# Active Learning Interface (for) Circuits (and) Electronics:

## Objective:

This document serves as a User’s Guide for the Universal ALICE software interface. The basic idea is to have one hardware agnostic user interface that can be connected to a set of hardware specific functions to communicate with various target data acquisition hardware platforms. Python was chosen as the development language because it is open source and the most cross computer / OS platform compatible.

## Background:

The main Universal ALICE GUI is written in Python (3.x) and depends only on the more or less generic standardly available Python plugin libraries (such as Tcl/Tk, numpy, pyplot etc.). The hardware specific interface level would generally also be implemented in Python plus any plugin libraries (such as pyserial, pyaudio) with possible C extensions depending on the target hardware. Under the hardware driver level are any specific host USB drivers for the various operating systems / host platforms and of course the firmware that runs on the target data acquisition hardware board (micro-controller). Some hardware specific tool chains or drivers might need to be obtained directly from the hardware vendors and not necessarily hosted on this archive.

### What’s in a name?

Although the word ALICE can be spelled out from the title of this users guide, it is actually an allusion to the fantasy works of Lewis Carroll: 1865's Alice's Adventures in Wonderland and its 1871 sequel Through the Looking-Glass, and What Alice Found There. In these stories Alice explores a strange and wondrous world down a rabbit hole and on the other side of a mirror (looking glass). Hopefully, through the use of this software along with various data acquisition hardware boards / instruments, Students can explore the strange and wondrous world of Circuits, Electronics and Electrical Engineering.

## Functions:

The Universal ALICE Desktop software package provides the following functions:

* Up to Four Channel Oscilloscope for time domain display and analysis of voltage (current) waveforms depending on target hardware.
* Up to Two Channel Arbitrary Waveform Generator (AWG) controls.
* X-Y display for plotting captured Oscilloscope channel data.
* Phase Analyzer, Polar plotting of up to four voltage waveform amplitude and phase
* Two Channel Spectrum Analyzer for frequency domain display and analysis of waveforms.
* Bode plotter and network analyzer with built-in sweep generator.
* Impedance Analyzer for analyzing complex RLC networks and as a RLC meter and Vector Voltmeter.
* DC Volt meter, (up to four inputs) digital and analog meter face display
* DC Ohm meter, measures unknown resistance with respect to known external resistor

## Required files:

The ALICE Desktop program is written in Python and runs from the source code. Generally requires version 3.10 or greater of Python be installed on the user’s computer. The program only imports modules generally included with standard Python installation packages depending on the target data acquisition hardware.

Run ALICE source code from the Python 3.10 compatible source code with the following packages installed:

Python 3.10 (or higher, 64 bit version recommended)

numpy numerical package extension

matplot / pyplot

### Linux and OSX:

Most releases of the Linux operating system have Python included and many also include the numpy numerical package as well. Linux (including Raspberry Pi) and OSX users must manually compile

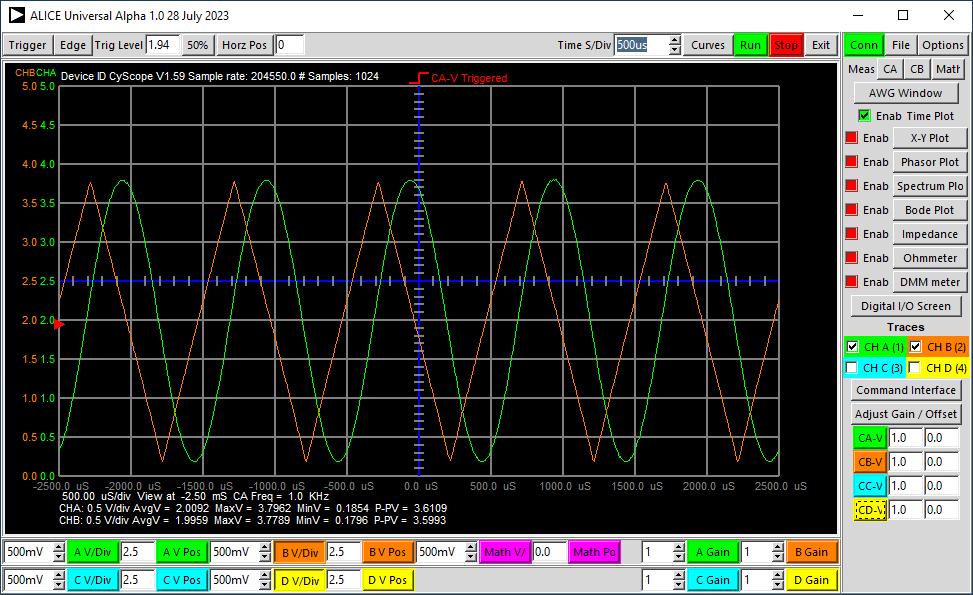
## Directions:

It is assumed that the reader is somewhat familiar with the functionality and capabilities of the targeted data acquisition hardware board being used.

First a few notes on nomenclature used in this document:

## Main Window:

Be sure that the data acquisition hardware board or instrument is plugged into a USB port before starting the program. Once the program is running the main window, as shown in figure 1, should appear. This is the main desktop window and serves as the Oscilloscope Tool Window as well as controls for opening the other display windows and certain common control functions. It is sub divided into 4 sections.



**File Drop Down**

**Options Drop Down**

**Open Digital PIO Window**

**Open / Sel Impedance**

**Open / Sel Spectrum**

**Open / Sel X-Y plot**

**Open AWG Ctrls**

**Curves Drop Down**

**Math Drop Down**

**Channels A, B, C, D Voltage and Math Range and Position controls**

**Trigger Drop Down**

**Time Axis Controls**

**Open / Sel Bode Plot**

**Enter Channel Gain and Offset values**

**Open / Sel Ohmmeter**

**Enable Time Display**

**Optional Function Buttons Will Appear Here**

**Open / Sel Phase Analyzer**

**Select Traces**

**Open Input Divider Calculator**

**Buttons to reset Gain and Offset values**

**Optional Input PGA controls if supported by hardware**

**Open / Sel Voltmeter**

Figure 1, Universal ALICE Desktop main window

The menu section along the top, shown in figure 2, contains various buttons and drop-down menus that control Triggering, Horizontal time base, Horizontal position, how and what signal traces are displayed, and run acquisition / stop acquisition / exit program.



Figure 2 Top menu buttons

The Trigger button is a drop down menu listing which signal to trigger on, None or CA-V, CB-V, CC-V or CD-V. The Edge button is a drop down menu listing either the rising or falling edge for triggering. The Trigger Level entry window contains the trigger level in volts for CA-V and CB-V. The 50% button sets the trigger level to the midpoint (50% point) of the selected trigger waveform. i.e. to the (maximum + minimum)/2.

The Time mS/Div spinbox entry window is used to set the horizontal time base in the standard 1, 2, 5 step increments. Other values maybe entered manually.

The Curves drop down menu button allows the selection of which signal waveform traces will be displayed vs time. The All button selects all four curves to be displayed and the None button clears all four curves. Plot the Math-X and Math-Y formulas by clicking on those options. ALICE can automatically adjust the trace vertical position to center the CA-V and /or CB-V traces on the midpoint of the waveform each sweep by clicking on either or both of the options below the –Auto Vert Center- heading.

When using external Resistor Divider attenuators the frequency response of the Channel may be reduced. The software High Pass frequency compensation can be selected for either channel by clicking on the buttons below the –Input HP Comp- heading. See the section on Analog Inputs below.

It is also possible to select which of the possible stored reference time traces, if saved via the Snap-Shot option, will be displayed.

At the bottom are options to enable the vertical Time and horizontal Voltage cursors.

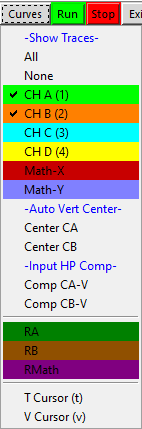


Figure 3 Curves Drop Down Menu

The green Run button starts continuous looping acquiring input samples. The red Stop button stops the acquisition looping. The Exit button exits (kills) the program.

### The Right Side Menu Section

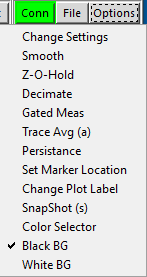
The Green Conn button in the top row indicates that the target hardware board is connected and ready to go. If the button is red and says Recon then a board was not found. Connect board and click on the button to connect to board.

The File drop down menu lists commands for saving and loading configuration settings (.cfg file), saving the graphics display area to an encapsulated postscript file (.eps), and saving the captured channel A, B, C, D voltage data to a coma separated values file (.csv). Depending on the specific capabilities (that is the size of the sample buffer) for most Time/Div settings the number of sample points is 2 screen widths with a minimum of 1024 samples and a maximum of 90,000. This saved table of raw sample values can then be loaded into other programs for analysis such a spreadsheet program or math program like MATLAB, or Octave.

On most operating systems there is a way to capture a bit map graphic of any of the display windows at any time. Some are built in or done through a support program or application. In Windows:

Press the <alt> and <print screen> keys to capture the currently selected window in the copy buffer (clip-board). Then start a program such as Word or Paint (any similar program). Use Paste to place the screen shot into your document or drawing etc. Then save that file to disk.

The Options drop down menu, figure 4, lists a command for enabling smoothing where spline curves are used to connect the input sample points rather than the default straight lines. A second option for connecting the sample points is to use a zero order hold function where a horizontal line and a vertical line are used. This looks like a stair step waveform much like the output of the Digital-to-Analog converters used to generate the AWG output signals actually produce.



**Draw Traces with Spline curves**

**Draw Traces with Stair Steps**

**Turn on Trace Averaging**

**Turn on Decimation**

**Turn on Gated Measurements**

**Save Trace Snap Shot**

**Use Black Background**

**Use White Background**

**Turn on Trace Persistence**

**Open Color Selector**

**Open Settings Controls**

**Set Location for Marker Text**

**Enter New Plot Label Text**

Figure 4, Options Drop Down Menu

The Trace Avg button turns on trace averaging. The number of sweeps to average can be set in the Change Settings Window. The width of the traces in pixels can be also be set using the Change Settings Window.

The currently displayed traces will be saved via the Snap-Shot option as reference traces. They can be added to the graphics plot area by selecting the desired trace from the Curves drop down menu for time plots. They will be drawn in a darker color corresponding to the matching live waveform trace.

The Graphics display area can be drawn with either a Black (default) or White background. Use these two buttons to select which is used.

The CA and CB measure drop down menus, figure 5, list which vertical measurements for the Channel A and B voltage and current signals are to be displayed along the bottom of the graphics display area.

**Vertical Voltage Measurements**

**Horizontal Time Measurements**

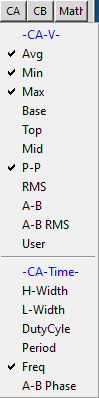


Figure 5, Measurements Drop Down Menu ( CA )

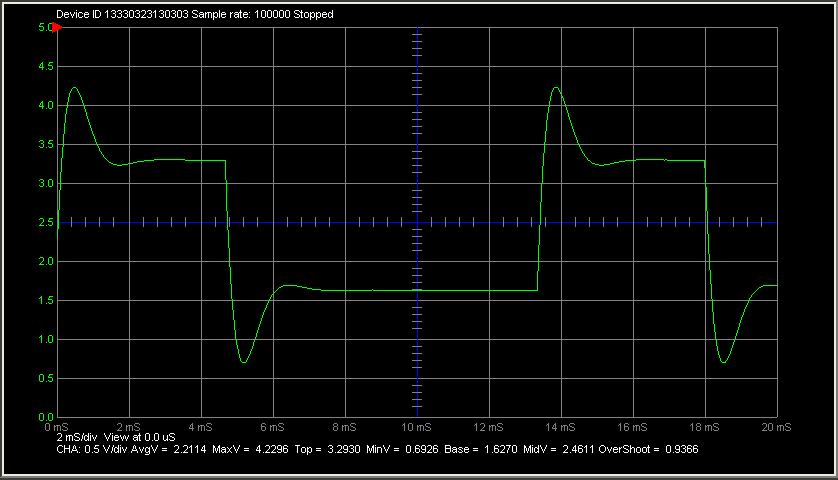
The displayed vertical measurements can be the following:

* Average, which is the sample by sample sum of the data record divided by the number of samples. For most Time/Div settings the number of samples is 2 screen widths.
* Minimum, which is the minimum value within the data record.
* Maximum, which is the maximum value within the data record.
* Base, used mainly for square waves it is the voltage level of the lower flat portion of the wave which may be different from the Min value due to undershoot.
* Top, used mainly for square waves it is the voltage level of the upper flat portion of the wave which may be different from the Max value due to overshoot.
* Midpoint, which is the maximum value plus the minimum value divided by two.
* Peak-to-Peak, which is the maximum value minus the minimum value.
* RMS, which is the square root of the sum of the sample by sample data record squared divided by the number of samples.
* CA-CB and CB-CA differences of the Average ( DC ) voltage values of the channels.
* Display User defined measurement.

The displayed horizontal measurements for the voltage traces can be the following:

* High pulse width ( time waveform is above the mid-value )
* Low pulse width ( time waveform is below the mid-value )
* Duty Cycle ( percent of time waveform is High )
* Period ( time between 2 rising edges where waveform crosses mid-value )
* Frequency ( 1 / period )

Figure 6 shows examples of many of the possible waveform measurements. Six of the vertical measurements are derived directly from the waveform data array. These are Avg, Min, Max, Top, Base and RMS. The rest are calculated from these six. P-P is obviously Max – Min. Mid is (Max + Min / 2). CA-CB is CA Avg – CB Avg.



**Max**

**Top**

**Base**

**Min**

**Avg**

**Mid**

**Low Pulse Width**

**High Pulse Width**

**P-P**

Figure 6, Measurement examples

The User measurement option allows the user to calculate any other measurements based off these constants. When clicked on the user is prompted for a label to be used while displaying the value and a formula for calculating the value. For example the overshoot can be calculated by the formula:

(MaxV1 –VATop)/(VATop-VABase)

Crest factor is the ratio of peak-to-RMS values. This is 1.414 for sinewave inputs (1/0.707), but can be as high as five or more for random noise.

Another example would be gain where channel A is considered the input and channel B is the output. The gain would be the ratio of the two P-P values:

(MaxV2-MinV2)/(MaxV1-MinV1)

Waveform calculated Vertical measurement constants:

DCV1 is the channel A Average voltage

MinV1 is the channel A Minimum voltage

MaxV1 is the channel A Maximum voltage

VATop is the channel A Top voltage

VABase is the channel A Base voltage

SV1 is the channel A RMS voltage

DCV2 is the channel B Average voltage

MinV2 is the channel B Minimum voltage

MaxV2 is the channel B Maximum voltage

VBTop is the channel B Top voltage

VBBase is the channel B Base voltage

SV2 is the channel B RMS voltage

DCV3 is the channel C Average voltage

Minv3 is the channel C Minimum voltage

Maxv3 is the channel C Maximum voltage

SV3 is the channel C RMS voltage

DCV4 is the channel D Average voltage

Minv4 is the channel D Minimum voltage

Maxv4 is the channel D Maximum voltage

SV4 is the channel D RMS voltage

Waveform calculated Horizontal measurement constants:

CHAHW is the channel A High Pulse Width

CHALW is the channel A Low Pulse Width

CHADCy is the channel A Duty Cycle

CHAperiod is the channel A Period

CHAfreq is the channel A Frequency

CHABphase is the channel A to channel B relative phase angle

CHBHW is the channel B High Pulse Width

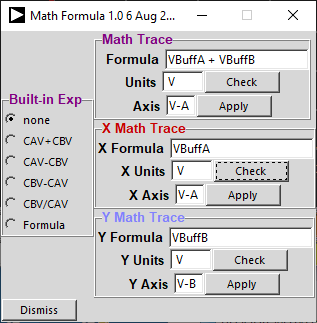
CHBLW is the channel B Low Pulse Width

CHBDCy is the channel B Duty Cycle

CHBperiod is the channel B Period

CHBfreq is the channel B Frequency

The Math menu button, figure 7, opens a control screen that lists which sample point by sample point calculated waveform combining the Channel A and B voltage signals is to be displayed vs time.



**Enter Formula for Math Waveform**

**Show Math Formula Waveform**

**Math Waveform Selector**

**Enter Formula for X Math Waveform**

**Enter Formula for Y Math Waveform**

Figure 7, Math Drop Down Menu

One of the following built-in calculated waveforms can be displayed at a time:

* CA-V + CB-V, the sum of the channel A and B voltage waveforms
* CA-V – CB-V, the difference of the channel A and B voltage waveforms
* CB-V – CA-V, the difference of the channel B and A voltage waveforms
* CB-V / CA-V, the ratio of the channel B voltage and channels A voltage waveforms which is instantaneous voltage gain assuming CA-V is input and CB-V is output

The first three calculations result in voltages and share the corresponding left side voltage scale on the display grid. These calculated waveforms can produce strange looking results for periodic waveforms driving non-resistive loads such as capacitors or inductors. The ratio calculation can be used to calculate voltage gain and is dimensionless.

If Formula is selected then the mathematical formula entered in the top formula, will be plotted vs time. This allows greater flexibility in waveform plotting at the expense of typing in the function to be plotted. See section on Advanced Math Traces below on how to enter formulas. Any one of the four channel vertical axis controls can be chosen for the Formula axis using the Math Axis entry. Generally when plotting using Formula, one or the other of the four channels are not being displayed and its axis controls will be available to be used. Two more math formulas, X Math Trace and Y Math Trace can also be entered / edited through these controls.

The AWG control Window is opened by default when the program is started. Since all of the displays use the AWGs in some fashion, it is important that this window be available to all. If you dismiss (minimize to the tool bar) the AWG control window, clicking on the AWG Window button will bring back the window.

The X-Y Plots Window button opens the X vs Y display window.

The Phasor Plot button opens the phase analyzer window.

The Spectrum Window button opens the Spectrum Analyzer display window.

The Bode Plot Window button opens the Bode Plot display window.

The Impedance Window button opens the Impedance Analyzer display window.

The Ohmmeter button opens the DC Ohm Meter window.

The DMM meter button opens the DC Volt Meter window.

To update the display window for a particular tool (when running) the matching Time Plot, X-Y Plot, Freq Plot, Bode Plot and/or Impedance Plot enable check boxes must be selected. More than one display can be selected at a time but some combinations such as X-Y and Spectrum or X-Y and Impedance would not make much sense while Time and X-Y or Time and Spectrum might.

### Input Divider Calculator

A simple calculator window has been included to make calculating an input resistor divider's Gain and Offset values easy based on the resistor values used and offset connections. The button directly above the Gain and Offset entries will open the calculator.

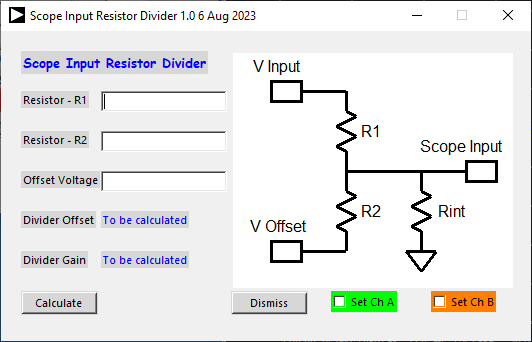
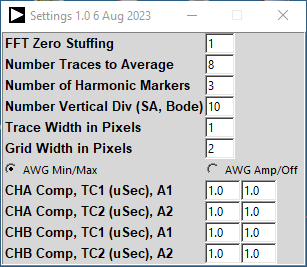
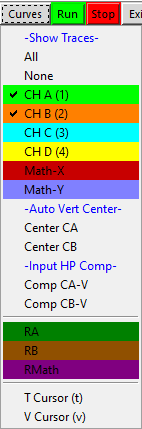


Figure 8, Input Divider Calculator.

Values for divider resistor R1 and resistor R2 are entered as well as any offset voltage that is applied to the bottom of the divider. The Exact values, as measured with a bench DMM, can be entered for R1 and R2 to calculate more accurate gain and offset results. The Rint internal resistance of the channels (which might be1 Meg Ω but can be set in the hardware specific configuration file) is taken into account in the calculation as this will have a significant effect for higher values of R1 and R2. Click the Calculate button to calculate the values. The Channel A or B entries can then be set to the calculated values using the Set CH A and Set CH B buttons respectively. These values can then of course be tweaked as needed for even better accuracy. Entries for any third and fourth channels can be entered manually from the displayed calculated values.



**Frequency compensation check boxes**

**Frequency compensation time constant and gain settings**

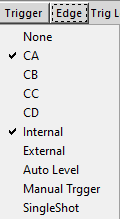
Figure, Entering input divider frequency compensation values.

At the bottom of this section are entry windows which allow input gain and offset correction for any external resistor divider attenuator networks that might be added to the channel inputs. Save and Load Cal buttons can be found under the File drop down menus. For more on the use of input attenuators please refer to the following two documents:

### The Top Menu Section

The menu section along the top contains various buttons and drop-down menus that control Oscilloscope Triggering, Horizontal time base, Horizontal position, how and what signals are displayed, and run acquisition looping / stop acquisition looping / exit program.

The Trigger button is a drop down menu listing which signal to trigger on, CA, CB, CC, CD or none depending on the number of hardware channels. The Auto Level option automatically set the trigger level to the selected waveform midpoint on each sweep. The trigger level will thus track any changes in the input waveform. The Single shot option allows a single sweep to be captured each time the Run button is clicked. The kind of triggering used is hardware dependent. Some hardware might support hardware triggering which could be either internal or external and some might use software triggering. In the case of software triggering the internal / external selection is meaningless.



**Enable Trigger Filtering**

**Manual Triggering**

**Set Auto Trigger Level mode**

**Signal Selectors**

**Single Shot Triggering**

**Select internal or external triggering if supported**

Figure 10, Triggering drop down menu.

The Edge button is a drop down menu listing either the rising or falling edge for triggering. The Trigger Level entry window contains the trigger level in volts. The 50% button sets the trigger level to the midpoint (50% point) of the selected trigger waveform. i.e. to the (maximum + minimum)/2.

The Hold Off entry window, in mS, is used to shift the horizontal position (apparent time 0 start point) within the acquired sample point buffers being displayed. The data used for the vertical and horizontal waveform calculations is also shifted by that amount. Due to the discontinuous nature of the AWG outputs this allows the user to skip over any initial transients that might appear if the system being measured has “inertia” or “state” that needs to settle out.

The Horz Pos entry window is used to change the horizontal position of the time trace. Normally, with the Horz Pos set to 0 the left edge of the grid is “time 0”. Setting Horz Pos to something else shifts the 0 time point on the grid by that amount (in mSec).

The Time mS/Div spinbox entry window is used to set the horizontal time base in the standard 1, 2, 5 step increments. Other values maybe entered manually.

The Curves button allows the selection of which signal waveforms will be displayed when plotting vs time. The All button selects all four curves to be displayed and the None button clears all four curves. It is also possible to select which of the possible stored reference time traces, if saved via the Snap-Shot option, will be displayed.

It is good practice to turn off or disconnect any power supplies when making any modifications to the circuit under test.

The green Run button starts continuous looping acquiring input samples. The red Stop button stops or pauses the acquisition looping. The keyboard space bar toggles between Run and Stop. The Stop button also serves as a sort of refresh button. If the Stop button is clicked when stopped the graphics display is redrawn using any new settings that might have changed but using the existing data buffers. The Exit button exits (kills) the program.

### The Bottom Menu Section

The menu section along the bottom contains the range (V/Div) and position controls for however channels the target hardware supports (up to four). The entry labels are color coded to match the waveform trace colors.

The channel V/Div spinboxs set the corresponding vertical ranges in the standard 1, 2, 5 step increments. Other values maybe entered manually. The position entry windows determine the vertical position of their scales with respect to the blue center line on the grid. That is to say the value entered corresponds to the number displayed next to the blue center line. The software allows support for hardware channels the might include programmable gain (PGA) functionality. The PGA controls on the right will not appear if not enabled in the hardware level specific configuration file.

The default arrangement of the entry widgets and the buttons that serve as labels for the entries is for the labels to be to the right as shown in figure 11. In the alice\_init.ini file an optional flag can be set to swap this arrangement with the labels to the left as shown in figure 12.



Figure 11, Default arrangement for Channel control buttons; ButtonOrder = 0



Figure 12, Alternate arrangement for Channel control buttons; ButtonOrder = 1

### The Graphics Display Area

The graphics display area, show in figure 13, is where the various signal waveforms are plotted on either a black or white background depending on which is selected under the Options drop down. It consists of a main 10 by 10 grid with the center vertical and horizontal grid lines drawn in dark blue. Each major grid is sub divided into 5 sub grids by the short tick marks along the blue center lines. The horizontal grid lines are labeled with color coded text to match the corresponding waveform trace with the voltage scales on the left for the first two input channels and the third and fourth input channels (if supported) scales on the right along with the Math trace if selected.

The red triangle, drawn on the left side in the example shown because the trigger input is set to CA-V indicates the trigger level.

Above the main grid area is a line of text showing the device ID and Sample rate and if the acquisition loop is running or stopped. Below the main grid are three lines of text which will display various information about the displayed plots. The first line shows the current time per division setting and the horizontal position of the left most grid line with respect to 0 time i.e. the trigger point.

The second and third lines of text are for displaying information related to Channel A and Channel B respectively. The selected V/Div is displayed along with any of the vertical measurements selected for that channel.

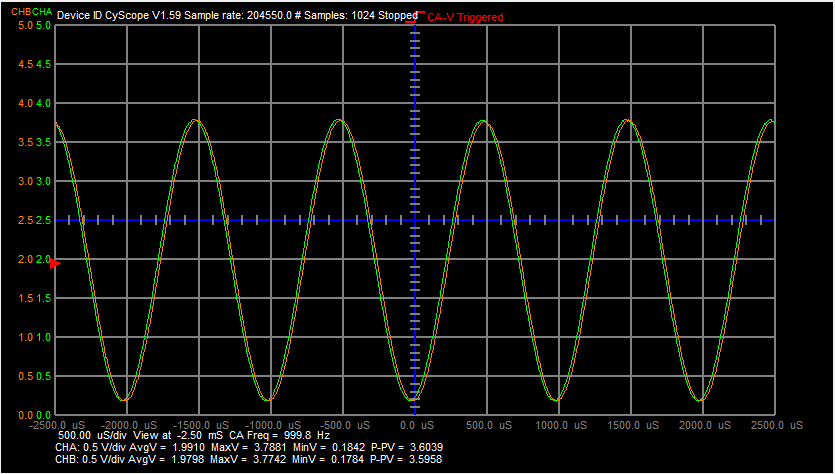


Figure 13 Graphics display area

### Grid Markers and Cursors

While stopped (red Stop button clicked) if you left click anywhere within the display grid a numbered marker “x” point will appear at that position. In the upper left corner of the display grid the maker number along with the vertical (voltage) and horizontal (time) values will also appear. For marker points > 1 the vertical and horizontal delta to the previous point will also be displayed. Clicking the red Stop button again will clear the markers. Clicking on the green CA V/Div or orange CB V/Div buttons will select which vertical range / position axis will be used and the marker will be drawn in that color.

Under the Curves Drop down menu there are selectors for displaying the T cursor (time) and V cursor (voltage). When selected, if you right click anywhere within the display grid either a vertical or horizontal cursor line, or both, will be drawn at that location. The vertical, horizontal, or both values for that point will be displayed. Scrolling with the mouse wheel will move the vertical line left–right when only the T cursor is selected and the horizontal line up-down when only the V cursor is selected. When both are selected the mouse wheel moves the vertical line left–right.

### Advanced Math Traces

In addition to the pre-programed Math traces, Universal ALICE provides a method of plotting user defined equations or formulas using the channel waveform buffers. The formulas are written in conventional Python syntax which is basically the same as you would expect to write any math expression. Any of the Python math (and numpy) module functions can be used such as math.sqrt() or math.sin() etc. Any of the ALICE global variables can be used but below is a list of the most useful available variables and constants:

Waveform Buffers:

VBuffA is the Channel A voltage sample array (in volts)

VBuffB is the Channel B voltage sample array (in volts)

VBuffC is the Channel C voltage sample array (in volts)

VBuffD is the Channel B voltage sample array (in volts)

VmemoryA is the Channel A voltage memory array used for Trace Averaging

VmemoryB is the Channel B voltage memory array used for Trace Averaging

VmemoryC is the Channel A current memory array used for Trace Averaging

VmemoryD is the Channel B current memory array used for Trace Averaging

SAMPLErate is the sampling rate

Vertical Position variables:

CHAOffset is the value in the channel A voltage position entry window

CHBOffset is the value in the channel B voltage position entry window

CHCOffset is the value in the channel C voltage position entry window

CHDOffset is the value in the channel D voltage position entry window

As a simple example, to plot the difference between the channel A and B voltage waveforms the following formula would be used:

(VBuffA - VBuffB - CHAOffset)

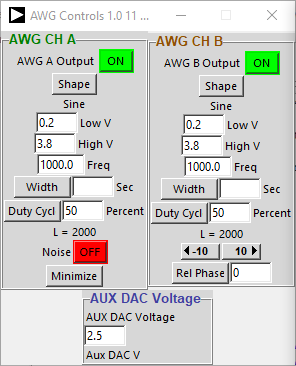
As the program iterates over the array (time) index, the channel B (voltage) value is subtracted from the channel A (voltage) value and then offset on the screen by the channel A position variable. This replicates the built-in math trace CA-V – CB-V.

Figure 14, Calculating the slew rate

A few words of caution, care must be taken when writing the formula to not cause a Python syntax error or other math exception such as divide by zero. If you make a mistake ALICE will stop and put up the math formula entry window so you can find and correct your mistake.

## AWG Controls Window:

The AWG controls window is shown in figure 15.



**Waveform Minimum value**

**Waveform Maximum value**

**Waveform Shape Menu**

**Waveform Frequency value**

**AWG B Waveform phase or delay shift**

**Square Wave Duty-cycle value**

**Length of Waveform Array is Shown Here**

**Enable / Disable AWG Outputs**

**Set Channel A as Noise**

**If hardware supports a variable DC Aux Output**

**Aux DAC output voltage**

Figure 15, AWG Controls window

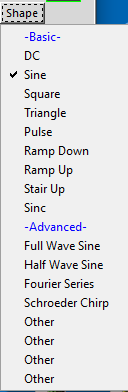
There are two identical sets of controls for configuring the Channel A and B outputs. Depending on the capabilities of the target hardware different features and controls may not be functional.

The Min and Max entry windows program the minimum and maximum values for the output waveform. If the value entered in the Min window is higher (more positive) than the value entered in the Max window the apparent phase of the output wave is inverted. While this is somewhat redundant for the Sine, Triangle and Square wave shapes, given the Phase control described later, it is useful for determining if the Sawtooth or Stairstep shapes are rising or falling ramps.

The Freq entry window programs the frequency of the waveform in Hertz. Based on the specific hardware maximum sample rate, the maximum possible frequency is, by definition, ½ the AWG sample rate but the practical upper limit is more like 1/5 or even 1/10 the sample rate.

The relative timing between the two AWG channels can be set. The left and right +/- 10 arrow buttons shift waveform B with respect to waveform A between the two. The entry window programs smaller or larger shifts. The Duty Cycle percent entry window applies to the Square shape and programs the duty cycle in percent from 0% to 100%. For the Triangle wave shape the percent entry window applies to the percentage of the period for the rising slope vs the falling slope of the triangle.

The Shape drop down menu is used to select the shape of the output waveform. The number of waveforms supported is hardware dependent. When DC is selected the constant value of the output voltage is set by the value in the Max entry window.



**Basic Shapes**

**Advanced Shapes**

**Full Wave Rectified Sine**

**Sine**

**Build waveform using Fourier Series**

**Square**

**Multi-Tone Chirp**

**Sin X/X waveform**

**Triangle Waveform**

**Number of Available Shapes is hardware dependent**

**Down Ramp Waveform**

**Half Wave Rectified Sine**

**Up Ramp Waveform**

Figure 16, AWG Shapes drop down menu

The Fourier Series shape builds a waveform based on the Fourier series of cosines for a square wave. The number of odd harmonics of the fundamental, is entered when prompted by the program. The minimum and maximum values of the fundamental are set using the Min and Max entries and the fundamental frequency is set using the Freq entry. Entering 1 for the number of harmonics will result in just the cosine wave at the fundamental frequency. Entering 3 for the number of harmonics will include the third harmonic, entering 5 for the number of harmonics will include the third and fifth harmonics and so forth. More information on this can be found in the Advanced Users Guide.

## DC Example:

To demonstrate how to use the Oscilloscope Tool as a DC voltmeter consider the resistor voltage divider network, shown in figure E1. We wish to measure the voltages at the 4 nodes and the voltages across the 6 resistors. In the figure the nodes are numbered from N0 to N4 with N0 being the ground or common node that all the voltage measurements will be made with respect to. With the Oscilloscope Tool we can measure two node voltages at a time and the voltage difference between those two nodes. From the Meas CA menu select from the –CA-V- section the Avg and CA-CB check boxes. Likewise from the Meas CB menu select from the –CB-V- section the Avg and CB-CA check boxes.

R2

10KΩ

GND

+5V

CB-In

CA-In

R1

470Ω

R3

2.2KΩ

R4

1KΩ

R5

470Ω

R6

4.7KΩ

**N0**

**N1**

**N2**

**N3**

**N4**

Figure E1, Test resistor network, measuring nodes N1 and N2

We start with the network powered from a fixed +5 volt power supply at node N1 and the channel A input also connected to N1. The channel B input is connected to node N2. Click on the Run button and the N1, N2 node voltages will be displayed along with the difference between them as CA-CB and CB-CA.

We can now proceed around the network measuring pairs of nodes until we can fill out table 1 below. Figure E3 shows the voltmeter inputs connected to nodes N3 and N4. Any combination of two nodes can be measured and the voltage difference between the two nodes will be displayed.

R2

10KΩ

GND

+5V

CB-In

CA-In

R1

470Ω

R3

2.2KΩ

R4

1KΩ

R5

470Ω

R6

4.7KΩ

**N0**

**N1**

**N2**

**N3**

**N4**

Figure E2, Test resistor network, measuring nodes N3 and N4

|  |  |
| --- | --- |
| Node | Voltage |
| N0 | 0.00 |
| N1 | 4.931 |
| N2 | 3.958 |
| N3 | 1.770 |
| N4 | 0.779 |

Table 1 Node voltages

From the measured node voltages (and the difference voltages) we can get the voltages across the 6 resistors shown in table 2.

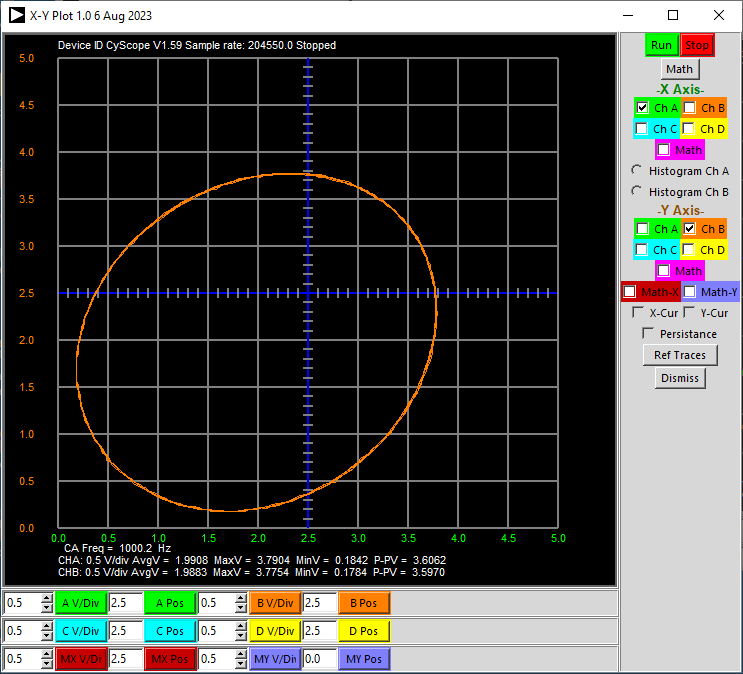
|  |  |
| --- | --- |
| Resistor | Voltage |
| R1 | N1 – N2 =0.972 |
| R2 | N2 – N0 =3.958 |
| R3 | N2 – N3 =2.188 |
| R4 | N3 – N4 =0.991 |
| R5 | N4 – N0 =0.779 |
| R6 | N2 – N4 =3.179 |

Table 2 Resistor voltages

From these voltages and the values of the resistors the currents through the resistors can be calculated.

## The X-Y Plotting Tool:

When the X-Y Plot Window button is clicked in the Main Window the X-Y display Window should appear, as shown in figure 19. It is sub divided into 3 sections.



**X Axis waveform selector**

**Y Axis waveform selector**

**Show Y Axis Cursor**

**Show X Axis Cursor**

**Show Reference Trace and Save Reference Snap Shot**

**Channels A, B, C, D Range and Position controls**

**Open Math controls**

**Close Window**

**Display as Histogram**

**Turn on waveform Persistence**

Figure 19, X-Y Plots window

The menu on the right allows selection of which of the four possible input channel waveform signals or Math formula is to be used for the X and Y axis. Given four possible signals, Channel A, B, C, D, there are in theory 16 possible combinations for X and Y. Not all 16 have been implemented since, for example, plotting a signal vs itself such as CA vs CA is a rather meaningless straight line.

Under the -X Axis- heading there are two options to display the histogram of either the channel A voltage or the channel B waveforms. The horizontal axis is in volts and controlled by either the CA or CB V/Div and V Pos controls. The vertical axis is the histogram count or number of hits at a given voltage level. The vertical axis scale is controlled by the CA or CB control.

It is also possible to select Math on one or the other or both axis. If Math is selected for just one axis then the selected trace from the Math drop down menu is used. Only a few of the built-in Math traces are supported. If Math is selected for both axis then the entered X formula and Y formula, using the Enter X or Y Formula buttons, will be plotted. This allows greater flexibility in X-Y plotting at the expense of the typing in the function to be plotted. See the earlier section on how to enter Advanced Math Traces for the Oscilloscope display. The same applies here to the X and Y formulas.

Any one of the four vertical axis controls can be chosen for the X and Y axis using the Math X or Y Axis buttons. Generally when X-Y plotting using Math one or the other of the four channels are not being displayed and its axis controls will be available to be used.

The X-Cur and Y-Cur check boxes select vertical and horizontal cursor lines which operate much the same as the T and V cursors in the Time display grid.

There is also a check button to display the saved X-Y reference trace (see note above in Oscilloscope section on Snap-Shot option).

## Oscilloscope and X-Y Plot Examples:

To demonstrate some of the features of the ALICE Oscilloscope and X-Y Plot Tools the following example circuit is offered. In figure E3 we see a simple NPN transistor (2N3904) in the common emitter configuration with a 100 KΩ resistor used to bias the base and a 1 KΩ resistor as the collector load. The collector load is supplied from the fixed +5 V power supply. We will use the ALICE software to plot IB vs VBE. We will also determine the value of CA-V corresponding to IC = 2 mA and then measure the input to output voltage gain around that operating point.

DUT

100K

IC

IB

VBE = CB

IB = (CA - VBE)/100K

GND

CB

CA

VCE = CB

1K

+5V

IB = (CA - CB)/100K

IC = (5V - VCE)/1K

IC = (5V - CB)/1K

Figure E3, NPN common emitter amplifier

To plot VBE and IB we first start out with the channel B input (CB-V) connected to the base of the transistor. As the formula in figure E3 states, IB can be calculated by taking the difference of CA-V and CB-V and dividing by the 100 KΩ resistor value. 100 KΩ is chosen to simplify the calculations so that the current is found by just moving the decimal point of the measured voltage (i.e. 1 V = 10 uA).

Set up the AWGs as follows: Channel A, Mode set to SVMI, Shape set to Triangle, Min set to 0.0, Max set to 5.0, Freq set to 100. Channel B Mode set to Hi-Z. Be sure that the Sync AWG box is selected.

The time base should be set to 0.5 mSec/Div so that the rising half cycle from 0 to 5.0 volts will fill the grid. Set the Hold Off to 10 mSec so the start of the second cycle will be displayed. Under the Curves menu select CA-V and CB-V. Under the Math menu select CAV - CBV.

Press the green Run button. You should see something like figure E4.

Figure E4, VBE and IB plot

The green CA-V trace is the 0 to 5 V ramp that is being applied to the 100 KΩ resistor. The orange CB-V trace is the voltage on the base of the transistor or VBE. The magenta CAV-CBV math trace is the voltage across the 100 KΩ resistor and represents IB as 10 uA/V.

Pause or Stop the program (red Stop button)

To make an XY plot of IB vs VBE open the X-Y Plot Window and check the X-Y Plot box. In the X-Y Display window press the CB-V button in the -X Axis- section and the Math button in the -Y Axis- section. In the X-Y Window set the CB V Pos entry to 0.5 and the CB V/Div to 0.1.

Press the green Run button. You should see something like figure E5.

Figure E5, IB vs VBE

The base current is very small when VBE is less than 0.6 V so there is likely to be noise in that part of the trace. Remember that the vertical voltage scale (0.5 V/Div) is divided by the 100 KΩ resistor so it is 5 uA per division.

To plot the collector current move the Channel B input to the collector of the transistor.

Now we need to go back to the time display window. Uncheck the X-Y Plot box and make sure the Time Plot box is checked. Under the Math menu select none for now.

Press the green Run button. You should see something like figure E6.

Figure E6, VCE plot

To turn the plot of VCE into IC we can use the gain and offset calibration equation for channel B to calculate the equation for IC in figure E3. The “calibration” equation is as follows:

V\_dis = (V\_raw – Offset )\*Gain

Where:

Vdis is the “calibrated” (scaled by some gain factor) value to be displayed

Vraw is the measured value

Offset is the “calibration” value entered

Gain is the “calibration” value entered

If we set the Offset equal to the actual value of the +5 V supply divided by the Gain calibration (1.0) factor and change the sign of the Gain factor (i.e. make it -1.0) we have the formula for IC from figure E3. After changing the channel B offset and gain factors press the green Run button and you should see something like figure E7.

Figure E7, IC plot

IC should be nearly zero where CA-V is less than 0.6 V. You may need to tweak the Offset factor to get IC to be exactly on the 0.0 grid line. An easy way to check this is to temporarily move the channel B input to the +5 V power supply. Now the difference between CB-V and the supply is exactly zero.

Remember that the vertical voltage scale (0.5 V/Div) is divided by the 1 KΩ resistor so it is 0.5 mA per division. With the program paused, under the Options menu press the SnapShot button. This saves a copy of the displayed CA-V and CB-V traces. Under the Curves menus select RB-V. This will now display the saved IC plot.

Now move the channel B input back to the base of the transistor. Under the Math menu select the CAV-CBV math trace. Reset the Channel B Offset and Gain calibration factors to their normal values.

Press the green Run button. You should see something like figure E8.

Figure E8, IC, IB and VBE plot

Now we have plots of IC (dark orange), IB (magenta) and VBE (orange) on the same grid as the base resistor bias CA-V is swept from 0 to 5 V (green).

Under the Curves Menu select the V cursor. Right click on the dark orange IC curve where it crosses the 1200uSec time grid. The voltage value at that point will appear next to the horizontal cursor line. The Use the mouse wheel it adjust the cursor up or down so it lines up exactly where the IC curve cross the time grid line. It should look like figure E9.

Figure E9, voltage cursor added.

The beta of the transistor can now be calculated by scrolling the cursor down till it lines up exactly where base current (magenta trace) at the same time grid line. The displayed voltage will represent the base current. Beta will be IC / IB. For this example IB is about 13 uA so beta will be around 154.

The CA-V value where the green trace crosses the same Time Grid as IC = 2 mA should correspond to the base bias point where IC is equal to 2 mA. This is the bias point we would like to center our input signal on for the next measurement of the amplifier gain.

Move the channel B input back to the collector of the transistor.

Calculate new Min and Max values for Channel A by adding and subtracting 0.25 V to the 2 mA bias point we just measured. Enter these for Channel A. Set Channel A mode to sine wave. Under the Curves menu turn off the RB-V trace and under the Math menu select none. Set the time base to 2.0 mS/Div and the Hold Off to 0.0 so that two cycles of the input waveform are displayed. Under the Meas CA and CB menus in the -CA V- and -CB V- sections select Avg and P-P to be displayed.

Press the green Run button. You should see something like figure E10.

Figure E10, Amplifier input / output gain

The DC average of the output waveform should be at about 2.5 V (2.5 mA in the collector load resistor) below the +5 V power supply or about +3 V. The voltage gain of the amplifier will be the Channel B P-P value (0.9) divided by the Channel A P-P value (0.5). Which for this example is about 1.8.

## The Phase Analyzer:

There are many times where you are interested in measuring the phase angle and magnitude difference between two signals (sine waves). In most Circuits classes, students calculate phase difference using an oscilloscope by measuring the time difference between the zero crossings of two signals. This process can be tedious and prone to errors. Measuring phase is part of teaching phasor notation. Students have trouble going back and forth between phasor notation and time signals, at least in the beginning. Students are exposed to the same sort of gain / phase information from a Bode plot but in a different form.

There are specialized instruments for that, known as a Phase Analyzer. This is typically used in teaching power systems and power electronics circuits but general circuits classes can benefit as well. A phase analyzer shows the phase relationships between two or more sinusoids in the complex plane, on a polar plot. Many professors teaching introductory circuits courses find a tool like this useful when students are first learning about phasors.

The multi-channel voltage measurement support of Universal ALICE makes adding this polar / vector display very inviting. So here is the ALICE take on such a virtual instrument display mode.

The corresponding time domain scope display for the example test circuit from figure 22 is shown first in figure 20 for comparison.

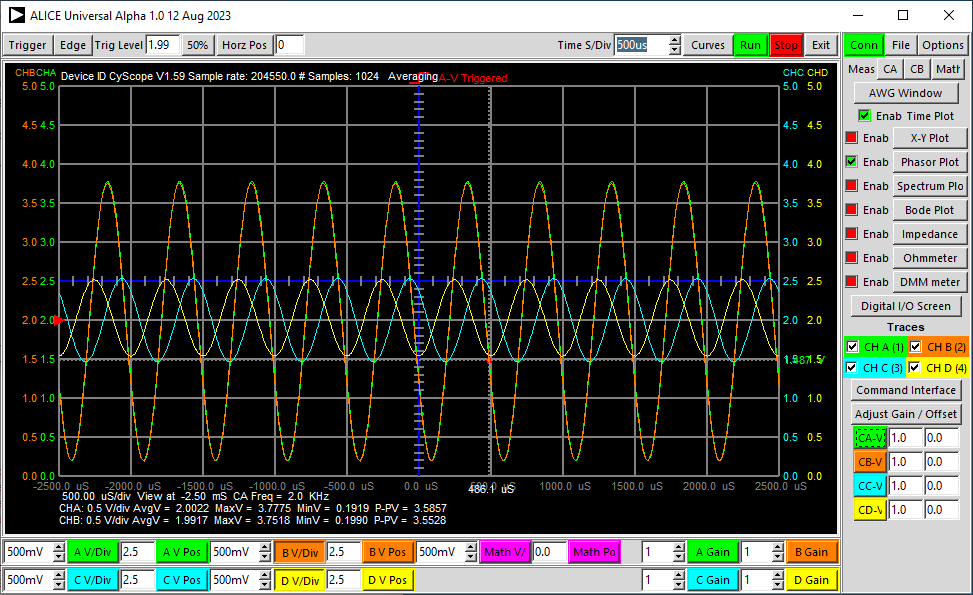
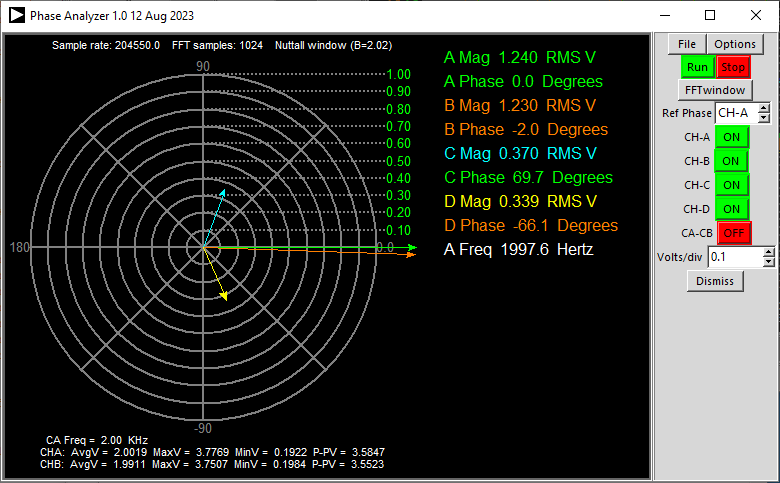


Figure 20, Time domain scope display.

The Phase Analyzer uses the Time domain waveform data to calculate an FFT and extracts amplitude and phase information from that. The Time display must be actively running to display Phase information. The Phase display for the time waveforms in figure 1 is shown in figure 2.



**Close Window**

**Select signal as reference phase**

**Enable Channels**

**Set RMS V/Div**

Figure 21, Phase Analyzer screen.

In the Phase Analyzer you choose one of the input signals to be the "reference." It is assigned 0 degrees and all other signals are measured with respect to this. The reference phase can be any of the four signals. The voltage vectors can be turned on or off. The scale for the RMS voltage can each be adjusted. Other controls for the FFT work much like the impedance analyzer and spectrum analyzer.

### Use Examples

**RC and RL Circuits**

In the circuit example shown in figure 22, we have the AWG A output driving a series RC circuit (R=1350 C=0.22 uF) and the AWG B output is driving a series RL circuit (R=510 L=12mH) at a frequency of 2 KHz. The C-A voltage vector is used as the reference phase (set to 0). The sine wave amplitude for both AWG channels is set to 1.2 Vrms (C-A green vector, C-B orange vector), the voltage across C1 is the blue vector and the voltage across L1 is the yellow vector. In the AWG controls the B output phase is set to 0 degrees relative to the A output. Those two voltage vectors are therefore both at 0 degrees and are plotted on top of each other. Because the AWG outputs have a DC offset component and the scope inputs only measure positive voltages the ends of C1 and L1 are connected to a fixed +2.0 V DC source in this example.

C1

R2

L1

CA-In

+2.0V

CB-In

R1

CC-In

CD-In

AWG A

AWG B

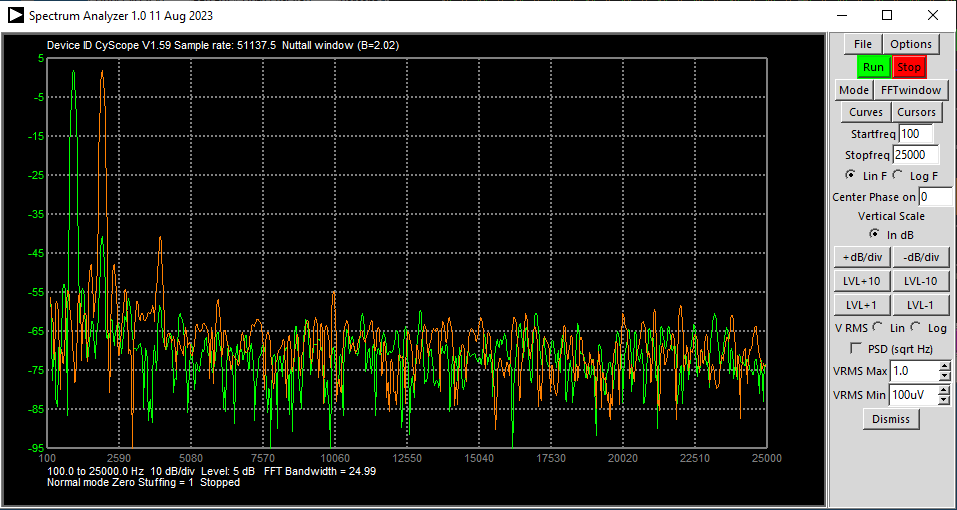
Figure 22, Example Circuits

In this circuit example the channels are used independently to measure two separate circuits.

## The Spectrum Analyzer:

### Window Setup:

When the Spectrum window button is clicked in the Main Window the Spectrum display Window should appear, as shown in figure 23. It is sub divided into 2 sections.



**FFT window selector**

**Mode selector**

**Cursors Menu**

**Curves Menu**

**Start Frequency**

**Stop Frequency**

**Lin / Log Freq Axis**

**Set dB/Div**

**Set dB Reference Level**

**Close window**

**Angle to center Phase Axis on**

**Vertical scale in dB**

**Vertical scale in Vrms**

**Set Vrms Reference Level**

**Vertical scale as PSD**

Figure 23, ALICE Spectrum analyzer window

### The menu buttons:

The following sections cover the functions of the various menu buttons. All of the program controls can be found under the buttons, there are no scrollbars, or rotating knobs.

**File drop down menu**

Save Config

Load Config, commands for saving and loading configuration settings (.cfg file)

Save Screen, command for saving the graphics display area to an encapsulated postscript file (.eps)

Save Data, command for saving the captured channel A and B amplitude vs frequency data to a coma separated values file (.csv). The amplitude data can be saved as magnitude in Vrms ( type a 0 ) or in dBV ( type a 1 ).

**Options drop down menu**

Smooth, an option to enabling smoothing where spline curves are used to connect the FFT frequency points rather than the default straight lines.

Cut-DC, an option that will remove the DC component from the sampled data record. It element by element subtracts the average value of the sample record.

Store trace, no explanation required, you can store a reference trace with it.

Screen setup, to select the number of vertical divisions on the grid. Also for selecting smaller grids. Can be handy when you want to capture smaller spectrum analyzer bit map pictures for documentation or a website.

Setup, in the setup menu, you can input the desired Zero Stuffing factor (power of 2).

**Run, Stop**

Start and stop buttons for the sweep.

**Mode drop down menu**

In Normal mode the trace is continuously refreshed.

In Peak hold, the peak or maximum value for each frequency bin is remembered. For each sweep if the new value is higher, then that new data point of the trace is saved and displayed.

In Average mode, the trace values are averaged. This smooths out the randomness in the noise floor.

In Single Shot mode a single sweep is obtained each time the Run button is pressed.

**FFT window drop down menu**

Used to select an FFT window function. It is generally better not to select the "Rectangle window" or no window. This window has a poor dynamic range due to the high side bands that are generated with no weighting function in the FFT calculation. The Flat Top window gives the highest amplitude accuracy but also has a large bandwidth, so less selectivity.

The number of samples in the FFT calculation is generally fixed based on the buffer size supported by the target hardware. More samples provides higher frequency resolution but a slower update rate for the screen. Fewer samples provides a lower frequency resolution, but a faster update rate for the screen.

**Curves drop down menu**

The Curves button allows the selection of which signal waveforms will be displayed. The All button selects all four curves to be displayed and the None button clears all four curves. The Marker option turns on a text marker which displays the amplitude and frequency at the peak of the displayed signal. Options to display the difference (subtraction) of the CA-dBV – CB-dBV traces or the CB-dBV – CA-dBV traces. It is also possible to select which of the possible stored reference traces, if saved via the Store trace option, will be displayed.

The color of the CA-dBV and CB-dBV traces will turn red if the input signal goes beyond the 0 to +5 V analog input signal range.

Under the Curves Drop down menu there are selectors for displaying the F cursor ( frequency ) and dB cursor (amplitude). When selected if you right click anywhere within the display grid either a vertical or horizontal cursor line, or both, will be drawn at that location. The vertical, horizontal, or both values for that point will be displayed. Scrolling with the mouse wheel will move the vertical line left–right when only the F cursor is selected and the horizontal up-down when only the dB cursor is selected. When both are selected the mouse wheel moves the vertical line left–right.

**Start freq and Stop freq**

Used to set the start and stop frequency of the display. The stop frequency can not be higher than ½ the sample rate. If a greater value is entered the highest frequency displayed will limited to ½ the sample rate. The start frequency can not be lower than one FFT frequency bin width. If a lesser value is entered the lowest frequency displayed will limited to one FFT bin width.

**Lin F and Log F selector**

Select linear or logarithmic horizontal frequency axis scale.

**dB/div +/- buttons**

Used to set the dB's per division. Can be 1, 2, 3, 5, 10, 15, or 20 dB/Div.

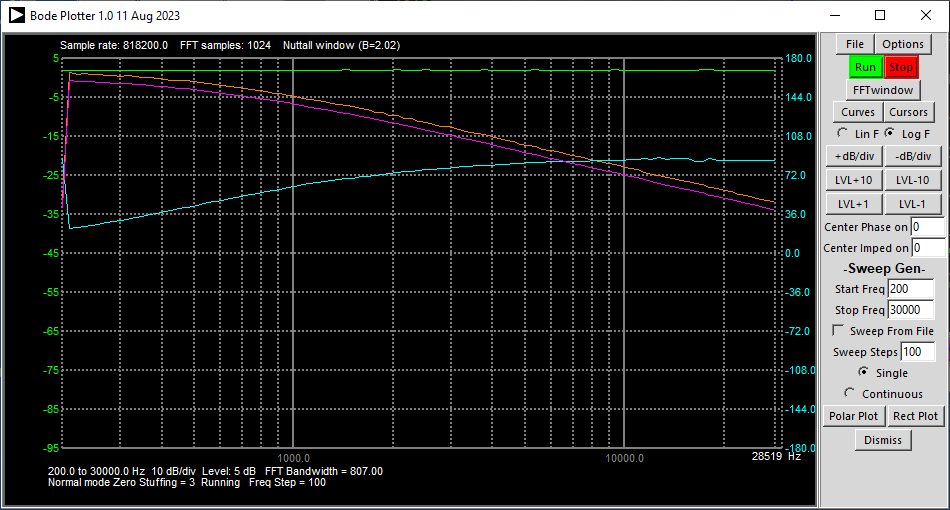
**LVL +/- buttons**

Used to set the top line of the grid or reference level. Sometimes called the “sensitivity”. 0 dB is equal to an input amplitude of 1 Vrms.

## The Bode Plotter:

### Window Setup:

When the Bode Plot window button is clicked in the Main Window the Bode Plot display Window should appear, as shown in figure 24. It is sub divided into 2 sections.



**FFT window selector**

**Curves Menu**

**Start Frequency**

**Stop Frequency**

**Lin / Log Axis**

**dB/Div**

**Set Reference Level**

**Close window**

**Number of steps**

**Open Rect Plot**

**Open Polar Plot**

**Cursors Menu**

**Angle to center Phase Axis on**

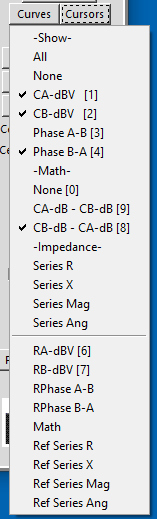
Figure 24, ALICE Bode plot window

**FFT window drop down menu**

Used to select an FFT window function. It is generally better not to select the "Rectangle window" or no window. This window has a poor dynamic range due to the high side bands that are generated with no weighting function in the FFT calculation. The Nuttall window function is set by default and is generally the best overall option. The Flat Top window gives the best amplitude accuracy but can have strange phase results at some frequency steps.

**Curves drop down menu**

The Curves button allows the selection of which signal waveform traces will be displayed. The All button selects all four curves to be displayed and the None button clears all four curves. The Marker option turns on a text marker which displays the amplitude and frequency at the peak of the displayed signal. Options to display the difference (subtraction) of the CA-dBV – CB-dBV traces or the CB-dBV – CA-dBV traces. It is also possible to select which of the possible stored reference traces, if saved via the Store trace option, will be displayed.



**Used in conjunction with Impedance Analyzer when in Sweep mode**

**Select All Traces or None**

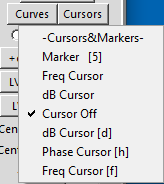
**Display Snap Shot Reference Traces**

**Use Math Traces for relative gain and phase**

Figure 25, Bode plot Curves Drop Down

The color of the CA-dBV and CB-dBV traces will turn red if the input signal goes beyond the 0 to +5 V analog input signal range.

Under the Curves Drop down menu there are selectors for displaying the F cursor (frequency) and dB cursor (amplitude). When selected, if you right click anywhere within the display grid either a vertical or horizontal cursor line, or both, will be drawn at that location. The vertical, horizontal, or both values for that point will be displayed. Scrolling with the mouse wheel will move the vertical line left–right when only the F cursor is selected and the horizontal up-down when only the dB cursor is selected. When both are selected the mouse wheel moves the vertical line left–right.



**Show Fundamental Frequency / Harmonics Markers**

**Select between dB or Phase cursor**

**Select Cursors**

**Turn On / Off All Cursors**

Figure 26, Bode plot Cursors Drop Down

**Lin F and Log F selector**

Select linear or logarithmic horizontal frequency axis scale. This also determines how the frequency steps are spaced, linearly or logarithmically.

**dB/div +/- buttons**

Used to set the dB's per division. Can be 1, 2, 3, 5, 10, 15, or 20 dB/Div.

**LVL +/- buttons**

Used to set the top line of the grid or reference level. Sometimes called the “sensitivity”. 0 dB is equal to an input amplitude of 1 Vrms.

Under the Sweep Gen menu section are controls for generating frequency sweeps of the analog output sources. The screen is up-dated after each frequency step. First are radio buttons to select which AWG output channel, or none will be swept. When using the Bode plotter the selected AWG will be forced into SVMI mode with a Sine Shape. Use the AWG controls window to set the source amplitude. The other channel will be forced into Hi-Z mode. The selected output will be swept from the Start Frequency to the Stop Frequency. The number of steps can be set using the Sweep Steps entry. Lastly, there is a radio button selector for single or continuous sweep. The frequency sweep is started, or restarted from the beginning each time the Run button is pressed.

**Bode Polar and Rectangular Plots**

The standard Bode plot for a parallel LC tank circuit is shown in figure 27, where the relative gain or CB-dBV – CA-dBV trace and relative Phase B-A trace are plotted on the log frequency axis.

Figure 27, Bode Gain, Phase log Frequency Plot

It is also possible to plot the relative gain and relative phase measurement on a polar axis as the frequency is swept. Figure 28 plots the CB-dBV – CA-dBV gain trace as a function of the relative phase angle.

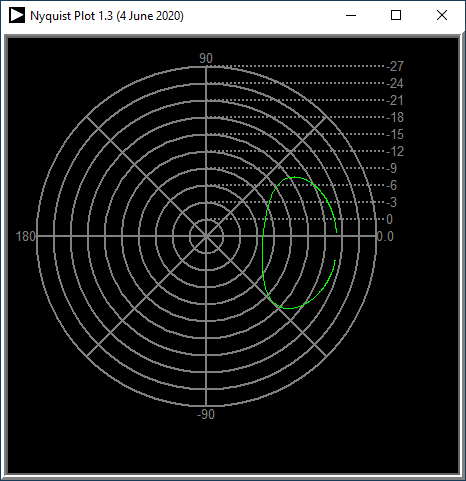


Figure 28, Bode Polar Plot

It is also possible to plot the relative gain and relative phase measurement on a rectangular axis as the frequency is swept. Figure 29 plots the CB-dBV – CA-dBV gain trace (vertical) as a function of the relative phase angle (horizontal).

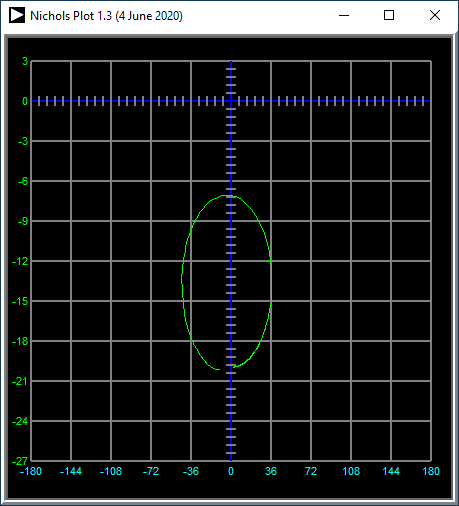


Figure 29, Bode Rectangular Plot

## Frequency Analysis:

The ALICE Desktop program uses the Fast Fourier Transform (FFT) to produce the frequency spectrum of a set of time samples of the input signals. The FFT takes as an input a set of time samples at a given sample rate and produces a set of frequency samples or values from DC ( 0 Hz ) to one half of the sampling frequency. In the case of the ALM1000 the sample rate is fixed at 100 KHz so the highest frequency will be one half of that or 50 KHz. The number of individual frequency bins the FFT produces is one half the number of time samples that are used. So the width of the bins or frequency resolution will be 50 KHz divided by one half the number of time samples taken. The number of time samples can be set from 64 ( 26 ) to 65536 ( 216 ) in the program.

### What is an FFT window function?

In ALICE you can choose from a number of FFT window functions. But what is an FFT window and what is it doing? The principle is very simple. The program reads a number of samples from the Hardware and puts them in an array. The size of the array is generally a power of 2 (but not required) for the FFT calculation, for example 1024. With no window weighting function, all samples have an equal contribution or weight in the FFT calculation. You should expect to have an optimal result, but that is not the case if there is not an exact number of repeating cycles in the array. Another way of thinking about this is the starting value of the time waveform must be the same as the ending value. The end of the waveform will line up with the beginning if wrapped around on itself. This will almost never be the case in actual practice.

An FFT windowing function weights the samples from the beginning of the array to the end. With higher weights at the center and weights closer to zero near the start and end. The samples at the beginning and at the end of the array, that probably don’t line up, hardly contribute to the FFT calculation. Why would we use only part of the samples or even not at all? There are even FFT window functions in which some sample points counteract with the other sample points.

The reason why we need an FFT window can be seen figures 30-35 in the various spectrums using different FFT window functions. No FFT window (also called a Rectangular window), generates many side bands in the spectrum of the FFT calculation. That is very visible in the first spectrum plot of the Rectangular (dark orange) and Cosine (light orange) window functions. Very low amplitude signals close to the main signal cannot be measured. So the dynamic range around the large main signal is low. By using an FFT window, the side bands are much more attenuated, how much depends on the type of FFT window. The increased side band suppression is at the expense of the selectivity. FFT windows with a very high side band suppression and therefore a very high dynamic range, have much less selectivity.

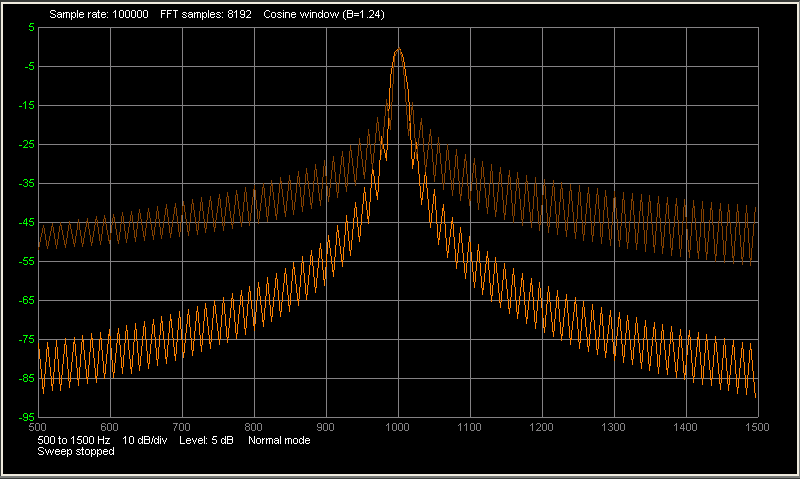


Figure 30, Rectangular vs cosine window function

A Cosine window is a good compromise between a good selectivity and a good dynamic range.

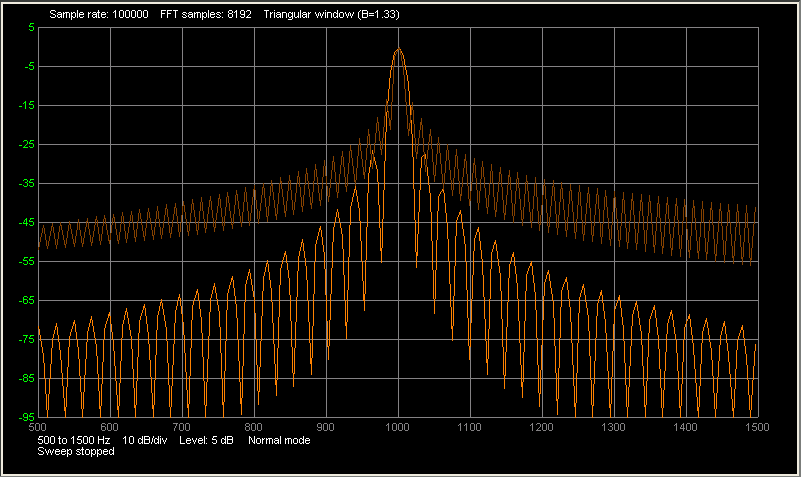


Figure 31, Rectangular vs Triangle window function

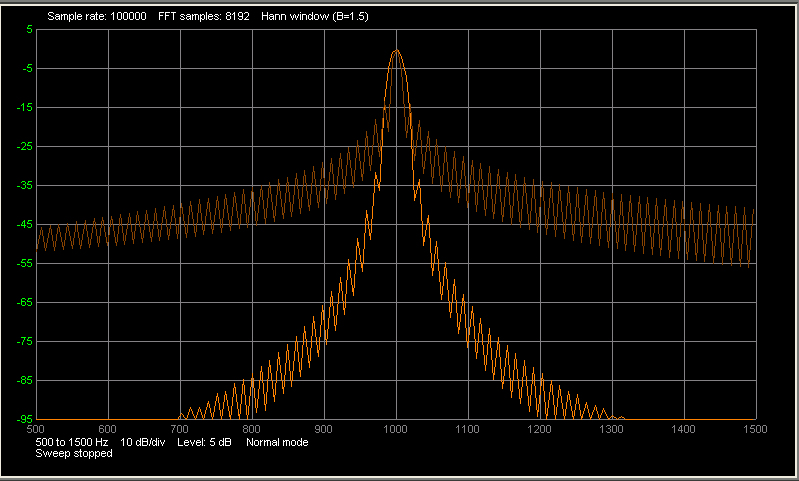


Figure 32, Rectangular vs Hann window function

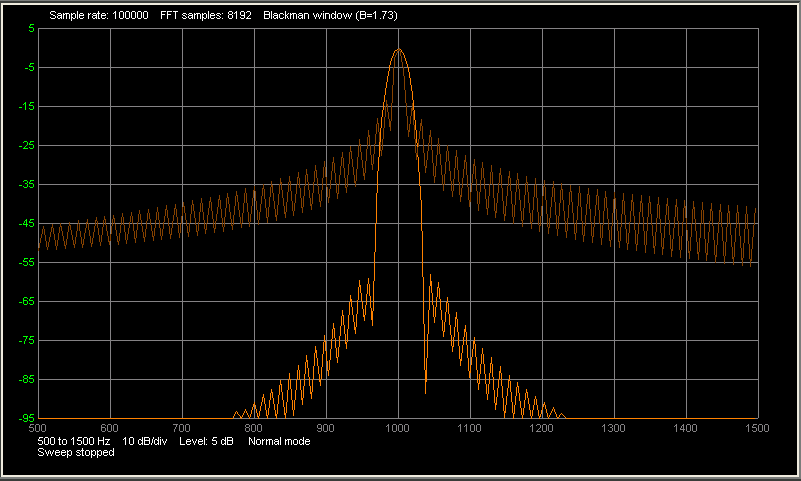


Figure 33, Rectangular vs Blackman window function

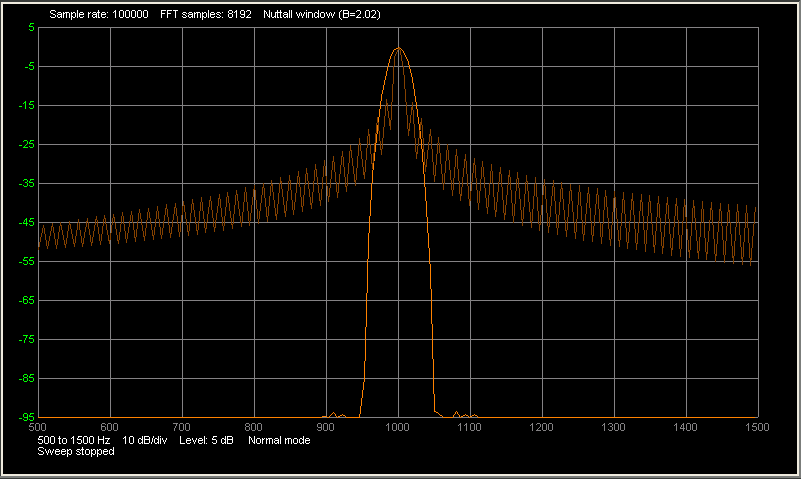


Figure 34, Rectangular vs Nuttall window function

At the expense of a little wider bandwidth the Nuttall window function provides the best side band reduction and may be the optimal compromise between good selectivity and good dynamic range.

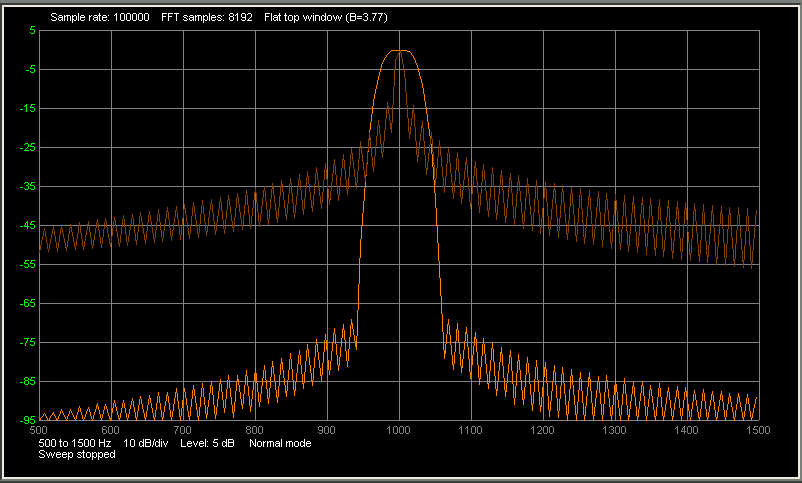


Figure 35, Rectangular vs Flat Top window function

A special filter is the Flat Top filter. It has a flat top as the name implies. That is why it is very usable for accurate amplitude measurements. The peak of the signal does not have to be exactly on the center of an FFT frequency bin.

ALICE has 7 built in windowing functions.

Rectangular, no window function B=1

Cosine window function, medium-dynamic range B=1.24

Triangular non-zero endpoints, medium-dynamic range B=1.33

Hann window function, medium-dynamic range B=1.5

Blackman window, continuous first derivate function, medium-dynamic range B=1.73

Nuttall window, continuous first derivate function, high-dynamic range B=2.02

Flat top window, medium-dynamic range, extra wide bandwidth B=3.77

Zero Stuffing

With the menu button "Setup" you can set the factor for the Zero stuffing. What problem are trying to solve by Zero stuffing? The bandwidth of the FFT depends on the choice of the FFT window function. For a narrow FFT filter, the bandwidth is slightly larger than the difference between 2 FFT frequency bins. When the signal frequency is exactly between the 2 FFT frequency bins, the signal will be displayed lower than its actual value because half of the signal appears in each of the two bins. Figure 8 shows good example of this. The signal is slightly more than 1 KHz and lies exactly between the two FFT frequency bins. The actual peak value should be equal to 0 dB, but the displayed value of the two adjacent samples is lower. The signal level is not displayed correctly by either of the FFT frequency bins. This is called Scalloping loss.

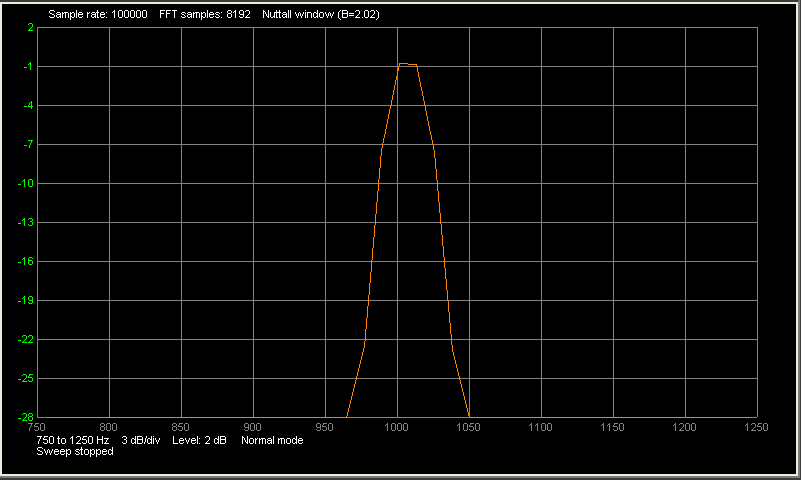


Figure 36, Fundamental frequency not centered, no zero stuffing

Zero stuffing provides a simple solution to this problem. For 1x Zero Stuffing, we double the size of the time sample array. The original array was say 2048 samples. We add 2048 samples with the value zero and we get a new array with 4096 samples. This may seem counterintuitive, when we add zero's we do not add extra measurement data. However, something happens in the FFT calculation with twice as many samples. The effect can be seen in figure 16. Extra FFT frequency bins have been added. Coincidentally, here the extra frequency bin coincides with the frequency of the signal and the level of the signal is displayed correctly. Also even if the signal frequency does not coincide with the frequency of the extra FFT bin, the measured error will be smaller. As we add samples with the value zero, the bandwidth of the FFT filter remains the same.

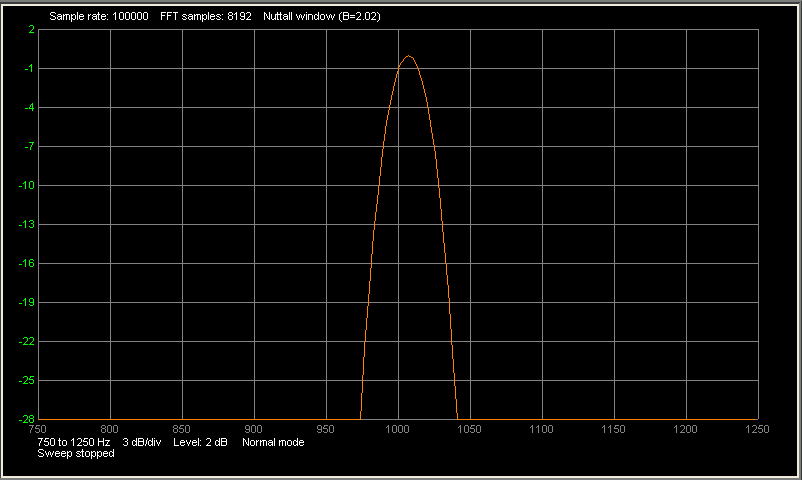


Figure 37, Fundamental frequency not centered, with zero stuffing

In the program, you can choose a value between 0 and 5 for the Zero Stuffing. As it is a power of 2, it is a value between 1 and 32. So 0x - 31x points will be added. As a result, the FFT calculation time will be up to 32x longer as well and the spectrum analyzer screen update rate will slow down considerably. One extra point (a value of 1 for the Zero Stuffing) is often good enough to keep the Scalloping loss acceptable. As an alternative, what you can do is set Zero Stuffing to 0, and use a Flat top window. The flat top is so wide, that even without Zero Stuffing, you will have little Scalloping loss, but you will have less frequency selectivity.

Spectrum Examples:

The following example shows a technique where the ALICE spectrum analyzer tool can be used to measure the amplitude vs frequency response of two simple RLC configurations. Shown in figure E11, first on the left is a parallel LC bandpass configuration and second on the right is a series LC bandstop configuration. Indicated by the green boxes are the connections to the ALM1000. Channel A is setup to output the driving function of the network. Channel B is setup as an input to measure the response seen across the LC network. For this example R1 is 1 KΩ, L1 is 15 mH and C1 is either 0.22 uF or 0.44 uF.

L1

C1

CB-In

+2.5V

CA-In

R1

L1

C1

CB-In

+2.5V

CA-In

R1

Parallel LC Bandpass

Series LC Bandstop

AWG

AWG

Figure E11 RLC circuits

In a linear system, the frequency response can be obtained by sweeping sinusoidal inputs over a range of frequencies. This series of sinusoidal signals at different frequencies can then be used to compute the frequency response. While the ALICE Desktop Bode Plotter does include a sweep generator function, a sweep with many frequency points using large FFT sample sizes can take many seconds or even minutes to up-date the plot. However, FFT analysis can be used to obtain the transfer function for a network from its impulse response. We can generate a test signal with a wide frequency content, a very narrow square pulse, which will produce a plot from a single sample record at a much higher up-date rate.

Using the FFT to get a transfer function of a system is not overly complex.

There must be an input to the network which you can observe and record. There must be an output from the network which you can observe and record. The input and output data has to be able to be read into an analysis program, such as ALICE Desktop, that can take the Fast Fourier Transform of both input and output data records. A basic concept of linear analysis is that the unit impulse response of a network and the transfer function of the network are a Laplace transform pair, or said another way, the transfer function is the Laplace transform of the unit impulse response. The implication is that we can obtain the transfer function by getting the Laplace transform of the unit impulse response.

**Practical implementation.**

If we set the number of FFT samples to 8192 the total sample time will be 81.92 mSec which is the same as one cycle at 12.2 Hz. By setting the Channel A function generator to a 12.2 Hz square wave with a very narrow duty cycle of only 4 – 6 samples wide the resulting test signal will contain frequency content every 12.2 Hz with nearly equal amplitude out to high frequencies. At 12 Hz each 10 uSec sample period is equal to about 0.012 % of duty cycle. We can set the duty cycle to anything from 0.012% to 0.08% and get similar results. The only difference is how fast the signal level falls off with increasing frequency. For a given pulse amplitude, the narrower the pulse the less energy in each 12.2 Hz spaced frequency but the flatter vs frequency they will be. The wider the pulse the more signal energy but a faster frequency roll off. 0.08% gives an acceptable frequency roll off out to 10 KHz.

The detailed settings for Channel A are as follows:

Shape - Square

Mode - SVMI

VMIN = 1.3

VMAX = 3.7 (pulse amplitude set to allow some headroom for overshoot and ringing)

Freq = 12.2

Phase = 180 (phase is set to 180 degrees to center the pulse in the time sample record)

DutyCycle = 0.08 (can be adjusted down to 0.012% for flatter input signal energy)

Channel B is set in Hi-Z mode as an input.

Other Settings:

FFT Window – Flat top (has a wide FFT bandwidth which is wider than 12 Hz)

FFT Samples = 8192

Start Freq = 100 (set to something other than 0, to ignore DC content)

Stop Freq = 10000

ZeroStuffing = 0 (can be adjusted but generally has little effect on resultant plot)

Below in figure E12 is a screen shot for the bandpass RLC configuration of figure E1. The green trace for channel A is the narrow pulse forcing function response. The light and dark orange traces are the output responses seen by channel B for C1 = 0.44 uf and 0.22 uF respectively. The light and dark magenta traces are the subtraction of the Channel A trace (n dBV) and the Channel B trace (in dBV). As we know subtraction in dB (logs) is the same as division in magnitude. The magenta traces are the actual input to output transfer function of the RLC network. The Yellow trace is the phase response.

Figure E12, Bandpass response

Similarly in figure E13 is a screen shot for the bandstop RLC configuration of figure E11.

Figure E13 Bandstop response

**For Further Reading:**

https://en.wikipedia.org/wiki/Fast\_Fourier\_transform

http://www.analog.com/media/en/training-seminars/design-handbooks/MixedSignal\_Sect5.pdf

https://en.wikipedia.org/wiki/Window\_function

https://en.wikipedia.org/wiki/Spectral\_leakage

http://docs.scipy.org/doc/numpy/reference/generated/numpy.fft.fft.html

## Impedance Analyzer / LCR Meter

### Background:

The basic concept that is used to make gain/phase, impedance and RLC measurements using ALICE Desktop is shown in figure 38. Channel A of the hardware AWG is used to apply a known frequency sine wave at VA and measure the applied voltage waveform. Channel B is used to measure the voltage waveform seen across the network under test. FFTs are calculated on the two waveforms which provide amplitude and phase information at the applied frequency. From these the relative gain ( CHB amplitude / CHA amplitude ) and relative phase ( CHB phase – CHA phase) are obtained. Further these values can be used to calculate the impedance (RLC) of the network in the dashed box.

The resistor, REXT, is a known value. For the audio frequency range measurements possible with the ALM1000 hardware it can be adjusted as needed depending on the magnitude of the impedance being tested. Impedances in the range of about 0.1 to 10 times REXT can be accurately measured. REXT can range from 50 Ω to 50 KΩ.

The unknown impedance to be measured is modeled as a series circuit consisting of an unknown series resistance, RX, and an unknown series reactance, jXX. The magnitude of the impedance is ZX.

RX

XX

REXT

VA

VZ

VI

VRX

jVXX

Figure 38: Basic Concept

Three voltages are measured:

1. VA is the applied voltage (from Channel A).

2. VZ is the voltage across the unknown impedance (from Channel B).

3. VI, the voltage across the known resistor REXT is calculated from VA and VZ and is related to the current in both REXT and the unknown impedance.

These three voltages are actually vectors and indicated in figure 39.

VA

VI

VRX

jVXX

VZ

Common

Θ

Φ

Figure 39: Vector Diagram

Using the law of cosines and referring to figure 38 the magnitude of VI can be calculated as:

VI = sqrt(VA^2 + VZ^2 – 2\*VA\*VZ\*cos(Φ))

The angle Φ is the measured relative phase between channel B and channel A. The law of cosines is used to calculate the cosine of the angle, Θ.

cos(Θ) = (VA^2 + VI^2 – VZ^2)/ (2 \* VA \* VI)

The magnitude of the total impedance (including REXT) can be calculated as:

Za = R­­EXT \* VA / VI

We note from figure 1 that the sum of REXT and RX can be found by:

REXT + R\_X = Za \* cos(Θ)

Thus, we can solve for RX by:

RX = Za \* cos(Θ) – REXT

Taking possible measurement errors into account it is possible that RX could compute to be a negative value which is not likely to be the case. The thing to do if that happens is to set RX to zero. The impedance is purely reactive.

The magnitude of the unknown impedance can be calculated as:

ZX = REXT \* VZ / VI

The magnitude of the unknown reactance can be calculated as:

XX = sqrt(ZX^2 – RX^2)

Again taking possible measurement errors into account it is possible that the square root of a negative number might occur. If that happens then XX should be set to zero.

Once we have a value for XX, we can calculate either the series capacitance (when XX is negative = XC) or series Inductance (when XX is positive = XL).

C = -1/(2π f XC)

L = XL/(2π f )

### Making Measurements:

Connections to the hardware and the network to be measured are shown in figure 40. In this case we show a simple series connected resistor and capacitor. REXT is 1350 Ohms and the series resistor RS is 100 Ohms and the capacitor CS is 0.22 uF. The AWG generator output should always be set to be a sine wave shape. The user can control the output voltage amplitude and offset with the Min and Max entry slots as when using the scope and spectrum analyzer displays. A good place to start is with Min set to 0.2 and Max set to 3.8 which produces a 1.2 Vrms amplitude centered on 2.0 V DC. The Channel B analog input is used with the Impedance Analyzer and it always considered as an input.

CS

CA-In

+2.0V

CB-In

REXT

RS

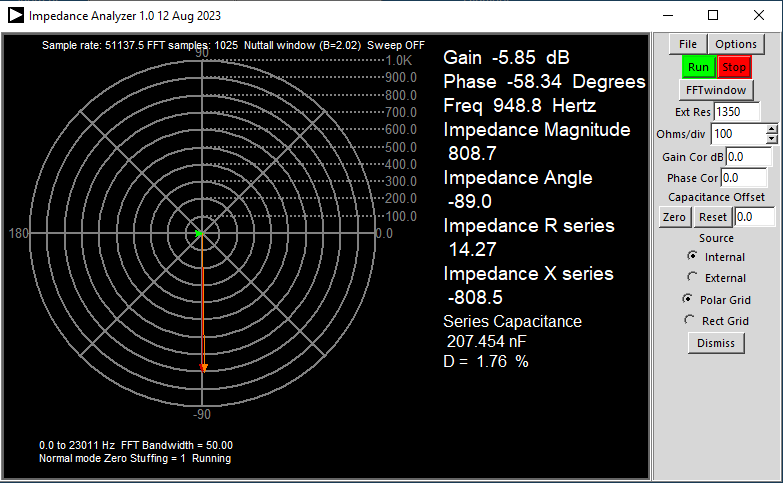
Test Circuit

AWG

Figure 40, Measurement setup

### Window Setup:

The main impedance analyzer window should appear, as in figure 41. It is sub divided into 2 sections.



**File Menu**

**Options Menu**

**Enter External Resistor value here**

**Set Ohms per Division**

**Enter Gain Correction value here**

**Enter Phase Correction value here**

Figure 41, ALICE Impedance Analyzer window

### The Right Side Menu Section

**Run, Stop**

Start and stop buttons for continuously taking readings.

**FFT window drop down menu**

Used to select an FFT window. It is generally better not to select the "Rectangle window" or no window. This window has a poor dynamic range due to the high side bands that are generated with no weighting function in the FFT calculation. The Flat Top window gives the highest amplitude accuracy but also has a large bandwidth, so less selectivity. Using the narrowest bandwidth FFT window and increasing the zero-stuffing factor can improve the measurement results. The program starts up set to the Nuttall window (BW=2.02).

**File drop down menu**

Save Config Load Config buttons. Commands for saving and loading the configuration settings to a file. (.cfg file)

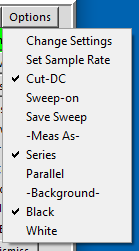
Save V-Cal, Load V-Cal buttons. ALICE-VVM uses the same calibration file as the Voltmeter Tool. To load the saved calibration factors press the Load button. To save the calibration values to the file for future use, press the Save button. The values are saved to a file with a unique name board based on the first 9 characters of the board device ID serial number. For example something like: 203131543\_V.cal.

Save Screen button. Command for saving the graphics display area to an encapsulated postscript file (.eps).

The Help button will open a web browser to this document on the ADI Wiki site.

**Options drop down menu**

Cut-DC, an option that will remove the DC component from the sampled data record. It sample by sample subtracts the average value of the sample record. Any DC offset in the FFT could result in that being the peak amplitude and resulting in meaningless measurements. The program starts up with this turned on. This is important given the positive only analog input range of some data acquisition hardware and the DC offset.



**Open Sample Rate Controls**

**Open Settings Controls**

**Save Reference Sweep Trace**

**Turn on Impedance sweep in conjunction with Bode Sweep**

**See section on calculating parallel impedance**

Figure , Impedance Analyzer Options Drop Down

The section along the right hand side contains the controls for making the measurements. There is a place to enter the external resistor value. The program starts up with this set to 1000. Next is a spin box to set the number of Ohms/div for the polar (circular) grid.

Analog Inputs

Low Capacitance FET Input Buffers

Breadboard Adapters

### Main Graphics area

The main graphics area is where the measured results are displayed. The impedance magnitude and angle along with the real and imaginary parts are drawn on the polar (circular) grid in Ohms. The real, series resistance component is drawn in green at 0 degrees phase. The imaginary part of the series impedance is drawn in red at either +90 degrees or -90 degrees depending on the sign. A positive impedance is inductive and an negative impedance is capacitive. The combined magnitude of the total series impedance is drawn in orange and at the measured angle.

To the left of the grid the relative gain of Channel B to Channel A is displayed in dB. Next the relative phase is displayed in degrees. Next the measured frequency in Hz is displayed. Next the measured Impedance Magnitude, Angle, R series and X series are displayed. Finally the calculated capacitance (if X series is negative) or inductance (if X series is positive) is displayed.

To convert the series values to the equivalent parallel values see section on Calculating Parallel Impedance further down in this document.

Additional setting information is also shown.

### Impedance Analyzer Examples:

**Example 1:**

As an example to show the frequency dependent impedance of a series LC circuit we will use the ALICE impedance analyzer tool to examine the combination shown in figure E14 with L1 equal to 6.5 mH and C1 equal to 1 uF. We will use a 1000 Ω REXT to be in line with the expected impedance level of the circuit.

L1

C1

CA-In

+2.0V

CB-In

REXT

Series LC

AWG

Figure E14 Testing a series LC circuit

The LC circuit is tested at three different frequencies, the first much lower than the resonate frequency where the impedance is dominated by the capacitor shown in figure E15.

Figure E15 Measured results at low frequency, 500 Hz

The second much higher than the resonate frequency where the impedance is dominated by the inductor shown in figure E16.

Figure E16 Measured results at high frequency, 8500 Hz

The third at the resonate frequency where the negative impedance of the capacitor nearly cancels the positive impedance of the inductor shown in figure E17.

Figure E17 Measured results at resonate frequency, 2191 Hz

In these cases the series R measured stays nearly the same at about 11 Ω.

**Example 2:**

We can use ALICE Desktop to measure the input capacitance of channel B. We know that the input capacitance is small so we will need to use a large value for REXT and measure at a high frequency. In figure E18 we show the connections used which is simply to connect CHA to CHB with a 47 KΩ resistor.

RIN

CIN

CA-In

CB-In

REXT

Channel B Input

**47 KΩ**

Figure E18 Measure CH B input capacitance

In the ALICE Impedance Analyzer screen shot shown in figure E19 we see that Ext Res is set to 47000 and the test frequency is set to 19000 Hz. The calculated capacitance is 394 pF which agrees nicely with the capacitance reported in the document on the ALM1000 analog inputs.

Figure E19, Measured results for CH B input capacitance

If we use the formula from Calculating Parallel Impedance to convert the series R to the parallel resistance we get around 1 MΩ. This is right in line with the known design value.

To measure capacitors around the same value as the input capacitance or even smaller it would be useful to null out this stray parasitic capacitance. This can be done using the Gain Cor and Phase Cor Entry widgets to enter correction factors for the gain and offset. If we enter 7.74 (dB) from the measured Gain for the Gain Cor entry and 65.35 (degrees) for the Phase Cor entry we get the result shown in figure E20.

Figure E20 Gain and Phase have be corrected

Now the Measured Gain difference is 0.0 dB and the Measured Phase difference is 0.0 degrees. The calculated capacitance is 1.0pF. If we now add a 39 pF ceramic cap, from the Analog Parts Kit, from the channel B input to ground we get the results shown in figure E21.

Figure E21, 39 pF cap added to CHB.

Now the calculated capacitance reported is 37 pF which is what we can expect from a +/- 20% tolerance on the capacitor.

### Calculating Parallel Impedance:

The method used in the ALICE impedance analyzer tool determines the series resistance and reactance. Sometimes the equivalent parallel impedance of a resistance and reactance are needed. All that is required is a mathematical series to parallel conversion as follows. The concept is to relate the real and imaginary conductance of the parallel network to the conductance of the series network. The numerator and denominator of the series network conductance is multiplied by the complex conjugate of the denominator to put the result in normal form.

1/R\_P + 1/jX\_P = 1/( R\_S + jX\_S) = (R\_S – jX\_S)/( R\_S^2 + X\_S^2)

where RS and XS are the series values and RP and XP are the parallel values.

By equating the real part we have the equivalent parallel resistance and by equating the imaginary part we have the equivalent parallel reactance:

R\_P = (R\_S^2 + X\_S^2) / R\_S

X\_P = (R\_S^2 + X\_S^2) / X\_S

Note that since the polarity of XS was known then the polarity of XP is also known and is the same sign.

### Impedance Sweeps:

The impedance Analyzer can be combined with the Bode plotter to perform sweeps of the network impedance. In this example we use the same parallel LC tank circuit with 50 ohms of series resistance added. We will do the same 100 Hz to 10 KHz sweep as before. Note: both the Bode and Impedance screens must be open and selected for this test.

From the Bode plot Curves Drop Down (Figure 14C) along with the CHB – CHA dB gain curve, under Impedance, the Series Mag and Series Angle are selected.

In the Impedance Analyzer Options Drop Down the Sweep On and Save Sweep options are selected. The external test resistance is the same 1000 ohms and the Ohms / Div is set to 100.

When a sweep is run we get the following two plots in the Bode window, figure E22, and the Impedance Analyzer window figure E23.

Figure E22, Impedance vs log frequency sweep

The plot colors might be a little confusing in figure E22. The magenta curve is the same relative gain in dB from the earlier Bode plot example (CB dB – CA dB) and uses the green dB vertical scale. The purple and darkish green curves are the impedance magnitude in ohms and phase in degrees respectively and use the magenta ohms and cyan angle vertical scales respectively.

Figure E23, Polar plot of Impedance for frequency sweep

## Digital I/O controls Windows:

If the target hardware provides digital input / output pins ALICE supports these through the Digital I/O controls.

Figure 27 Digital I/O connector

## Customizing ALICE Desktop

This section has been moved to the ALICE Advanced User’s Guide.

There are a number of variables that the user can use to customize the appearance of the user interface. These variables are located near the top of the Python program file.

These pairs of variables for each display window set the size of the graphics drawing area in screen pixels. The default values are sized to optimally fill a screen with 1024X600 resolution. The menu buttons surrounding the graphics area need this much space to be properly displayed on most screens so using sizes smaller than the default may result in mangled menus.

GRW = 720 # Width of the Time grid

GRH = 390 # Height of the Time grid

GRWN = 720 # Width of the spectrum grid 720 default

GRHN = 390 # Height of the spectrum grid 390 default

GRWBP = 720 # Width of the Bode Plot grid 720 default

GRHBP = 390 # Height of the Bode Plot grid 390 default

GRWXY = 420 # Width of the XY grid 420 default

GRHXY = 390 # Height of the XY grid 390 default

GRWIA = 400 # Width of the Impedance grid 400 default

GRHIA = 400 # Height of the Impedance grid 400 default

The colors that are used to draw the various parts of the screen can be modified.

Color = "#rrggbb" rr=red gg=green bb=blue, Hexadecimal values 00 - ff

COLORframes = "#000080" # 50% blue

COLORcanvas = "#000000" # 100% black used for background color

COLORgrid = "#808080" # 50% Gray used for grid lines

COLORzeroline = "#0000ff" # 100% blue used for vertical and horizontal center grid lines

COLORtrace1 = "#00ff00" # 100% green CH A voltage trace

COLORtrace2 = "#ff8000" # 100% orange CH B voltage trace

COLORtrace3 = "#00ffff" # 100% cyan CH A current trace

COLORtrace4 = "#ffff00" # 100% yellow CH B current trace

COLORtrace5 = "#ff00ff" # 100% magenta Math trace

COLORtrace6 = "#ff0000" # 100% red

COLORtrace7 = "#8080ff" # 100% purple

COLORtraceR1 = "#008000" # 50% green CH A voltage snapshot trace

COLORtraceR2 = "#804000" # 50% orange CH B voltage snapshot trace

COLORtraceR3 = "#008080" # 50% cyan CH A current snapshot trace

COLORtraceR4 = "#808000" # 50% yellow CH B current snapshot trace

COLORtraceR5 = "#800080" # 50% magenta Math snapshot trace

COLORtraceR6 = "#800000" # 50% red

COLORtraceR7 = "#4040a0" # 70% purple

COLORtext = "#ffffff" # 100% white used for Text display

COLORtrigger = "#ff0000" # 100% red used for trigger point

COLORsignalband = "#ff0000" # 100% red

Alternatively, for users of the Windows executable version of ALICE Desktop, a file named alice\_init.ini can be created and placed in the same directory with the alice-desktop-1.1.exe executable file. The alice\_init.ini file is read, if found, when ALICE Desktop starts and before any of the windows are created. If no init file is found the internal default settings are used.

Inside the alice\_init.ini any of the above variables can be set using the example format shown here:

global GRW; GRW = 720

global GRH; GRH = 390

global GRWN; GRWN = 720

global GRHN; GRHN = 390

global GRWXY; GRWXY = 420

global GRHXY; GRHXY = 390

global GRWIA; GRWIA = 400

global GRHIA; GRHIA = 400

## Using the numpy library

This section has been moved to the ALICE Advanced User’s Guide.

ALICE includes the numpy numerical library of array creation and manipulation functions. The reader is directed to the [[https://docs.scipy.org/doc/numpy/reference/index.html |numpy documentation]] for complete details on these functions. Here we will point out some of the more useful functions for creating and manipulating waveform sample arrays. Numpy contains many more than can be covered here. However, be sure to only use functions that return 1 dimensional arrays.

In these example we use AWGAwaveform as the array variable but any of the ALICE internal waveform arrays can be of course used.

### Array Creation:

numpy.ones(length) Return a new array of given length filled with ones.

numpy.zeros(length) Return a new array of given length filled with zeros.

numpy.full(length, fill\_value) Return a new array of given length, filled with fill\_value.

numpy.linspace(start\_value, stop\_value, num=length) Return a new array of given length of evenly spaced numbers between start\_value and stop\_value.

numpy.logspace(start\_value, stop\_value, num=length, base=log\_base) Return a new array of given length of numbers spaced evenly on a log scale. The base of the log can be optionally specified such as 10 or 2 etc.

### Trigonometric functions:

numpy.sin(x) Trigonometric sine, element-wise.

numpy.cos(x) Cosine element-wise.

To create one cycle of a sine wave 400 samples long you will first create an array of values from 0 to 2\*pi and then send it to the sine function like this.

numpy.sin(numpy.linspace(0, 2\*numpy.pi, 400))

The waveform values will be from -1 to 1 so additionally you will need to scale and or offset the values to be between 0 than 5 for the AWG. In this example we create the sine wave centered on 2.5 V with a P-P of 4 V.

(numpy.sin(numpy.linspace(0, 2\*numpy.pi, 400)) \* 2) + 2.5

numpy.sinc(x) Return the sinc function.

Much like the trig functions the input to the sinc function is a linear spaced array of points.

numpy.sinc(numpy.linspace(-4, 4, 400)) will product 4 “cycles” 400 samples long.

The values will be between -1 to 1 so additionally you will need to scale and or offset the values to be between 0 than 5 for the AWG. In this example we create the sinc pulse centered on 2.5 V with a peak value of 4.5 V.

(numpy.sinc(numpy.linspace(-4, 4, 400)) \* 2)+2.5

### Rearranging Sample points:

numpy.roll(AWGAwaveform, shift) Roll array elements by shift points. This will in effect change the relative timing delay or phase of the waveform.

### Functions to extend waveform:

numpy.concatenate((AWGAwaveform, AWGBwaveform,…)) Join a sequence of arrays.

numpy.repeat(AWGAwaveform, repeats) Repeat elements of an array. This will effectively lower the sample rate of the waveform. If repeat is 2 the frequency of the new waveform will be ½ what the original was.

The pad function adds samples to the beginning and end of the array.

numpy.pad(AWGAwaveform, (100, 100), 'edge')

numpy.pad(AWGAwaveform, (100,100), 'maximum')

The first argument is the array variable, next is a list of the number of points to add. In the case of our one dimensional waveforms this is just two values for the beginning and end of the array. The third argument tells the function what values to use to extend the array. How the array is extended can be one of the following:

‘constant’ - Pads with a constant value.

‘edge’ - Pads with the edge values of array.

‘linear\_ramp’ - Pads with the linear ramp between end\_value and the array edge value.

‘maximum’ - Pads with the maximum value of all or part of the vector along each axis.

‘mean’ - Pads with the mean value of all or part of the vector along the axis.

‘median’ - Pads with the median value of all or part of the vector along the axis.

‘minimum’ - Pads with the minimum value of all or part of the vector along the axis.

‘reflect’ - Pads with the reflection of the vector mirrored on the first and last values of the vector along the axis.

‘symmetric’ - Pads with the reflection of the vector mirrored along the edge of the array.

‘wrap’ - Pads with the wrap of the vector along the axis. The first values are used to pad the end and the end values are used to pad the beginning.

### Window functions:

numpy.bartlett(length) Return the Bartlett window.

numpy.blackman(length) Return the Blackman window.

numpy.hamming(length) Return the Hamming window.

numpy.hanning(length) Return the Hanning window.

numpy.kaiser(length, beta) Return the Kaiser window

### Random functions:

Many of the random number functions return arrays of random numbers. Here are a few examples:

numpy.random.standard\_normal(8000)+2.5 will return a 8,000 sample array of random numbers with a normal distribution, standard deviation = 1, centered on 2.5.

numpy.random.uniform(1,4,10000) will return a 10,000 sample array of random numbers with a uniform distribution between 1 and 4.

numpy.random.triangular(1, 2.5, 4, 10000) will return a 10,000 sample array of random numbers with a triangular distribution between 1 and 4, centered on 2.5.