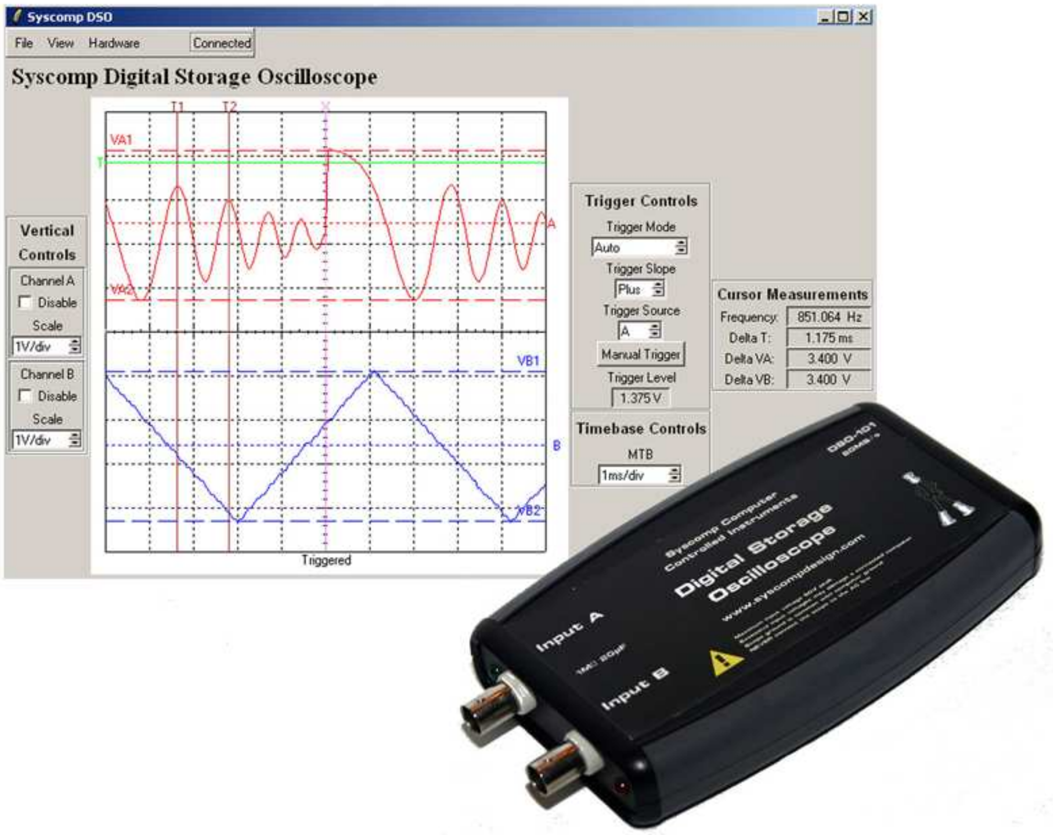
Syscomp Computer Controlled Instruments
DSO-101 Oscilloscope Manual

Syscomp Electronic Design Limited
`http:\\www.syscompdesign.com`
Revision: 1.07
August 15, 2007



Revision History

Version	Date	Notes
100A	January 2006	First revision
100B	2 May 2006	Added material on Debug Console
102	17 May 2006	Added detail to the 'Commands' Section
103	13 Aug 2006	Added info on Linux USB permissions
104	30 Aug 2006	Added: Manually selecting COM port in XP
105	14 Apr 2007	Added: XP installation screen shots
		Added: Additional Features (V1.7)
107	13 Aug 2007	Minor corrections

Contents

1 Overview **1**

2 Features and Specifications **1**

3 Oscilloscope, Basic Controls **2**
3.1 Amplitude 2
3.2 Timebase 3
3.3 Triggering 3
3.4 Display 4
3.5 Save/Load Settings 5

4 Additional Features **5**
4.1 Measurements Screen 5
4.2 Waveform Zoom/Hi-Rez (High Resolution) Mode 6
4.3 Spectrum Display 7
4.4 XY Mode 11
4.5 Histogram 12
4.6 Waveform Math 13
4.7 Screen Capture 14
4.8 Export to Postscript 15
4.9 Windows: `PrtSc` Method 16
4.10 Linux: Grab Screen Method 16
4.11 Vertical Calibration 16

5 Aliasing **17**

6 Safe Measurement Technique **18**
6.1 Floating Power Supply 18
6.2 Grounded Power Supply 19
6.3 Battery and AC Adaptor Power Supplies 19
6.4 Russian Roulette and AC Line Voltage 20
6.5 Removing the Ground 21
6.6 Observation of AC Line Voltages 21

7 Overview of USB Operation **21**

8 Troubleshooting **22**
8.1 Microsoft Windows 22
8.1.1 Windows XP 22
8.1.2 Manually Assigning a COM Port Number in Windows XP 23
8.2 Linux 25
8.2.1 Manually Changing Device Port Permissions 26
8.2.2 Setting Default Port Permissions to User Mode: Suse 9.2 27
8.2.3 Setting Default Port Permissions to User Mode: Fedora Core 6 27
8.2.4 Running the Program 28
8.2.5 Device Properties using `usbview` 28

9 Technical Description **29**
9.1 Input Channel Overview 30

10 Adjustments **31**
10.1 Input Compensation Capacitors 31
10.2 Offset Adjustment Resistors 32

11 Oscilloscope Commands **33**
11.1 Using the Debug Console 33
11.2 General Commands 34
11.3 Control Register Commands 34
11.4 Memory Commands and Organization 37
11.4.1 Memory Organization 37
11.4.2 Memory Commands 38
11.5 Other Commands 38

12 Example: Performing a Capture **39**
12.1 Preamp Settings 40
12.2 Trigger Level Settings 40
12.3 Timebase and Trigger Point Settings 40
12.4 Control Register 41
12.5 The Capture 42
12.6 Plotting the Data 42

13 Manual Operation **42**
13.1 Windows 42
13.2 Linux 43

14 Modifying the Tcl/Tk Software **44**

15 Sources of Information **44**
15.1 News Groups 44
15.2 Websites 44
15.3 Paper 45
15.4 Textbooks 45

16 Windows XP Appendix, Screenshots and Technical Details **46**
16.1 Checking the Installed Files . . . 46
16.2 Obtaining Details of the USB-Serial Port . . . 46
16.3 Adjusting the COM Port Selection . . . 48
16.4 XP Installation Instructions and Screenshots . . . 48
16.4.1 Preliminary: Checking for Installed Drivers . . . 49
16.4.2 Starting the Installation . . . 50
16.4.3 Continuing the Installation . . . 51
16.4.4 Installing the Drivers . . . 54

List of Figures

1 Oscilloscope GUI . . . 3
2 `Measurements` Screen . . . 5
3 High-Res Mode Display . . . 6
4 Square Wave Spectrum . . . 8
5 XY Display Mode . . . 11
6 Diode Voltage-Current Characteristic . . . 12
7 Histogram Display . . . 13
8 Waveform Math Tool . . . 14
9 Screen Capture . . . 15
10 Calibration Controls . . . 17
11 Floating Power Supply . . . 18
12 Floating Power Supply . . . 19
13 AC Line Voltage . . . 20
14 Usbview . . . 29
15 Oscilloscope Block Diagram . . . 29
16 Scope Input Compensation Network . . . 32
17 Preamp Control Register . . . 34
18 Scope Control Register . . . 36
19 Memory and Associated Parameters . . . 37
20 Example Waveform Showing Capture Parameters . . . 40
21 Example Internal Memory Map Showing Parameters . . . 41
22 My Computer . . . 46
23 Control Panel . . . 47
24 Add or Remove Programs . . . 47
25 System Properties . . . 48
26 System Properties, Hardware (Device Manager) . . . 49
27 Serial Port Properties . . . 50
28 Serial Port Properties, Advanced . . . 51
29 Install Screen . . . 52
30 My Computer . . . 52
31 CD Directory . . . 53
32 Windows Software/Driver Installation . . . 53
33 Windows Software/Driver Installation . . . 54
34 Welcome to the Syscomp .. Setup Wizard . . . 54
35 Select Destination Location . . . 55

36 Select Additional Tasks 55
37 Ready To Install 56
38 Completing the Syscomp .. Setup Wizard 57
39 Step 2: Install the Drivers 58
40 Found New Hardware Wizard: 1 59
41 Found New Hardware Wizard: 2 59
42 Found New Hardware Wizard: 3 60
43 Found New Hardware Wizard, Progress Bar 60
44 Found New Hardware Wizard, Final Screen 61
45 Found New Hardware Wizard, Final Screen 61
46 Port Settings Dialog 62

Caution: Never connect this oscilloscope to the AC line. Doing so may result in personal injury and extreme damage to the operator, the oscilloscope and to an attached computer. See section 6 on page 18.

1 Overview

The Syscomp DSO-101 is a general-purpose two-channel oscilloscope for the teaching, development and debugging of electronic circuits. It is one of a series of instruments from Syscomp Electronic Design.

Channels A and B are sampled simultaneously and stored in the oscilloscope memory before being sent for display to the host computer. Consequently, they are always time aligned and triggered from the same trigger signal.

The oscilloscope timebase frequency is derived from a crystal oscillator, so it can be expected to be precise and stable. The displayed amplitude is determined by 1% resistors and analog-digital conversion.

A PC host displays a graphical user interface for the oscilloscope with frequency readouts, sliders, clickable buttons and various other controls. The PC connects to the oscilloscope hardware via a USB cable so that no other power source is required.

A Graphical User Interface program for operating the oscilloscope is provided on the accompanying CDROM. The program is written in the Tcl/Tk language. It is completely Open Source, so it may be modified and distributed freely.

The USB interface emulates a serial port so that the oscilloscope may be accessed as a serial port device.

2 Features and Specifications

The features and performance specifications are as follows:

Channels	2 independent channels sampled simultaneously
Sampling Frequencies	20 MSamples/second maximum in 13 ranges stepped x2
Vertical Resolution	8 bits per channel (1:256)
Vertical Bandwidth	2 MHz
Vertical Input Range	±250mV to ±25V full scale
Vertical Gain Settings	7 settings, 50mV/div to 5V/div, in 1:2:5 sequence (Note 1)
Vertical Scale	10 major divisions
Horizontal Time Settings	21 settings, 200mSec/div to 50nSec/div, in 1:2:5 sequence
Horizontal Scale	10 major divisions
Input Impedance	1M Ohm parallel 27pF
Triggering	Digital comparison with input signal
Trigger View	Pre and Post trigger simultaneously viewable
Trigger Controls	Source (A, B, Manual), level and slope select
Memory Depth	32K Samples each channel
Display resolution	8 bits vertical (1:256)
	Adjustable 1 to 2^{15} horizontal
Indicators	Power LED (Green)
	Activity LED (Red)
Interface	USB 2.0: Emulated serial port
Physical Dimensions	3” x 5”
Note 1	Vertical preamp gain settings, not *software zoom*.

3 Oscilloscope, Basic Controls

The oscilloscope *Graphical User Interface* (*GUI*) is shown in figure 1. The exact design of the GUI is subject to change and development as features are added, but figure 1 will provide some guidance.

Many of the scope controls are similar in function to those of the classic analog oscilloscope. Other controls are unique to the DSO-101. They provide functions - such as pre-trigger display - that are only possible in a digital oscilloscope.

The scope controls divide into various groups: *amplitude*, *timebase*, *triggering* and *display*.

3.1 Amplitude

There are two input channels, A and B, each with identical controls.

- **Disable** Click on this button to disable display of the corresponding waveform. This is useful to reduce clutter on the display.

- **Scale** Sets the vertical scale factor of the display between 50mV per division and 5 volts per division in the traditional oscilloscope 1:2:5 sequence. The amplitude may be read off the display by measuring the number of divisions and multiplying by the scale factor. Alternatively, the *display amplitude cursors* may be used (see *display* below.)

  Changes to the *scale* control adjust the sensitivity of the front-end of the oscilloscope and adjust the amplitude of the signal into the A/D converter in order to maximize the resolution of the A/D converter. This is not a scale change on the display.

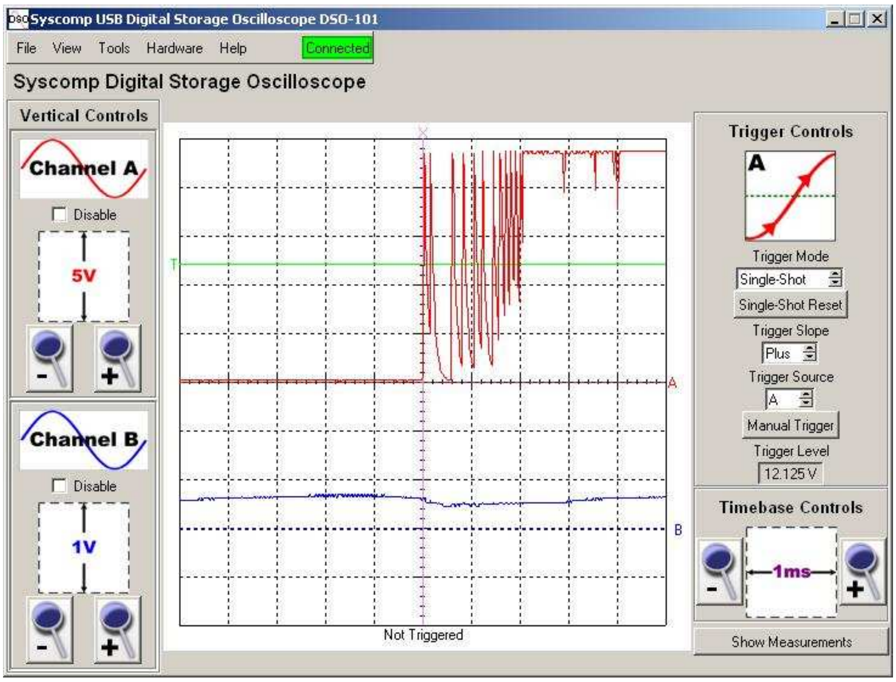


Figure 1: Oscilloscope GUI

3.2 Timebase

The *Main Time Base* (*MTB*) can be varied between 50nsec/division and 200msec/division in the traditional oscilloscope 1:2:5 sequence. At longer time scales, there is a noticeable pause between display updates as the buffer memory fills with a new set of waveform samples.

3.3 Triggering

In order to present a stable waveform display, each display update must start at the same point on a waveform. The trigger functions determine how that *trigger point* on the waveform is selected.

The trigger controls must meet certain requirements in order to generate trigger signals for waveform capture. If these controls are not set properly, it is possible that the scope will not capture waveforms and the display will appear to be frozen. Alternatively, the display may appear to jump between captures, without a stable waveform display.

- **Trigger Time** The *trigger time* is marked by a vertical cursor with an **X** symbol at the top. The trigger point may be dragged left or right to reveal more or less of the waveform preceeding or following the trigger point. This ability to view the waveform preceeding the trigger point is one of the advantages of a digital oscilloscope.

- **Trigger Level** The *trigger level* is marked by a horizontal cursor with a **T** symbol at the leftmost edge of the screen. The trigger level may be dragged vertically to set the amplitude on a waveform that establishes the trigger point. In order to cause triggering, the trigger level cursor must be positioned within the amplitude of the triggering waveform.

- **Trigger Mode: Auto/Normal/Single/High-Res**
  - In the **Normal** position, the scope hardware *must* get a proper trigger signal in order to display a new waveform. Without a trigger signal, the display simply waits. (The **Manual Trigger** button can be used to force a trigger event, that is, display one capture.)
  - In the **Auto** position, if there is a trigger signal, the scope uses the trigger signal to synchronize waveform capture. If there is no trigger signal the scope hardware waits for a period of time and then generates a trigger signal internally. That way, there are periodic updates to the waveform display, even if triggering is not occurring from an input waveform.

    In general, the most convenient position is **Auto**. However, there are two situations where **Normal** triggering is necessary:

    * For very low frequency waveforms, the trigger signals occur infrequently. If **Auto** triggering is enabled, the scope will decide that trigger signals are not present and generate them internally. This is not what is wanted: the scope should wait for a waveform trigger signal.
    * If the scope is being used to capture a single-shot event, then it should not trigger itself: it should wait for a waveform trigger signal, regardless of how long it takes for that trigger signal to occur.
  - In the **Single** position, the scope waits for a trigger signal. (This is known as the **Armed** state.) When a trigger signal occurs, the software captures and displays that waveform and disables further triggers. The **Single-Shot Reset** button clears the display and returns the scope to the **Armed** state.
  - The **High-Res** (High Resolution) trigger mode is described in section 4.2, page 6.
- **Trigger Slope** The *trigger slope* control selects a positive-going or negative-going slope at the trigger point. This allows one to trigger off the leading or trailing edge of a positive pulse waveform, for example.
- **Trigger Source** The trigger signal may be derived from the Channel A waveform or the Channel B waveform. Generally, it is easier to obtain a stable trigger signal from the simpler of the two waveforms.
- **Manual Trigger** Actuating this button generates a trigger signal. This is sometimes useful to cause the scope to capture one waveform.
- **Trigger Level (Readout)** This display shows the amplitude of the trigger level setting.

3.4 Display

- **Vertical Position** A waveform may be moved in vertical position. At startup, the **A** and **B** cursors - which are the zero reference for the channel – are placed at centre screen. Using the mouse, drag the cursor vertically to change the position of a waveform.
- **Time Cursors** Under the **View** menu, it is possible to enable and disable various cursors. The time cursors show as vertical lines with labels **T1** and **T2**. The time between those cursors shows in the **Delta T** readout.

  The inverse of the value of **Delta T** is shown in the **Frequency** readout. For the frequency value to be meaningful, the **T1** and **T2** cursors should be placed at the start and finish of one complete cycle of a waveform.
- **Amplitude Cursors** There are two amplitude cursors for each waveform: **VA1** and **VA2** for channel A, **VB1** and **VB2** for channel B. The **Delta VA** readout shows the voltage between the **VA1** and **VA2** cursors. The **Delta VB** readout shows the voltage between the **VB1** and **VB2** cursors.

  For example, to measure the peak value of a waveform, put cursor **VA1** at the maximum value of waveform A, and cursor **VA2** at the zero (ground) value of the waveform.

3.5 Save/Load Settings

Use the button `File -> Save Settings` to save the oscilloscope control settings to a configuration (.cfg) file.

Use the button `File -> Load Settings` to restore the oscilloscope control settings from a text file.

All configuration files are text readable, that is, they can be examined by loading them into a text editor such as `Notepad` under Windows.

Saving and loading the oscilloscope configuration is useful for those learning to use the oscilloscope. Once the scope is set up and working for a particular measurement, save that configuration. Then return to that configuration by loading the same file.

For those requiring multiple oscilloscope control settings, save each configuration as a different file. Then select the desired configuration to return the scope to those settings.

4 Additional Features

Section 3 described the basic controls of the oscilloscope. In this section we describe additional controls for more specialized measurements.

4.1 Measurements Screen

The Measurements Screen shows cursor and automatic measurements of the most common features of a waveform: amplitude and time.

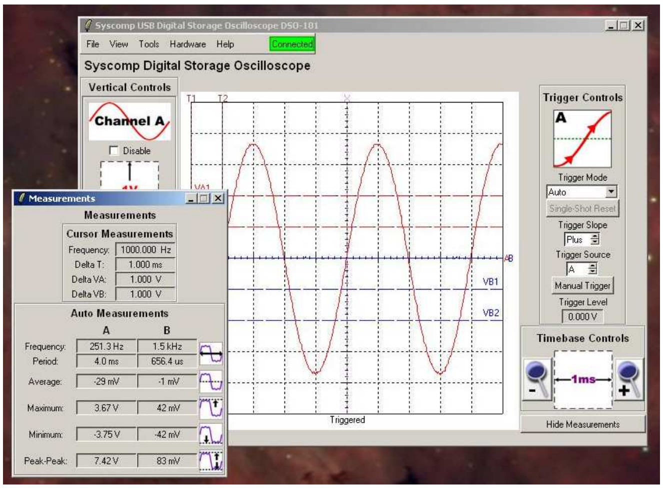


Figure 2: `Measurements` Screen

As shown in figure 2, the Measurements Screen is in a separate window. This window defaults to being displayed, that is, it is displayed when the program is first started.

The Measurements Screen may be hidden with the standard X window closer or the `Hide/Show Measurements` button on the lower right corner of the main screen.

The Measurements Screen is divided into two areas. The upper area shows the screen cursor readouts (section 3.4, page 4).

The lower area shows the `Auto Measurements`, in which the software automatically measures waveform parameters. The `Frequency` and `Period` readouts are based on the zero-crossings of the waveform. Consequently the waveform must be periodic and the main display must show at least one complete cycle of the waveform for the frequency and period values to be meaningful. As well, the frequency and period measurement can be fooled if the waveform includes noise that causes multiple spurious zero crossings.

For multiple measurements on a reasonably clean waveform, the Auto Measurements can be a real time saver. However, it is best to verify that the measurement is reliable before relying on it extensively.

4.2 Waveform Zoom/Hi-Rez (High Resolution) Mode

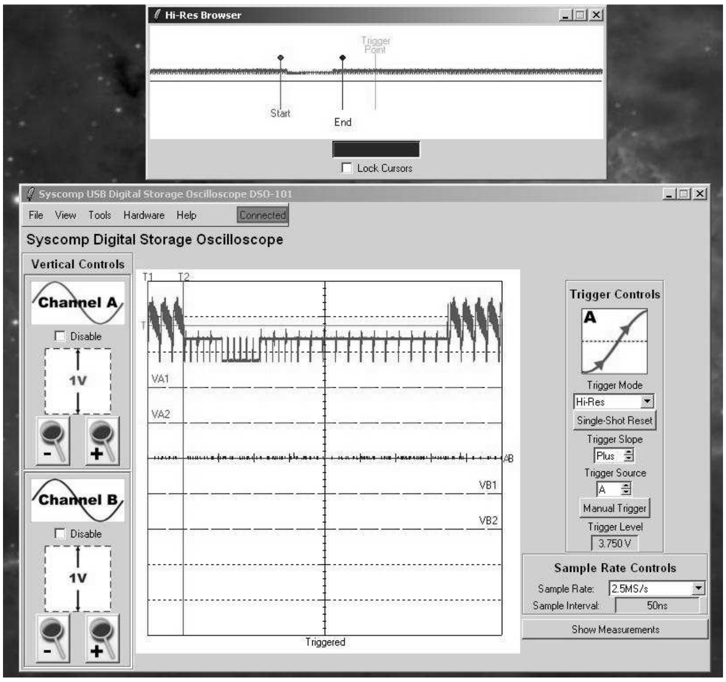


Figure 3: High-Res Mode Display

The high-resolution browser allows one to capture a lengthy waveform and then zoom into portions of the waveform. This is particularly useful when the waveform has details of interest that are offset some distance from a trigger point.

This facility is somewhat similar to the *delayed sweep* function found on some analog oscilloscopes.

Figure 3 shows an example[1]. This video waveform has a complex structure which can be examined in detail by scrolling the high-res display.

The small display in the upper part of the image contains 32k samples of thhe video waveform. The main display shows a small area of that waveform.

To use high-res mode:

1. Set up a display of the waveform on the main screen so that it is triggering correctly.

[1] This image is from the paper *Using the High Resolution Mode of the DSO-101 Oscilloscope to Display Video Waveforms*, which can be downloaded from the Syscomp web site.

2. Click on 'Trigger Mode'. Select the entry 'Hi-Res'. A smaller `High Res Browser` window will appear.

3. New `Sample Rate Controls` will appear in the bottom right corner of the scope control panel, replacing the timebase selector. Select a suitable sample interval.

4. Hit the `Single Shot Reset` control and when it receives the next trigger signal the scope will initiate a new capture of 32k samples.

   The normal scope display is 500 samples, so acquiring a high-res display takes significantly longer, especially at low sample rates. A progress bar on the High Res Browser window indicates the acquisition.

5. There are two cursors on the High Res Browser window, marked `Start` and `End`. The region between these two cursors, in the high res window, appears in the main display.

   You can drag these cursors separately to vary the amount of the high-res waveform in the main display. Or you can lock the two cursors together and drag the two of them the length of the high-res waveform.

6. The number of samples between the two cursors appears in a readout `Number of Samples Selected`. This is of interest when the spectrum display is in use.

7. You can change the sample rate to some other value and then click on `Single Shot Reset` to force the acquisition of another waveform.

To close the high-res display, use the `Trigger Mode` selector to choose `Auto`, `Normal` or `Single-Shot` mode.

4.3 Spectrum Display

A complex waveform may be treated as being composed of a number of sinusoid waveforms. These sinusoids are of various phases, frequencies and amplitudes. The description of the magnitude, phase and frequency of these various waves is known as the *spectrum* of the signal, by analogy with the spectrum of light.

Spectrum Analysis or *Fourier Analysis* is the process of analysing some time-domain waveform to find its spectrum. We also say that the time domain waveform is converted into a frequency spectrum by means of the *Fourier transform*.

Clicking on `Tools -> Spectrum Analysis` brings up the spectrum analysis display of figure 4. The displayed spectrum in this image is a 10kHz square wave of amplitude 8.8 volts peak.

The theory of Fourier Analysis shows that a square wave is composed of a fundamental of magnitude E volts at frequency f (10kHz in this case) with the following harmonics: $E/3$ magnitude at frequency $3f$, $E/5$ magnitude at frequency $5f$, $E/7$ magnitude at frequency $7f$, and so on. The spectrum display shows this pattern.

Each vertical line represents one of these frequency components. The height of the line is proportional to the magnitude of that particular component. The horizontal axis is a linear scale of frequency, with zero frequency (DC) at the left edge.

The vertical cursor can be dragged horizontally to determine the frequency and magnitude of a component of the spectrum.

The spectrum display and main waveform display are active at the same time, allowing one to simultaneously observe a waveform in the time domain and frequency domain.

Interpreting the Display

Because of fundamental limitations in a sampled-data system, it is possible for the display to be misleading. Here are some important points to keep in mind when using spectrum analysis based on digital methods:

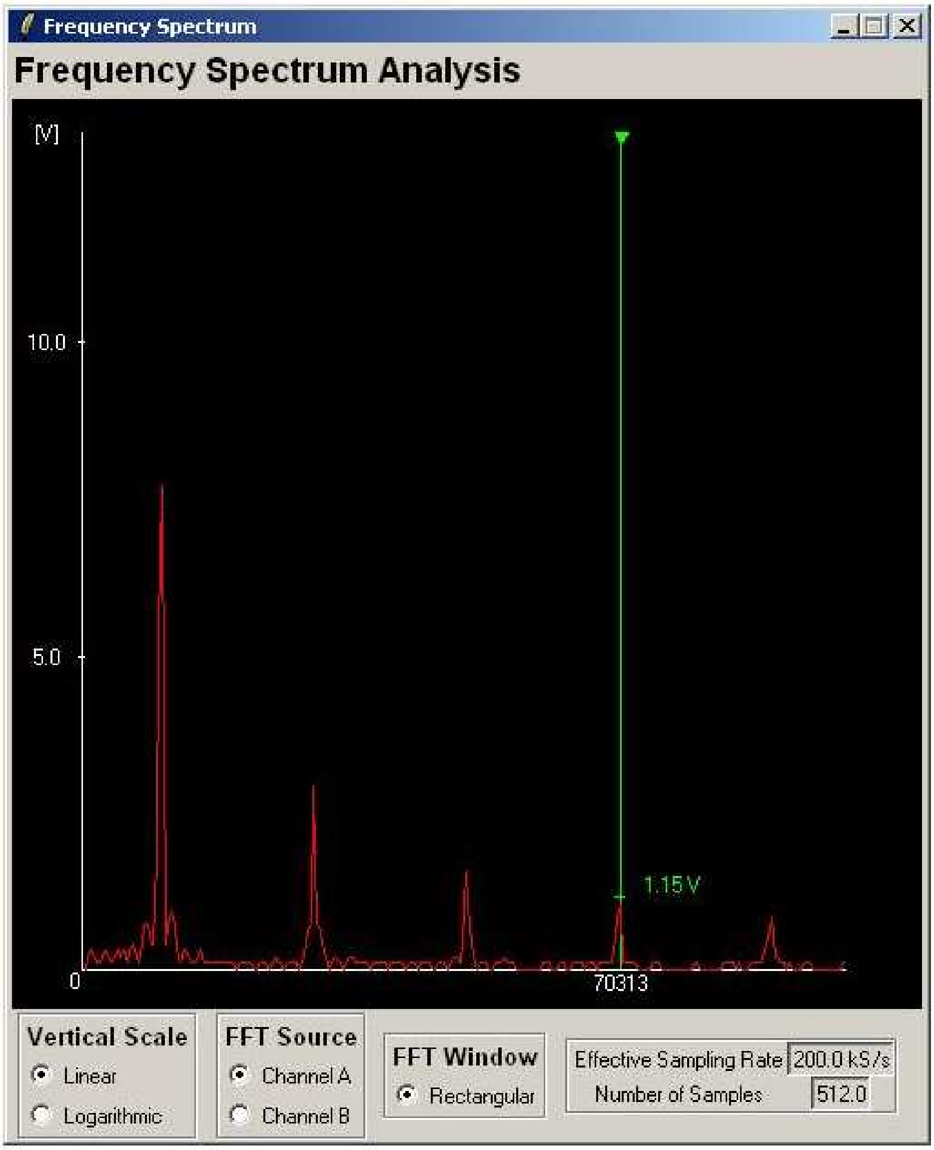


Figure 4: Square Wave Spectrum

- The `Effective Sampling Rate` is shown in a readout at the bottom right corner of the spectrum display. This is important: the sampling rate must be at least twice the frequencies being analysed to avoid aliasing. Put another way, there must not be frequency components above the Nyquist rate, which is half the sampling rate. In the example shown in figure 4, the sample rate is 200kHz. The frequency components range from 10kHz to 90kHz, below the Nyquist rate of 100kHz[2].

[2] Harmonics of the square wave extend to much higher frequencies but we assume their amplitude is small enough to be ignored.

- A *sweeping type* analog spectrum analyser moves a bandpass filter across a range of frequencies to determine the spectrum. A digital spectrum analyser such as this one divides up the frequency range into a number of *bins* and then measures the energy in those bins.

  The main oscilloscope display of the DSO-101 is 500 points. This is padded to 512 points[3] by appending zeros to the waveform record. As a result, there are 256 frequency bins when using the main oscilloscope display.

  The centre frequency of each of these bins may not coincide exactly with the frequency components present. If that is the case, then the displayed amplitude will be incorrect and should only be regarded as an approximation of the true situation.

- As the readout cursor is dragged higher in frequency it jumps from bin to bin, reading out the centre frequency of each bin. A given frequency component may not be centred in its bin, so the frequency readout will be only approximate. For example, in figure 4, the square wave frequency (generated by a Syscomp WGM-101 waveform generator) is at a frequency 10kHz to within a fraction of a Hz. The 9th harmonic is at 90kHz. The spectrum display readout puts the 9th harmonic at 90234 Hz, which is only approximately correct.

Frequency Scale, Bin Spacing

It is sometimes useful to be able to determine the resolution of the frequency axis. Each frequency bin has a width $\Delta f = 1/T$ Hz where T is the length of the data record in seconds. If there are N points in the data record, then $N/2$ points are displayed as positive frequency. (The other $N/2$ points are redundant.)

Example

Determine the frequency resolution (bin spacing) for the case of the display of figure 4.

Solution

The sample rate is 200kS/sec. The sample interval ΔT is the reciprocal of this:

$$\begin{aligned} \Delta T &= \frac{1}{200 \times 10^3} \\ &= 5\ \mu\text{Sec} \end{aligned}$$

The number of points N in the data record is 512 points, so the total length of the data record is:

$$\begin{aligned} T &= N\Delta T \\ &= 2.56\ \text{mSec} \end{aligned}$$

The resolution Δf is the reciprocal of the record length:

$$\begin{aligned} \Delta F &= \frac{1}{T} \\ &= 390.625\ \text{Hz} \end{aligned}$$

[3]The FFT routine requires that the number of points be a power of 2.

For example, the 7th harmonic should appear at $f_7 = 70$kHz. The spectrum display actually puts it at 70313 Hz, which is bin 180.

$$\begin{aligned} F_7 &= 180 \times 390.625 \\ &= 70313 \text{ Hz} \end{aligned}$$

The maximum frequency f_{max} on the display occurs at bin 256:

$$\begin{aligned} f_{max} &= 256 \times 390.625 \\ &= 100 \text{ kHz} \end{aligned}$$

Windows

Window or *weighting* functions are often applied to the time-sequence data prior to transformation into the frequency domain. All window functions taper the data down to zero at its ends. Then the discontinuity caused by a finite record length does not affect the shape of the transform.

The choice of window function depends on the application, and all window functions are a compromise of some sort. For example, some window functions provide very accurate amplitude readings, others are best for separating closely spaced frequencies. A collection of window functions is shown at `http://en.wikipedia.org/wiki/Window_function`.

The current spectrum analysis routines have only one window function, the *rectangular window*. This is in fact a non-window, it does not shape the time function, all points on the time record are weighted equally. Other weighting functions may be added in future versions of the software.

Applications

Spectrum analysis has a number of applications in electronics and mechanical engineering:

- A pure tone has no harmonics and will show up on a spectrum display as one single vertical line. Distortion of a sine wave will create additional harmonics. Consequently, a measure of the magnitude of the harmonics is a measure of the magnitude of the *harmonic distortion*.

- In a distortion-free (linear) system, two separate input tones (single frequencies) will emerge as the same two tones at the output. If the system is distorting (non-linear), then the system will generate other tones at the sum and difference frequencies of the input signals. A measure of these extra signals is a measure of the *intermodulation distortion*.

- The existence of certain frequencies in a signal may give some clues as to its source. For example, if a signal contains the power line frequency (eg, 60Hz in North America, 50Hz for the UK), then it is probably picking up interference from the AC power line.

- Power systems frequently manipulate waveforms by chopping them or combining them with other signals. Spectrum analysis allows one to measure the harmonic content of a signal, which may be specified as a requirement.

- The analysis of a mechanical system for resonances can be done by driving the system with a wide-band excitation signal, an impulse hammer blow or random noise from a shaker. Microphones or accelerometers convert the mechanical vibration of the system to an electrical signal. The spectrum analysis of this signal indicates the mechanical resonances in the structure.

- The extraction of signals from noise may require some knowledge of the spectrum of the signal and the noise.
- It is useful to see the spectrum diagram for modulation and other signal manipulations.

Further information on spectrum analysis is in the paper *Introduction to Digital Spectrum Analysis*, which is on the Syscomp web site.

4.4 XY Mode

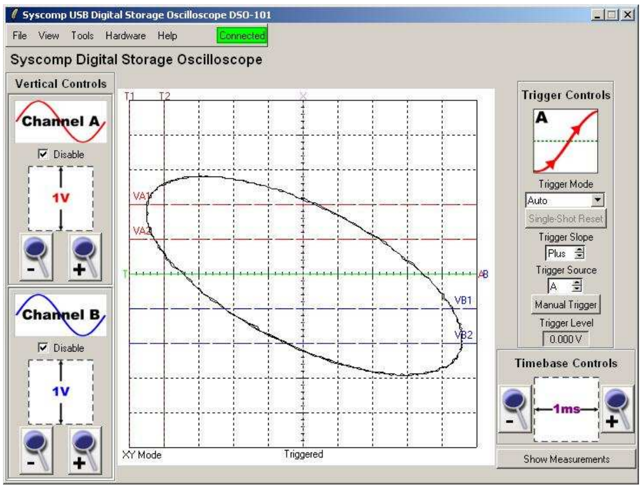


Figure 5: XY Display Mode

The usual oscilloscope display shows a plot of the two signal signal amplitudes, voltage on Channel A and Channel B, vs time. It is also possible to plot the two voltages against each other: Channel A as the X axis and channel B as the Y axis.

Select `View -> XY Mode` to enable the XY display.

The DSO-101 can simultaneously display both the XY display and the conventional voltage-time waveforms, which is useful in teaching situations and for debugging purposes.

Lissajous Figures

When the two signals are sine waves of the same frequency, with a phase shift between them, the display is as shown in figure 5. This type of looping display is known as a *lissajous figure*.

If the two frequencies different but integer multiples of each other, then the lissajous figure will have multiple nodes. In the early days of oscilloscopes, lissajous figures were used in this manner for frequency measurement. The vertical amplifiers of the day could not work at high frequencies, so the signals were applied directly to the deflection plates of the cathode ray tube. The lissajous figure gave an indication of frequency ratio and relative phase.

To form a complete lissajous loop, the timebase setting must be such that both waveforms show at least one complete cycle.

Magnitude Measurement

If the two signals are exactly in phase, the XY plot is a straight line. If the two signals are of exactly the same magnitude, the angle of the straight line is $45°$. If the magnitudes are different, then the line is at some other angle. This is a sensitive method of comparing the amplitude of two waveforms, which need not be sine waves. Any waveshape should function in this measurement.

General Purpose Plotting

The XY Mode display may be used for a variety of applications where a plot of some kind is required.

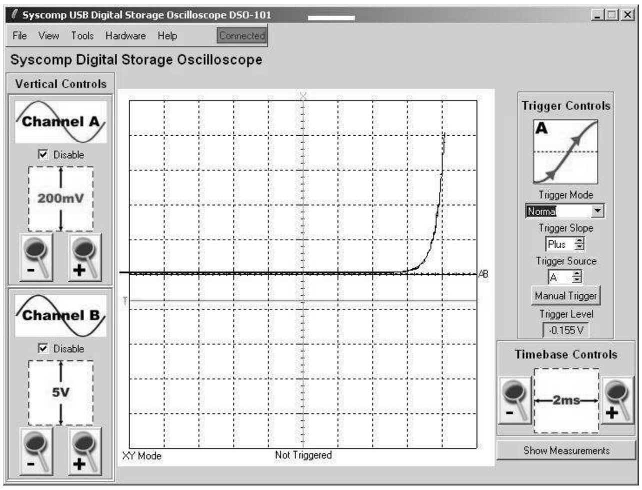


Figure 6: Diode Voltage-Current Characteristic

Figure 6 shows an example. The DSO-101 has been configured to plot the voltage-current curve of a silicon diode. Notice that the diode threshold is around 0.6 volts. The vertical scale is voltage measured across a current sensing resistance, equivalent to 10mA per division.

4.5 Histogram

A *histogram* shows the probability of finding a particular level in the waveform. For example:

- A triangle waveform has a uniform probability distribution: all the levels are equally likely.
- A square wave is most likely to be at its positive maximum or negative maximum. Levels in between these two are unlikely.
- A ramp waveform is the same as a triangle waveform: all the levels are equally likely.

The histogram display is most useful on noise-like waveforms. For example, random noise has a so-called *gaussian* probability distribution. Its most probable value is around zero. Moving away from zero, positive or negative, the probability decreases.

Minimizing electrical noise, including interference from other sources, is a difficult challenge. When the noise level is low, it can be difficult to tell whether some remedial measure (shielding, moving a ground connection,

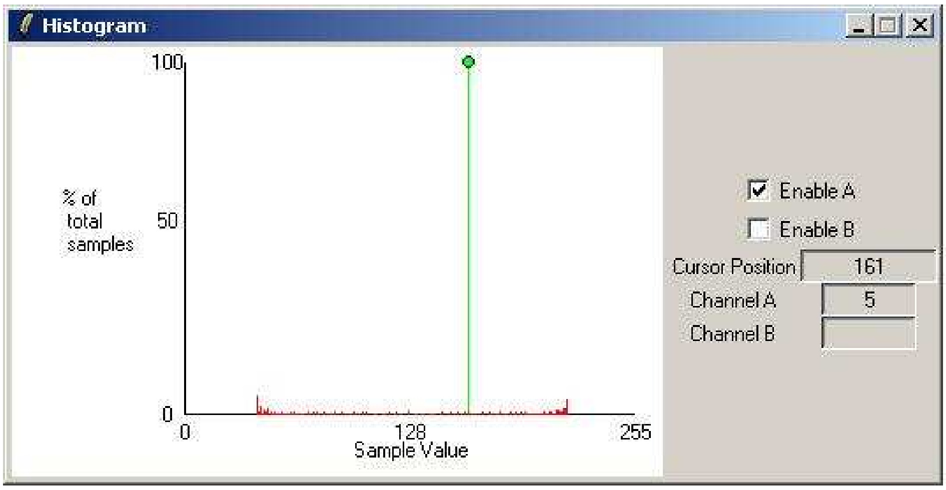


Figure 7: Histogram Display

adding bypassing) causes an improvement. The probability distribution can then be a very useful form of the display. For example, to minimize the noise in an amplifier, ground the input and examine the output histogram display. Ideally, the output should be zero: the probability of the zero value should be very high and the probability of other values should be zero. If there is noise present, there is a finite probability of other values. Then the histogram is one large spike at zero with smaller spikes, adjacent to the main spike. As you reduce the system noise, the small spikes will decrease in amplitude and the horizontal range of non-zero spikes will decrease.

In the DSO-101, the histogram display is generated by plotting A/D values against the number times each value occurs. The 8-bit A/D converter can produce values from 0 to 255, so this is the horizontal scale, the range of possible values. (The mid-point, corresponding to an input value of zero volts in the waveform, is the A/D value 128.) The software examines a waveform, counts the number of occurrences of each of these possible values, and expresses the result as a percentage of the total.

Click on `Tools -> Histogram` to bring up the histogram display,

Figure 7 shows the histogram display of a sine wave. Notice that the extreme values are slightly more probable than the central values. That is, a sine wave is slightly probable to be found near its maximum positive or negative value, less likely to be found near zero. Put another way, the sine wave spends more time near the peaks than in the transition region between peaks.

The histogram `Enable` controls select the displayed channel(s). The cursor may be dragged left or right to obtain a numeric readout of the `Percent of Total Samples` at that particular A/D value.

The histogram display operates in real-time, updating with each waveform capture.

The Tcl/Tk code for the histogram display (`histogram.tcl`) is a relatively simple example that illustrates how to add a display feature to the oscilloscope program.

4.6 Waveform Math

The math toolbox generates a waveform display that is some function of the waveforms on Channel A and Channel B.

Clicking on `Tools -> Math Toolbox` brings up the display of figure 8. At this time, there are two math functions: `Add: A+B` and `Subtract: A-B`. The software does a point-by-point addition or subtraction of

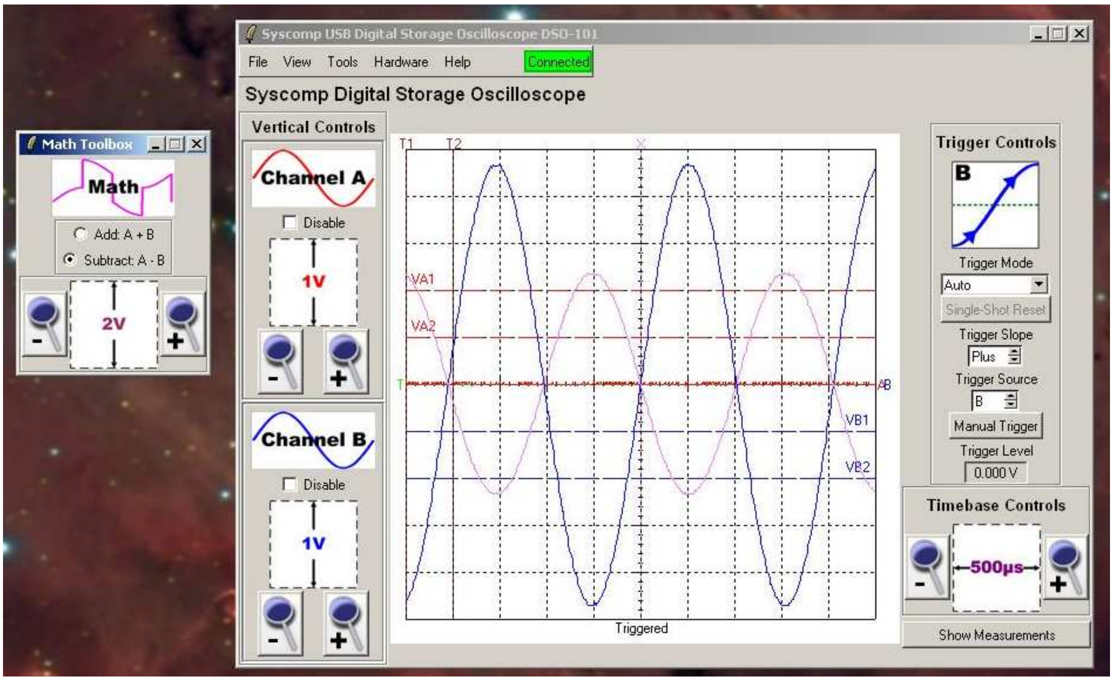


Figure 8: Waveform Math Tool

the input waveforms and generates a display of the result on the main display.

The vertical scale factor control (2 volts/div in figure 8) can be used to adjust the displayed magnitude of the result.

The display of figure 8 shows an input sine wave on channel B. The channel A input is zero. The output of the math calculation is $A - B = 0 - B = -B$, that is, an inverted form of the waveform on channel B. The inverted waveform is half the height of the input waveform because the math result vertical scale factor is has been set to twice the channel B vertical scale factor.

Subtraction of two waveforms is useful in taking a differential voltage measurement across some component, when neither terminal of the component is grounded.

It is a relatively simple matter to add other functions to the math toolbox. (see the source code file `math.tcl`). For example, a *product* function, would support the multiplication of two waveforms. If one waveform is proportional to voltage and the other to current, then the product is a waveform of power.

4.7 Screen Capture

It is extremely useful to be able to capture oscilloscope screen shots. One or more screen shots may be used to document a particular measurement situation as a record of the measurement or to capture the result for a larger document.

The oscilloscope main window or any of its subsidiary windows (measurements panel, histogram, spectrum display) can be captured to a JPEG image file. Figure 9 shows an example[4].

[4]Acknowledgement: Special thanks to John Foster who helped develop the screen capture code.

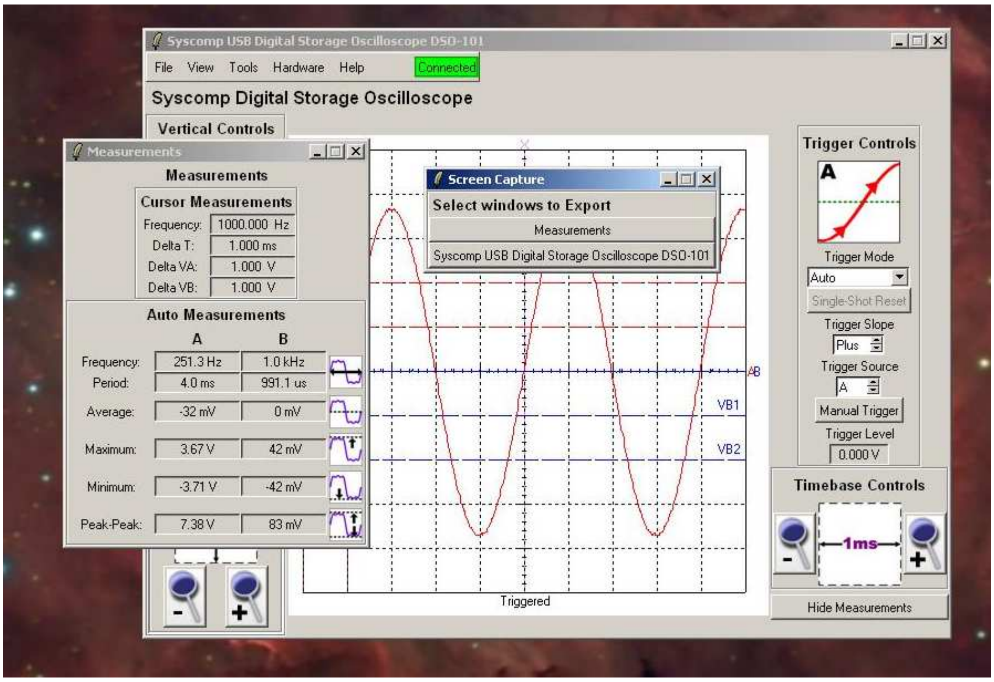


Figure 9: Screen Capture

The main DSO-101 screen is in the background, on the computer desktop. The `Measurements` panel is currently also being displayed.

Selecting the menu item `Tools -> Screen Capture (jpg)` brings up the small dialog in the centre of figure 9. In this case, the menu shows two screen entries: `Syscomp Digital Oscilloscope DSO-101` and `Measurements`.

Clicking on one of these selections then brings up the standard `File Save` menu, and the file may be named and saved where required.

The screen shot can only be saved in JPEG format. If it is needed in another format, load it into a drawing or image processing program and save it in that format.

Under the Windows operating system, you can use `Paint` for this purpose.

Under Linux, the program `Imagemagick` can convert to a variety of imagee formats.

4.8 Export to Postscript

Section 4.7 on page 14 described the oscilloscope the `Screen Capture` features. These features support the capture of any oscilloscope screen window to a JPEG image file.

It is also possible to capture the waveform display area of the main oscilloscope display as a postscript file.

Select the menu item `Tools -> Export to Postscript (PS) File` brings up the standard `File Save` menu, and the file may be named and saved where required.

What are the relative advantages of JPEG screen capture vs exporting the display area to a Postscript file?

- `Screen Capture` grabs everything inside the selected window. This may be useful if you need to document the control settings. Furthermore, `Screen Capture` can capture any of the scope windows.

Export to Postscript grabs only the waveform display area on the main oscilloscope screen.

- Certain writing tools may require image format to be postscript. It is entirely possible to convert a JPEG image to Postscript, but the file size is quite large. If you need the waveform in Postscript format and file size is an issue, then you might be better off capturing just the waveform area in Postscript format.
- Both the JPEG capture and Postscript capture are in colour.

4.9 Windows: PrtSc Method

For completeness, we mention that the computer screen may be captured using the PrtSc keyboard button[5]

- Press the PrtSc keyboard button to save the entire desktop screen to an operating system buffer.
- Open My Computer -> Programs -> Accessories -> Paint.
- Use Edit -> Paste this image into the paint program.
- Save this image in the desired format.

The key combination <alt><PrtSc> causes only the top window to be saved in this fashion.
The images of section 16.1 were obtained using this technique.

4.10 Linux: Grab Screen Method

Under Linux, there are several alternatives. With the GNOME desktop, select Accessories -> Take Screenshot. With the KDE desktop, select Utilities -> Desktop -> KSnapshot. The shareware program xv is suitable. The **Gimp** can capture a desktop.

4.11 Vertical Calibration

Revision 1.7 of the oscilloscope software added the vertical calibration controls shown in figure 10.

Click on Tools -> Calibrate Vertical Sensitivities to bring up this display.

Each slider adjusts the vertical display calibration by $\pm 20\%$. In the mid-position, the calibration is 1.0, and the adjustment varies between $\times 1.2$ at the top to $\times 0.8$ at the bottom.

The gain is established in the hardware with 1% tolerance components and the A/D converter has a somewhat larger variation in gain. Syscomp ships the software in its default gain state.

If you wish improve on the vertical accuracy of each channel, there are two methods: AC and DC.

The AC method avoids any problems of DC Offset.

- Apply an AC voltage of known amplitude to the input. For example, you could use a signal generator to produce a sine wave and measure the amplitude with a digital voltmeter. The peak value of the sine wave will be $\sqrt{2}$ times the measured amplitude. The peak-peak value will be twice this.

  There is the potential for serious inaccuracy. You must do this at a frequency for which the voltmeter AC voltage reading is accurate.
- Adjust the appropriate calibration slider until the displayed amplitude is exactly correct. You can use the vertical display cursors to read out the amplitude.

[5]The keyboard label PrtSc was originally chosen when this keyboard button caused the screen to be printed. That function has been replaced by capturing it to a buffer.

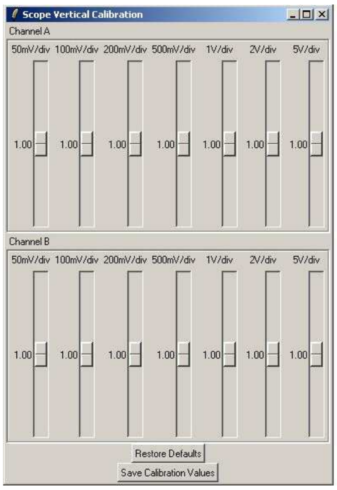


Figure 10: Calibration Controls

The DC method is simpler but can be in error because of DC offset.

- Short the input to the scope channel and verify that the trace is zeroed.
- Remove the short circuit and apply a voltage of known magnitude to the input. The calibration source could be provided by an adjustable DC power supply and a digital voltmeter. (Any meter on the power supply should be checked for accuracy. Analog power supply voltmeters can be extremely incorrect.)
- Adjust the appropriate calibration slider until the displayed amplitude is exactly correct. Then the vertical calibration of the oscilloscope will be as accurate as the digital voltmeter.

The oscilloscope software copies the calibration values to a text file `calibration.cfg` in the current directory when requested by the `Save Calibration Values` button, and when the oscilloscope program is shut down.

The scope software checks at startup for the existence of a text file named `calibration.cfg`. If that file exists, it reads the values into the calibration sliders.

The `Restore Defaults` button resets all the calibration values to 1.0, ie, centre position.

5 Aliasing

The oscilloscope is a *sampled-data-system*. It works by taking a series of samples of the input waveform and displaying them. However, when the signal contains high frequency components compared to the sampling rate, the display may be incorrect. In theory, at least two samples per cycle of the highest frequency present in the waveform are required to reconstruct the waveform correctly.

Some sampled-data-systems have a constant sampling rate. For example, audio is typically sampled at 44.1k samples per second. In that situation, usual practice is to incorporate a low-pass filter such that frequencies above 22 kHz are prevented from entering the system[6]

[6] In practice, the lowpass cutoff frequency is set to somewhat less than half the sampling frequency to allow for the finite rolloff rate of the filter.

Most – if not all – digital oscilloscopes do not incorporate an anti-aliasing filter. The sample rate of a digital scope varies over a wide range of frequencies, and so the cutoff frequency of the anti-aliasing filter would have to do so as well. Combined with the bandwidth requirement, that is a difficult technical challenge. Instead, the oscilloscope relies on the operator to recognize when aliasing is occurring and increase the sample rate until the effect disappears.

A useful strategy in measuring an unknown waveform is to approach it from a high sampling rate (aka time-base setting) and reduce the setting until a readable display appears. It is also required of the operator to know (approximately) the frequency of the waveform that is being observed. That is often the case.

A useful rule of thumb is this: the display must contain about 10 samples per cycle of the waveform to reconstruct it. For this oscilloscope the maximum sampling rate is 20MSamples/second, so it can usefully observe frequencies up to about 2MHz. The analog bandwidth has been designed to be 2MHz to meet this requirement.

6 Safe Measurement Technique

These notes are included for the benefit of those who are new to using an oscilloscope. The information is not unique to this oscilloscope, but applies to most oscilloscope measurement instruments.

Rather than simply state rules and prohibitions, we explain why certain procedures are dangerous and why some techniques should be avoided. This information is provided for guidance in using the oscilloscope and is not intended to replace proper training in working around high voltage circuits.

In general, this oscilloscope may be used safely to observe signals in low-voltage circuits where the power supply is *floating* from the AC line.

6.1 Floating Power Supply

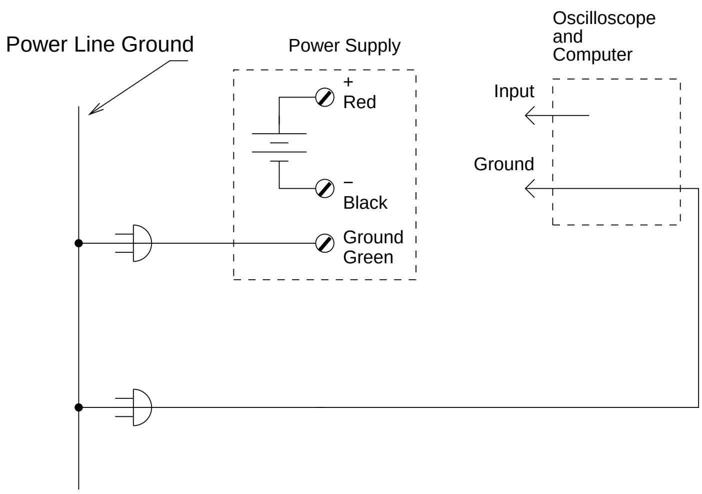


Figure 11: Floating Power Supply

In this context, *floating* power supply is one in which neither terminal is connected to the power line ground.

Consider the circuit shown in figure 11. Like many lab power supplies, the power supply has three terminals: positive, negative and ground. The ground terminal is connected to the third prong on the line cord, which connects to the power line ground wire.

The oscilloscope has two connections: the *input* terminal and *ground* terminal. On the front panel BNC connector, the inner contact is the input, the outer ring is ground. The ground connector finds its way to the AC ground line via the third prong on its line cord.

As shown in figure 11, *either* lead on the oscilloscope can be safely connected to the positive or negative terminal of the power supply. With proper care to avoid short-circuits of the power supply, this is a safe measurement situation.

6.2 Grounded Power Supply

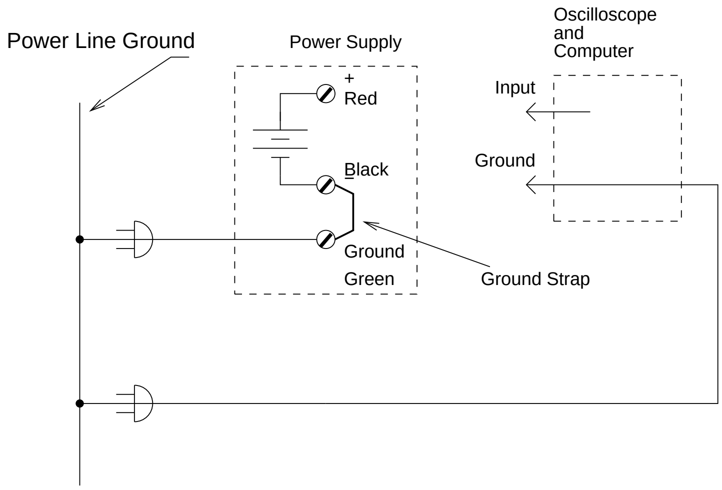


Figure 12: Floating Power Supply

Now consider that the negative terminal of the power supply is connected to its `Ground` terminal, as shown in figure 12.

If the ground terminal of the oscilloscope is connected to the ground terminal of the power supply, then the circuit is in no danger and will work properly. However, **if the ground terminal of the oscilloscope is inadvertently connected to the positive terminal of the power supply, then the power supply will be connected to a short circuit. The power supply short circuit will drive current around the ground connections of the power supply and oscilloscope.** Since lab power supplies are usually current limited to less than an ampere of current, the equipment will likely survive. However, the circuit will not function properly because the power supply is in a short-circuited condition.

To avoid this situation, **do not connect the positive or negative terminal of the lab power supply to the ground terminal. Leave the supply floating.**

6.3 Battery and AC Adaptor Power Supplies

If batteries are used to power the circuit under test, the problematic situation of section 6.2 is not likely to occur, because batteries do not normally have a ground connection to the AC line[7].

[7]The disadvantages of batteries are (a) they run down and (b) they are not current limited. A short circuited battery can do significant damage.

An *AC Adaptor* is essentially a small transformer coupled DC power supply. These are usually supplied with a two-prong line cord, so there is no ground connection to the AC power line ground. The supply is floating so the problem of section 6.2 cannot occur.

6.4 Russian Roulette and AC Line Voltage

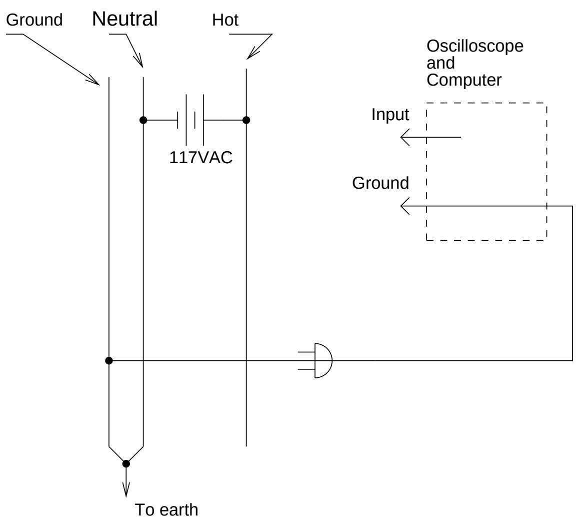


Figure 13: AC Line Voltage

The unsafe situation of an oscilloscope being used to measure AC line voltage, is shown in figure 13. The AC line consists of three connections: the *hot* line, the *neutral* line, and the *ground* wire. For safety reasons, the neutral and ground are connected together and to an earth ground at the system AC distribution panel. The hot and neutral line carry load current in the system – normally the ground wire does not carry any current. Because the neutral is carrying current and because the neutral wire has resistance, at any given point in the system there likely will be a small voltage difference between the neutral and ground wires.

If the ground wire of the oscilloscope is inadvertently connected to the *hot* wire of the AC line, an extremely large short circuit current will flow through the ground connection. Eventually, a circuit breaker will open, a fuse will blow, or the short-circuit current will destroy a conductor. However, until that occurs, the short circuit current can be in the order of hundreds of amperes. This current will destroy the oscilloscope and computer, and the resultant flaming debris may cause injury to nearby living organisms, including humans. It may also start a fire.

Furthermore, connecting the ground lead of the oscilloscope to the AC neutral line causes another problem - it effectively connects the neutral and ground AC lines at that point. Now the neutral current has another path, and some of it will flow through the oscilloscope and computer ground leads. If this current is sufficient, it may damage the oscilloscope and computer.

Notice that *this same situation can occur with equipment that is not transformer isolated from the AC line*. For example, some electronic equipment has a direct connection to the AC line, so that the chassis is connected directly to the neutral line of the AC system. To safely observe the signals in this device with an oscilloscope *the equipment must be isolated from the AC line by a transformer*. The transformer must function as an *isolation* *transformer*, the secondary winding must not have an electrical connection to either of the primary leads, and the transformer must consist of a separate primary and secondary winding.

An *autotransformer* (common trade name *Variac*) is an adjustable transformer that is often used for adjusting line voltage. An auto transformer does *not* have an independent secondary winding and cannot be used to isolate electronic equipment from the AC line.

6.5 Removing the Ground

The potential for a short circuit is reduced if the ground connection is removed from the computer. However, this is extremely dangerous because metallic connections on the computer (such as the shell around a connector) are connected to the ground line. A connection to the AC line puts those metallic points at line potential, presenting a serious shock hazard to the user and possible short circuit if attached equipment is grounded.

The AC ground connection (the 'third prong' on a plug) is there specifically to prevent the chassis of the equipment from assuming a potential that is above ground, and therefore dangerous to a human operator. Removing that ground connection removes any grounding protection. This is a serious violation of health and safety regulations.

Similarily, a battery-powered laptop computer, when disconnected from its line-operated charger, is not connected to the ground line of the AC power system, so it is less likely to cause the kind of short circuit described in the previous section. However, it is extremely dangerous to rely on this. The laptop may itself become live at the AC line potential, which makes it hazardous to the operator and any attached equipment (such as a line-operated video monitor).

6.6 Observation of AC Line Voltages

If you must observe line voltage, here are the rules:

- The oscilloscope must be able to cope with the peak value of the input AC voltage. The Syscomp DS-101 is certified to reliably accept up to 50 volts on its input terminal.

  A *times-ten* oscilloscope probe increases this by a factor of ten. It is absolutely essential to use a probe that can withstand this voltage, and it essential to ensure that the probe cannot inadvertently be switched to a *times-one* setting.

  Notice that the peak value of a sinusoidal voltage is 1.41 times the RMS value. So a 117VAC line voltage will peak at around 170 volts.

- There must be no direct connection to the AC line. If the equipment is line operated, then it must be powered by an isolation transformer (see above).

- It is possible to obtain electronic probes that provide an *isolation barrier* between the line circuit and the oscilloscope. For example, the measurement signal is transferred from the AC line side to the oscilloscope side by means of an optically coupled circuit. There is no electrical connection between the oscilloscope and the AC line. The signal is transferred over a beam of light. This method removes all possibility of short-circuiting the line voltage to ground. See for example `http://www.powertekuk.com/`.

7 Overview of USB Operation

In general, the operation of the USB connection is seamless and invisible to the user. Operation of the oscilloscope is usually as simple as plugging it in to an USB port and running the oscilloscope GUI software. However, it may be useful to understand some of the details.

The USB interface uses a USB-Serial chip FT232BM from FTDI. This chip, with the appropriate driver software on the host PC, emulates a serial port. Consequently, the Tcl/Tk GUI software can access the hardware just as if it was accessing a device connected to the host serial port. This is orders of magnitude simpler than dealing with USB, which is extremely complicated. We refer to this as a `USB-serial` interface.

The `USB-serial` has major advantages over the traditional serial port. First, the data transfer rate is much faster (especially using the USB2.0 standard). Furthermore, power is transmitted from the host to the hardware over the USB cable so that an AC adaptor is not needed. Third, the USB system handles enumeration automatically so that multiple devices are accessed correctly without manual configuration.

Under Windows, the FTDI `USB-serial` drivers are automatically loaded by the `Install` program, so the user should not normally be required to intervene.

Under Linux, the FTDI drivers are included in the Linux kernel since 2.4, so they do not need to be installed under Linux.

When a `USB-serial` device is plugged into the host computer for the first time, the host USB system detects a new device and allocates it to a serial port. In Windows, this is a COM port. In Linux, this is a device such as `/dev/ttyUSB0`.

Thereafter, the operating system always associates that hardware with that serial port, even if it is plugged into a different USB port.

In Linux, the default permission of the `/dev/ttyUSBx` ports is set for root access only, so the permissions must be changed as described under 8.2. (The `x` in `/dev/ttyUSBx` represents a number for a ttyUSB port, something like `ttyUSB0` or `ttyUSB1`, and so on.)

8 Troubleshooting

In addition to the information provided here, you may also find useful information and screenshots in sections 16.1, 16.2 and 16.3.

8.1 Microsoft Windows

The system requirements for running the host software are:

Processor	Pentium, 233MHz minimum or equivalent
RAM	64MB
Hard Drive space	5MB
Video	800 x 600 minimum resolution, 16 bit colour
Ports	One USB port
Operating Systems	Windows 98SE, Windows ME, Windows 2000, Windows XP

Instructions for installing the software are on the CDROM included with the hardware. Use a text editor to open and read the file README, and the follow directions from there.

Installation screenshots are given in section 16.4, page 48.

8.1.1 Windows XP

1. Install the oscilloscope software per the instructions on the CDROM.
2. Boot up your computer and change to the directory where you installed the scope software.
3. Using the supplied USB cable, plug the oscilloscope into a computer USB port. If you have more than one USB port, you can chose any port.
4. The operating system should now recognize that you have plugged in a USB device and issue a chime noise.

5. Go to: Start -> Settings -> Control Panel -> System -> Hardware -> Device Manager
6. On the Device Manager panel, click on `Ports (COM and LPT)`. You should see an entry like `USB Serial Port (COM 5)`. This is the COM port that the operating system has assigned to the oscilloscope for this session. Make note of the port, in this example COM 5.
7. Start the oscilloscope Tcl program by double clicking on its icon. The exact name will vary, but it should be something like `scope-100.tcl`. It may complain with an `Unable to Connect` warning message. Ignore this and close the warning message. The oscilloscope GUI should now be on the screen.
8. On the oscilloscope GUI, open Hardware -> Connect. Select the COM port that you found previously. In our example, that would be COM 5. Click on `Save and Exit`. This causes a small text file `scopeport.cfg` to be written to the directory where the scope program was launched. Thereafter, the oscilloscope GUI Tcl program will read this file and automatically select that particular COM port.
9. Open `Hardware` again and click on `Connect`. The oscilloscope GUI status message at the top of the screen should show `Connected`. This is a Happy Moment, because your computer is now talking properly to the oscilloscope hardware.
10. Operate the frequency, amplitude or offset controls on the oscilloscope GUI. As you do so, the GUI sends commands to the hardware and you should see flashing from the `Activity` LED on the hardware front panel.

From now on, it should be sufficient to boot up your computer, plug in the oscilloscope and double-click on the desktop icon for the oscilloscope.

8.1.2 Manually Assigning a COM Port Number in Windows XP

It may be useful to know how to set the COM port manually.

For example, under certain circumstances, it is possible for the operating system to assign a COM port number that too high to be usable by the host software. In our situation here at the Syscomp factory, we test each instrument by plugging it into a USB port. Each time the operating system sees a new instrument, it assigns it a new COM port number. If the COM port number is greater than 9, it cannot be selected by the host software. Then you must re-assign the port number manually.

1. Plug in the problem instrument to a USB port.
2. Go to: Start -> Settings -> Control Panel -> System -> Hardware -> Device Manager
3. Double-click on `Ports (COM and LPT)`. This opens to show any USB-serial port assignments. Let's say that it shows `USB Serial Port (COM12)`. This exceeds COM9, so we have to manually reset the COM port number.
4. Double-click on the entry `USB Serial Port (COM12)`. This opens a new dialogue box, `USB Serial Port (COM12) Properties`.
5. Select `Port Settings` and click on `Advanced`.
6. This opens a new dialog box, `Advanced Settings for COM12`. In the upper left corner, there is a scrollbox `COM Port Number`. Use the up/down arrows to scroll through the possible COM port assignments.

7. For the new COM port assignment, it's best to choose a COM port number 4 or larger. (Lower numbers may conflict with a USB keyboard or mouse). Suppose we decide to move to COM5. The scrollbox says `COM5 (in use)`. Select it anyway. Click `OK`.
8. A warning pops up that the COM port is in use and asks if you want to continue. Click on `Yes`.
9. Back out through the menues until you have closed the Device Manager panel. Re-open it and examine the `Ports (COM&LPT)`. This time it should read `COM5`.
10. Back out to a clean desktop and restart the instrument program. This time, it should connect properly.
11. If you are connecting multiple instruments, you may need to do this for each instrument. However, having once done the assignment for a given instrument, the operating system associates that instrument with the chosen COM port and connection should be automatic.

8.2 Linux

These troubleshooting notes are specific to Suse Linux 9.2, but should apply in general. They also assume a working knowledge of Linux and its variants.

Overview

If the software does not operate correctly, here are some things to check. They are subsequently explained in detail.

- The operating system is too old and does not contain the necessary drivers for the usb-serial ports.
- The operating system for some reason is not recognizing the usb device and assigning it to a usb-serial port. This can occur if the usb device has `root` ownership and permissions. The permissions must be changed to allow a user-mode program to access the usb port.
- The operating system is assigning the hardware to some usb-serial port but the Tcl/Tk program is not automatically selecting that particular port. You'll need to select the usb-serial port manually, using the controls in the Tcl/Tk program.
- The `wish` program, which is the interpreter for all Tcl/Tk programs, is not being found by the operating system. Locate it and change your path so that it is found.
- The device is being recognized and connects properly, but does not respond properly to certain controls. Use the instructions in section 13.2 to send commands to the hardware to determine how it is functioning.

1. **Check the kernel version**.
   The drivers for the FTDI USB-Serial interface are a standard part of the Linux kernel from version 2.4 onward. To check that you have a sufficiently modern kernel, run the `dmesg` command piped to the `more` command.

   ```
   phiscock@panther: dmesg | more
   ```

   Examine the first few lines, which should be something like this:

   ```
   Linux version 2.6.8-24-default (geeko@buildhost) (gcc version 3.3.4 (pre
   3.3.5 20040809)) #1 Wed Oct 6 09:16:23 UTC 2004
   ```

   In this case, the kernel is 2.6.8-24, so it contains the FTDI drivers.

   If your kernel version is older than this, you may have to upgrade the kernel or install a driver module.

2. **Install the software**:
   per the instructions on the CDROM.

3. **Determine the serial port used by the USB driver**. In this step, we'll use the `dmesg` command to determine which serial (COM) port is being assigned to the oscilloscope when it is plugged in.

   Execute `dmesg` to get an idea of the most recent kernel messages. Using the USB cable, connect the scope hardware to a USB port. Execute `dmesg` again, and you should see something like this as the last entry in the `dmesg` printout:

```
usb 4-2: new full speed USB device using address 4
usb 4-2: Product: USB <-> Serial Cable
usb 4-2: Manufacturer: FTDI
usb 4-2: SerialNumber: 00000001
ftdi_sio 4-2:1.0: FTDI FT232BM Compatible converter detected
usb 4-2: FTDI FT232BM Compatible converter now attached to ttyUSB0
```

Unplug the USB cable and run `dmesg` again and see something like this:

```
usb 4-2: USB disconnect, address 4
FTDI FT232BM Compatible ttyUSB0: FTDI FT232BM Compatible converter now
disconnected from ttyUSB0
ftdi_sio 4-2:1.0: device disconnected
```

Evidently the USB device is being assigned to device `ttyUSB0`.

This shows that the USB device is being recognized by the operating system and assigned to a usb-serial port.

4. **Set the permissions for the USB-Serial port**
   The default situation is that root is the owner of the USB serial port `ttyUSB0` and operation is restricted to root. For an ordinary user to access the port, the permissions must be changed.

   First, we will show how to do this manually in section 8.2.1. However, Linux is usually set up so that the permissions revert back to root mode every time the USB is plugged and unplugged, and every time the system is rebooted. Therefore, we need to modify the system so that this is done automatically, ie, so that the port permissions are set to user mode by default. This is shown in section 8.2.2 below.

8.2.1 Manually Changing Device Port Permissions

Change to the `/dev` directory.

```
phiscock@linux:~> cd /dev
```

Check the permissions on the `ttyUSB` ports:

```
phiscock@linux:/dev> ls -l ttyUSB*
crw-rw----  1 root uucp 188, 0 2005-11-07 18:25 ttyUSB0
crw-rw----  1 root uucp 188, 1 2004-10-02 01:38 ttyUSB1
<others deleted>
```

In this case, the owner (root) has read-write access. The group that root belongs to, uucp, also has read-write access. Others (that's you) have no access at all. To open up the port to user access, enter root mode using the `su` command:

```
phiscock@linux:/dev> su
```

The system asks for the root password. Enter it. Now you can change the permissions (mode) for the ports. In this case, we'll use the `chmod` command to add read and write permission for 'others'. For example, the first command below says: *change the mode of device ttyUSB0 to add read permission for 'others'*. The second command does the same for write permission.

```
linux:/dev # chmod o+r ttyUSB0
linux:/dev # chmod o+w ttyUSB0
```

Check the permissions again:

```
ls -l ttyUSB*
crw-rw-rw-  1 root uucp 188, 0 2005-11-07 18:25 ttyUSB0
crw-rw-rw-  1 root uucp 188, 1 2004-10-02 01:38 ttyUSB1
```

That's it. You should now be able to access those ports from user mode. Exit from root to user mode.

Incidentally, you may be able to change the permissions by logging in as root and then using the features of the KDE or Gnome desktop to change the permissions.

8.2.2 Setting Default Port Permissions to User Mode: Suse 9.2

This change will ensure that the serial-usb ports are always created with user mode access.
The default permissions for user devices are contained the file: `/etc/udev/permissions.d/50-udev.permissions`
We have to modify the entry for the `ttyUSBx` ports so that the default is user mode.

1. Change to the directory `/etc/udev/permissions.d` and check that the file `50-udev.permissions` exists.

2. If the file exists, [8] enter root mode, and copy the existing file so you have a copy of the original.

   ```
   cp 50-udev.permissions 50-udev.permissions-orig
   ```

3. Now open the file `50-udev.permissions` with your favourite editor. Find the entry that says:

   ```
   ttyUSB*:root:uucp:660
   ```

   Change that to read:

   ```
   ttyUSB*:root:uucp:666
   ```

   Save the file. Now every time a `ttyUSBx` port is created, you should be able to access that port without problems.

8.2.3 Setting Default Port Permissions to User Mode: Fedora Core 6

This note[9] applies to Fedora Core 6, kernel 2.6.19-1.2911.fc6.

Look in `/etc/udev/rules.d/50-udev.rules` for the line:

```
KERNEL=="tty[A-Z]*", NAME="%k", GROUP="uucp", MODE="0660"
```

Change the mode value to 0666.

---

[8] If the fi le does not exist, please let us know the name of the Linux distribution and we'll look for another solution.
[9] Kindly supplied to us by John Foster.

8.2.4 Running the Program

1. Change to the directory where the program resides. The `wish` interpreter is required to run the tcl program. It is normally included with a Linux distribution, so it is probably present on your system. You can find out by issuing the `which` command.

   ```
   phiscock@linux: which wish
   /usr/bin/wish
   ```

   If this doesn't turn it up, use the 'find' command, starting at the root directory '/'. If it is on the system, then add that location to your path.

   ```
   phiscock@linux: find . -name wish
   <much deleted>
   /usr/bin/wish
   ```

2. Start `wish`.

   ```
   phiscock@linux:~/eelab/demos> wish
   ```

3. A new small window will appear. This is the container for any program that `wish` executes. The cursor remains where the `wish` command was run.

4. Click in that window and run the command:

   ```
   % source main.tcl
   ```

   using the correct name for the oscilloscope program. The oscilloscope GUI should now run correctly.

There are many other ways to start the program. For example, the command `wish scope-101.tcl` (substitute the correct name of the tcl program) can be used. As well, the KDE and Gnome graphical user interfaces be used to set up an icon on the desktop. Then clicking on that icon will start the program.

8.2.5 Device Properties using `usbview`

In general, it's not necessary to know anything about the USB properties of the hardware in order to use it. However, if you do want to inspect those properties, `usbview` is useful.

It is likely that you will have to install `usbview` from your Linux distribution disks.

Once `usbview` is installed, (figure 14) you can use it to determine whether a USB device is recognized by the operating system USB. As a USB device is plugged and unplugged, an entry appears and disappears in the `usbview` window.

Clicking on an entry opens up a list of USB properties of the device.

Notice that `usbview` does not indicate the serial port (`/dev/ttyUSB2` or whatever) that the operating system has assigned to this device. You must use `dmesg` for that purpose.

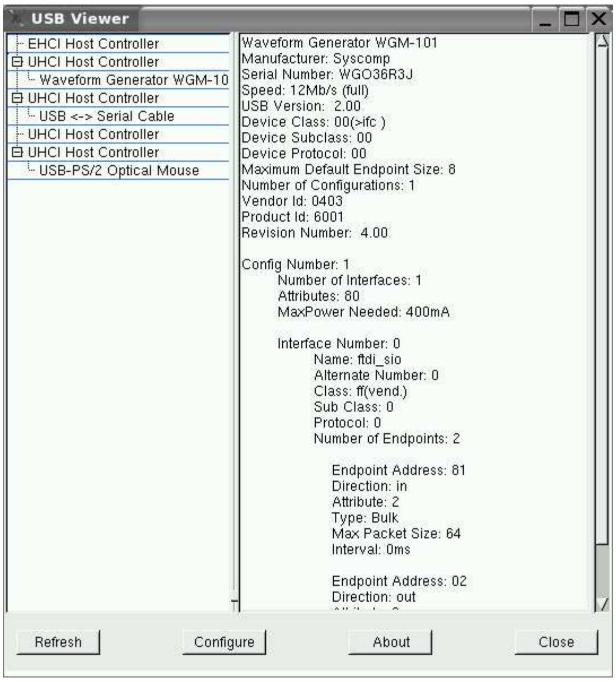


Figure 14: Usbview

9 Technical Description

A block diagram of the oscilloscope is shown in figure 15.

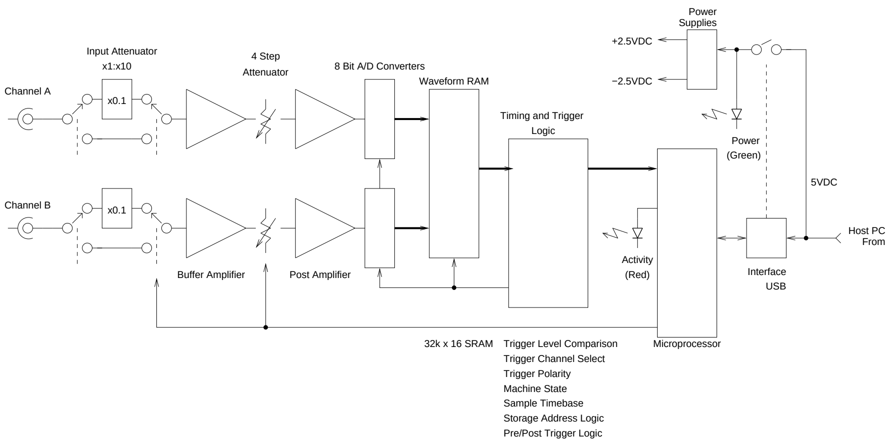


Figure 15: Oscilloscope Block Diagram

9.1 Input Channel Overview

There are two identical input channels, **Channel A** and **Channel B**. Each channel has its own 7 step gain adjustment circuitry and A/D converter.

The input channel circuitry serves several purposes:

- Protect the internal circuitry from an excessive input voltage.
- Present a constant, large impedance to the circuit under test, regardless of gain setting.
- Provide the necessary voltage and current gain to drive the A/D converter.
- Adjust the amplitude of the signal so that it is a good fit to the input range of the A/D converter. Too large a signal will overload the A/D converter and lead to clipping distortion. Too small a signal will fail to mask the quantization noise of the A/D converter.

The circuitry must provide these functions over the analog bandwidth of the instrument.

Each input appears as a 1MΩ resistance in parallel with approximately 27pF capacitance.

A compensated **x10** probe may be attached to the input. A **x10** probe allows larger input signals (by a factor of 10) and reduces the load capacitance on the circuit under measurement.

The description of each functional block is as follows:

Input Attenuator Signals enter the oscilloscope at the left edge of the diagram. The **x1:x10** attenuator is selected by a two position relay and adjusts the amplitude of the input signal to avoid overloading the gain adjustment circuitry.

Buffer Amplifier Presents a large impedance to the input signal and provides current gain to drive subsequent circuitry.

4 Step Attenuator This electronic attenuator adjusts the magnitude of the signal in a 1:2:5 sequence as required to optimize the signal-noise ratio into the A/D converter. With the input attenuator, this gain switch provides 8 steps of gain. One step overlaps, so that there are 7 distinct gain settings.

Post Amplifier This amplifier provides voltage and current gain to drive the A/D converter.

A/D Converter The A/D converter samples the input analog signal at a rate suited to the timebase setting. It outputs 8 bit digital values.

Waveform RAM During capture of waveform data, samples from both A/D converters are simultaneously stored in this memory. Up to 32k samples are stored from each channel.

Timing and Trigger Logic The trigger circuitry is entirely digital. The trigger logic operates by comparing 8 bit digital values from the A/D converter with a preset 8 bit digital value.

This unit also contains the high-speed timing logic, which generates the A/D and memory control signals.

Microprocessor The microprocessor receives and decodes commands from the USB interface. The microprocessor flashes the *Activity* LED each time a command is received.

The microprocessor then adjusts the settings of the input channel and digital timing circuitry as requested by host commands.

USB Interface This device handles the USB communication tasks and translates incoming messages into serial asynchronous format.

Power Supply Power (5VDC) is provided to the unit via the USB cable. When the scope is first plugged into a USB connector on the host computer, the majority of the scope is disabled. The scope USB interface requests a supply current 500mA. If that is acceptable to the host, the host then allows the USB interface to energize the remainder of the circuit. This causes the scope front-panel *Power* LED to illuminate.

The power supply also generates voltages other than 5VDC as required by the circuitry.

10 Adjustments

The oscilloscope has been adjusted before shipping, so it should not need adjustment before use. These instructions are provide for reference purposes.

10.1 Input Compensation Capacitors

The schematic of the input circuitry of the oscilloscope vertical preamplifier is shown in figure 16.

The two channels are identical. Referring to channel A, the input network is selected by a two-position relay. In the `x1` position, the signal goes directly to the buffer amplifier U3A. In the `x0.09` position, the signal is attenuated by that factor.

Stray capacitance at the output of the `x0.09` voltage divider has the potential for limiting the bandwidth. To make the divider frequency independent, the resistive divider R1, R2 is accompanied by a capacitive voltage divider VC1, C1. The capacitive divider must be adjusted to have the same division ratio as the resistors. This is accomplished by adjusting VC1.

To do so, apply a square wave input signal to Channel A. Set the vertical gain to its least sensitive position, 5 volts/division. This ensures that the relay is selecting the `x0.09` voltage divider. Adjust the magnitude of the square wave such that it makes a suitable display. Adjust the oscilloscope timebase such that the leading edge of the square wave is visible.

Locate VC1 on the circuit board. With a tiny screwdriver, adjust VC1 until the square wave shows the fastest possible rise time without overshoot. It is best to use a screwdriver with an insulated shaft because a non-insulated screwdriver will connect human body capacitance into the circuit which affects its operation. If the screwdriver is not insulated, make an adjustment and then remove the screwdriver to see the effect.

In the `x1` position, the variable capacitance VC2 should be adjusted so that the input capacitance seen at the BNC connector J1 is the same in both positions of the relay. This is not particularly important if the oscilloscope is connected to a circuit with a direct connection. However, it will affect the operation of a `x10` probe if that is used. It will also affect the waveform if the source impedance approaches the input impedance of the oscilloscope.

To adjust VC2:

- Adjust VC1 as described previously.
- Attach a `x10` scope probe to channel A.
- With the preamp set to the 1 V/div position, connect the scope probe to a square wave signal. Adjust the `x10` scope probe compensation to obtain the best possible square wave.
- Change the preamp vertical gain setting to the next most sensitive position, 500mV/div. (You should hear the relay click as you switch between these positions).
- Identify adjustable capacitance VC2. Adjust the variable capacitance VC2 so that the waveform has the same appearance that it did in the 1 V/div position.

The Channel B capacitors VC3 and VC4 are adjusted in the same manner.

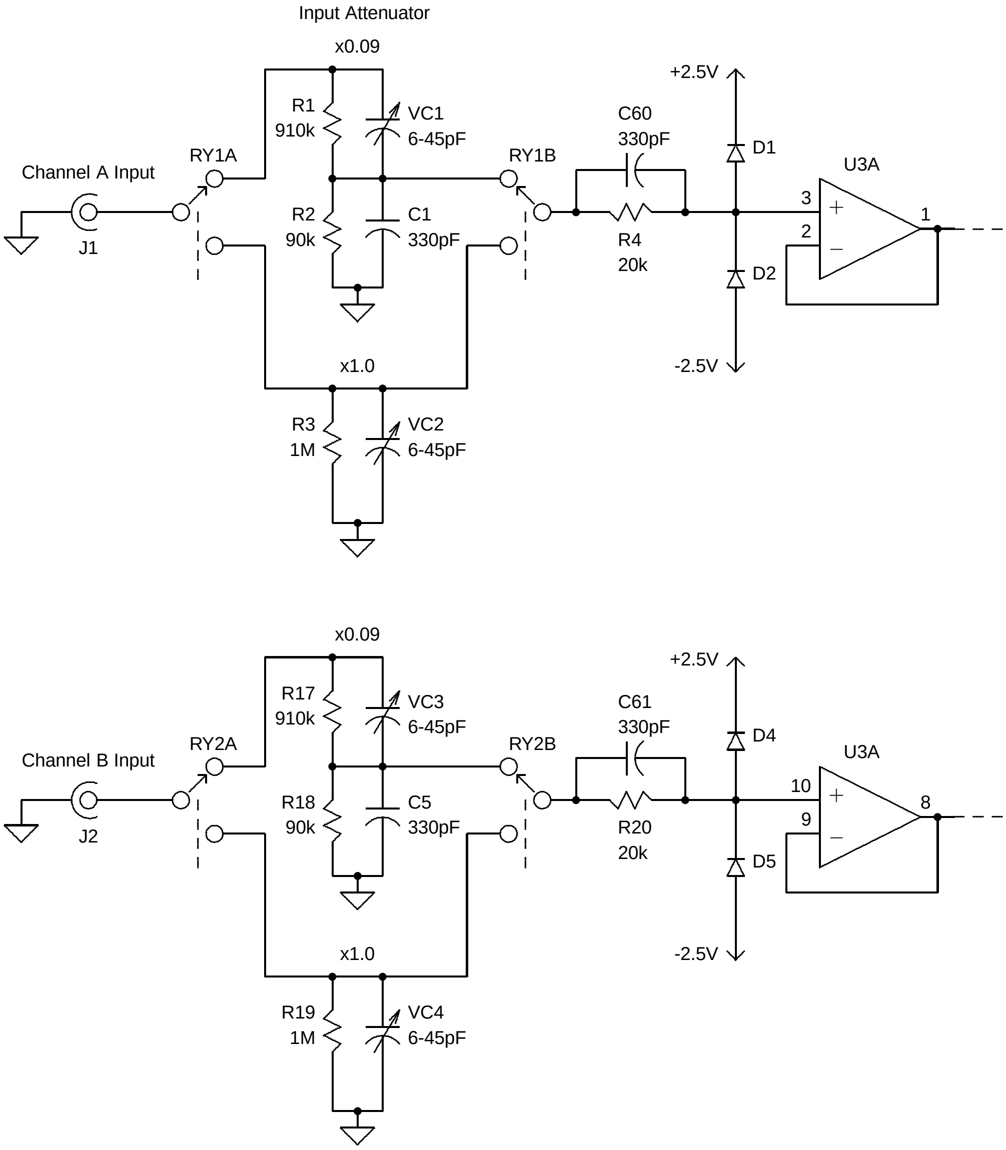


Figure 16: Scope Input Compensation Network

10.2 Offset Adjustment Resistors

Again, this adjustment is done before shipping the oscilloscope, so it should not be required by the user.

When the input to the oscilloscope preamp is zero volts, the input to the 8 bit A/D converter should nominally be 1.03VDC. The 8 bit A/D output can assume values between 0 and 255, so a zero volt input to the preamp should correspond to the mid-point, a reading of about 127. However, because of tolerances in the component values and in the A/D, this may require adjustment.

This is not a critical adjustment, because the zero line may be adjusted in software. However, if a zero input does not correspond to a count of 127, then the dynamic range of the signal is assymetrical and is more likely to clip on one peak than the other. As a result, significant offsets are undesireable.

For Channel A, the offset potentiometer RV1 is used to adjust the offset. Set the trigger mode to `AUTO` so that the scope continuously acquires a waveform. With no input to the preamplifier, adjust RV1 until it corresponds to the zero line on the display.

Similarily, use RV2 to adjust the offset on Channel B.

11 Oscilloscope Commands

These commands are low-level instructions to the scope hardware. The commands (and where relevant, messages back from the hardware) are ASCII strings so that they can be generated easily by software or a human operator.

There are two situations where the interface commands may be useful.

- The scope hardware may be operated directly from a terminal emulator program such as `Hyperterminal` under Windows, `Minicom` or `Seyon` under Linux.
- The commands must be known to create a scope control program with different functionality. For example, a program could be created to sweep the read the oscilloscope and plot the results on a strip-chart type of display. This new program needs to issue commands to the hardware.

  Should you decide to attempt such a project, the Tcl source code for the Oscilloscope GUI, which is provided on the accompanying CDROM, will be a useful source of ideas in controlling the scope.

  There is no requirement that the controlling program be written in the Tcl language. Any program that can issue ASCII strings to a serial port will be capable of controlling the scople hardware. (i.e `Matlab`, `Visual Basic`)

11.1 Using the Debug Console

When you send a command to the scope, it does not echo any confirmation back to the host terminal. This is because you are doing with the terminal exactly what is done with a control program, and responses from the scope would slow down the overall operation of the system.

Each command consists of an ascii string of characters, such as T55`<cr>` to set the trigger level to 55, where `<cr>` is a carriage return character.

In addition to the material in the user manual, you can see commands being sent to the oscilloscope by selecting

```
View -> Debug Console.
```

This brings up a terminal screen which lists commands as they are being sent to the scope hardware, and some other debug information.

This information scrolls past rather quickly when the scope is in auto trigger mode, because it is repeatedly obtaining data from the hardware. To slow this down, put the scope in 'Manual Trigger' mode. Now each time you hit the 'manual trigger' button, the debug screen will show the commands that were sent to the scope hardware.

Now you can change control settings on the scope GUI and see the corresponding commands as they are sent to the scope hardware. For example, move the trigger level cursor on the scope screen and you will see a series of trigger level commands like T55 T43 T27 being sent to the hardware. Similarily, changing the vertical preamplifier gain settings to show the corresponding hardware commands.

11.2 General Commands

Command Name: Identify
Usage: i
Description:
Request device identification. Returns device type and version.

Command Name: Query Trigger State
Usage: q
Description:
Returns the current state number of the internal state machine. The states are defined as follows:

S0	The scope is capturing samples
S1	The scope is capturing 16k samples before a trigger can occur
S2	The scope is resetting is waiting for the previous trigger to clear
S3	The scope is capturing 16k samples after the trigger
S4	The scope has completed the capture
S5, S7	The scope is reading data from the SRAM
S6	The scope is armed and waiting for a trigger event

For example, the scope will return ”6” when it is armed and waiting for a trigger event. The scope will return ”5” when the capture process is complete and data has been returned to the PC.

11.3 Control Register Commands

Command Name: Set Trigger Level
Usage: tX
Description:
Set the trigger level. The argument X must be an integer $(0 < X < 255)$. The trigger level voltage is determined by the following formula:

$$V_{TL} = (128 - X) * V_{EAD} \tag{1}$$

where V_{TL} is the trigger level voltage, X is the integer value of the trigger level register, and V_{EAD} is the effective A/D step size based on the current preamp setting (refer to Tables 1 and 2).

Command Name: Modify Preamp Settings
Usage: pX
Description:

Change the preamp settings. The argument X is the integer to write into the preamp control register. The preamp control register is shown in Figure 11.3.

7	6	5	4	3	2	1	0
VB0	VB1	VB2	ATTEN −B	VA0	VA1	VA2	ATTEN −A

Figure 17: Preamp Control Register

Bit 0 ATTEN-A Attenuation Select Channel A. Logic 1, x1 setting, relay engaged. Logic 0, x10 setting, relay released.

Bit 1:3 VA2:0 Channel A Vertical Setting.

Bit 4 ATTEN-B Attenuation Select Channel B. Logic 1, x1 setting. Logic 0, x10 setting.

Bit 5:7 VB2:0 Channel B Vertical Setting.

The preamp control values are shown in Table 1 for channel A, and Table 2 for channel B.

Preamp Gain	VA0:2	Atten-A	Register Value (Integer)	Effective A/D Step Size	Intended Vertical Deflection	Input Limit
1.0	000	1	1	1.92mV	50mV/div	500mVpp
0.5	100	1	9	3.84mV	100mV/div	1Vpp
0.25	010	1	5	7.68mV	200mV/div	2Vpp
0.1	110	1	13	19.2mV	500mV/div	5Vpp
0.05	100	0	8	38.4mV	1V/div	10Vpp
0.025	010	0	4	76.8mV	2V/div	20Vpp
0.01	110	0	12	192.0mV	5V/div	50Vpp

Table 1: Preamp Channel A Control Values

Preamp Gain	VB0:2	Atten-B	Register Value (Integer)	Effective A/D Step Size	Intended Vertical Deflection	Input Limit
1.0	000	1	16	1.92mV	50mV/div	500mVpp
0.5	100	1	144	3.84mV	100mV/div	1Vpp
0.25	010	1	80	7.68mV	200mV/div	2Vpp
0.1	110	1	208	19.2mV	500mV/div	5Vpp
0.05	100	0	128	38.4mV	1V/div	10Vpp
0.025	010	0	64	76.8mV	2V/div	20Vpp
0.01	110	0	192	192.0mV	5V/div	50Vpp

Table 2: Preamp Channel B Control Values

Example: We would like to set channel A to 100mV/div and channel B to 5V/div. To set preamp channel A to 100mV/div an integer value of 9 must be written into the register. To set preamp channel B to 5V/div an integer value of 192 must be written into the register. The value written into the register to set up both channels is the sum of these two integers $192 + 9 = 201$. The value is sent to the hardware by the command *p201*.

Command Name: Modify Scope Control Register
Usage: rX
Description:
The scope control register is used to configure capture settings such as sample rate, and trigger source. The argument X is the integer to write into the scope control register.

7	6	5	4	3	2	1	0
n/c	TRIGSRC	TRIGPOL	INTEXT	SAMRATE 3	SAMRATE 2	SAMRATE 1	SAMRATE 0

Figure 18: Scope Control Register

Bit 3:0 SAMRATE Sample rate. Refer to Table 11.3

Bit 4 INTEXT Internal/External Trigger Select. When triggering off of channel A or B this bit must be set to one. For manual triggering, set this bit to zero.

Bit 5 TRIGPOL Trigger polarity select. Set this bit to zero for rising edge trigger and one for falling edge.

Bit 6 TRIGSRC Trigger source select. Set this bit to zero to trigger off channel A. Set this bit to one to trigger off channel B.

Bit 7 Not Used

Sample Rate (Samples/S)	SAMRATE3:0
20M	0000
10M	0001
5M	0010
2.5M	0011
1.25M	0100
625k	0101
312.5k	0110
156.25k	0111
78.125k	1000
39.0625k	1001
19.53125k	1010
9.765625k	1011
4.8828125k	1100

Table 3: Oscilloscope Sample Rates

Command Name: Change the Baud Rate
Usage: bX
Description:
Change the baud rate at which the scope transmits and receives. Table 11.3 gives the relationship between the parameter X and the baud rate setting. The scope defaults to a baud rate of 38400 when it powers up. For example, to set the communication baud rate of the scope to 115kbps use the command *b3*.

The scope defaults to a baud rate of 38400 on startup. If you issue the command ”B3” it will immediately switch to 115kbps. After issuing the B3 command you need to reconfigure the COM port on the computer to 115kbps in order to talk to the device. It is a good idea to send a few line ends after changing the baud rate to flush out the buffers on the PC and the scope and to make sure the UARTs are aligned.

Parameter (X)	Baud Rate (bps)
191	2400
95	4800
47	9600
31	14.4k
23	19.2k
15	28.8k
11	38.4k
7	57.6k
5	76.8k
3	115.2k
1	230.4k

11.4 Memory Commands and Organization

11.4.1 Memory Organization

The oscilloscope performs a capture by writing data from the A/D converters into an internal memory block. Each input channel of the scope has 32kB of storage memory. When a capture is initiated, the scope constantly captures data at the specified sample rate and waits for a trigger event. The memory acts like a circular buffer during this phase of the capture. The memory address is incremented each sample until the counter reaches the top of the memory (32kB) at which point the counter rolls over and begins again at zero. The memory address where a trigger event occurs is called the *trigger point*. The scope will capture 16kB of data after the trigger point. This provides 16kB of data before the trigger, and 16kB of data after the trigger.

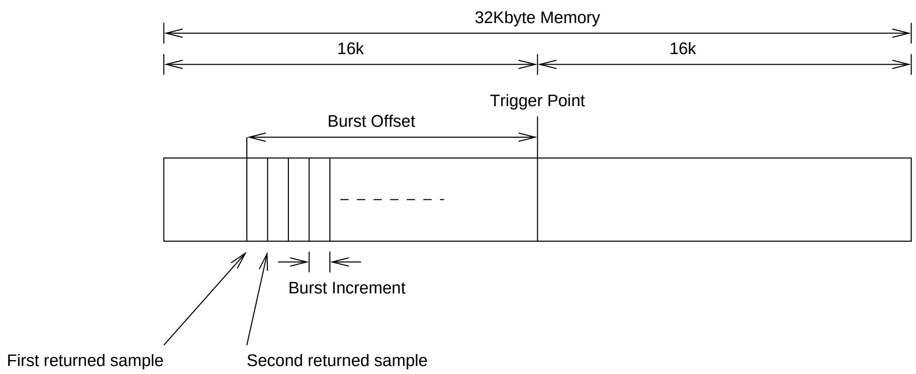


Figure 19: Memory and Associated Parameters

Data is sent from the scope to the PC in bursts. Each burst contains data from both channels. The *burst offset* defines how many samples, or memory locations, the scope will go back in memory from the trigger point to begin reading back data. The burst offset allows the user to start reading back data from any point in the internal memory. The *burst increment* defines the spacing (or number of memory locations) between samples in the internal memory.

The burst increment allows you to subsample from the SRAM. In the scope software we subsample the SRAM using different burst increments to create different timebase settings by reading out samples spaced at different intervals.

Setting the burst increment to one would read back each successive sample, setting the burst increment to 3 would read back every third sample, etc. The *burst count* defines how many samples are read back from each channel. The total number of samples returned is two times the burst count, since data is returned from both channels.

11.4.2 Memory Commands

The registers which contain the burst count, the burst increment, and the burst offset are 16-bit registers. They are accessed by writing to the lower eight-bits and upper eight-bits separately.
Command Name: Change low byte of burst increment. (That's the letter *el*, not the number *one*).
Usage: lX
Description:
X is the integer value to be written into the register.

Command Name: Change high byte of burst increment.
Usage: hX
Description:
X is the integer value to be written into the register.

Command Name: Change low byte of burst count.
Usage: cX
Description:
X is the integer value to be written into the register.

Command Name: Change high byte of burst count.
Usage: vX
Description:
X is the integer value to be written into the register.

Command Name: Change low byte of burst offset.
Usage: dX
Description:
X is the integer value to be written into the register.

Command Name: Change high byte of burst offset.
Usage: uX
Description:
X is the integer value to be written into the register.

11.5 Other Commands

Command Name: Initiate a capture.
Usage: g
Description:
This command will begin a capture sequence. The data acquisition system will be armed and will wait for a trigger event (on channel A, B, or manual, depending on the current control register settings). After the trigger

event the system will complete the data capture and send back the data to the PC.

Command Name: Manual Trigger.
Usage: m
Description:
Manually trigger the scope. For best results, switch to external trigger select (i.e. set INTEXT in the scope control register to 0) before initiating a manual trigger.

Command Name: Set ASCII Data Mode.
Usage: A
Description:
Issuing this command to the scope will cause it to return data in ASCII format after a capture. This is the default mode and is slower than binary mode. The format for ASCII data mode is:
D A0 B0 A1 B1 A2 B2 ...
The first character 'D' indicates that the scope is returning a data stream. A0 is the first sample from channel A, A1 is the second sample from channel A, B0 is the first sample from channel B, etc. For example:
D 128 64 134 59 144 55 ...

Command Name: Set Binary Data Mode.
Usage: B
Description:
Issuing this command to the scope will cause it to return data in binary format after a capture. Only sample data is affected, text strings are unaffected. Binary data is returned interleaved starting with one sample from channel A, then one sample from channel B, then the next sample from channel A, etc. Binary data mode is faster since each 8-bit character returned represents one sample.

12 Example: Performing a Capture

The following steps must be taken to set up the scope for a capture:

1. Write to the preamp control register to set up the hardware preamp for channels A and B.
2. Write to the trigger control register to set up the trigger level.
3. Write to the scope control register to set up the sample rate, and trigger settings.
4. Write to the memory control registers to set up the parameters for reading back data after the capture has completed.
5. Initiate the capture.

In this example we will set up the scope with the following parameters:

Parameter	Value
Preamp Settings	CHA: 500mV/div CHB: 2V/div
Trigger Level	750mV
Trigger Source	Channel A
Trigger Slope	Rising (Plus)
Timebase	1ms/Div, 10 samples per division
Trigger Point Display	End of second division

12.1 Preamp Settings

From table 1 a register value of 13 is required to set channel A to 500 mV/div, and a value of 64 is required to set channel B to 2 V/div. The total register value is $13 + 64 = 77$ and is written into the register by sending the command *p77*.

12.2 Trigger Level Settings

Using the equation for the trigger level voltage and the effective A/D step voltage from table 1:

$$V_{TL} = (128 - X) * V_{EAD} \quad (2)$$

$$750\text{mV} = (128 - X) * (20\text{mV}) \quad (3)$$

$$X = 90.5 \quad (4)$$

$$X = 90 \quad (5)$$

$$(6)$$

Note that X must be an integer. This value is written to the trigger level register with the command *t90*.

12.3 Timebase and Trigger Point Settings

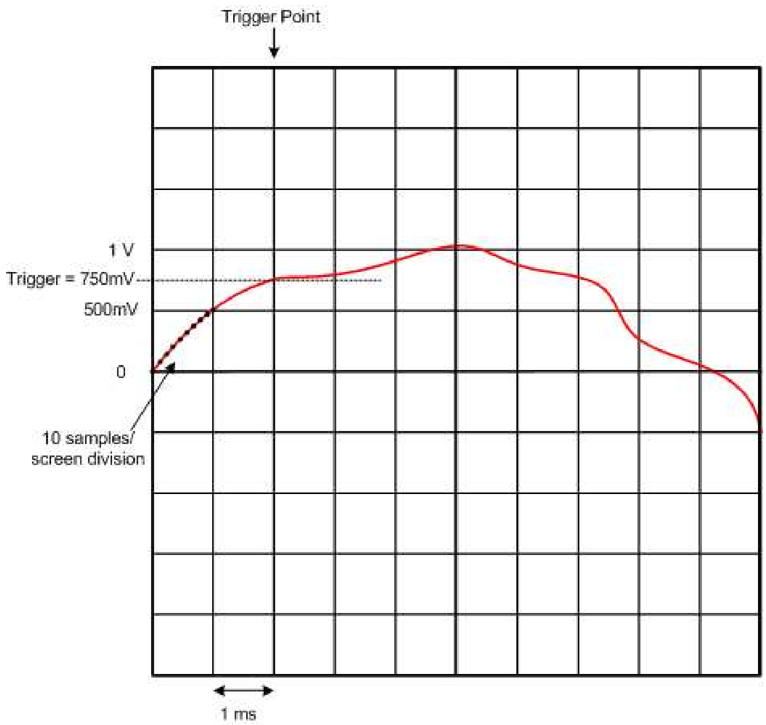


Figure 20: Example Waveform Showing Capture Parameters

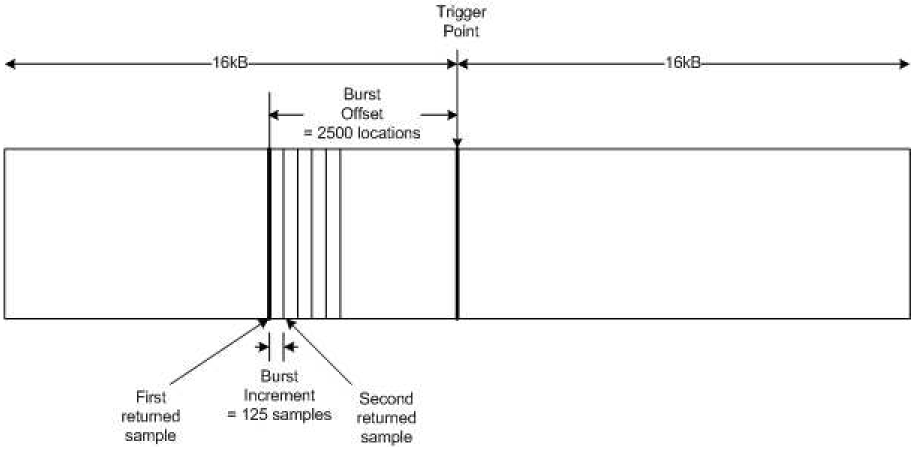


Figure 21: Example Internal Memory Map Showing Parameters

In this example, the user would like a timebase of 1ms/div. Assuming a standard display plot of 10 divisions horizontally, a minimum of 10 ms of sampling must be conducted. We would like to choose the fastest sample rate that satisfies this condition. The sample period, T_s, for a sampling rate (or frequency) of $f_s = 1.25MHz$ is $T_s = 1/(f_s) = 800$ns. The internal memory for each sample channel holds 32kB of data and each sample is 1 byte. The total amount of sampling conducted t_{sample} is given by $t_{sample} = 32768 \cdot T_s = 26.2$ms.

We have specified 10 samples per screen division. The burst count is obtained by multiplying the number of screen divisions by the number of samples per division. In this case, 10 samples per screen division times 10 screen divisions gives a total of 100 samples. In hexadecimal, 100_{10} is represented as 0x0064$_8$. The upper byte is 0x00$_8$ $= 0_{10}$ and the lower byte is 0x64$_8$ $= 100_{10}$. The commands *c100* and *v0* write these values into the registers.

Each screen division is 1 ms and contains 10 samples. Therefore, each sample plotted on the screen should be spaced at 0.1 ms intervals. The sample period at 1.25 MHz was 800ns. The burst increment must be $0.1\text{ms}/800\text{ns} = 125$. In other words, we would like to read back every 125_{th} sample. By doing so, we are reading back samples at 0.1 ms intervals. The commands *h0* and *l125* write these values into the registers.

Lastly, we would like the trigger point to appear at the end of the second division on the screen. In other words, we would like two divisions worth of samples before the trigger point and eight divisions afterwards. The spacing between each sample was determined to be 125 memory locations (or 0.1 ms). Two divisions on the screen contains 20 samples. So, we would like to offset 20 samples or 20x$125 = 2500$ memory locations. This offset defines where the scope will start reading back the data. In hexadecimal, $2500_{10} =$ 0x09C4$_8$. The upper byte of the offset is 0x09$_8$ $= 9_{10}$ and the lower byte of the offset is 0xC4$_8$ $= 196_{10}$. These values are written into the burst offset registers with the commands *u9* and *d196*.

12.4 Control Register

The final step in preparing for the capture is to write to the scope control register. The scope control register is shown in Figure 11.3. In this example, we would like to trigger off of a rising edge on channel A with a sample rate of 1.25 MHz. These settings correspond to a binary control register value of 00010100_2 or an integer value of 20. These settings are written into the control register using the command *r20*.

12.5 The Capture

Now that all of the parameters for this capture have been configured we can initiate the capture by sending the command *g*. The scope is now armed and will wait for a trigger event. Once the trigger occurs, the scope will automatically capture that data, format it and return it to the PC in a burst transfer. The scope default to ASCII mode so data will be returned in the format *D A0 B0 A1 B1 A2 B2 ...* where *D* indicates the start of a data stream. *AX* and *BX* are samples in integer format where *A0* is the first sample returned from channel A, *B0* is the first sample returned from channel B, etc.

Subsequent captures can be performed by issuing the *g* command again. It is not necessary to set up all of the control registers each time a capture is performed, they registers retain there values from capture to capture.

12.6 Plotting the Data

The capture data is 8-bit and the A/D converter actually returns values between 0 and 255. The input signal is offset to the middle of the A/D range, so that 0 to 255 actually represents the range -128 to +127. To convert 0 to 255 to the correct range, subtract 127 from each data value.

The effective A/D step size is multiplied by the integer value returned by the scope. The ”setVertical” procedure in the oscilloscope source code gives the details. That code has the values for each preamp setting as well as the formula for calculating the effective A/D step size.

13 Manual Operation

The oscilloscope may be operated by sending it commands from a terminal emulator. This can be useful for debugging.

13.1 Windows

The `Hyperterminal` program that is supplied as part of Windows operating systems is suitable for this.

1. To simplify matters, it is probably best to unplug any USB devices that are connected.
2. Plug the scope hardware into a USB port on the computer.
3. Using the steps described in section 8 above, determine the COM port that the scope is currently connected to.
4. Start Hyperterminal:
   Start `->` Programs `->` Accessories `->` Communications `->` Hyperterminal
5. Hyperterminal starts with a ’Connection Description’ popup window. Cancel the popup.
6. Select File `->` Properties
7. Operate the `Connect Using` menu to select the COM port that you found previously.
8. Select `Configure` which pops up a `Port Settings` window. Most of the port settings can be left at their default values, but the baud rate (`Bits per Second)` must be changed to the correct value. At this time, the correct value is 38400. The correct port settings are:

Bits per Second:	38400 (probably)
Data bits:	8
Parity:	None
Stop Bit:	1
Flow Control	Hardware

If you have a version of the scope Tcl program that works correctly with the scope hardware, you can verify the baud rate. Load that code into a text editor and look for the baud setting. Search for a variable `baud` or a procedure `openSerialPort`.

Hit `OK` and back out to the hyperterminal screen.

9. Issue some command from the list in section 11. Type it in followed by the 'Enter' key. A good choice is `i`, which should result in a message from the scope indicating its version number.

13.2 Linux

For manual operation under the Linux operating system, you will need to communicate with the hardware using a *terminal emulator* program. There are two terminal emulators in common use under Linux: `seyon` and `minicom`. These may not be installed as part of your linux distribution. To check whether the program is installed, use the `which` command:

```
phiscock@panther:~> which seyon
/usr/X11R6/bin/seyon
```

Seyon must be properly configured to be used. This is described in the document *Seyon: Quick Start Guide* which is on the system CDROM.

Information on using `minicom` may be found at the following location:

Using Minicom and Seyon
Chapter 11 of *Learning Debian GNU/Linux*
Bill McCarty
O'Reilly Books, 1999
`http://www.oreilly.com/catalog/debian/chapter/book/ch11_07.html`

1. Plug the scope hardware into a USB port on the computer.
2. Run the command `dmesg` to identify the serial port that the is allocted to this USB device. It may be a few seconds before the operating system finishes its allocation, so run dmesg repeatedly until you see the serial port number, which will be something like `\ttyUSB1`.
3. Start the terminal communications program (Seyon or Minicom):
4. Set the port settings to:

Bits per Second:	38400 (probably)
Data bits:	8
Parity:	None
Stop Bit:	1
Flow Control	Hardware

If you have a version of the oscilloscope Tcl program that works correctly with the scope hardware, you can verify the baud rate. Load that code into a text editor and look for the baud setting. Search for a variable `baud` or a procedure `openSerialPort`.

5. Issue some command from the list in section 11.2. Type it in followed by the 'Enter' key. A good choice is `i`, which should result in a message from the oscilloscope indicating its version number.

   Now you can type in other commands.

14 Modifying the Tcl/Tk Software

The host software for the scope (and other Syscomp instruments in this series) is released in plain text format under the GPL (Gnu Public Lincense). Consequently, it is legal to modify the program in whatever way you may find useful. We also encourage you to share your work with others.

The software is written in Vanilla Tcl/Tk, that is, in the Tcl/Tk language without any third party packages or linked libraries. This makes it extremely simple to modify.

The program itself is in text form. The code does not need to be compliled or linked, it is executed directly from the text form by the `wish` interpreter.

Although the scope program is fairly complicated, it is possible to create a powerful Tcl/Tk program, with a sophisticated user interface, with only a few lines of Tcl/Tk code.

To set up for development, ensure that the `wish` interpreter is installed on your computer. Under Linux, it is probably already there. Under Windows, you need to download and install a file from the ActiveState website:
`http://www.activestate.com/Products/languages.plex?tn=1`
Download the appropriate .exe file for your operating system. Run the program to install Tcl/Tk.

Make a copy of the original code, of course, and put it in a safe place. Then, using a text editor, read and modify the existing code. When you want to test the code, run the `wish` interpreter. Execute the `source` command with the name of your file, and the program will execute. Repeat this cycle until you have the desired result.

We'd like to hear about your work. Check out our web page for the latest contact information.

15 Sources of Information

15.1 News Groups

`comp.lang.tcl`
Internet News Group

15.2 Websites

`http://www.syscompdesign.com`
Latest information on Syscomp instruments and supporting software.
Many useful application notes and project descriptions.

`http://www.tcl.tk/`
Home of the Tcl Developer Xchange. Pointers to information and software downloads.

`http://www.activestate.com/`
ActiveState is a commercial firm that sells various programming tools provides a home for the Tcl/Tk language. Free versions of Tcl/Tk are available for download from their site.

15.3 Paper

Scripting: Higher Level Programming for the 21st Century
John K. Ousterhout
IEEE Computer magazine, March 1998
Currently at: `http://home.pacbell.net/ouster/scripting.html`
Also at: `http://www.tcl.tk/doc/scripting.htm`
The definitive paper on Tcl/Tk and scripting languages in general. Ousterhoot shows a Table of Applications which have been coded in Tcl/Tk and in the C language, and the relative effort and time required for each implementation.

15.4 Textbooks

Practical Programming in Tcl and Tk, 4th Edition
Brent B. Welch & Ken Jones with Jeffery Hobbs
Prentice Hall PTR, 2003
The definitive reference for Tcl and Tk. Includes CDROM with Tcl and examples.

Tcl and the Tk Toolkit
John K. Ousterhout
Addison-Wesley, 1994
Now somewhat dated, but a still useful introduction to Tcl/Tk by the inventor of the language.

Graphical Applications with Tcl & Tk, 2nd Edition
Eric Foster-Johnson
M&T Books, 1997
Very accessible introductory textbook.

Tcl/Tk Tools
Mark Harrison
O'Reilly, 1997
Information on a number of extensions to Tcl/Tk.

Effective Tcl/Tk Programming
Mark Harrison, Michael McLennan
Addison Wesley, 1998
Techniques of design for Tcl/Tk programs.

Tcl/Tk for Programmers
Adrian Zimmer
IEEE Computer Society, 1998
An textbook on Tcl/Tk with an academic tone and exercises.

Tcl/Tk for Dummies
Tim Webster
IDG Books, 1997
A useful introduction to Tcl/Tk.

Tcl/Tk for Real Programmers
Clif Flynt
Academic Press, 1999
Medium to high-level material on Tcl/Tk

16 Windows XP Appendix, Screenshots and Technical Details

16.1 Checking the Installed Files

This section details how to determine that the files are installed correctly.

1. Double click on `My Computer` to obtain a screen like figure 22.

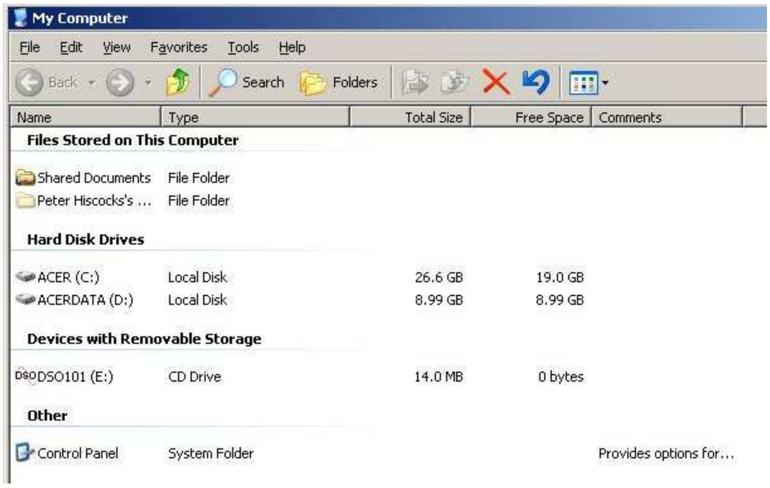


Figure 22: My Computer

2. Double click on `Control Panel`, obtaining a screen like figure 23.

3. Double click on `Add or Remove Programs`, obtaining a screen like figure 24.

   Figure 24 lists the FTDI drivers, so they are installed. Similarily, the Tcl/Tk program for the unit should be shown on the same list under `Syscomp`.

16.2 Obtaining Details of the USB-Serial Port

1. On the `Add or Remove Programs` list of figure 24, find the entry `System`. Double click on it to obtain the `System Properties` screen of figure 25.

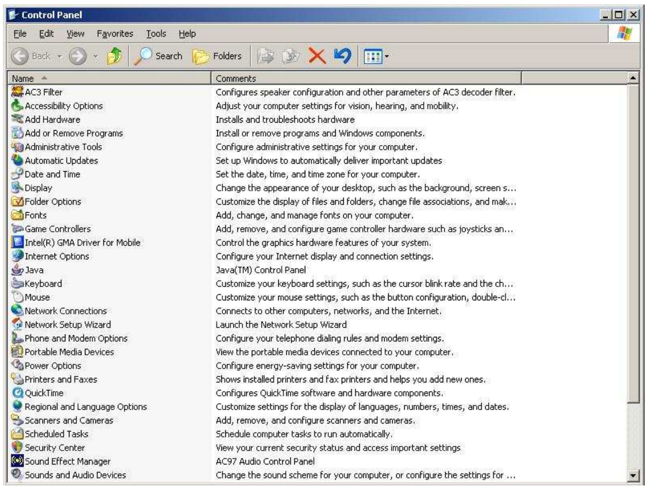


Figure 23: Control Panel

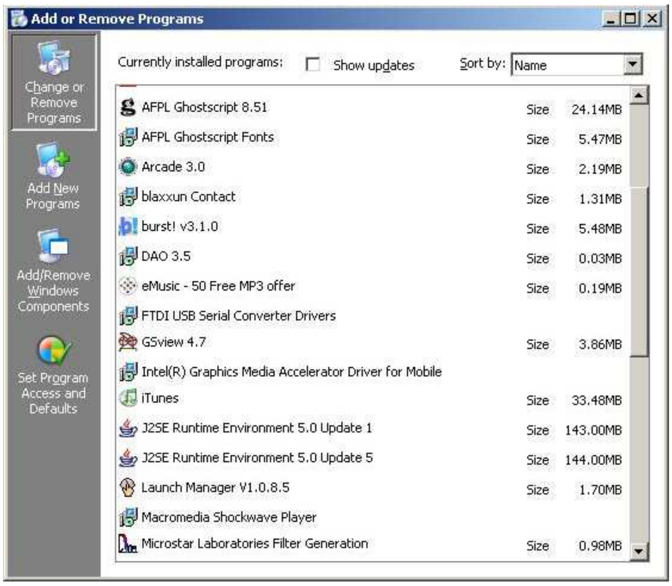


Figure 24: Add or Remove Programs

2. Double click on the `Hardware` tab to obtain the `Device Manager` screen of figure 26.

3. **Ensure that the hardware unit (DSO-101 oscilloscope, WGM-101 waveform generator, etc) is plugged into a USB port.**

   In the System Properties, Hardware panel of figure 26, there should be an entry `Ports (COM & LPT)`. Double click on that and it should open to an entry like `USB Serial Port (COM4)`.

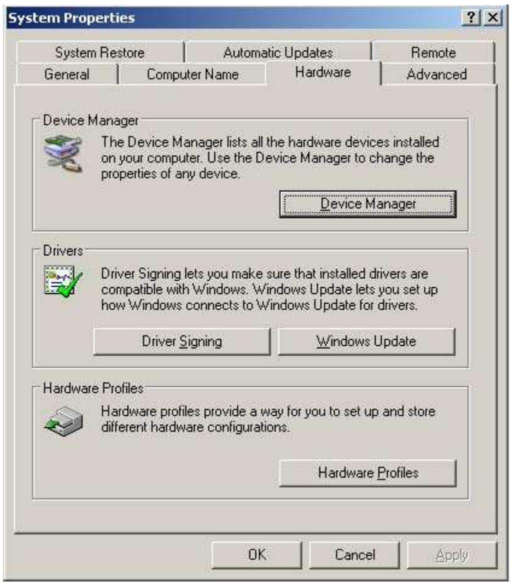


Figure 25: System Properties

This specifies the COM port that should be selected in the Tcl/Tk program in order to communicate with the USB hardware unit.

16.3 Adjusting the COM Port Selection

1. **Ensure that the hardware unit (DSO-101 oscilloscope, WGM-101 waveform generator, etc) is plugged into a USB port.**
2. In the System Properties, Hardware panel of figure 26, there should be an entry `Ports (COM & LPT)`. Double click on that and it should open to an entry like `USB Serial Port (COM4)`.
3. Double click on this last entry to obtain the Serial Port Properties screen of figure 27.
4. Click on the tab`Port Settings`. You should not need to change any of these settings.
5. Find and click on the `Advanced` button to obtain the `Advanced Serial Port Properties` screen of figure 28.
6. In figure 28, the `COM Port` selection is highlighted. Click on the arrow to show a selection of `COM` ports. Use the up and down arrow to scroll through all the `COM` ports. If you wish to force the hardware to use a different `COM` port, you may select that port number here.

16.4 XP Installation Instructions and Screenshots

The installation procedure should be semi-automatic. Insert the CD in the CD drive, and the installation procedure should run. Answer the prompts, and the software should be completely installed and ready to run.

In general, there should be no need to install drivers from the Web.

Figure 26: System Properties, Hardware (Device Manager)

In this section, for troubleshooting purposes and for those who wish more detail, we describe the installation procedure in detail.

16.4.1 Preliminary: Checking for Installed Drivers

The software consists of two parts: a Tcl/Tk program, and two FTDI drivers. Normally, you will not have any of these installed. However, the FTDI drivers are used by other equipment from other vendors and in the course of installing that equipment, you may have obtained the FTDI drivers. If they are installed it is not necessary to install them again.

You can check for these drivers by going to:
`My Computer -> Control Panel -> Add or Remove Programs`

Normally, there will not be FTDI drivers on the `Add or Remove Programs` list. In that case, proceed with the installation as listed below in section 16.4.2.

If **and only if** there are FTDI USB-Serial drivers on this list, then proceed as follows:

1. Open the CD directory:

   Start up your web browser and double-click on `File -> Open File...` Move to the CD drive and open it. You should see a listing like figure 31.

2. Double click on `setup.exe` (figure 31) to run it.

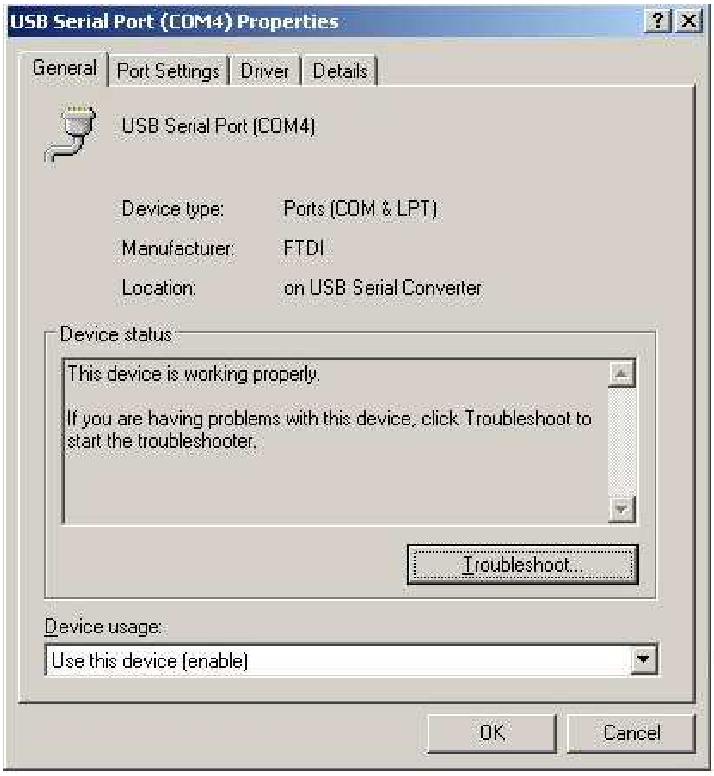


Figure 27: Serial Port Properties

When this completes, the installation should then be complete.

16.4.2 Starting the Installation

- Insert the CD into the CD drive. This should start up the installation procedure by showing the screen of figure 29.
- If that doesn't happen, double click on `My Computer` and you should see the installation CD listed under `Devices with Removeable Storage` (figure 30). Double click on the installation CD entry, and the screen of figure 29 should appear.
- If that still doesn't happen, right click on the installation CD entry of figure 30. This should bring up a directory listing of the installation CD, somewhat like figure 31. Double-click on `setup.htm`. This should bring up the install screen of figure 29.
- If that fails, start up your web browser and double-click on `File -> Open File...` Move to the CD drive and open it. You should see a listing like figure 31. Double-click on `setup.htm`. This should bring up the install screen of figure 29.
- If that fails, you have our permission to have a drink. Call or email us.
- **Note**: Do not run `setup.exe`. This file installs the Tcl/Tk program, but does not install the FTDI drivers.

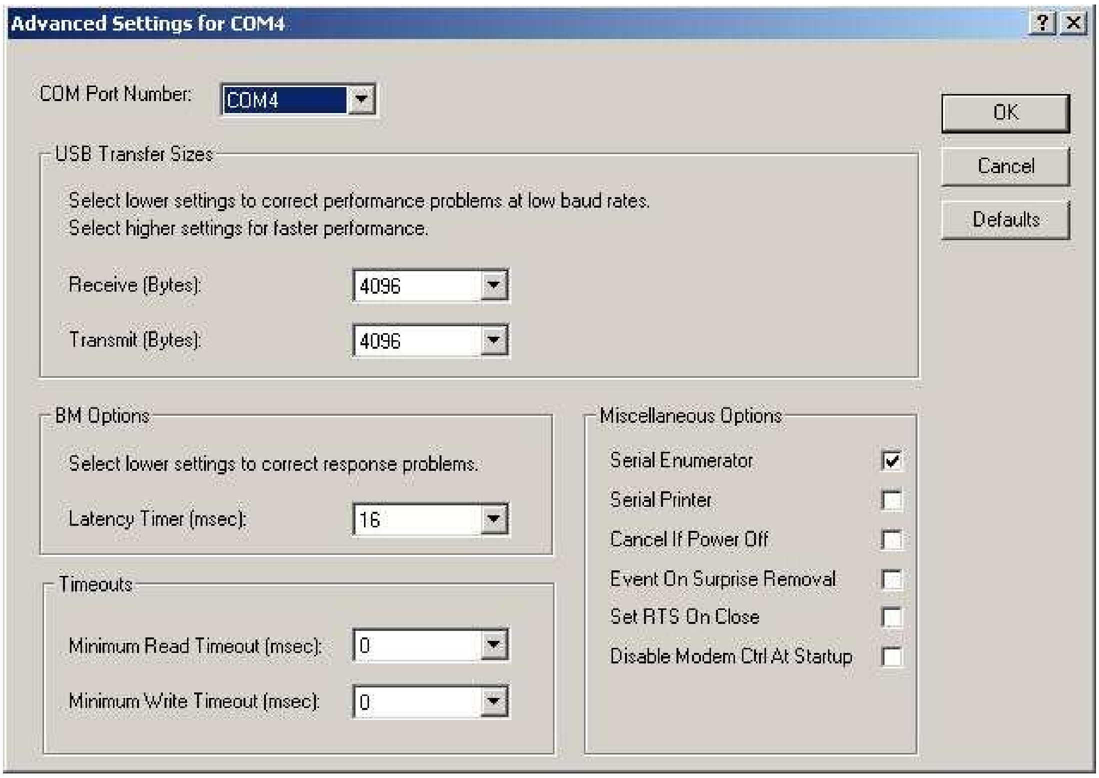


Figure 28: Serial Port Properties, Advanced

16.4.3 Continuing the Installation

1. We assume that the installation screen, figure 29, is showing.
2. Click on `Windows Installation: Click here`. The screen of figure 32 appears.
3. Click on `Next>>`. The screen of figure 33 appears.

   Click on the icon labelled `setup.exe` to run it (Windows Explorer browser) or cause it to be downloaded (Firefox browser).

   If it is downloaded, the default location is `Desktop` and the icon `setup.exe` appears on the computer desktop.
4. In either case, double click on the icon `setup.exe` to run it. The screen of figure 34 appears.
5. Click on `Next>>`. The screen of figure 35 appears.

   This screen shows the default directory where the software will be located, for example:

   `C:\Program Files\Syscomp\DSO-101`

   Unless you need to install into a different directory, we recommend that you accept the default.

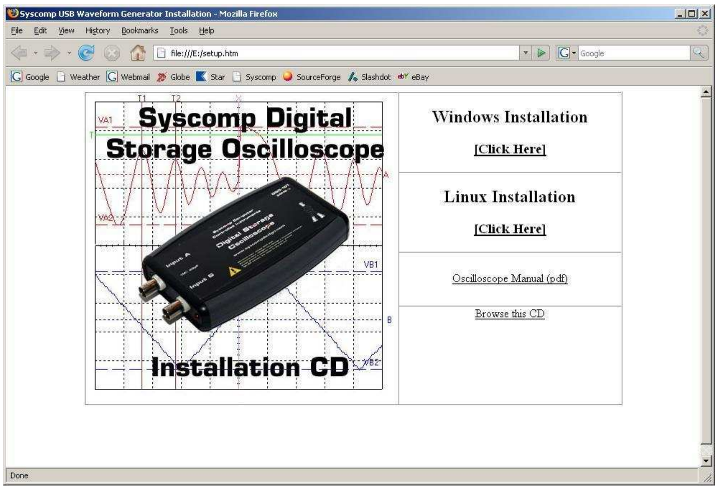


Figure 29: Install Screen

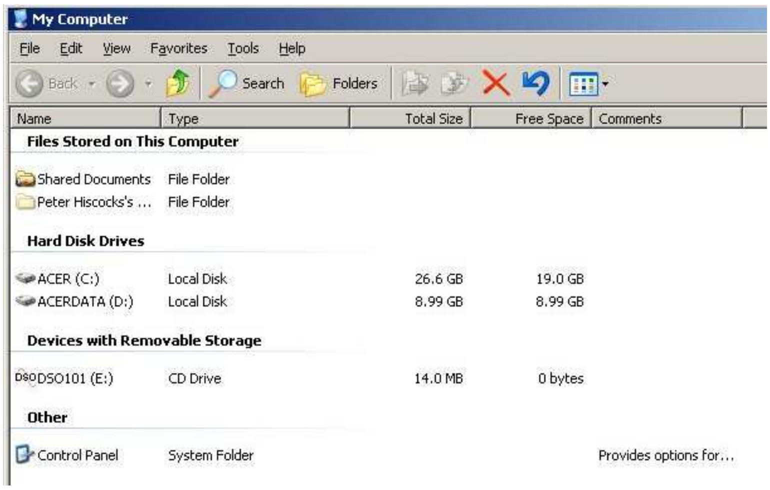


Figure 30: My Computer

6. Click on `Next>>`. If this directory already exists, the computer will ask you whether you 'really want to install to that directory anyway'. Select `Yes` unless you have good reasons not to.

   The screen of figure 36 appears.

   Check the `Create Desktop Icon` and `Create a Quick Launch Icon` if you wish those icons to be created at this point. (We recommend the Desktop Icon.)

7. Click on `Next>>`. The screen of figure 37 appears.

8. Click on `Install`. A progress bar appears and in a second or two, the installation completes. The screen of figure 38 appears.

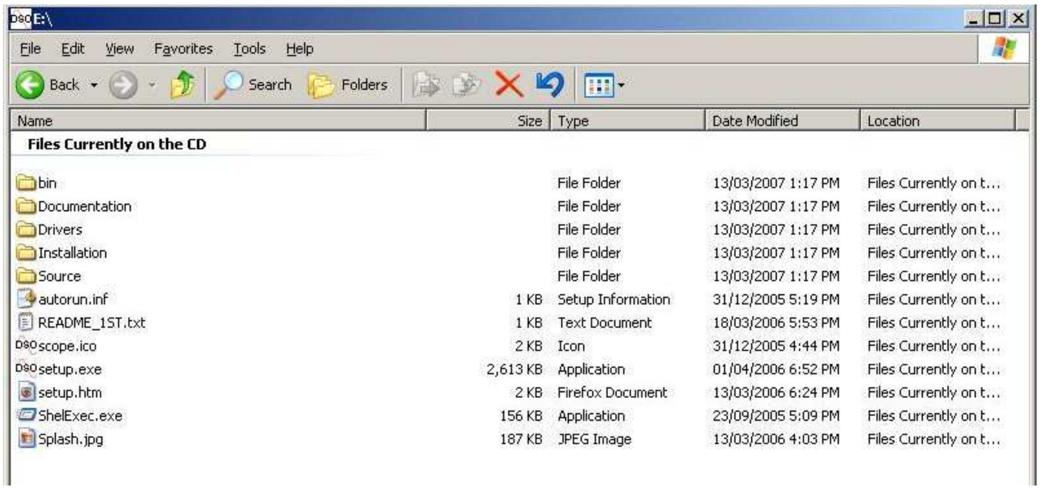


Figure 31: CD Directory

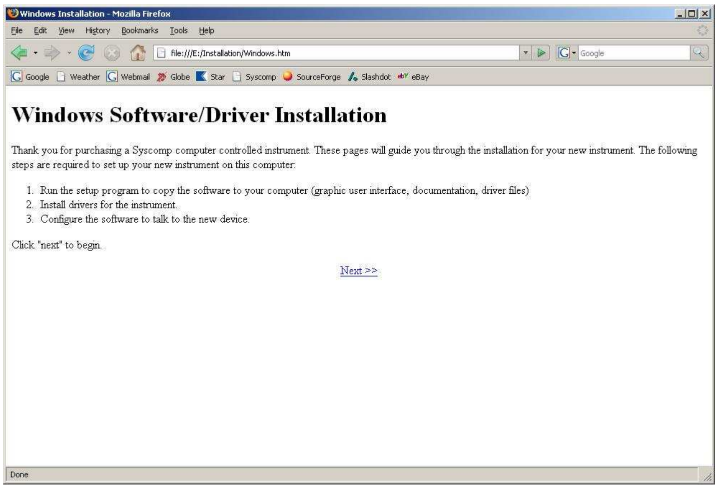


Figure 32: Windows Software/Driver Installation

If you elected to `Create Desktop Icon` in step 6, there will be an icon on the desktop labelled, for example, `Syscomp Digital Oscilloscope`[10].

Click on `Finish` to complete this phase of the installation.

No, we're not done yet. This completes the installation of the Tcl/Tk program, but the drivers still remain to be installed.

The screen shown in figure 32 reappears. (Actually, it's been there all along, it would be more accurate to say it is now *uncovered*.)

[10]Maybe that label is a bit long for an icon. We might change it.

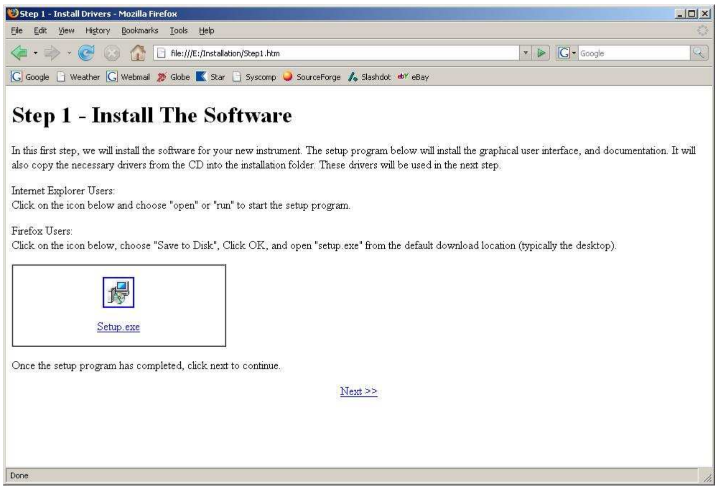


Figure 33: Windows Software/Driver Installation

Figure 34: Welcome to the Syscomp .. Setup Wizard

9. Click on `Next`. The screen of figure 39 appears.

   Select the version of your operating system. In this case, we assume `Windows XP`. Click on that selection.

16.4.4 Installing the Drivers

We are now going to install **two** drivers necessary for the correct operation of the device. The install procedure is the same for both drivers. Follow the steps below to install the first driver.

1. Plug the instrument into an available USB port.

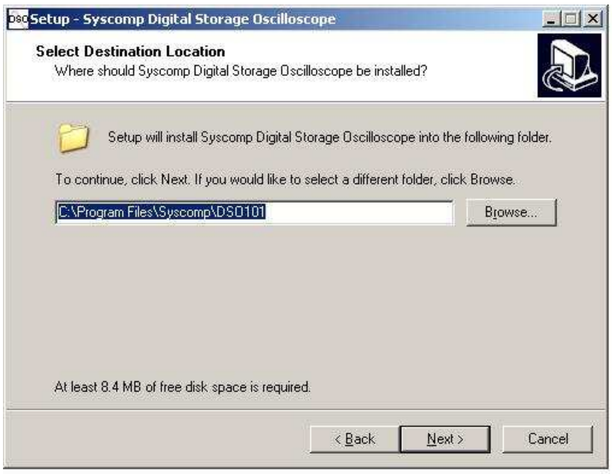


Figure 35: Select Destination Location

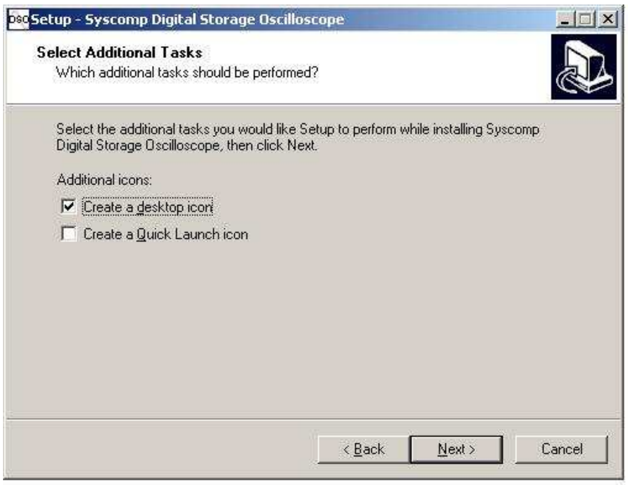


Figure 36: Select Additional Tasks

2. The Found New Hardware Wizard will appear, figure 40 on page 59.

   If the wizard does not appear, right-click on the `My Computer` icon (typically on the desktop or in the Start menu) and select `Properties`. The `System Properties` window (figure 25 on page 48) will appear.

   Click on the `Hardware` tab and click on the `Device Manager` button. When the Device Manager (figure 26 on page 49) appears, select the `Action` menu and then `Scan For Hardware Changes`. Windows should detect the device and the screen of figure 40 should appear.

   Click `Next`.

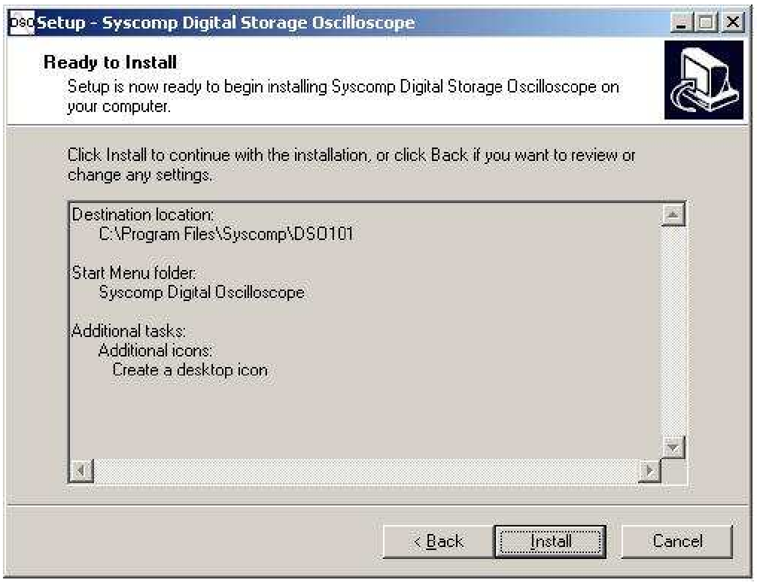


Figure 37: Ready To Install

3. Windows will show the screen of figure 41 and ask you if you would like to connect to Windows update to search for software. Choose `No, not this time` and click `Next`.

4. Windows will now show the screen of figure 42 on page 60 to ask how the software should be installed. Choose the radio button `Install from a list or specific location (Advanced)`” [11] and click `Next`.

5. Still in figure 42 on page 60:

   - Select the radio button `Search for the best driver in these locations`.
   - Uncheck the checkbutton `Search removable media`.
   - Check the checkbutton `Include this location in the search.`

6. Still in figure 42 on page 60:

   Click on the `Browse` button and navigate to the `Drivers` folder located in the installation directory on your hard drive. Typically, the install directory is `C:\Program Files\Syscomp\MODEL\Drivers` where `MODEL` is `WGM101` for the waveform generator and `DSO101` for the oscilloscope, and C: is the letter of the hard drive.

   Click `OK` to select this driver folder as the search location. Click `Next` to continue.

7. Windows will detect the driver and show a progress bar, figure 43 on page 60.

8. Windows now shows the screen of figure 44 on page 61.

   Click `Finish` to complete the installation of the first driver.

[11] Isn’t it nice to know you are an *advanced* user?

Figure 38: Completing the Syscomp .. Setup Wizard

Installing the Second Driver

Windows will now begin to install the second driver. The Found New Hardware Wizard will appear again. This is normal, repeat the above procedure steps 1 through 8.

The drivers should now be successfully installed for your device. You should only have to install the drivers once. From this point on, the computer will recognize your instrument when it is plugged into the USB port. Click `Next` to continue to the last step.

Configuring the Software

The `Configure The Software Screen`, figure 45 on page 61 appears.

1. Double click on the desktop program icon for the oscilloscope or waveform generator to start the instrument program. (If you didn't create a desktop icon during the installation, start the program using the `Start` menu.)

   At the top of the instrument is a `Connected` indicator. If it is showing green, then the instrument is connected to the hardware and ready to use. No further adjustments are necessary.

   If the `Connected` indicator is showning red, then it is necessary to select the `COM` port according to step 2, following.

2. Click on `Hardware -> Port Settings` to obtain the dialog shown in figure 46 on page 62.

3. Click on `Autodetect`. After a few seconds, an alert should appear, indicating that the instrument has been found on a particular COM port. Click on `Save and Exit`. This will cause that selection of COM port to be saved in a text file, `scopeport.cfg` for the oscilloscope or `waveport.cfg` for the waveform generator. Thereafter, when you start the program, it will read this file and set the `COM` port appropriately.

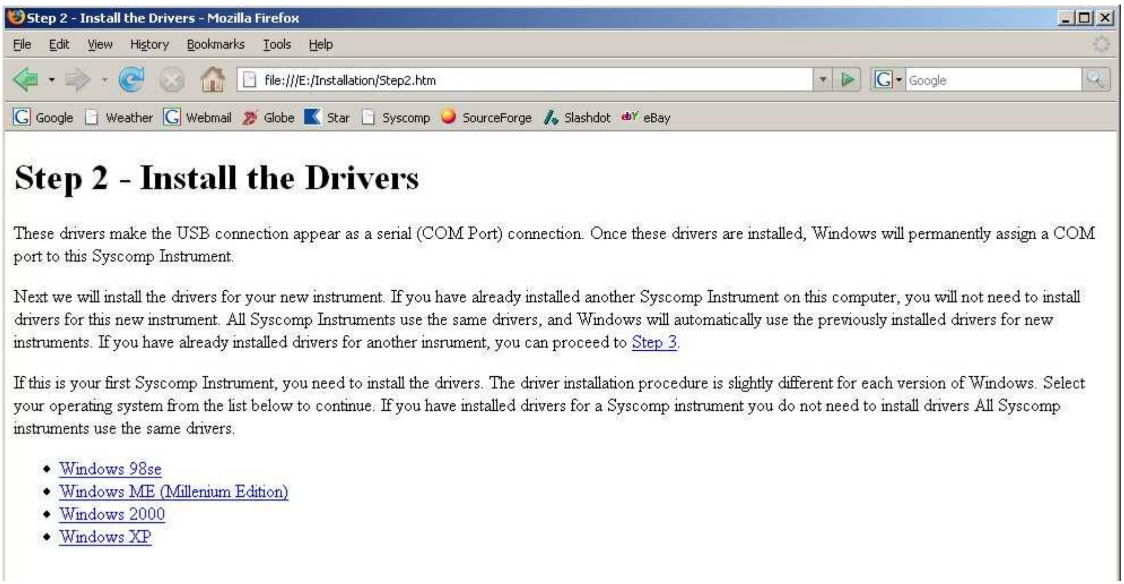


Figure 39: Step 2: Install the Drivers

If the instrument does not connect properly at this stage, then you must manually adjust the port settings.

The most likely problem is that `COM` ports 0 to 9 are occupied by other devices. Early versions of the software required that the Syscomp equipment occupy one of these `COM` ports. Consequently, it was necessary to move other (more flexible) devices to available `COM` ports above number 9. The detailed instructions for that process are given elsewhere in this manual.

The most recent version of the software is less restrictive. It has provision for manually entering the number of a `COM` port into this dialog. View the `Device Manager`, `COM & LPT`, properties, advanced settings and choose a suitable available unused `COM` port. Then enter that number into the `COM` port dialog of the Syscomp software.

Figure 40: Found New Hardware Wizard: 1

Figure 41: Found New Hardware Wizard: 2

Figure 42: Found New Hardware Wizard: 3

Figure 43: Found New Hardware Wizard, Progress Bar

Figure 44: Found New Hardware Wizard, Final Screen

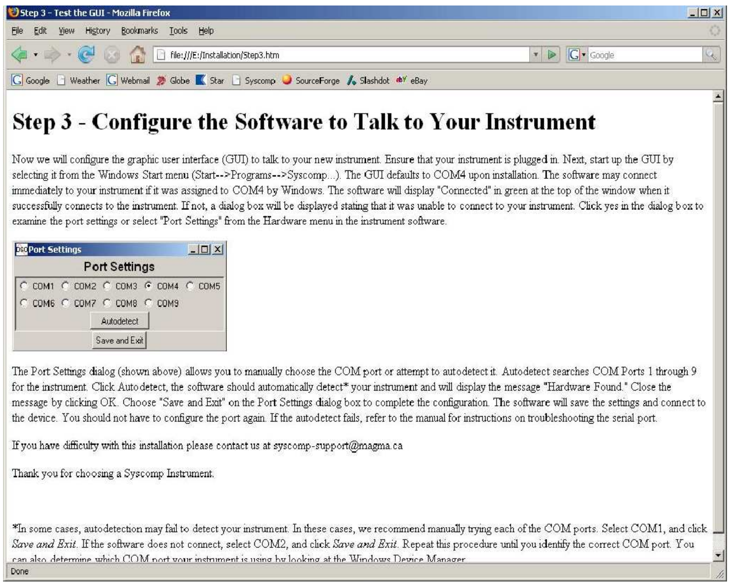


Figure 45: Found New Hardware Wizard, Final Screen

Figure 46: Port Settings Dialog