

PCB scope User Manual

Version 1.1.0
December, 2013

1 Introduction

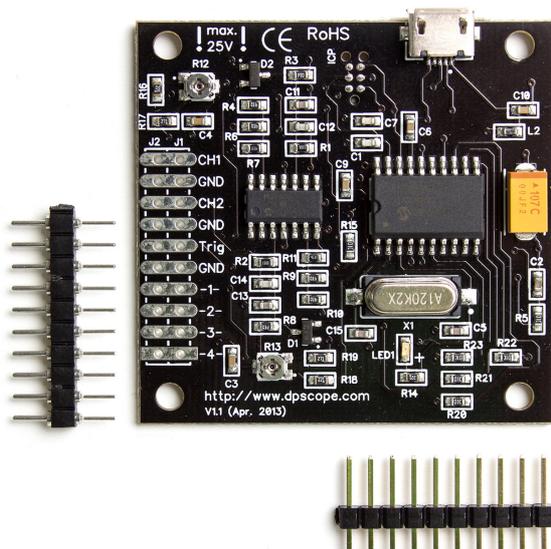
The PCB scope is a low-cost, microcontroller based digital oscilloscope intended for hobby and educational use. It is a low cost PCB version of the 'DPscope SE' and so shares the same circuit and software. If you are new to oscilloscopes, we recommend the following introductory application notes (created by Tektronix, one of the major oscilloscope manufacturers for the professional, high-end market) which will supply all the knowledge necessary to understand this user manual and get the most from your PCB scope:

[The XYZs of Oscilloscopes](#)
[The ABCs of Probes](#)

You can download both documents either from the Tektronix website (<http://www.tektronix.com>) or from the PCB scope website (<http://www.dpscope.com> → PCB scope → Downloads).

If after reading through the documentation you still have questions regarding the PCB scope or want to provide feedback or suggestions for improvement, do not hesitate to contact us at

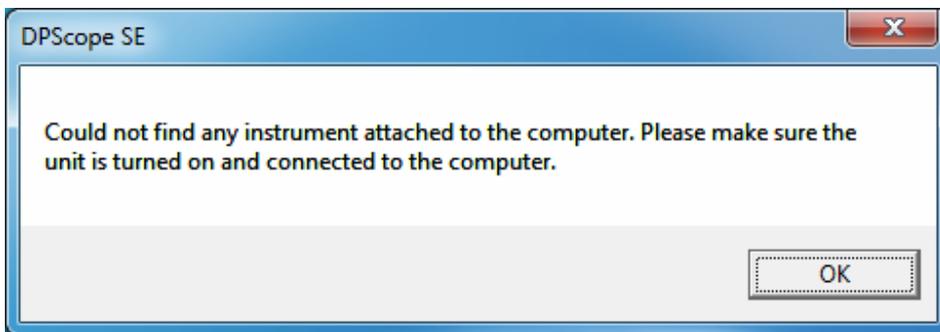
support@dpscope.com



2 Software Startup

After you installed the PCB scope software (see the assembly guide for details) plug in your PCB scope and wait a short time so the computer has a chance to recognize the scope. The PCB scope's power LED on the front panel should blink quickly a few times and then stay solidly on – this is the sign that the scope has successfully powered up. The PCB scope does not need any special drivers since it uses the standard USB HID driver that comes with Windows.

Then launch the PC software (Start → PCB scope → PCB scope). After a few seconds the PCB scope program should appear on the screen. If the program cannot find and connect to the PCB scope you will get a popup (see below) that you need to confirm in order to continue. The software will still start after that, but of course you will not be able to run the acquisition. In this case, close the software, unplug the oscilloscope, plug it in again (you may want to try a different USB port if that does not help), and launch the software again. (This problem is very unlikely to happen. More likely is that you simply haven't attached the scope to your computer).

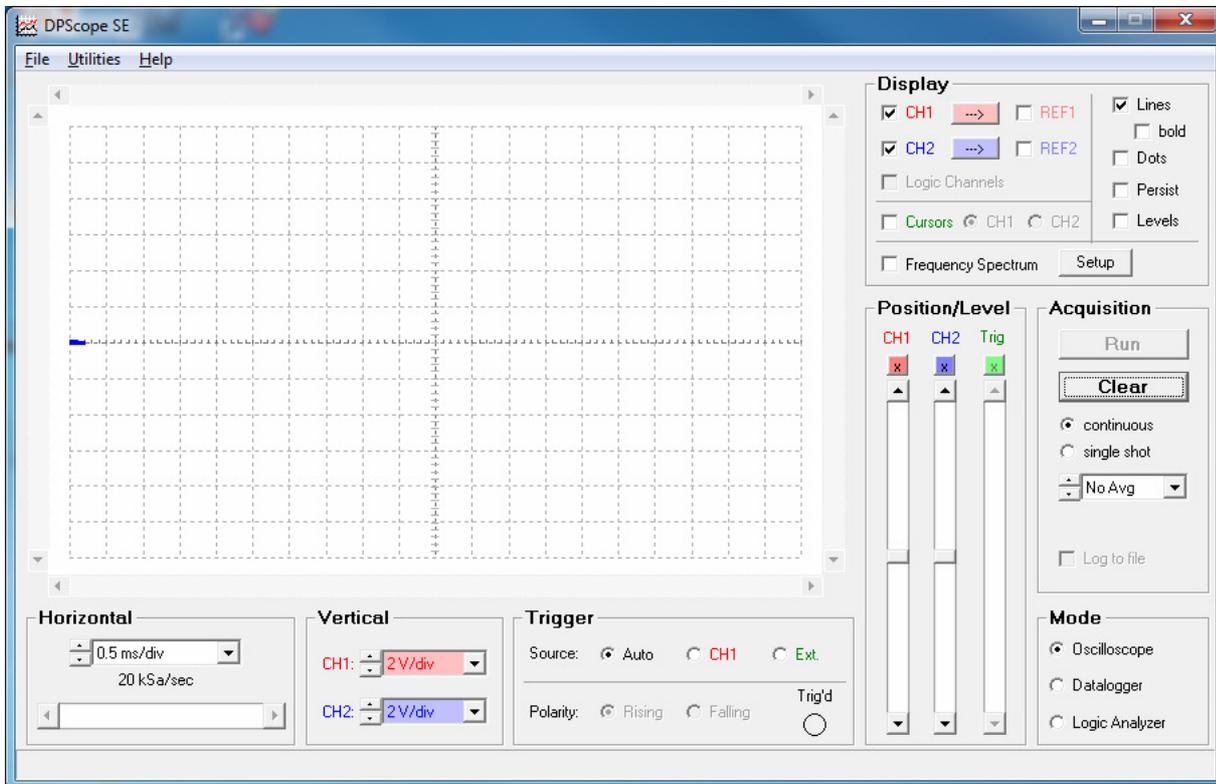


The PCB scope should always be attached to a USB port directly at the computer, or through a powered USB hub. Unpowered USB hubs tend to have excessive supply voltage drop which needs to be compensated for (and which will reduce the scope's maximum voltage range). If in doubt you can check the actual USB supply voltage by selecting Utilities → Check USB Supply. The nominal value is 5V but in reality can be as low as 4.3V.

The **minimum required screen resolution** for the PCB scope software is **800x600 pixels**.

3 The Main Screen

After startup the screen should look as shown below:



The left side of the display is mostly taken up by the waveform display. To the right and on the bottom you can see that the controls are grouped in several functional blocks (we will explain each of them in more detail later):

- **Mode:** Selects the main functionality (oscilloscope, datalogger, or logic analyzer).
- **Display:** Determines how the acquired data is to be displayed.
- **Acquisition:** Main control of acquisition process
- **Position/Level:** Sets the voltage offset for each channel, and the trigger threshold level.
- **Vertical:** Controls the input amplification/attenuation for each of the two input channels.
- **Horizontal:** Controls the time base, sampling speed, and delay.
- **Trigger:** Defines the trigger source.

On the bottom there is a status line that displays the cursor information (right now the cursors are turned off so it is empty), which you can use to make measurements on the displayed waveforms (e.g. to determine the signal amplitude or frequency).

On the top of the screen you find the menu containing three entries, File (to save the displayed waveforms in numerical format, and to save and restore the PCB scope's software settings), Utilities (measurements, DMM display, and supply voltage check), and Help (displays general info about the software).

4 First Acquisition

After startup the software is set up in a way to maximize your chance to see *something* useful on the screen right away: Both scope channels are turned on, the voltage range for both is set to maximum, ground levels centered on the screen, continuous mode and trigger in Auto mode (meaning the acquisition will run without waiting for a trigger event), and so on.

Attach the analog channels to some signal source, e.g. the toggling output of a microcontroller. Press the “Run” button (its caption will change to “Stop”). You should see a waveform on the screen. Since the trigger mode is “Auto”, most likely the waveform will not be stable but rather jump around.

Now play around a bit with some of the controls. Don't be afraid! You can't break the scope no matter what settings you change. The worst that can happen is that you no longer acquire a waveform and/or see it on the screen. If you get completely stuck, simply close down the PCB scope program and start it again.

Move the offset sliders and see how the waveforms follow. Now select “CH1” in the trigger control menu. Move the trigger level slider so the trigger level (indicated by a green triangle on the left border of the waveform display) is somewhere in the middle of the CH1 waveform (the red waveform). That waveform should now be stable (assuming your waveform is periodic). Try switching between rising and falling edge trigger.

Now change the sample rate in the Horizontal control display and see how the waveform changes (it's like zooming in and out on the waveform timing).

In the Display control box, turn the waveforms on and off by clicking on the appropriate checkbox (you can still trigger on a waveform even when it is not displayed). Change the display style to points and/or infinite persistence.

Congratulations – you have just measured your first signal with the PCB scope!

5 General Remarks

The PCB scope is a digital storage oscilloscope (refer to the “XYZs of Oscilloscope” mentioned in the introduction if you are unsure what that is), based on a Microchip PIC18F14K50 microcontroller. Most of the oscilloscope functionality is actually integrated in this single chip – analog-to-digital converters, sample logic, trigger logic and threshold generation, memory, logic analyzer, USB interface. Only the analog frontend (input amplifier/attenuator) consists of external components.

5.1 Acquisition Modes

The PCB scope’s acquisition engine has four different modes (again, refer to the “XYZ of Oscilloscopes” document for further details):

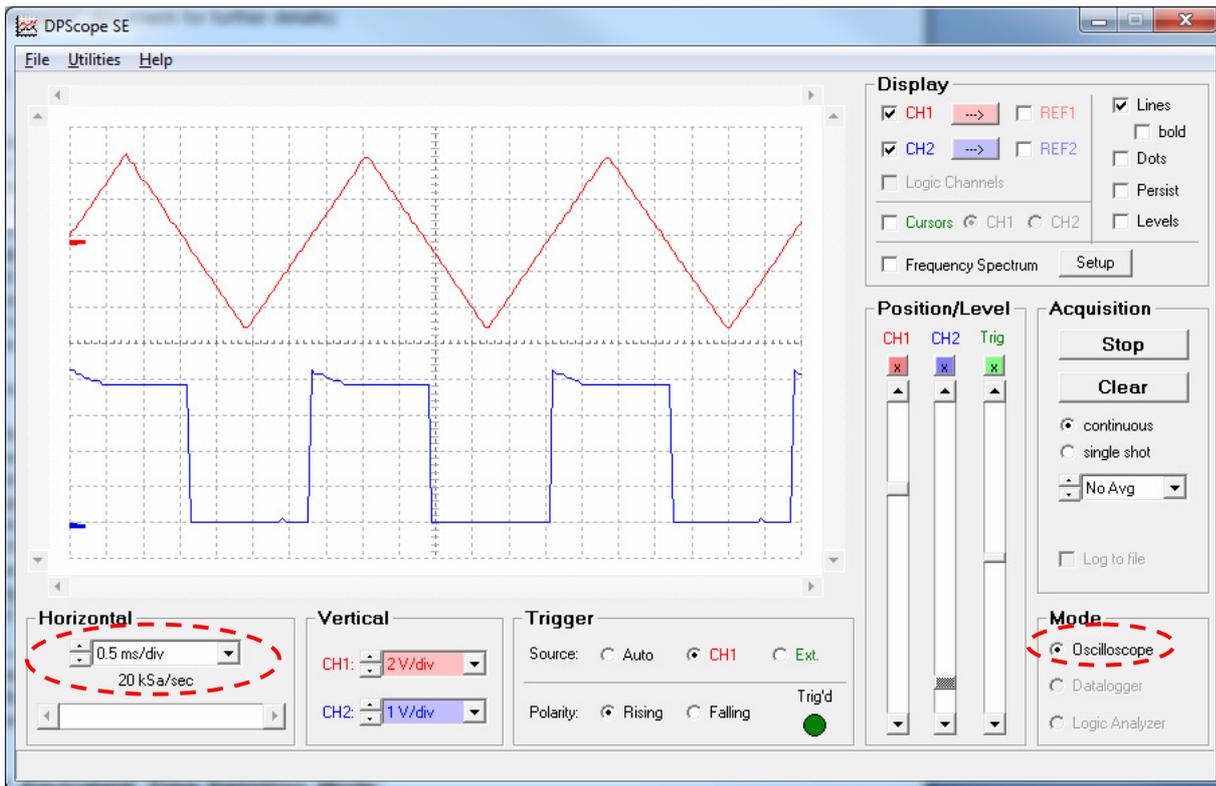
5.1.1 Normal Mode:

Acquisition is started by a trigger event (or automatically right after the previous acquisition if the trigger mode is “Auto”) and samples (digitizes) the signal in real time. This is very similar to how a classic analog oscilloscope operates. Real-time acquisition means the acquisition is very fast (up to approx. 20 records per second on a sufficiently fast computer) and you can capture single-shot events. (Note that the acquisition rate will decrease for slow sample rates – slower than about 5 kSamples/sec or 2 ms/div – because it takes some time to simply capture each full record if only a few samples are taken per second). Also, since this mode uses an interrupt generated by the internal hardware comparator the timing is very tight (i.e. reaction to the trigger event is almost instantaneous), resulting in very low jitter against the trigger – in practice that means there is very little horizontal (timing) “wobble” of the displayed waveform.

This mode has two limitations:

First the maximum sample rate is 20 kSa/sec (20000 samples per second) if acquiring both channels, because the controller has to digitize and store the data as fast as it comes in. The main limitation here is the speed of the built-in analog-to-digital converter; it can go up to about 50 kSa/sec and is shared between the two scope channels. For up to 20 kSa/sec on two channels the acquisition is interleaved, i.e. the scope acquires one sample on CH1, then one on CH2, then one on CH1 again, and so on, at 40 kSa/sec, so each channel gets digitized at 20 kSa/sec. At 50 kSa/sec the acquisition is performed as in an alternating fashion, i.e. after the first trigger only CH1 is captured (full record, 200 points, at 50 kSa/sec), then at the second trigger event CH2 is digitized (again full record).

Second, since the trigger *initiates* the capture process, you can only look at things that happen *after* the trigger unless you can supply a suitable pre-trigger signal (to CH1 or external trigger) that precedes the signal to be captured (on CH1 or CH2) by some margin.



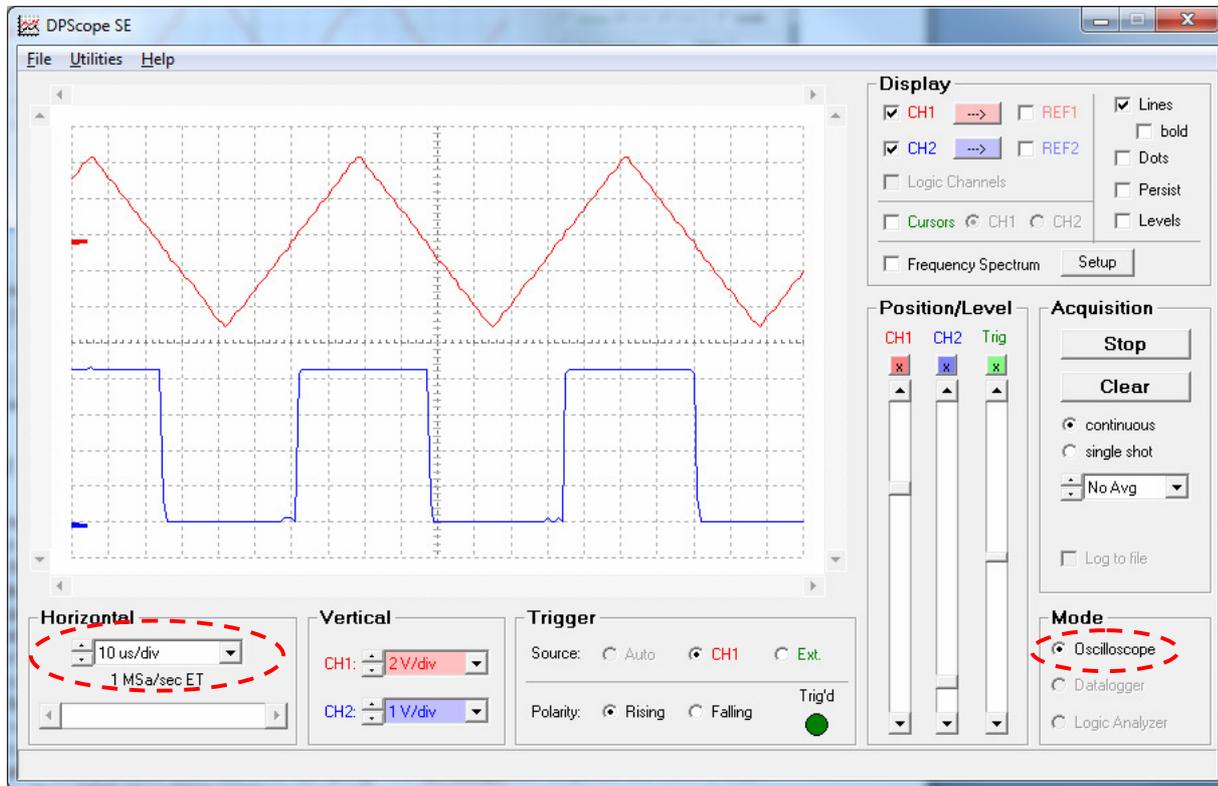
5.1.2 Equivalent Time Sampling Mode:

This mode works similar to Normal Mode in the sense that the acquisition is initiated by the trigger event, so again you can only look at things that happen some time after the trigger. But other than Normal Mode in Equivalent Time Sampling Mode the scope makes no attempt to sample the signals in a single sweep. Instead it waits a defined time after the trigger event and then just captures a single sample on a single channel. Then it waits for the next trigger and captures a second sample on this channel, and so on. For each sample it waits for a slightly longer time after the trigger; this delay can be controlled in increments much finer (down to 500ns or half a μ s) than the smallest real-time sample interval (20 μ s). That way – assuming the waveform is repetitive with respect to the trigger – the sample points of subsequent acquisitions will lie in regular intervals along the repetitive waveform. Plotting them together (suitable shifted by the respective added delay versus the trigger) produces a picture of the signal with finer resolution than possible with real-time sampling. The PCB scope acquires first CH1 in this fashion, then CH2, then CH1 again, and so on.

The main advantage is that you can resolve the signal in much finer increments (up to 2 MSamples/sec, or equivalent-time sample intervals of just 500 nanoseconds) than possible with real-time sampling.

There are a number of tradeoffs connected with equivalent time sampling. First there is lower acquisition speed (because the PCB scope only acquires one sample per trigger event, instead of a full record like in real time sampling), especially noticeable at slower timebase settings and slower trigger rates. Second there is the necessity that the signals be repetitive with respect to the trigger. Thirdly you always need to have a trigger source with fixed timing relationship to the signal to be measured in order to get usable waveforms, otherwise subsequent partial captures will be unrelated timing-wise, and the composite waveform on the screen will look like random noise rather than the

waveform you expect. Also it is obvious that you can't make single-shot acquisitions in this mode (the software allows single shot mode, but here it means that one full waveform – consisting of many single-sample captures – gets acquired and then the acquisition stops). Still, given the high acquisition speed of the PCB scope and the fact that most signals one encounters are repetitive anyway (or can be made repetitive) means these restrictions are far less serious than you might think.

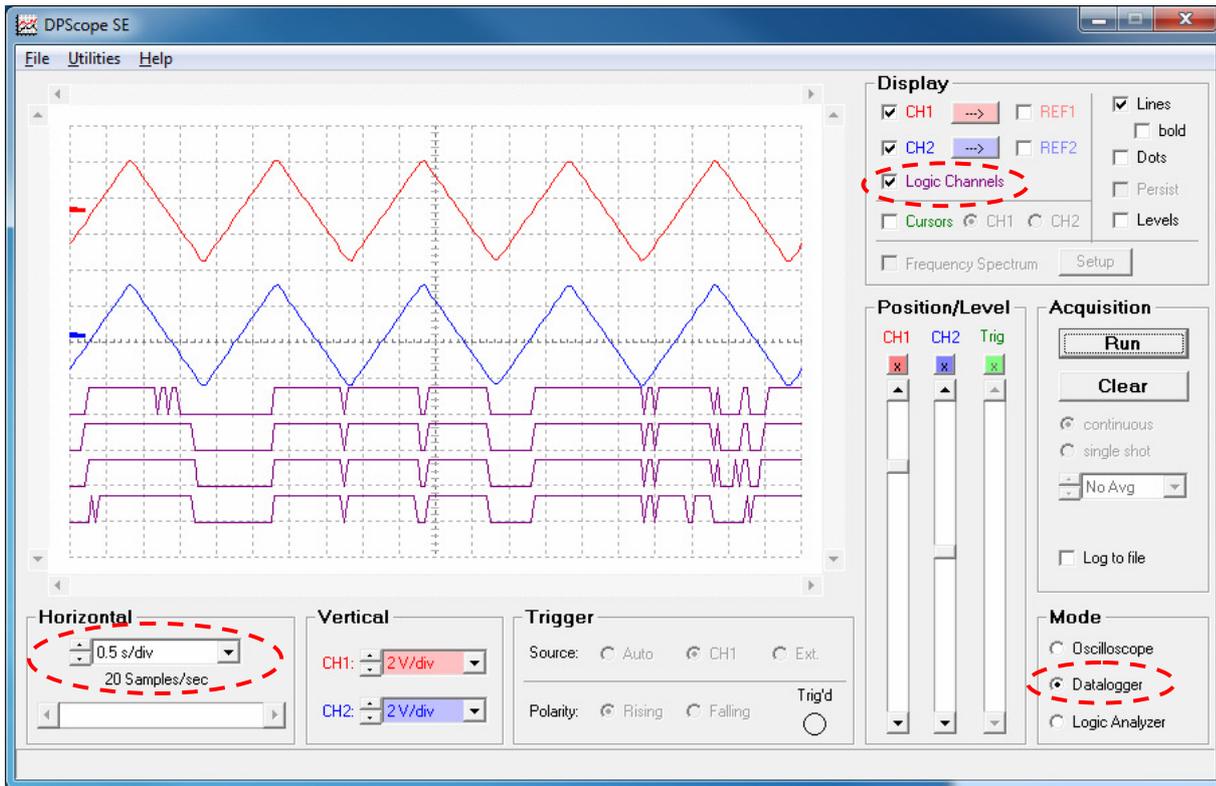


5.1.3 Datalogger (Roll) Mode:

For very slowly varying signals it takes a long time before a full record has been acquired. E.g. at 1 sec/div timebase each acquisition (20 horizontal divisions) takes 20 seconds, plus the time it takes before the first trigger event after the end of the previous acquisition happens – so you may wait close to a second before the screen refreshes just once. In this case it is usually better to display the sampled data as it comes in. This is what the Datalogger mode (often also called Roll Mode) is for: The scope acquires the signals continuously and plots the samples on the screen right away. The waveform scrolls to the left and the new points get added to the right, similar to how a classic datalogger would draw curves on a moving band of paper from a roll (that's where the Roll Mode has its name from).

There is no trigger available in this mode (since we do not want to wait for a trigger event anyway). But you can log the captured data into a text file on your hard drive, so unlike the other three modes you can capture data sets of arbitrary length, only limited by the free space on your hard drive. Clearly this mode only makes sense for relatively slowly varying signals, so the sample rate is limited to 20 samples per second (timebase 0.5 sec/div) or slower.

Since sampling speed is slow and the scope can transmit the data as it comes in, i.e. does not have to store full records in its small internal memory, you can not only measure and display the analog signals on the scope inputs but also the four digital logic analyzer inputs at the same time.

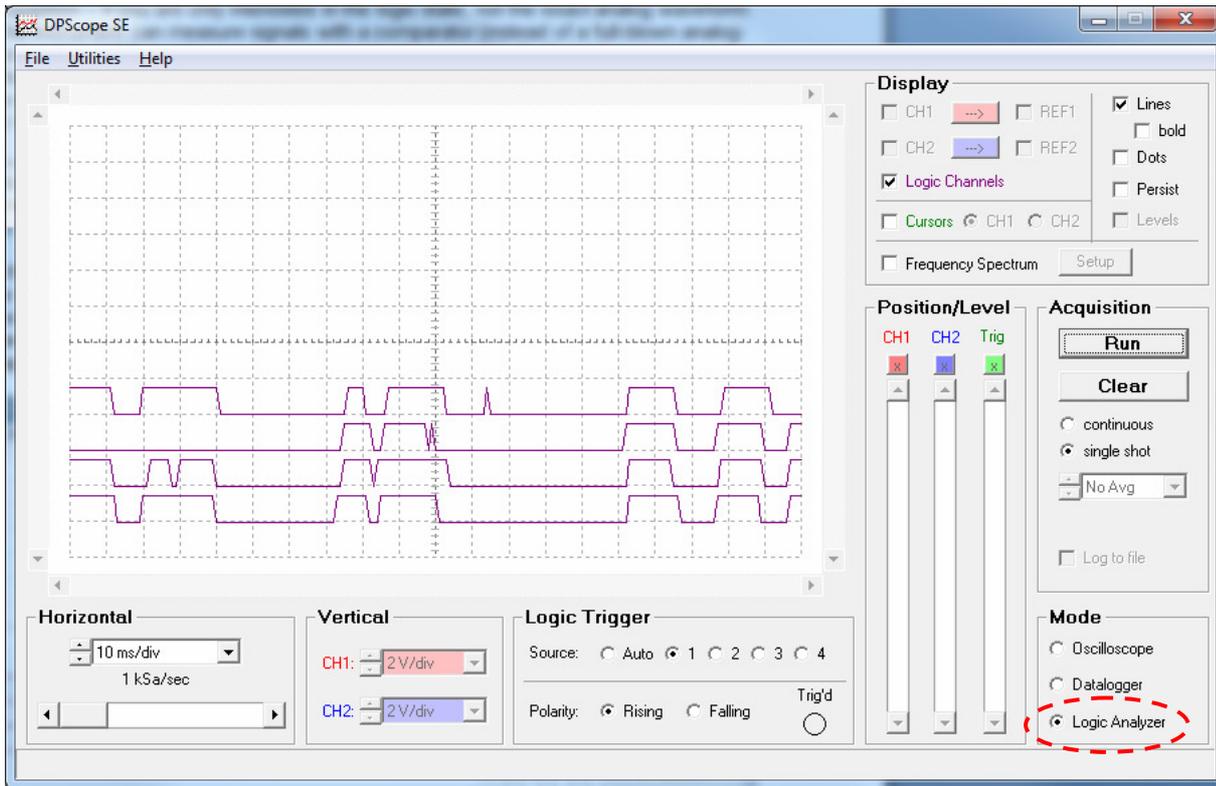


5.1.4 Logic Analyzer Mode:

Many signals you encounter are digital in nature, i.e. they are either high or low. This can greatly simplify acquisition – if you are only interested in the logic state, not the exact analog waveform or levels, the instrument can measure signals with a comparator (instead of a full-blown analog-to-digital converter) and for each sampling instant only store one bit of information per channel. It can do this much faster (up to 1 MSA/sec real time sampling rate for the PCB scope) and store more sample on more channels (up to 850 samples on 4 channels) than in scope or datalogger mode.

Input in this mode is through the logic analyzer connector on the back of the instrument. The input channels are labeled “1”, “2”, “3”, and “4”, respectively. You also need to connect ground (labeled “G”) to the ground of your circuit under test.

Otherwise this mode is very similar to the normal (real time) scope mode: You can trigger on each of the four channels, either on a rising or a falling edge, or have the acquisition in free running mode, and either single shot or repetitive capture. That said, the most common usage of a logic analyzer is in triggered, single shot mode – i.e. set up a trigger, wait for the trigger event to happen, capture one full record on all channels, then stop and analyze the data.



5.2 Probe, Vertical Gain Setup and Trigger Setup:

When you want to acquire a new (unknown) signal, it is best to maximize your chances of seeing *something* before fine-tuning your settings. Choose the widest voltage range available (1 V/div) and set the channel offset to the center of the screen. That way your visible range spans from -10V to +10V which will capture most signals you encounter. For larger signal, use a 1:10 probe to widen the range even further.

If you use a new 1:10 probe, first perform a probe calibration (described in the PCB scope Assembly Guide). 1:1 probes (ie straight wire connections) should not need any calibration because the PCB scope's input is compensated for them by design.

Set the time scale to 0.5ms/div (20 kSa/sec) so you are in real time sampling mode (and triggering is not mandatory).

It is usually best to set the trigger to "Auto" mode – otherwise you'd have to hunt for a valid trigger level on the invisible waveform before the scope displays anything. The waveform timing will not be stable that way (but will move randomly in time), but you will be able to see which voltage range the signal spans. Using the "Persistence" display mode can sometimes be helpful if looking at rare spikes where the signal amplitude is difficult to judge otherwise.

Once you know the range you can adjust vertical gain and offset to center the signal vertically and make it fill a good portion (at least a quarter) of the vertical display range.

To set up the trigger, move it to the center of the trigger channel's signal band (i.e. the green triangle on the left should be somewhere between maximum and minimum level of the signal on CH1) and

turn the trigger on. Switching on the “Levels” display can be helpful there as well. (If you are using the external trigger, which is fed in through the connector on the back of the instrument – pins “T” for the trigger signal and “G” for ground, then the trigger level is fixed at around 1.5V – good for TTL, 5V CMOS, 3.3V CMOS and 2.5V CMOS).

Now you can fine-tune your setup (e.g. use a different sampling mode, change the trigger polarity, etc.). When you expect to measure the same signal again later you may want to save the complete setup to a file (File → Save Setup) that can be loaded back later. That way you do not have to repeat the full setup procedure over and over.

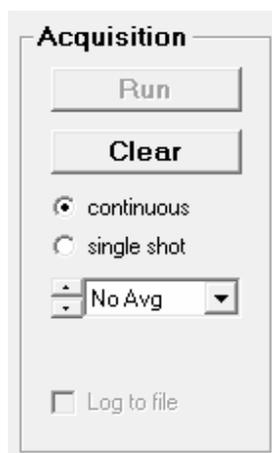
5.3 Undersampling / Aliasing

One very common trap that many beginners (but not only they!) fall into with digital sampling oscilloscopes is so-called aliasing. This effect can happen whenever the sample rate is too slow compared to the signal frequency. To pick a simple example, assume you have a 1 kHz sine wave that you sample with a 1kSample/sec sample rate. In this case each acquisition will hit the same point in the signal period (just one period later each time), so the displayed waveform on the screen will be a flat, horizontal line (i.e. look like DC) rather than a sine wave – clearly wrong! If the signal frequency is a steady 1.01 kHz and the sample rate is 1 kHz each sample will be 1/100th of a period offset on the, and the resulting displayed waveform will be a very convincingly-looking 10 Hz sine wave – this can be very misleading! (Incidentally, this is in fact a variation of the equivalent-time sampling process; it is called “coherent undersampling” and is widely used whenever one can’t get sufficient sample rate to sample incoming high-speed signals in real time).

The solution is of course to sample the waveform with a sufficiently high sample rate. So if in doubt, change the sample rate and see if the waveform behaves “normally” (e.g. when doubling the sample rate the displayed frequency of the signal should not change). And as a general guideline when working with an unknown signal always start looking at it with maximum sample rate and only then change it to something slower.

6 Description of User Controls

6.1 Acquisition



6.1.1 Run/Stop

Starts and stops the acquisition process – data is captured (if trigger event occurs) and displayed on the screen. If “continuous” is selected the acquisition will repeat until you press the button (labeled “Stop” while acquisition is in progress) is pressed. In case of “single shot” acquisition the scope acquires and displays only one record and then stops if “No Avg” (no averaging) is selected; otherwise it will acquire the given number of records to average over and then stop. This means that if you want to do a true single shot capture you need to set the averaging to “No Avg”.

In datalogger (roll) mode single shot and averaging is not available; in this case this button simply starts (and stops) the continuous acquisition.

6.1.2 Clear

Clears (deletes) the currently stored traces. One typical use is when averaging is selected and the measured waveform has changed, as this clears out the old information so the waveform only consists of an average of the new (changed) signal.

6.1.3 Continuous / Single Shot

Selects whether the PCB scope shall continuously acquire new waveforms, or only acquire once and then stop. Continuous acquisition is useful for repetitive waveforms when you want to have a “real time” image of the signal and to see if or how the signal changes over time. Single shot capture is useful e.g. for event that happen only once (e.g. a sudden pulse), or when you want a steady picture to take measurements on.

6.1.4 Averaging

Selects how many waveforms to average over (1, 2, 5, 10, 20, 50, or 100). Averaging reduces random noise on the waveform, resulting in a much cleaner picture that allows you to see small details that may otherwise be hidden under the noise. For averaging to make sense you need to have a basically stable, repetitive waveform, so you will need to turn the trigger on (i.e. auto mode won't work because the sample timing would be asynchronous to the signal period).

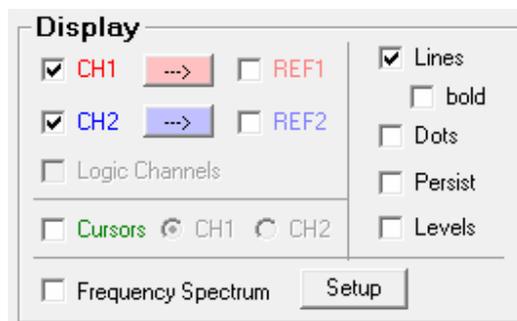
Note that the averaging process used is not a simple arithmetic average over the given number of samples; instead it uses exponential weighing, i.e. earlier acquisitions have less weight than the more recent ones. This is the digital equivalent to an analog R-C low-pass filter (but unlike a filter in your signal path it will *not* degrade your signal bandwidth or rise time; it will just filter out fast changes from one capture to the next, but not fast changes along the waveform).

Averaging is only available in scope mode, not in datalogger mode or logic analyzer mode.

6.1.5 Log to File

This function is only available in datalogger mode. It allows you to write all the captured samples directly into a file. This way you can record signals of arbitrary time spans and with an arbitrary number of sample points, no longer restricted to the PCB scope's 200 points per channel record length.

6.2 Display



6.2.1 CH1 / CH2

Turns the display of scope channel 1 and 2 on or off, respectively. Note that this does not affect the signal capture itself, meaning e.g. you can still trigger on CH1 even when it isn't displayed.

6.2.2 REF1 / REF2

Turns the display of reference waveform 1 and 2 on or off, respectively. To copy the currently displayed waveform (CH1 or CH2, respectively) into the respective reference waveform you use the appropriate "→" buttons.

Reference waveforms are very helpful e.g. to judge small signal changes by comparing the stored waveform to the displayed waveform.

Another important application is when you want to compare more than two signals; one example would be an SPI bus consisting of clock, data, and chip select. In this case you can first set up the PCB scope so CH1 looks at the chip select and CH2 looks at the clock signal, and it triggers on the falling edge of the chip select signal. Capture both signals with single shot mode, and save CH2 to REF2. Now connect CH2 to the data line instead, and repeat the capture. You now have all three signals (clock, data, chip select) on the screen at the same time. (Note that if the signals are digital in nature it may be easier to use logic analyzer mode instead where you can capture up to 4 signals concurrently).

6.2.3 Lines

Checking this option will cause the scope to connect the captured data points with lines. This makes the actual waveform much easier to see, especially when there are fast changes (which would make subsequent points fall on places far away on the screen). Check the "Bold" checkbox to make the lines wider, e.g. to be able to see them at a distance.

6.2.4 Dots

Checking this option will cause the scope to indicate the actual sample points with small circles. This can be very useful if you want to check if your sample rate is truly high enough for an accurate waveform display.

Either lines or dots (or both) must be turned on, as unchecking both will result in an empty display which could be confusing to the user; thus the PCB scope enforces this automatically.

6.2.5 Persist

Normally the screen is cleared whenever the scope has a new acquisition to display. Checking this option will make the scope keep all waveforms on the screen until you either press "Stop" and then "Start" again, or until you press "Clear". This allows you to see over which range the waveform is changing; important cases are measurement of peak-to-peak noise or timing jitter. This is also a great tool to capture rare glitches (sudden waveform aberrations) that you may miss otherwise because without persist mode they may disappear from the screen again so quickly that you can't see them. When persist mode is turned on however even a single such event will remain easily visible.

Finally, this mode is useful to generate so-called "eye diagrams" of digital data streams.

6.2.6 Levels

Turns level markers on and off; while the levels (CH1 and CH2 offsets, and the trigger threshold) are also indicated with colored triangle markers on the left edge of the waveform display, sometimes it is helpful to have rulers across the whole screen.

6.2.7 Cursors, CH1, CH2

Turns the measurement cursors on and off, respectively, and selects which channel (CH1 or CH2) they refer to.

The cursors are the solid horizontal and vertical lines in the waveform display which appear whenever you check the "Cursors" checkbox. You can move them with the slider bars on top/bottom and left/right of the waveform display. Their current level and time positions as well as the difference in their positions are shown in the status line on the bottom of the scope window:

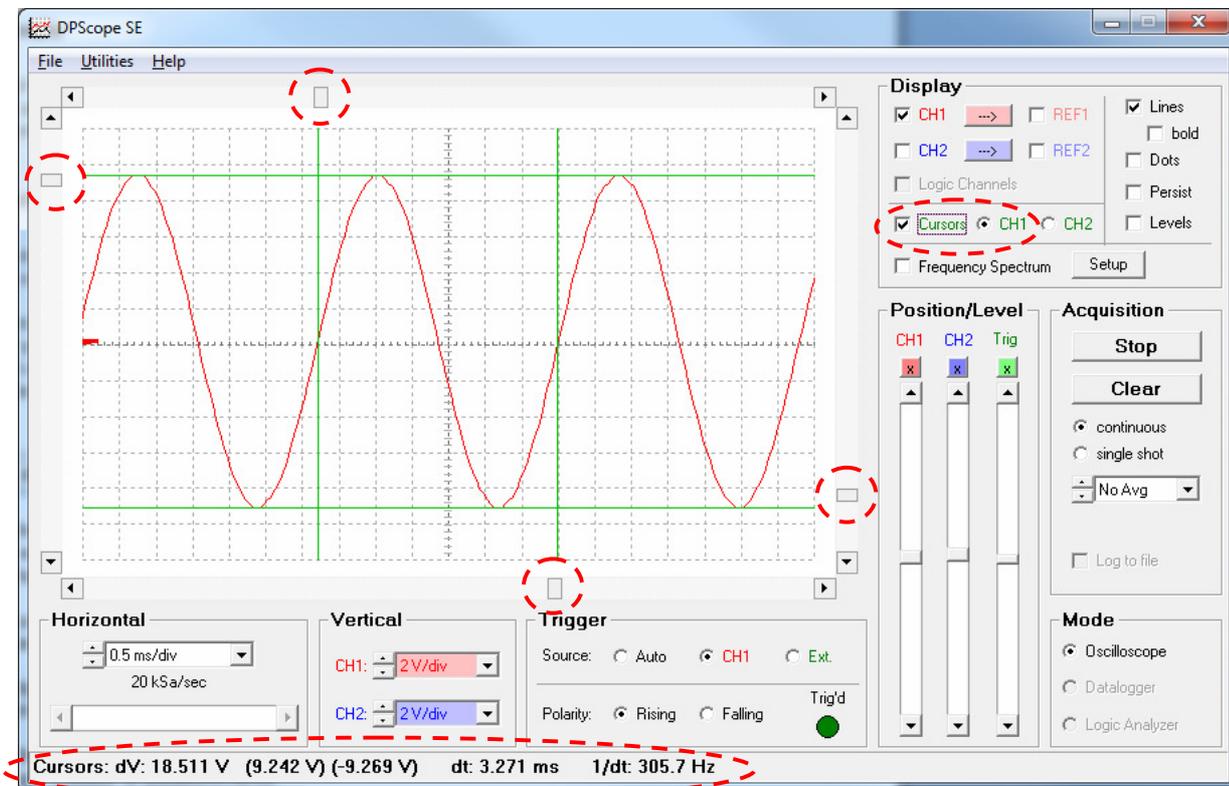


Because the voltage scales can be different between CH1 and CH2 you need to select which of these two waveforms the cursors shall refer to. (The time scale is the same for both so this selection has no bearing on timing measurements, i.e. on the vertical cursors).

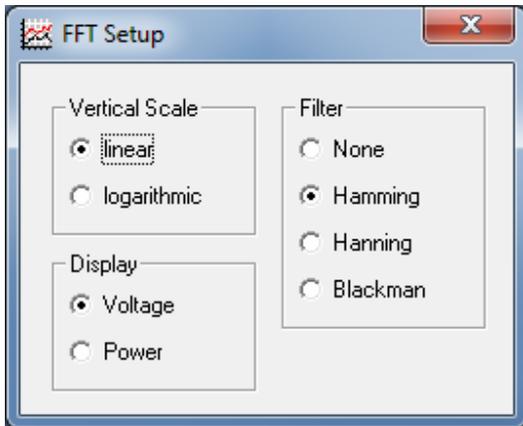
With the cursors you can perform many different measurements on the displayed waveforms, for example:

Levels: amplitude, high and low level, average level, overshoot and undershoot
Timing: period, frequency, delay, phase, rise and fall time

Note that the PCB scope also offers a set of fully automated measurements. You can access them through Utilities → Measurements.



6.2.8 Frequency Spectrum / Setup



Checking the “Frequency Spectrum” option will change the display to frequency domain: The oscilloscope performs a real-time FFT (Fast Fourier Transform) on each captured record, which shows the frequency components contained in the acquired signals. This is a highly useful mode to analyze periodic signals or to find small periodic components (e.g. periodic noise from the power line) hidden in the main signal – it is much easier to see an isolated 60 Hz component (spectral line) in a frequency plot than it is to see it among all the random noise riding on the signal.

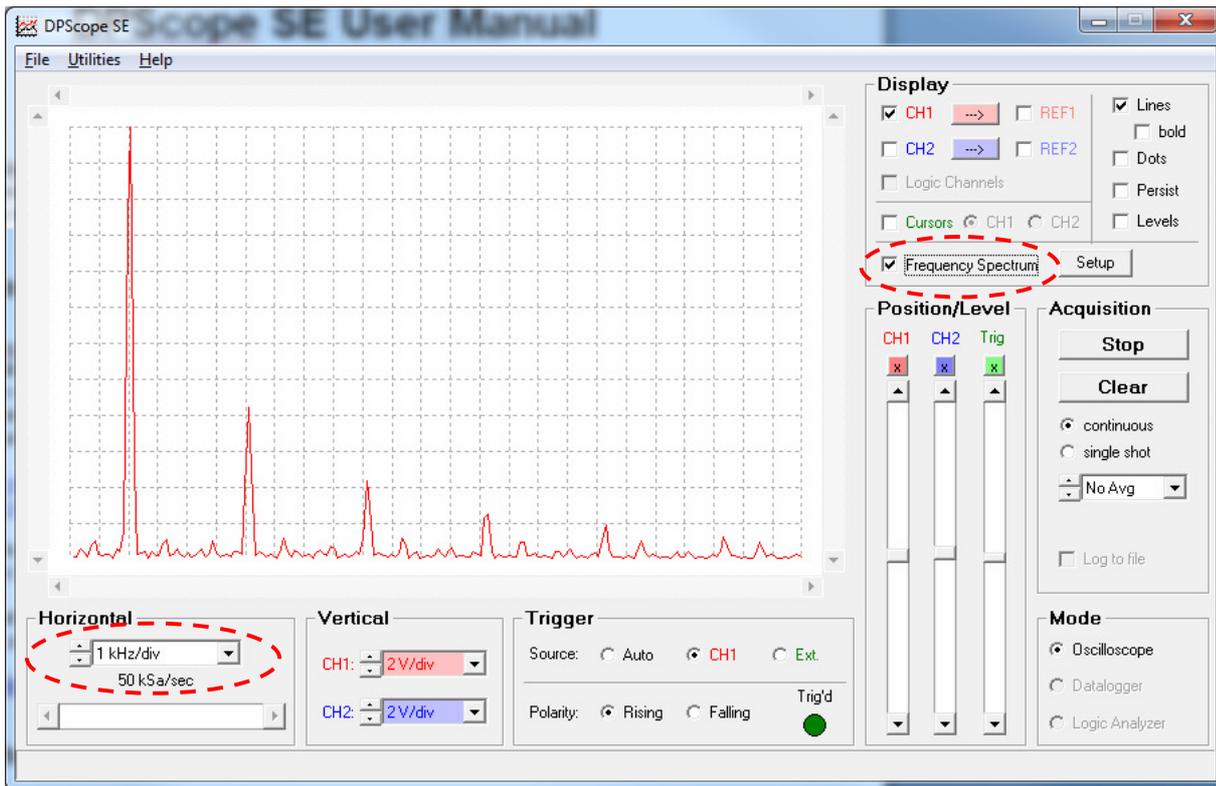
Note that in frequency spectrum mode the scope alternates the signal capture between channels, i.e. only captures one channel at a time. It does that so it can acquire a longer data record (410 points), which results in finer frequency resolution. You can still freely choose which channel (CH1 or external trigger input) to trigger on, i.e. also trigger on a channel when it is not displayed.

It is usually easiest to first set up the waveforms in normal (time domain) mode and only then change to frequency spectrum mode. Also, in most cases frequency spectrum mode works fine in “auto” trigger mode (as long as the waveform is periodic), and with sample rates low enough so a few signal periods are displayed on the screen. (However, make sure that your sample rate is at least 3-4 times your signal period, otherwise aliasing will occur, resulting in erroneous frequency domain pictures).

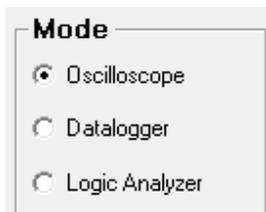
You can use the cursors to make measurements of frequency and relative power. Note that the sample rate box changes to frequency units (Hz and kHz).

Pressing the “Setup” button brings up a small panel where you can modify the settings used for the conversion: Linear (V) vs. logarithmic (dB) scaling, voltage vs. power, and the filter type to use. (The filters reduce artifacts caused by the finite length of the data set. The available types from top to bottom – none, Hamming, Hanning, and Blackman – result in increasingly accurate amplitudes for the price of increased spectral line width).

At the moment frequency spectrum mode is only available in real-time sampling mode, i.e. for sample rates up to 50 kSa/sec – well sufficient for audio range (the sample rate must be more than twice the highest frequency of interest – traditionally 20 kHz for audio).



6.3 Mode



6.3.1 Oscilloscope / Datalogger / Logic Analyzer

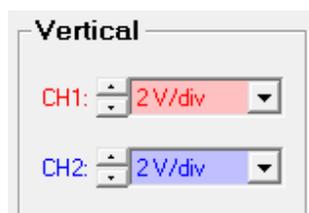
In oscilloscope mode the PCB scope will always acquire a full data set (200 points per channel) and store it in its internal memory before transmitting it to the computer for display. This enables very fast sample rates (up to 50 kSa/sec real time and 2 MSample/sec in equivalent time) because there is no limitation from the transmission speed between scope and computer. The downside is that the record length is limited to about 200 points per channels, and for very slow sample rates the time between subsequent transfers becomes larger (because it takes a while to capture 200 points at a slow rate).

Datalogger (roll) mode on the other hand is limited to relatively slow sample rates (20 samples/sec maximum, which corresponds to 0.5 sec/div), but you can have the PCB scope the captured data directly into a text file and thus have virtually unlimited storage. It also allows to log the four digital channels together with the two scope channels.

In Logic Analyzer mode you can capture the digital states on four channels (connected to the logic analyzer header at the back side of the instrument) with sample rates up to 1 MSA/sec. Other than that it works very similar to oscilloscope mode.

You can change modes only when the acquisition is stopped.

6.4 Vertical



6.4.1 Scale

The vertical gain boxes set the input amplifier gain, so you can look at very small (Millivolts) as well as quite large (several Volts) signals. The maximum signal range (in the 2V/div setting) that can be displayed is from -24V to +24V. Never apply signals exceeding this range to the scope input – doing so risks damage to the PCB scope as well as to your computer.

The scaling is displayed in V/div (Volts per division) or mV/div (Millivolts per division), so e.g. at 2V/div each vertical unit (interval between two horizontal dashed grid lines) corresponds to 2 Volts.

6.4.2 Probe Attenuation

The probe attenuation settings (“1:10” checkboxes) do not have any bearing on the actual amplifier setup, but rather tells the scope software how it must scale the measured data on the screen. A 1:10 probe reduces the signal by a factor of 10, which allows you to measure much larger signals than would be possible with a 1:1 probe. The option to use 1:10 probes is only available in the upper range settings (0.5, 1 and 2 V/div) – in the lower ranges they would not make sense anyway since you could simply use a 1:1 probe/wire links and change the gain setting accordingly.

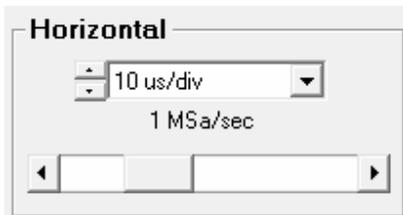
Important Warning: While the scope can display signals between -240V and +240V with a 1:10 probe (which reduces these signals to a range between -24 and +24V), you must exercise extreme caution when working with such large signals:

- As a rule of thumb, anything that exceeds 20V is potentially dangerous. If you accidentally touch the source you can seriously hurt yourself (or even die).
- To avoid damage to yourself, the PCB scope or your computer, always verify that the probe is truly set to 1:10 mode before contacting the circuit.
- Note that AC voltages (e.g. line voltage) have peak values that exceed the rated (effective, RMS) voltage value by a factor of $\sqrt{2}$ or about 40%. In other words a 110V AC signal oscillates between -156V and +156V, while 220V AC ranges from -311V and +311V. The

latter will thus exceed the PCB scope's input range even with a 1:10 probe. (There are 1:50 and 1:100 probes available that would allow to measure such signals).

Note: The PCB scope is intended for measurements on low-voltages only and we explicitly decline any responsibility for damage or injury resulting from work with high voltages (that is, any voltage exceeding 25V peak) – you do this at your own risk. Should you choose to perform such measurements you confirm that you have the necessary knowledge to do so and will always observe the required safety precautions.

6.5 Horizontal



6.5.1 Sample Rate

Determines the sample rate, i.e. how many times per second the signals are measured. The available range is dependent on the acquisition mode:

- Normal Mode: 1 sec/div to 0.5 msec/div (10 Sa/sec to 20 kSa/sec) – interleaved
0.2 msec/div (50 kSa/sec) – CH1 and CH2 alternating
- Equivalent Time Sampling Mode: 0.1 msec/div to 5 µsec/div (100 kSa/sec to 2 MSa/sec)
- Frequency spectrum: 0.5 Hz/div to 1 kHz/div (10 Sa/sec to 50 kSa/sec)
- Datalogger Mode: 0.5 sec/div to 1 hr/div
- Logic Analyzer Mode: 1 sec/div to 10 µsec/div (10 Sa/sec to 1 MSa/sec)

6.5.2 Horizontal Position

This slider is only active in Logic Analyzer mode, because the full record (850 samples per channel) is longer than what fits conveniently on the screen (200 samples).

6.6 Trigger / Logic Trigger

Oscilloscope mode:



Logic Analyzer mode:



This panel changes somewhat between oscilloscope mode and logic analyzer mode to account for the difference in available trigger sources.

6.6.1 Auto / CH1 / Ext. and Auto / 1 / 2 / 3 / 4

The Auto / CH1 / Ext. radio buttons let you select what the PCB scope shall trigger on. A trigger is basically a condition that determines when the oscilloscope starts the data acquisition. For a repetitive signal you use this to obtain a steady picture of the signals. For a one-time signal it allows you to have the scope wait until the event of interest occurs before it starts to capture the signals.

If set to “Auto” the PCB scope will be free running (constantly collecting data) and not wait for any special event before capturing data. This is useful when you want to look at an unknown signal, because you will always see *something* on the screen (the scope will not wait and not display anything just because some trigger condition is not fulfilled).

If set to “CH1” the scope will use the signal on scope CH1 as the input to its trigger circuitry. In this case you can use the trigger threshold slider (see explanation of the Position/Levels” frame) to determine the trigger threshold level.

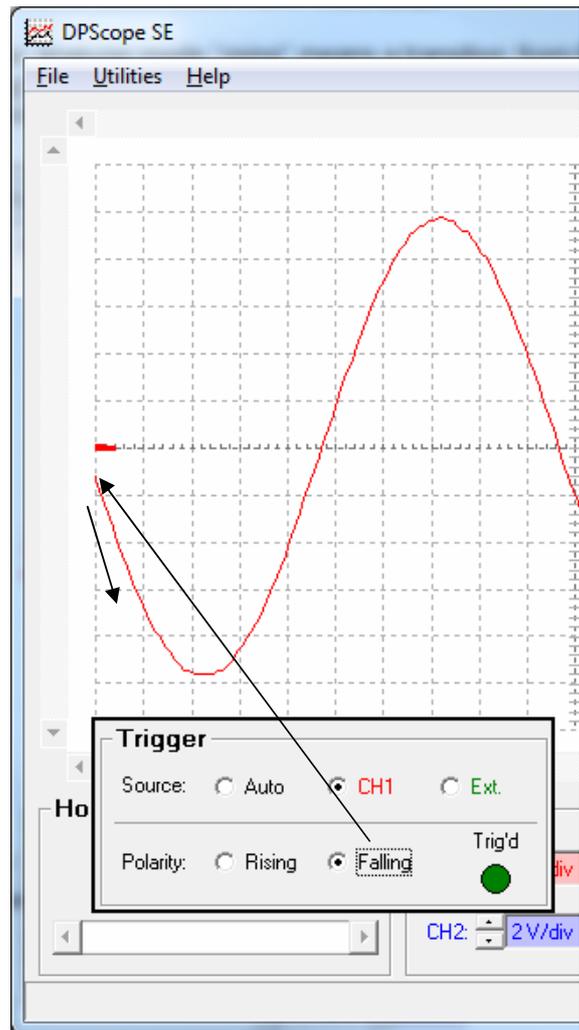
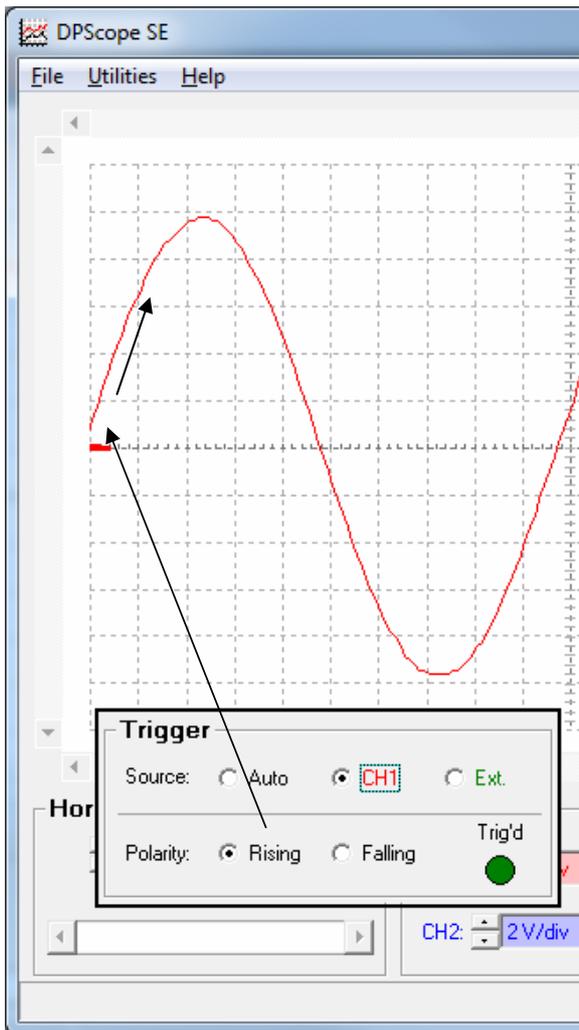
If set to “Ext.” then the scope triggers on the signal fed into the external trigger input (pin labeled “T” for the trigger signal and “G” for ground. The trigger threshold here is fixed at around 1.5V – good for TTL, 5V CMOS, 3.3V CMOS and 2.5V CMOS).

In logic analyzer mode you can either force constant acquisition (“Auto”) or trigger on any of the four digital input channels (1, 2, 3, or 4).

6.6.2 Rising / Falling

The trigger polarity further specifies the required trigger condition. “Rising” means the signal must cross the threshold from below and rise above the threshold (i.e. a positive slope) to cause a trigger. “Falling” means the signal must cross the threshold from above (i.e. a negative slope). The pictures below show this in more detail. Note that because of the noise reject feature (see below) you will not see the actual trigger instant because the acquisition of the sample data only starts a short moment after detection of the trigger.

In logic analyzer mode “rising” means a transition from 0 (logic low) to 1 (logic high), and “falling” is the opposite (transition from 1 to 0). One could conceptionally think of the threshold as being at 0.5 (of course there is no such digital value!).



6.6.3 Trig'd

This indicator (“triggered”) shows if the scope is getting trigger events. Each trigger will cause it to turn green for a short amount of time. (If it stays green that means the scope is getting triggered at a fast rate). This can be helpful e.g. when the signal has very little noise so it is hard to tell if the acquisition is stuck (Trig' is grey) or if the scope is actually constantly acquiring new data but each acquisition simply looks the same as the previous one.

6.6.4 Noise Reject

In scope mode if there is noise on the signal (and there is always some amount) this can cause false triggering. E.g. if triggering on a rising edge, the signal may actually be falling at a given instant, but a short little positive noise spike right when the signal has crossed the trigger threshold from above will look like a rising edge to the comparator and cause a trigger even though it is not a true positive-going edge. As a result the oscilloscope will effectively trigger on a falling edge instead of a rising edge.

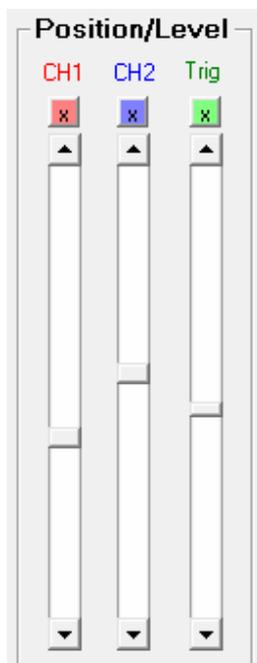
To avoid this situation the PCB scope automatically performs “Noise Reject” by requiring that the signal must remain above the threshold (in the case of a rising edge trigger; below for falling edge) for

at least one sample interval ($1/10^{\text{th}}$ of a division (i.e. 5 samples). If it fails to do so (e.g. because the edge really was just a short noise spike as discussed above), the trigger event will get suppressed and the PCB scope will keep waiting for a trigger.

Note that the absolute duration required depends on the sample rate (time base setting). This makes sense since at slow sample rates you usually aren't interested in changes (and doesn't want to trigger on) things that are too short to observe anyway.

This feature is especially valuable for slowly varying signals. Note however that if your signal period becomes too short the noise reject feature may prevent the PCB scope from triggering at all. In this case increase the sample rate so the period becomes a larger fraction of the screen width. (For fast sample rates the minimum duration is actually a bit longer than one sample interval because the processing overhead is no longer negligible).

6.7 Levels



6.7.1 CH1 / CH2 (Channel Positions)

Sets the channel offset for input channel 1 and 2, respectively. This lets you control where ground (zero Volts) is displayed on the vertical axis. First this expands the available voltage range for a given vertical resolution, e.g. at 0.1 V/div you could for example cover the range from 0V to + 1.2V (12 vertical divisions are visible), or alternatively from -1.2V to 0V.

A second use for the offset is to move the two waveforms (CH1 and CH2) apart on the screen so they are easier to discern, even though in reality they cover similar voltage ranges (e.g. 0 to 5V).

The zero position for each channel is indicated on the left side of the waveform display with a colored marker (red for CH1, blue for CH2). If "Levels" is turned on in the display control frame, two dashed

horizontal lines of same colors are drawn across the display as well at that vertical position. See picture below.

To reset the zero position back to the middle of the display press the respective “x” button.

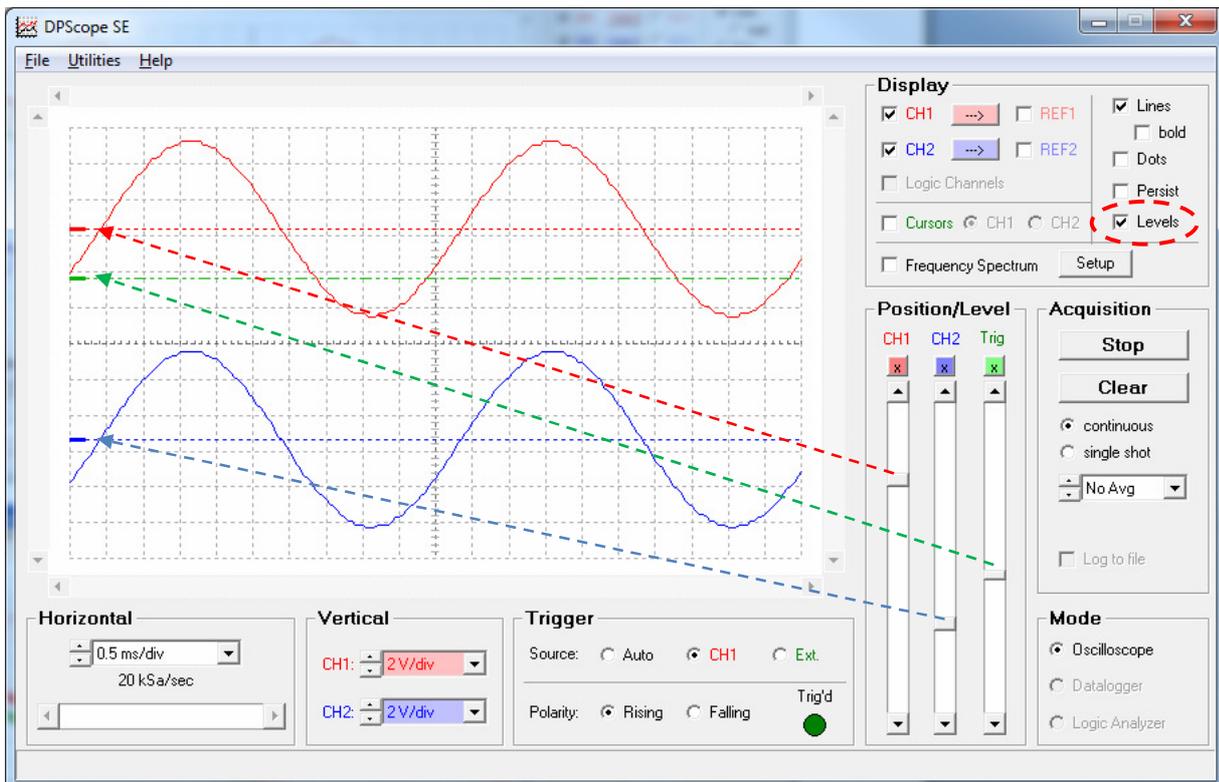
6.7.2 Trig (Trigger Level)

This slider controls the trigger threshold. If the PCB scope is set up to trigger on CH1, a trigger event will occur whenever the signal crosses this threshold with the desired polarity (rising or falling edge, respectively). If the trigger source is the external trigger input then this slider is not active (because the threshold on the external trigger input is fixed at about 1.5V). Same if the trigger mode is “Auto” since in this case the PCB scope does not wait for a trigger event at all.

The zero position is indicated on the left side of the waveform display with a green marker. If “Levels” is turned on in the display control frame, a green dashed horizontal line is drawn across the display as well at that vertical position.

Usually the best approach with an unknown signal is to first display it with trigger in “Auto” mode. That way – although the waveform on the screen will not be stable – you can at least judge the voltage range that the signal spans, and set the trigger level marker somewhere within that range before setting the trigger source to the appropriate channel.

See picture below.



6.8 File Menu

6.8.1 Load Setup

Recalls a previously saved setup (all settings like acquisition mode, vertical scales, time scale, trigger setup, etc.) from a file.

6.8.2 Save Setup As

Saves the current setup (i.e. all settings like acquisition mode, vertical scales, time scale, trigger setup, etc.) to a file.

6.8.3 Export Data

Exports the currently displayed waveforms to a text file. File format is standard CSV (comma separated values) format, which you can open and process in Microsoft Excel or any standard text editor (e.g. Notepad).

- In oscilloscope mode the file contains both scope waveforms (CH1 and CH2) as well as both reference waveforms (REF1 and REF2).
- In Logic Analyzer mode the file contains the four digital (logic) channels.
- In datalogger mode the file contains the two analog (oscilloscope) channels as well as the four digital (logic) channels.
- In datalogger (roll) mode it is usually more useful to select "Log to file" instead, because this will write all the data to the file, while "Export Data" only writes the latest screen interval.

6.8.4 Exit

Closes the PCB scope application.

6.9 Utilities Menu

6.9.1 Measurements

While it is possible to make measurements on the waveforms using the cursors, the PCB scope also offers a faster and much more convenient method: Fully automated measurements. Selecting Utilities → Measurements brings up the Measurement panel. Here you can select on which channel(s) (CH1 and/or CH2) the software shall take the measurements, and also which measurements to take. The values get updated after every acquisition. If you have already acquired the signal and only afterwards bring up the Measurements panel, simply press "Update Values" to perform the calculation based on

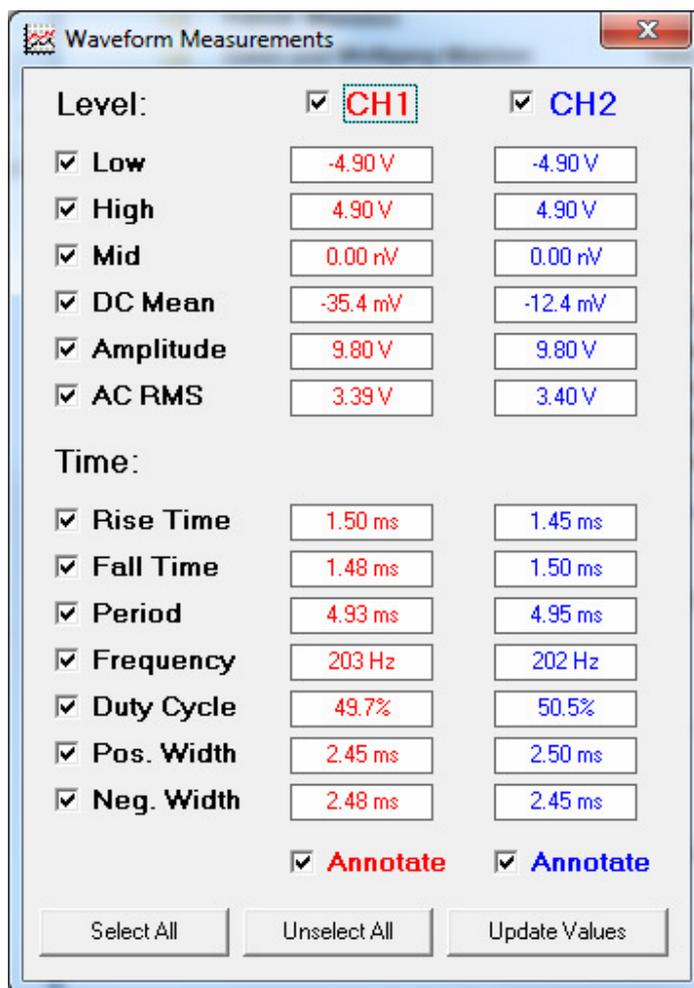
the currently displayed waveforms. (Do not change vertical or horizontal settings like gain or offset between the acquisition and the calculation).

The following measurements are available:

Level: low, high, midlevel, DC mean, amplitude, and AC RMS

Timing: rise time, fall time, period, frequency, duty cycle, positive width, negative width

You can turn the waveform annotations on and off for each channel. If turned on (and the channel is displayed) they will show the low, mid and high level that the software uses. This can be a valuable tool to troubleshoot your results.



Important notes:

- In order for the timing measurements to work, at least one full signal period needs to be displayed on the screen.
- The time measurements employ linear interpolation between the actual samples to improve resolution and accuracy of the results.
- Averaging can significantly improve the accuracy, especially for the level measurements.

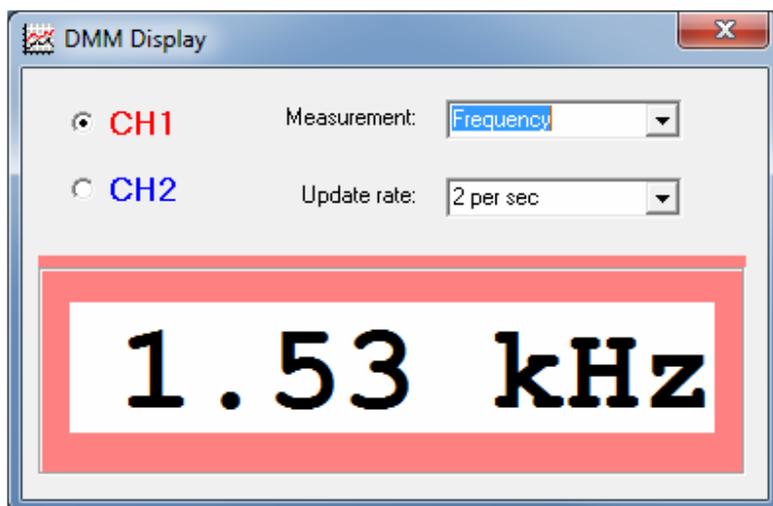
- Measurements slow down the acquisition process somewhat. Turn them off (close the measurement panel) for maximum screen update rate.

Short description of each measurement:

- **Low:** lowest value (minimum) on the screen.
- **High:** highest value on the screen.
- **Mid:** Midpoint between low and high, $\text{mid} = (\text{low} + \text{high}) / 2$
- **DC mean:** average voltage level, calculated over all points on the screen
- **Amplitude:** difference between high and low, $\text{amplitude} = \text{high} - \text{low}$
- **AC RMS:** RMS value calculated with the mid level taken as the reference

- **Rise time:** time to rise from 10% to 90% of the swing (low = 0%, high = 100%)
- **Fall time:** time to fall from 90% to 10% of the swing (low = 0%, high = 100%)
- **Period:** time between subsequent rising edges crossings of the 50% point (mid level) (or falling edges if there aren't two rising edges on the screen)
- **Frequency:** $1/\text{period}$
- **Duty cycle:** time spent above 50% during one period, relative to the length of the period.
 $\text{duty_cycle} = \text{pos. width} / \text{period}$
- **Pos. width:** time spent above 50% during one period
- **Neg. width:** time spent below 50% during one period

6.9.2 DMM Display



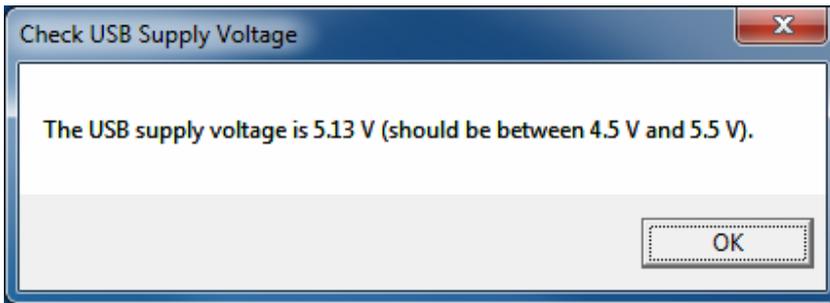
While the automated measurement panel is convenient, sometimes you may want to observe only a specific value. This could be e.g. to trim the frequency or the amplitude of an oscillator. Chances are you do this at some distance from the screen so it is important that the display is large enough to be observable without being right in front of the computer.

This is the purpose for the DMM Display window (DMM stands for **D**igital **M**ultimeter, because this mode has very similar functionality to such an instrument). There you can select a single measurement (out of all the measurements available in the main measurement panel) on one channel, which will then show up in very large digits.

The update rate lets you select how often the value will get updated (if the acquisition rate is slow then of course the update rate will be slower than selected, but it will never be faster). This makes the numbers easier to read since they don't jump around wildly.

6.9.3 Check USB Supply

Measures and displays the actual supply voltage from the USB port. This enables a quick check if the port voltage is close enough to its nominal value (5 Volts) – the actual value can be as low as 4.3V.



6.10 Help Menu

6.10.1 About

Here you can find version and build of your software and of the scope's firmware (the latter only if there is actually a scope connected) – please always include this information when submitting bug reports or support requests. The panel also shows the support contact info.



7 Appendix: Probe Calibration

As long as you use simple wires or 1:1 probes you do not need to perform any compensation since the PCB scope is compensated for them by design. This is different with 1:10 probes – some adjustment may be necessary to obtain correct waveform display and maximum bandwidth.

To perform this compensation connect the probe to the PCB scope and hook it up to a square wave signal (frequency about 1 kHz). This could e.g. be a function generator (set to produce a square wave) or a pin on a microcontroller that gets toggled 2000 times per second.

- Make sure the scope probe is set to 1:10 attenuation (not 1:1)
- Display the signal on the screen.
- Trigger turned on so the waveform is stable.
- Adjust the vertical scaling and offset so the signal spans about half the display height.
- Adjust the timebase so you see roughly two periods of the calibration signal.

On the probe there is typically a small screw (either at the probe head itself, or at a small block at the connector end) that you use to adjust the probe compensation. Since this is a trimmer capacitor you should use a non-metal (plastic or wood) screwdriver to adjust it – a piece of metal can detune the setting when it is close, which would make it hard to adjust accurately.

Refer to the picture below as to how the waveform shall look like after calibration.

