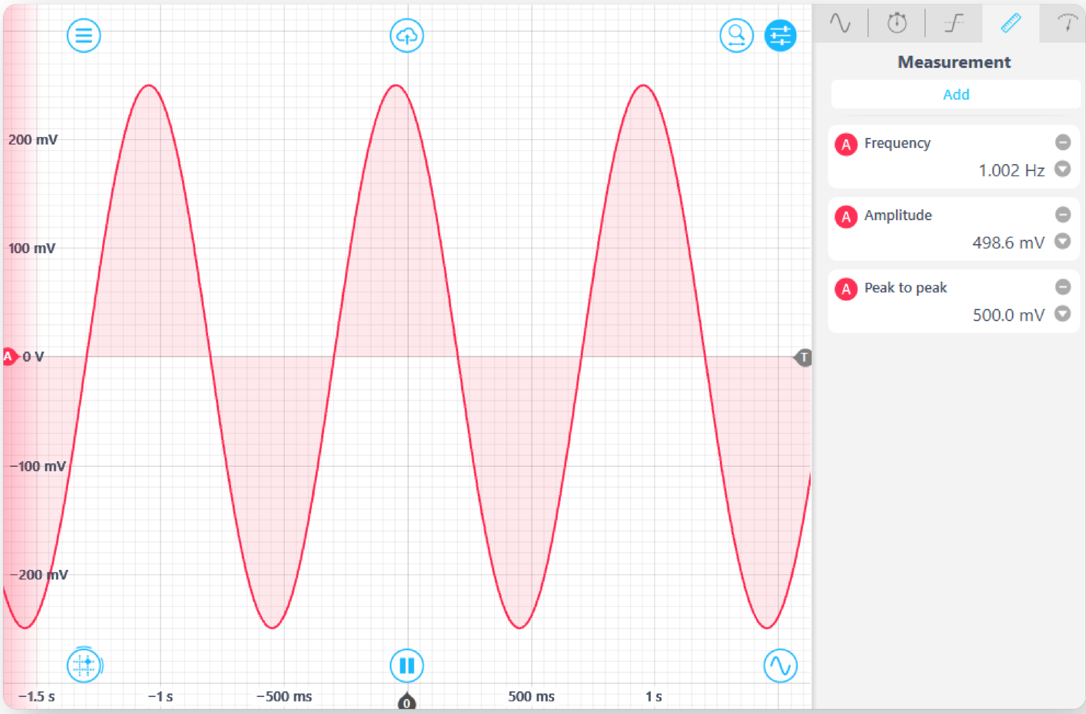


Figure 39. Oscilloscope graph settings to view the decoded PWM signal; a 1 Hz sine wave.

Additional controls

Main menu

The main menu can be accessed by clicking the icon on the top-left corner.

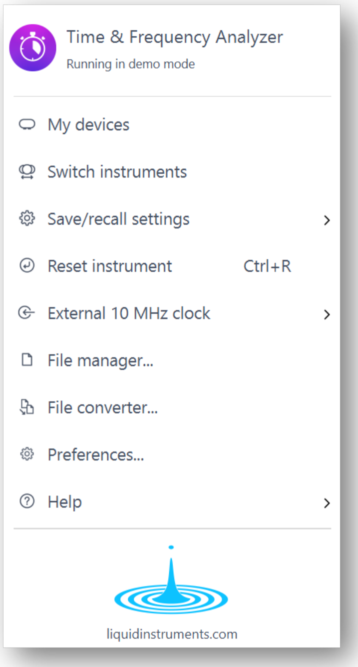


Figure 40. Main menu options for the Time & Frequency Analyzer.

AI Help... Opens a window to chat to an AI trained to provide Moku-specific help (Ctrl/Cmd+F1)
My Devices returns to device selection screen
Switch instrument to another instrument
Save/recall settings

- Save current instrument state (Ctrl/Cmd+S)
- Load last saved instrument state (Ctrl/Cmd+O)
- Show the current instrument settings, with the option to export the settings

Reset instrument to its default state (Ctrl/Cmd+R)
Sync Instrument slots in Multi-Instrument Mode*
External 10 MHz clock selection determines whether the internal 10 MHz clock is used.
Clock blending configuration opens the clock blending configuration pop-up *
Power Supply access panel*
File Manager access tool
File Converter access tool
Preferences access tool
* If available using the current settings or device.

Help

- **Liquid Instruments website** opens in default browser
- **Shortcuts list** (Crtl/Cmd+H)
- **Manual** Open the user manual in your default browser (F1)
- **Report an issue** to the Liquid Instruments team
- **Privacy Policy** opens in default browser
- **Export diagnostics** exports a diagnostics file you can send to the Liquid Instruments team for support
- **About** Show app version, check for updates or licence information

File converter

The File converter can be accessed from the main menu .

The File converter converts a Moku binary (.li) format on the local computer to either .csv, .mat, .hdf5 or .npy format. The converted file is saved in the same folder as the original file.

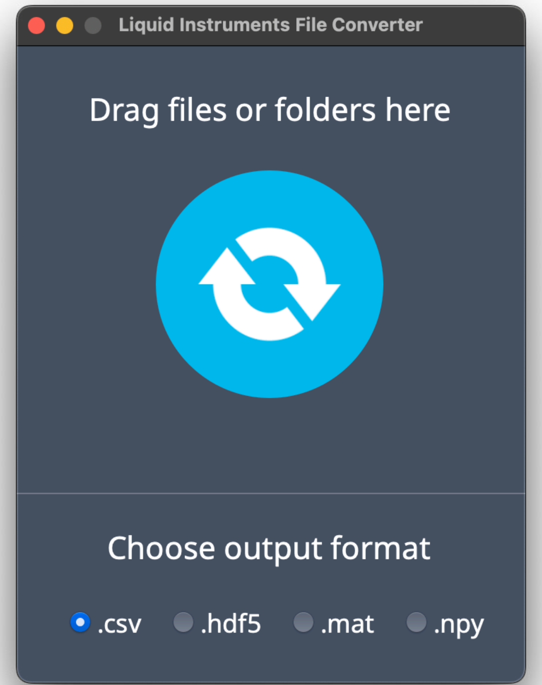


Figure 41. File Converter user interface.

To convert a file:

1. Select a file type.
2. Open a file (Ctrl/Cmd+O) or folder (Ctrl/Cmd+Shift+O) or drag and drop into the File converter to convert the file.

Preferences and settings

The preferences panel can be accessed via the Main Menu ☰. In here, you can reassign the color representations for each channel, switch between light and dark mode, etc. Throughout the manual, the default colors are used to present instrument features.

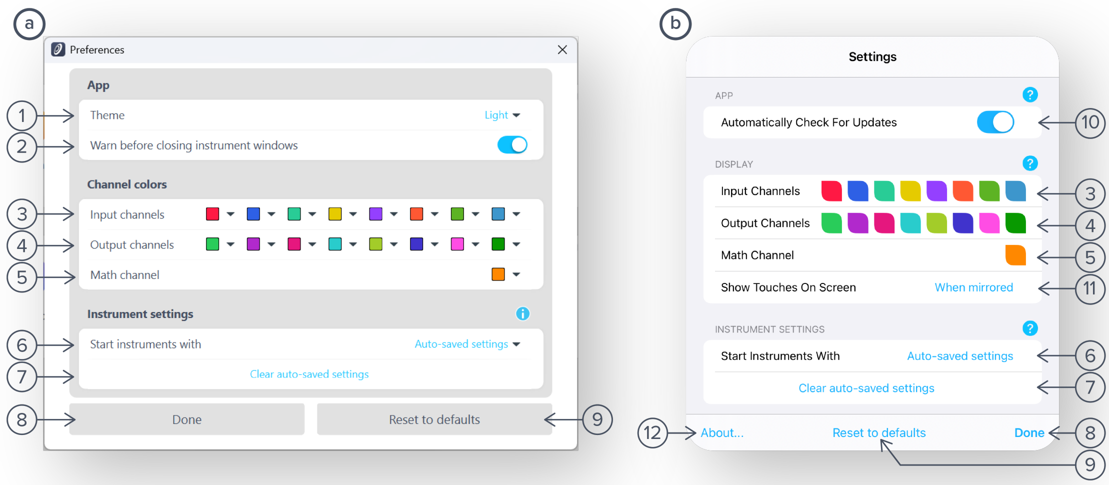


Figure 42. Preferences and settings for the Desktop (a) and for the iPad (b) App.

① Change the App theme, between dark and light mode.
② Choose if a warning opens before closing any instrument windows.
③ Tap to change the color associated with the input channels.
④ Tap to change the color associated with the output channels.
⑤ Tap to change the color associated with the math channel.
⑥ Select if instruments open with the last used settings, or default values each time.
⑦ Clear all auto-saved settings and reset them to their defaults.
⑧ Save and apply settings.
⑨ Reset all application preferences to their default state.
⑩ Notify when a new version of the app is available. Your device must be connected to the internet to check for updates.
⑪ Indicate touch points on the screen with circles. This can be useful for demonstrations.
⑫ Open information about the installed Moku application and license.

Shortcuts

Shortcuts are integrated to speed up your workflow on Moku. This includes graph zoom and scrolling, autoscaling, cursor, measurement, instrument, and general shortcuts. They are available from the main menu, as shown in Figure 43, or use the shortcut `Ctrl + H` to open the shortcuts dialog.

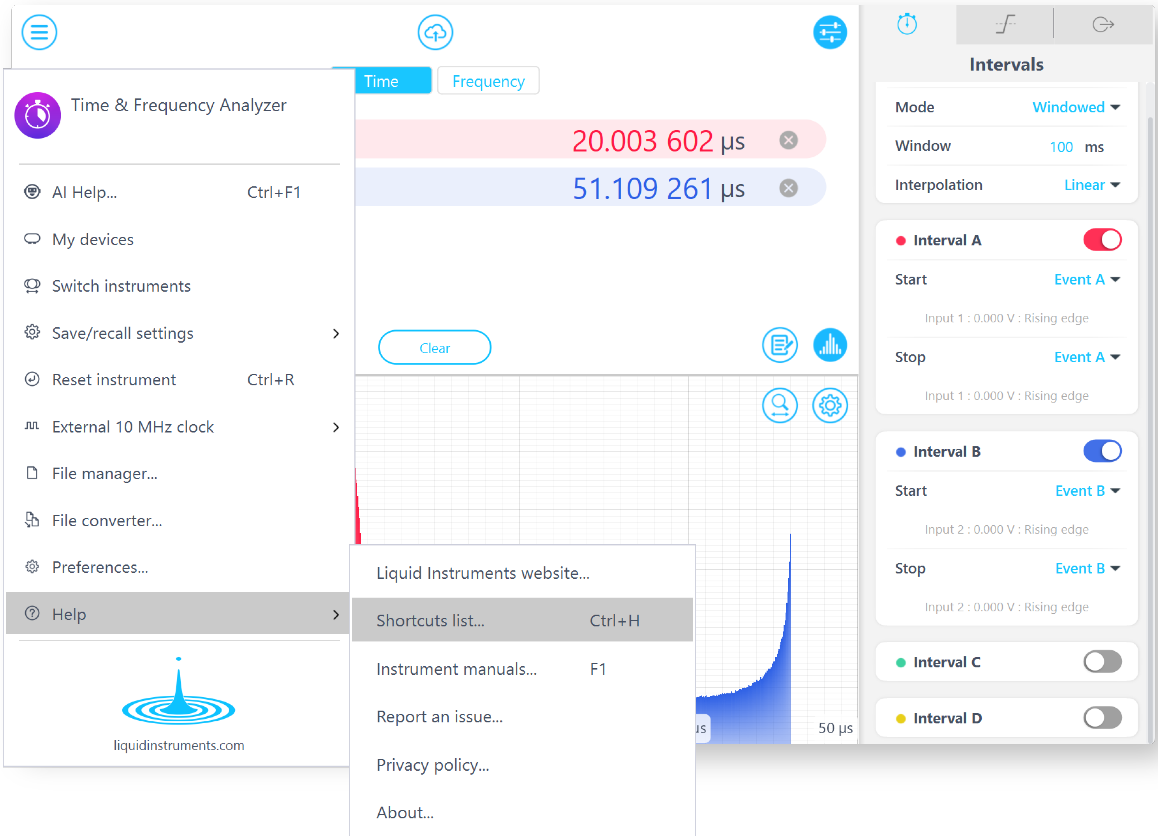


Figure 43. Main menu to access the help > shortcuts list.

External reference clock

Your Moku may support the use of an external reference clock, which allows Moku to synchronize with multiple Moku devices, other lab equipment, lock to a more stable timing reference, or integrate with laboratory standards. The reference clock input and output are on the rear panel of the device. Each external reference option is hardware dependent, review the available external reference options for your Moku.

Reference Input: Accepts a clock signal from an external source, such as another Moku, a laboratory frequency standard, or an atomic reference (for example, a rubidium clock or a GPS-disciplined oscillator).

Reference Output: Supplies the Moku internal reference clock to other equipment that require synchronization.

If your signal is lost, or is out of frequency, your Moku will revert to using its own internal clock until the reference signal returns. If this occurs, check the source is enabled, and that the correct impedance, amplitude, tolerance, frequency, and modulation are attached to the reference. Check the required specifications in the device specsheets.

When the reference returns within range, status changes to "validating" and then "valid" once lock is re-established.

10 MHz external reference

To use the 10 MHz external reference function, ensure "always use internal" is disabled in the Moku application, found in the main menu under "External 10 MHz clock". Then, when an external signal is applied to your Moku reference input and your Moku has locked to it, a pop up will show in the app. On some devices, the external reference information will be shown in the LED status as well, more information can be found in your Moku Quick Start Guide.

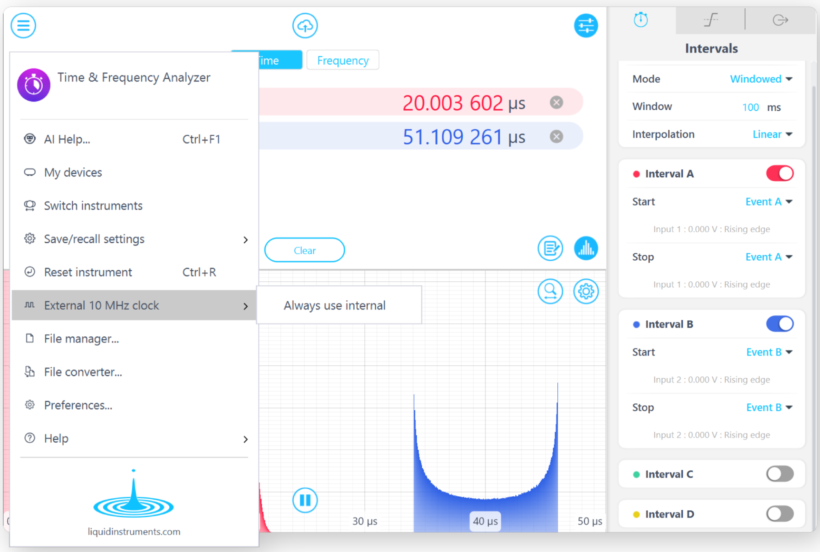


Figure 44. Moku main menu with "Always use internal" reference disabled and using an external reference.

Clock blending configuration

If available, Moku blends up to four clock sources simultaneously for more accurate phase, frequency, and interval measurements across all time scales. A low phase-noise Voltage-Controlled Crystal Oscillator (VCXO) is blended with a 1 ppb Oven-Controlled Crystal Oscillator (OCXO) for optimal wide-band phase noise and stability, which can be blended further with an external frequency reference and GPS disciplining to synchronize Moku with your lab and UTC.

The VCXO and OCXO will always be used for the clock generation signal. The external and 1 pps references are optional and can be enabled or disabled in the "Clock blending configuration..." settings from the main menu ☰. The loop bands are adjusted based on the different possible clock source configurations, shown in Figure 45, where the frequencies of the bands represent where each oscillator's phase noise dominates.

Read how the clock blending works on Moku:Delta for more details.

Figure 45. Moku clock blending configuration dialog with an external 10 MHz frequency reference enabled.

① **VCXO jitter reference** is always used for clock generation, handling high frequency jitter with the lowest noise.

② **OCXO jitter reference** is always used for clock generation, ensuring moderate term stability.

③ **External 10/100 MHz frequency reference** uses a "10 MHz" or "100 MHz" external reference to correct drift in the local oscillator, noting your Moku will have to be restarted after each change between a 10 MHz and 100 MHz source.

④ **1 pps synchronization reference** uses an "External" or "GNSS" reference to sync with UTC and correct drift in the local oscillator. The estimated clock stability is a measure of how much the reference performance deviates relative to the local OCXO/VCXO timebase (as currently blended and, if enabled, steered by the external 10 / 100 MHz External reference).

Troubleshooting

1. **No events or intervals are being registered.** Check the configuration of the event detectors to ensure your signal crosses the detector threshold, see Events and Acquisition.
2. **Data logging is cut short.** The event rate may be too high to save all the events. Review the specs for your device to ensure the event rate does not exceed the maximum achievable. Data saved up until logging ceased is accurate. Logging may also stop if device memory becomes full.
3. **Histogram is skewed or non-Gaussian.** The signal edge may be rising too quickly for the sample rate of the Moku to accurately capture the events. Check the signal rise time and ensure it is within the sampling capabilities of the event detecter. See Event settings and your Moku specifications.

If your issue is not listed above please visit our knowledge base for further information and support.

Tips and tricks

These tips and tricks are designed to help you upgrade your workflow to get the most out of your Moku.

- Scroll and pan the graph interface to move and view your waveform how you would like.
- Double click any axis to quickly center the signal, this will move the axis zero tick in the middle of the graph view.
- Share and export your data with ease from the Export button or the File Manager in the Data Logger. From the app, copy screenshots to clipboard, save to your local storage, or share directly to your team through email, airdrop and more.
- Adjust your parameters whichever way is preferable; click, drag, or scroll, arrows, numeric input to adjust parameters for quick and easy adjustments.
- Devices with an external input can be synced to other instruments and devices with the external 10 MHz signal found at the back of the Moku.
- The Moku application can be converted from light mode to dark mode in the setting preferences from the settings cog on the "Select your Device screen"