

Model 3820
Universal Counter
Hardware and Software

Instruction Manual

785080-A-MNL-D

Trademarks

AMETEK® and the **AMETEK**® and **A** logos are registered trademarks of AMETEK, Inc. Other product and company names mentioned are trademarks or trade names of their respective companies.

Company Names

SIGNAL RECOVERY is part of Advanced Measurement Technology, Inc, a division of AMETEK, Inc. It includes the businesses formerly trading as EG&G Princeton Applied Research, EG&G Instruments (Signal Recovery), EG&G Signal Recovery, **SENSE-actions**™ and PerkinElmer Instruments (Signal Recovery)

Table of Contents

Chapter One, Introduction

1.1 Overview	1-1
1.2 What's New in Version 3.0	1-2
1.3 Changes from the SENSE-ations Model 3820	1-3

Chapter Two, Installation & Initial Checks

2.1 Introduction	2-1
2.1.01 Media	2-1
2.1.02 Product Registration	2-1
2.1.03 Requirements	2-1
2.1.04 USB Computer Port	2-2
2.2 Installation	2-2
2.3 Initial Checks	2-2

Chapter Three, SR3820 Universal Counter Software

3.1 Introduction	3-1
3.2 Starting the Program	3-1
3.2 Program Operation	3-1
3.2.01 Introduction	3-1
3.2.02 Input	3-2
3.2.03 Information	3-3
3.2.04 Measurements	3-3
3.2.05 Measurement & Trendline	3-3
3.2.06 Data Logging	3-3
3.2.07 Data Log File Format	3-4
3.2.08 Connection Lost Dialog	3-5
3.2.09 Closing the Program	3-5

Chapter Four, SR3820Comms ActiveX Control and Demonstration Files

4.1 Introduction	4-1
4.2 Operation	4-1
4.2.01 Overview	4-1
4.2.02 Property Page	4-4
4.3 Control Methods	4-4
4.3.01 Connect	4-5
4.3.02 IsConnected	4-5
4.3.03 Disconnect	4-6
4.3.04 Reset	4-6
4.3.05 AutoInputAThreshold	4-6
4.3.06 AutoInputBThreshold	4-6
4.3.07 Start	4-7
4.3.08 ResetCounter	4-7
4.3.09 Stop	4-7
4.3.10 GetTimeAsString	4-7

TABLE OF CONTENTS

4.3.11	GetFrequencyAsString	4-8
4.3.12	GetPulseWidthHighAsString	4-9
4.3.13	GetPulseWidthLowAsString	4-9
4.3.14	GetPeriodAsString.....	4-9
4.3.15	GetDutyCycleAsString.....	4-9
4.3.16	GetCountsAsString.....	4-11
4.3.17	GetLogicLevelAsString.....	4-11
4.3.18	GetTimebaseCorrectionAsString	4-11
4.3.19	GetTimeAsDouble.....	4-11
4.3.20	GetFrequencyAsDouble	4-13
4.3.21	GetPulseWidthHighAsDouble	4-13
4.3.22	GetPulseWidthLowAsDouble	4-13
4.3.23	GetPeriodAsDouble.....	4-13
4.3.24	GetDutyCycleAsDouble.....	4-15
4.3.25	GetCountsAsDouble.....	4-15
4.3.26	GetLogicLevelAsDouble.....	4-15
4.3.27	GetTimebaseCorrectionAsDouble.....	4-15
4.3.28	GetSerialNumber	4-16
4.3.29	GetModelNumber.....	4-16
4.3.30	GetHWRevision	4-16
4.3.31	GetHWManufactureDate.....	4-16
4.3.32	GetHWCALibrationDate	4-16
4.3.33	GetSR3820CommsXVersion	4-18
4.3.34	GetUSBXVersion	4-18
4.3.35	AboutBox	4-18
4.3.36	DoDiagnostics	4-18
4.3.37	LoadEEPROM	4-18
4.4	Control Properties.....	4-19
4.4.01	Input	4-19
4.4.02	InputAThreshold	4-19
4.4.03	InputBThreshold.....	4-20
4.4.04	Rate.....	4-20
4.4.05	ResultsAsStrings	4-20
4.4.06	ResultsEvents	4-21
4.4.07	ResultsWithStatusEvents.....	4-21
4.4.08	Simulate.....	4-21
4.4.09	Test	4-21
4.4.10	bMonitorPNP	4-21
4.5	Control Events.....	4-22
4.5.01	Results	4-22
4.5.02	ResultsAsStrings	4-22
4.5.03	ResultsWithStatus	4-22
4.5.04	ConnectionLost().....	4-24
4.6	Demonstration Files	4-24
4.6.01	Visual C++	4-24
4.6.02	Visual Basic.....	4-25
4.6.03	VBScript (HTML).....	4-26
4.6.04	Excel 2003.....	4-26
4.6.05	LabVIEW 8.2	4-27

Appendix A, Specifications

A.1 Signal Inputs	A-1
A.2 Measurements	A-1
A.3 Indicators	A-1
A.3 Computer Interface	A-1
A.4 General.....	A-2

Appendix B, Software License and Support

B.1 Introduction.....	B-1
B.2 License	B-1
B.2.01 Updates and Upgrades	B-1
B.2.02 Term.....	B-2
B.2.03 General.....	B-2
B.3 Software Registration.....	B-2
B.4 Software Errors	B-3
C.3 Removing References to old Driver Files.....	C-2

Appendix C, Uninstalling Earlier Software Versions

C.1 Overview.....	C-1
C.2 Uninstalling the Operating Software	C-1
C.3 Removing References to old Driver Files.....	C-2

Index

Warranty	End of Manual
-----------------------	---------------

1.1 Overview

The **SIGNAL RECOVERY** Model 3820 Universal Counter allows up to eight measurement outputs to be generated as a function of time, these being:-

- Frequency
- Average period
- Average duty cycle
- Average pulse-high time
- Average pulse-low time
- Event counts
- Logic level (input voltage above/below threshold)
- Time

The counter is principally intended for counting bi-level signals, which have two distinct voltage levels and clean transitions between them. Such signals include those generated by all common logic families used in electronic circuits, as well as most **Trigger** or **Sync** outputs of common test instruments.

The unit is powered and controlled directly from the PC's USB port and so requires no additional power source. Operation is entirely via software, with no manual switches or settings on the module.

The supplied instrument control software consists of two layers. At the lower level, a dedicated ActiveX control known as "SR3820Comms" takes care of all communications to and from the instrument. At the upper level, a simple Windows dialog application, "SR3820Counter", offers a convenient panel that allows the input (**A**, **B** or one of three internal test sources) to be selected, an update rate to be specified and all eight output measurements to be displayed. A further display area shows one of the measurements in a larger font size, as well as displaying a graphical trendline display. The program also supports data logging to text file of the output measurements, with data being written directly in CSV (comma separated value) format for easy import to other programs. The software includes a sub menu where the voltage input thresholds can be set for the two inputs.

Two main modes of operation are therefore possible. Users who simply want to operate the counter "out of the box" need do no more than plug it in, install the driver and software, and then use the SR3820 Universal Counter software to control it. Alternatively, when the counter is to be used as part of a computer controlled test system, then the user can develop software to control it via the SR3820Comms ActiveX control. The control eliminates the need for users to write the low-level code needed to send commands to and receive responses from the counter, allowing them to concentrate on developing the higher level program to run their experiments.

Typical applications include:

- Photon counting
- Frequency measurement
- Test and measurement systems implement in LabVIEW where a **SIGNAL RECOVERY** counter can be used at the same time as instruments from different suppliers.
- Measurement system using scripted web pages (HTML files) operated via Internet Explorer.

The SR3820Comms control can of course also be used at the same time as other **SIGNAL RECOVERY** software ActiveX controls, such as SR3820Comms, allowing sophisticated systems to be assembled. For example, five APD's (avalanche photodiodes) could each be connected to the inputs of a model 3830 multiplexer, with the output being in turn connected to the A input of a model 3820 counter. Using both controls a user-developed application program could sequentially count the pulses being generated by each APD.

Both the top level SR3820 Universal Counter software (which this help file refers to) and the lower-level SR3820Comms ActiveX control include comprehensive on-screen help files, while examples of how to use the control in LabVIEW, Visual Basic, VBScript, Visual C++ and Excel are also supplied.

1.2 What's New in Version 3.0

Version 3.0 of the Model 3820 Universal Counter software and the underlying SR3820Comms ActiveX control are completely updated from revision 2.0. The control now uses a new USB driver, SRUSBXP.SYS, to communicate with the counter, and the driver has been fully tested for operation on Windows XP SP2 and Windows Vista in their common 32-bit versions.

Other changes to the software include:

- Support for underlying Windows themes, so that the dialogs and controls adopt the theme of the computer on which the software is installed.
- Updated Help system using compiled HTML help files replacing original WinHelp files. This ensures that the Help system is fully functional on Windows Vista machines.
- New setup package to make software installation, repair, and removal much easier.

We strongly recommend that users completely uninstall any existing software and drivers for the model 3820 Universal Counter that may be on their computers before installing the updated software. The procedure for doing this is given in Appendix C of this manual.

1.3 Changes from the ***SENSE-actions*** Model 3820

The model 3820 previously carried the ***SENSE-actions*** brand name. The hardware is unchanged from units carrying this older name, but the SR3820 Universal Counter software and SR3820Comms ActiveX control software are both new.

Software supplied with the ***SENSE-actions*** branded product included a program called PELauncher that was installed within the Startup programs group, as well as an ActiveX control (UCounterX) and application program (UCounter.exe). The PELauncher program was started when the computer booted and then ran in the background, monitoring USB bus activity. When a model 3820 Counter was plugged in the launcher program then started the UCounter.exe application software.

This approach is no longer used and the new SR3820 Universal Counter software is started in the same way as most other programs simply by clicking on the icon in the Start Programs menu. The SR3820Comms control is installed and registered during software installation and hence is available both to the SR3820 Universal Counter software and to user-written programs.

The new SR3820 Universal Counter application software generates data logging files that are compatible with those created by the older package, so in most cases users of the older software will get better performance by completely uninstalling it and replacing it with the new programs.

Installation & Initial Checks

2.1 Introduction

This chapter describes how to install the **SIGNAL RECOVERY** model 3820 Universal Counter and its supporting software on a personal computer.

2.1.01 Media

The software and device driver files are supplied on a CD, complete with setup program, or can be downloaded from the www.signalrecovery.com website.

2.1.02 Product Registration

The first time the SR3820Comms control (and hence also the SR3820 Universal Counter program that also uses it) is used following installation it will automatically attempt to connect to our website and display a form for you to enter the relevant details to register your purchase. If the computer has browser software (Internet Explorer or Netscape Navigator) but cannot establish a link to the site, perhaps because there is no network connection, then you will receive a “404 Page Not Found” or similar error and you should simply close the browser window to continue, and perform manual registration (see below) later. Similarly, if browser software is not found then a message is displayed inviting you to perform the registration manually.

Manual registration is possible simply by visiting www.signalrecovery.com/register and clicking on the relevant link.

Registration will ensure that you are kept informed of software enhancements and developments as they become available. In addition, should you need to contact us for support we will check your details against those we have on file, and, if your copy of the software is not registered then we will not be able to help you.

2.1.03 Requirements

The SR3820 Universal Counter and SR3820Comms ActiveX Control software require:-

- ◆ A **SIGNAL RECOVERY** or **SENSE-ations** model 3820 Universal Counter complete with USB cable
- ◆ A personal computer with a 200 MHz Pentium or better processor, hard disk, CD-ROM disk drive, free USB port and Microsoft Windows XP 32 bit SP2 or later operating system. 2 Mbytes free disk space is required for the full installation. . The Universal Counter can also be connected to a USB hub, with the hub in turn being connected to the computer.

2.1.04 USB Computer Port

Some older PCs were shipped with the USB port disabled. If the PC that will be used to control the Universal Counter has not been used with other USB devices, check to make sure the USB port is enabled. If the model 3820 is connected to the PC and Windows does not detect new hardware, go to Windows **Control Panel, System**, and click on the Device Manager tab. At the bottom of the list it should show that the “Universal Serial Bus controller” is present. If not, use the PC's BIOS setup menu to enable the USB port.

2.2 Installation

Before installing the new software, please ensure that any existing software and drivers for the model 3820 Universal Counter that may be on the computer is completely removed. The procedure for doing this is given in Appendix C of this manual.

- 1) Connect the model 3820 to the PC. Windows will detect it and display the "Found New Hardware" dialog box. If installing from a CD, insert this into the drive and follow the Windows prompts. If installing from the downloaded zip file, unzip the contents of this file to a new, empty, folder and then use the browse function to navigate to this folder. This process installs the device driver. Note that the driver is not digitally signed, but you can safely ignore any security warnings that might be displayed. You should also be running in an account with administrative privileges.
- 2) Next, browse the CD or the folder with the extracted setup files and locate the file `SETUP.EXE`
- 3) The installation program will now run, copying files to the computer's hard disk. Simply follow the on-screen instructions.
- 4) On completion, the Windows desktop returns.

Once the installation process has been completed, the SR3820 Universal Counter software can be started by clicking on the shortcut to the **3820 Universal Counter** folder on the Start - Programs menu, and then clicking the **SIGNAL RECOVERY**  **3820 Universal Counter** shortcut icon.

2.3 Initial Checks

When the SR3820 Universal Counter software is started for the first time, it will attempt to connect you to the relevant software registration page on the **www.signalrecovery.com** website by starting your default web browser. Enter the details requested and then close the browser window to continue loading the software.

Assuming that a model 3820 Universal Counter is found then the main program dialog will be displayed, as shown in figure 2-1 below.

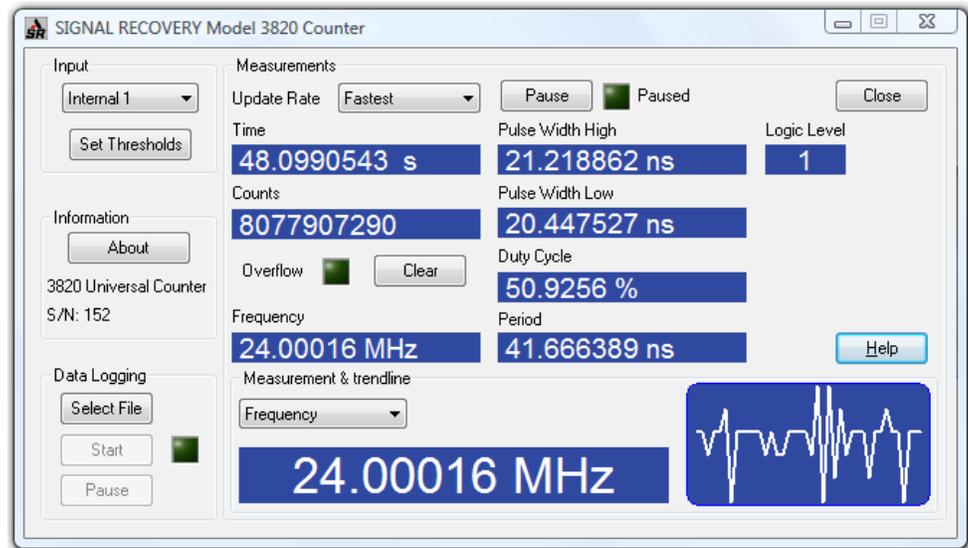


Figure 2-1, SR3820 Universal Counter - Main Dialog

Verify that the serial number (S/N) shown in the **Information** area is the same as that shown on the rear panel of the instrument, and then set the **Input** selector to **Internal 1**. This connects the counter's input to an internal signal generator of nominally 24 MHz. Check that the indicators in the **Measurements** area are being updated and that the displayed Frequency indicator is 24.0 ± 0.1 MHz.

In the **Measurements & trendline** area, select Counts as the output to be displayed and check that the trendline display shows a sloping line, as shown in figure 2-2.



Figure 2-2, SR3820 Universal Counter – Trendline Display

Click the **Close** button to terminate the program.

Assuming that the SR3820 Universal Counter program works as described then you can be certain that the underlying SR3820Comms ActiveX control has been installed correctly and is functioning properly.

3.1 Introduction

This chapter describes how to use the SR3820 Universal Counter application software to control the model 3820

3.2 Starting the Program

Having successfully completed the installation procedure as described in Chapter 2, the program may then be started by clicking on the shortcut to the **3820 Universal Counter** folder on the Start - Programs menu, and then clicking the **SIGNAL RECOVERY SR 3820 Universal Counter** shortcut icon.

Assuming that a model 3820 Universal Counter is found then the main program dialog will be displayed, similar to that shown in figure 3-1 below.

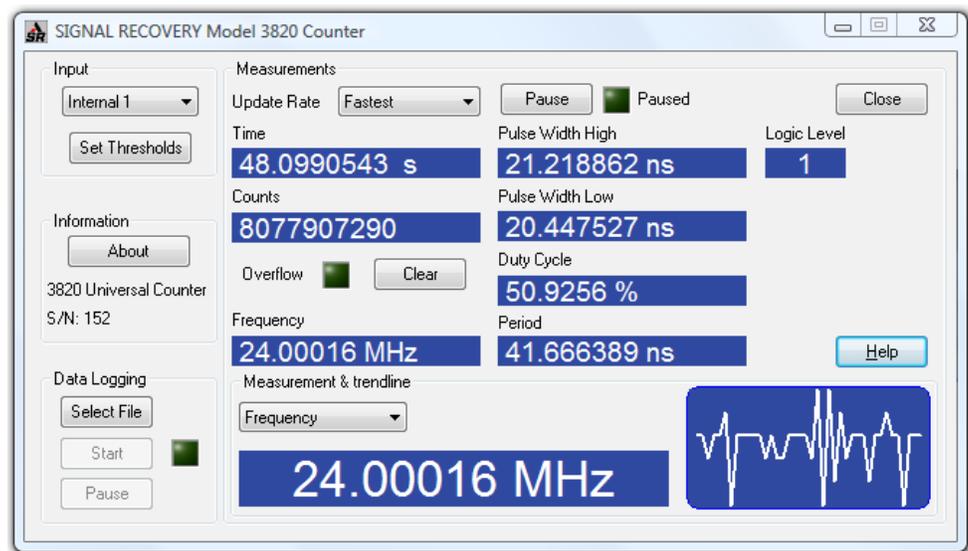


Figure 3-1, SR3820 Universal Counter - Main Dialog

If for any reason the Universal Counter is not found, then the **Information** area displays the message "Instrument not found". If this happens, close the dialog, check all the USB connections and then try restarting the program again.

3.2 Program Operation

3.2.01 Introduction

The main dialog is split into several clearly defined function areas, which are described in the following sections.

3.2.02 Input

The **Input** area includes a drop-down list box allowing the counter input to be selected. The **A** and **B** selections correspond to the counter's front panel BNC input connectors, while the **Internal 1**, **Internal 2** and **Internal 3** options use internal test signals of nominally 24 MHz, 190 MHz and 272 MHz respectively.

Clicking the **Set** button opens the dialog shown below in figure 3-2.

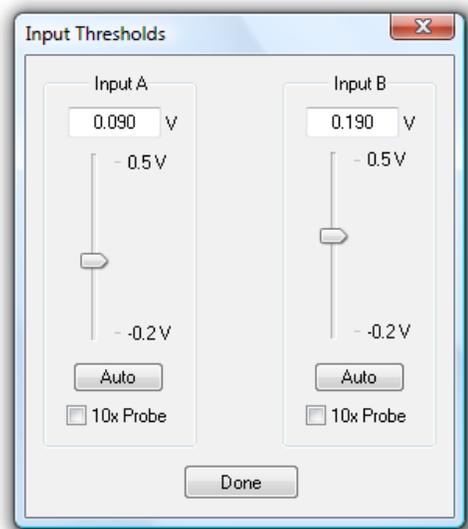


Figure 3-2, SR3820 Universal Counter – Input thresholds dialog

Use the sliders or edit boxes to set the thresholds as required. If using an external x10 oscilloscope probe, then check the **10x Probe** check boxes so that the corresponding thresholds are shown corrected for the attenuation of the probe. If the signal connected to the **A** or **B** input is DC or greater than 1 kHz then the corresponding **Auto** buttons will cause the optimum voltage threshold to be set for that input.

For reference, the typical threshold voltages and logic voltage swings for common logic families are:

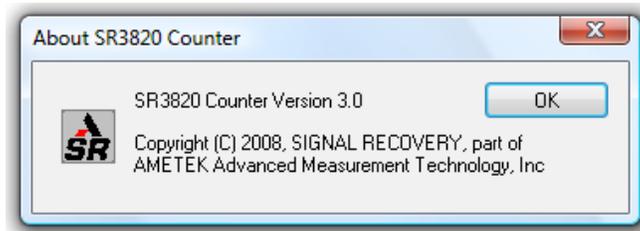
Family/Signal	Threshold	Voltage Swing
TTL	1.4 V	0 V to +3.5 V
CMOS	2.5 V	0 V to +5 V
LV-CMOS (3.3V)	1.4 V	0 V to 3.3 V
AC-coupled Sine	0.0 V	N/A
ECL: (-5.2v)	-1.35 V	-1.8 V to -0.9 V
PECL (+5V)	3.65 V	+3.2 V to +4.2 V
LVPECL (+3.3V)	2 V	+1.5 V to +2.5 V
Fast Negative NIM	-0.4 V	0 V to -0.8 V

Note that the counter is principally intended for counting bi-level signals, where there is a clean and fast transition between the two levels that crosses the specified threshold voltage. Cycles sine or triangular waves can also be counted, but typically only at frequencies above 10 kHz where the transition times become short enough for reliable operation.

Click the **Done** button to close the dialog.

3.2.03 Information

When an instrument is connected, the **Information** area shows its type and serial number; if an instrument is not found when the program starts then this area displays the message "Instrument not found" instead. If this happens, close the program using the **Close** button, make sure the USB and hardware connections are correct and then re-start the program. Clicking the **About** button shows the **About SR3820 Counter** dialog, shown below in figure 3-3, which gives program version and copyright information.



SR3820 Universal Counter – About Dialog

3.2.04 Measurements

The **Measurements** area has an **Update Rate** drop-down list box offering five choices of measurement update rate. When measurements are being taken the adjacent **Pause** button allows the screen update to be suspended (although the hardware continues to make measurements), with the pause mode being indicated by a flashing red "**Paused**" indicator. Screen update can be restarted by clicking the same button, which now has the caption **Continue**.

Eight indicators show the current measurement outputs. The **Time** and **Counts** displays can be reset at any time by clicking the **Clear** button. The **Overflow** indicator shows when the count has exceeded the maximum permitted, at which point it "wraps" and starts again from zero. Hence if the **Overflow** indicator is lit the **Counts** display does not show the true number of event counts since the counter was started.

3.2.05 Measurement & Trendline

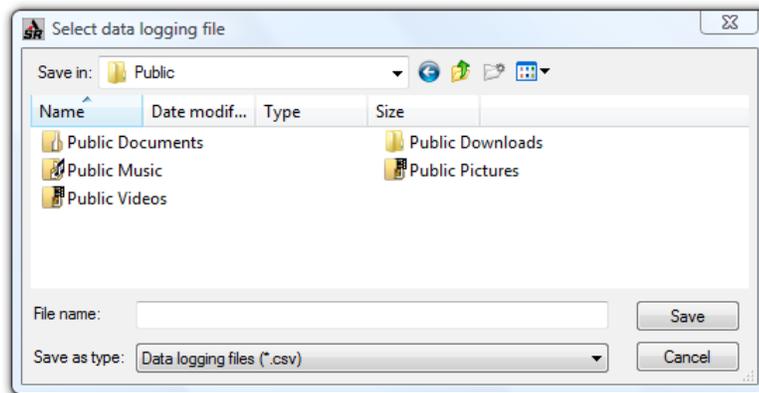
The **Measurement & Trendline** display area shows one selected output in a larger font size, while the adjacent graph display gives a recent "history" of the selected reading, allowing trends to be easily identified.

3.2.06 Data Logging

The program includes the ability to log the displayed measurement outputs directly to a text file, with a new set of readings being written to file each time the display is updated. Hence the **Update Rate** control in the **Measurements** area is used to define the rate at which readings are taken. The file format is described in section 3.2.07

Click the **Select File** button to open the file selection dialog, shown below in

figure 3-4.

**Figure 3-4, SR3820 Universal Counter – Select data logging file**

Enter the required filename or browse to an existing file and click **Save** to create or overwrite the file and write the header data to it. The dialog will close and the selected file will then be used for all further data logging until either the program is closed or a new logging file is selected. Alternatively, click **Cancel** to dismiss the dialog without changing the present logging file name.

Once a file has been specified, the **Start/Stop** and **Pause/Continue** buttons allow the eight instrument outputs to be recorded to file. A green flashing indicator shows when acquisition is running.

3.2.07 Data Log File Format

SR3820 Universal Counter data logging files are of plain ASCII text format and carry the filename extension “.csv”, meaning "comma separated values". This format is suitable for import to a wide variety of other software packages, including Microsoft Excel.

The log files have the following format:

The first line has the date and time at which the file was created. The date is given in Month/Day/Year format:

```
8/10/2004 9:37:12
```

The second line gives the data column identifiers:

```
ET (Sec) , Freq(Hz) , PWH(Sec) , PWL(Sec) , DutyCycle , Period(Sec) , Events , Logic
```

where ET is the experiment time (the Time output), Freq is Frequency in Hz, PWH and PWL are pulse width low and high times respectively, DutyCycle is the fractional duty cycle, Period is the pulse period in seconds, Events is the Counts output and Logic is the logic level output.

Following this line each further line gives a full set of outputs, with each value being separated from the next with a comma. For example, a run taken using the Internal 1 test signal might give a file as follows:

```

0,23999831,2.1340535e-008,2.0326426e-008,0.51216923,4.1666961e-008,0,0
1.0020071,23999761,2.1369484e-008,2.0297598e-008,0.51286251,4.1667082e-008,24047930,0
2.0030141,23999760,2.1370146e-008,2.0296937e-008,0.51287838,4.1667083e-008,48071860,0
3.0050212,23999761,2.1370029e-008,2.0297053e-008,0.51287559,4.1667082e-008,72119789,0
4.0060282,23999761,2.136876e-008,2.0298322e-008,0.51284513,4.1667082e-008,96143719,1
5.0080353,23999761,2.1370065e-008,2.0297016e-008,0.51287645,4.1667081e-008,1.2019165e+008,0
6.0090423,23999761,2.1370741e-008,2.029634e-008,0.51289268,4.1667081e-008,1.4421558e+008,0
7.0100493,23999761,2.1369555e-008,2.0297527e-008,0.5128642,4.1667081e-008,1.6823951e+008,0
8.0120564,23999761,2.1371843e-008,2.0295238e-008,0.51291914,4.1667081e-008,1.9228744e+008,0
9.0130634,23999762,2.1370666e-008,2.0296414e-008,0.51289089,4.1667081e-008,2.1631137e+008,1
20.519144,23999762,2.1369817e-008,2.0297264e-008,0.51287051,4.1667081e-008,4.9245456e+008,1
21.521151,23999761,2.1370785e-008,2.0296296e-008,0.51289374,4.1667081e-008,5.1650249e+008,0
22.522158,23999762,2.1370688e-008,2.0296392e-008,0.51289142,4.166708e-008,5.4052642e+008,0
23.523165,23999762,2.136977e-008,2.0297311e-008,0.51286938,4.1667081e-008,5.6455035e+008,1
24.525172,23999762,2.1369809e-008,2.0297272e-008,0.51287031,4.166708e-008,5.8859828e+008,0
25.526179,23999762,2.1371021e-008,2.029606e-008,0.5128994,4.166708e-008,6.126222e+008,0
26.528186,23999762,2.1371382e-008,2.0295698e-008,0.51290808,4.166708e-008,6.3667013e+008,0
27.529193,23999762,2.1370097e-008,2.0296982e-008,0.51287725,4.166708e-008,6.6069406e+008,1
28.5302,23999762,2.1369967e-008,2.0297113e-008,0.51287412,4.166708e-008,6.8471799e+008,0
29.532207,23999762,2.1371196e-008,2.0295884e-008,0.51290362,4.166708e-008,7.0876592e+008,0
30.533214,23999762,2.1371629e-008,2.029545e-008,0.51291402,4.166708e-008,7.3278985e+008,0

```

During this acquisition data logging was paused for about 10 seconds after the 9.0130634 s time point.

3.2.08 Connection Lost Dialog

If connection with the counter is lost while the program is running, a **Connection Lost** dialog, shown below in figure 3-5, is displayed. If this happens the program must be closed and re-started (after fixing the cause of the broken connection) in order to re-start communications with the hardware.

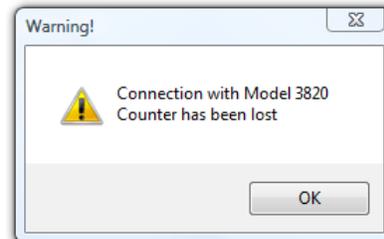


Figure 3-5, SR3820 Universal Counter – Connection Lost Dialog

3.2.09 Closing the Program

Clicking the **Close** button in the main dialog closes the program. All significant program settings are saved on exit and restored the next time the program is run. For example, if the **A** input has been used and the measurement is set to show Counts on the trendline display then these will be the settings used the next time the program is started.

3.2.10 On-Screen Help

Pressing the F1 key or clicking the **Help** button in the main dialog displays the program's Help file, shown below in figures 3-6. If the F1 key is pressed while the About or Input Thresholds dialog is visible, then the file opens automatically at the correct page.

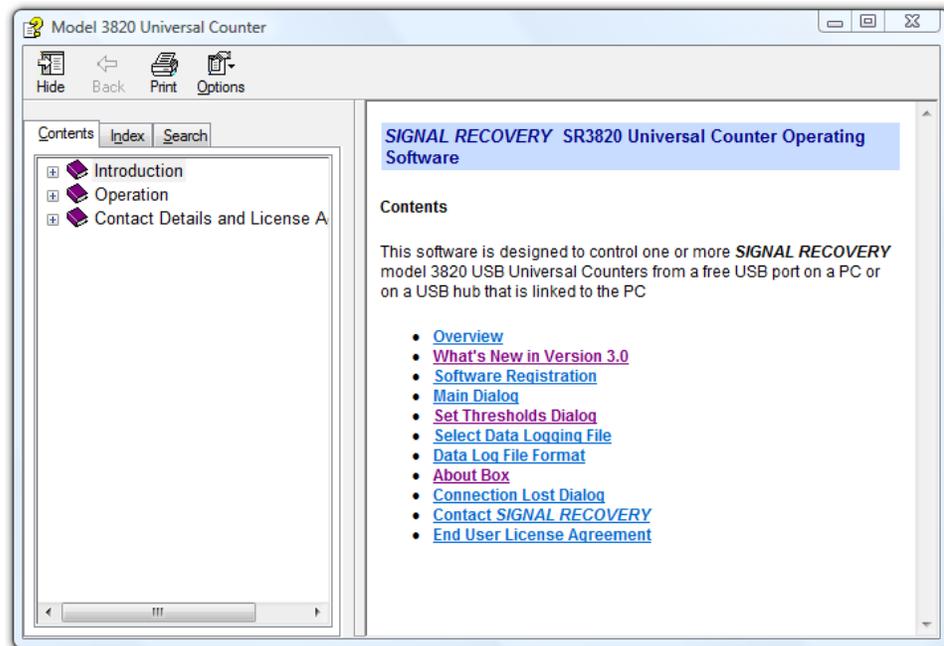


Figure 3-7, SR3820 Universal Counter – Help Screen

SR3820Comms ActiveX Control & Demonstration Files

4.1 Introduction

This chapter describes the SR3820Comms ActiveX control that can be used to control one or more **SIGNAL RECOVERY** model 3820 USB Universal Counters from a free USB port on a PC or on a USB hub that is linked to the PC. These counters allow up to eight measurement outputs to be generated as a function of time, these being:-

- Frequency
- Average period
- Average duty cycle
- Average pulse-high time
- Average pulse-low time
- Event counts
- Logic level (input voltage above/below threshold)

and

- Time

The ActiveX control supports the counter and features:-

- Support for multiple units simply by using multiple instances of the control and specifying the unit serial number that is to be operated.
- Ability to select the input channel to be used on the counter (**A** or **B**) or one of three internal test signals
- Selection of an automatic mode where the control generates notifications to the container application at regular intervals and posts an updated set of output readings
- The container application can poll the counter at any time to obtain the present value of given measurement. This approach allows the container application to fully control the overall program timing without the potential problems that can be caused if it is not able to service all notifications from the control.
- The control will notify the container application if communications are lost.

4.2 Operation

4.2.01 Overview

To use the control, first connect the model 3820 Universal Counter to a free USB port on the computer or on a USB hub connected to the computer. In the simple case when only a single Universal Counter is connected, your

application program need do only the following;

- 1) Call **Connect** using null parameter strings. The control will search for a connected Universal Counter and open a communications session with it. If this is successful, **Connect** will return TRUE; further communications should only be attempted if this is the case.
- 2) Call **Input** with an integer parameter value of 0, 1, 2, 3 or 4 to specify the required input which will be used as the input to the counter. Values of 0 and 1 correspond to the **A** and **B** input respectively, while values of 2, 3, and 4 select internal signal sources of nominally 24 MHz, 190 MHz and 272 MHz.
- 3) If using the **A** or **B** inputs, connect the external signal to one of them. The input thresholds should then be set in the range -0.2 V to $+0.5$ V using the **InputAThreshold** and **InputBThreshold** properties. If the applied signal is DC or greater than 1 kHz then it is also possible to use automatic functions called via **AutoInputAThreshold** and **AutoInputBThreshold** to set these thresholds. These functions set the threshold half-way between the minimum and maximum levels observed at the input, with each being measured over a number of milliseconds.
- 4) Call **Start** to start the counter taking measurements
- 5) To read a specific output, call **GetFrequencyAsString**, **GetTimeAsString**, **GetPeriodAsString**, **GetPulseWidthHighAsString**, **GetPulseWidthLowAsString**, **GetDutyCycleAsString**, **GetCountAsString** or **GetLogicLevelAsString**. These calls return values already formatted into strings, including suitable units. Alternatively, use the corresponding functions that return double values, such as **GetCountAsDouble** when it is necessary to perform arithmetic on the measured outputs.
- 6) To set the internal time counter and event counter back to zero, call **ResetCounter**.
- 7) Call **Stop** to stop the counter taking measurements
- 8) On program termination, call **Disconnect** to close communications with the Universal Counter.

Operation is similar when using more than one Universal Counter, but in such cases a separate instance of the SR3820Comms control is used for each unit and the **Connect** must specify the serial number of the unit to be controlled.

The above approach assumes that the application program will poll the control (and hence the counter) whenever it needs to read an output. An alternative approach is to set the control to generate notifications to the control container (i.e. your application program), and then get the latter to process these. To do this, the program needs to do the following:

- 1) Call **Connect** using null parameter strings. The control will search for a connected Universal Counter and open a communications session with it. If this

is successful, **Connect** will return TRUE; further communications should only be attempted if this is the case.

- 2) Call **Input** with an integer parameter value of 0, 1, 2, 3 or 4 to specify the required input which will be used as the input to the counter. Values of 0 and 1 correspond to the **A** and **B** input respectively, while values of 2, 3, and 4 select internal signal sources of nominally 24 MHz, 190 MHz and 272 MHz.
- 3) If using the **A** or **B** inputs, connect the external signal to one of them. The input thresholds should then be set in the range -0.2 V to $+0.5$ V using the **InputAThreshold** and **InputBThreshold** properties. If the applied signal is DC or greater than 1 kHz then it is also possible to use automatic functions called via **AutoInputAThreshold** and **AutoInputBThreshold** to set these thresholds. These functions set the threshold half-way between the minimum and maximum levels observed at the input, with each being measured over a number of milliseconds.
- 4) Set the **Rate** property to specify the rate at which the control will generate new output readings. The value is specified in milliseconds in the range 10 ms to 10,000 ms (10 seconds)
- 5) Set the boolean properties **ResultsAsStringsEvents**, **ResultsEvents**, and/or **ResultsWithStatusEvents** to specify the output types that will be generated by the control at each notification. **ResultsAsStringEvents** generates a full set of output readings already formatted into strings complete with units. **ResultsEvents** generates outputs expressed in numerical format (as doubles) suitable for numerical processing, while **ResultsWithStatusEvents** is the same as **ResultsEvents** but provides an additional status value for each output. This can be used, for example, to indicate when the internal counter has overflowed.
- 6) Call **Start** to start the counter taking measurements. The control will now start generating notifications to the container, depending on the boolean properties **ResultsAsStringsEvents**, **ResultsEvents**, and **ResultsWithStatusEvents**. The container application needs to process these events which are **ResultsAsStrings**, **Results**, and **ResultsWithStatus** respectively.
- 7) To set the internal time counter and event counter back to zero, call **ResetCounter**.
- 8) Call **Stop** to stop the counter taking measurements
- 9) On program termination, call **Disconnect** to close communications with the Universal Counter.

4.2.02 Property Page

The control includes a property page (figure 4-1) that shows the active input, input thresholds, update rate and selected output for the connected model 3820 Universal Counter. Note that the property page is only effective when a counter is active and has been opened with the **Connect** method.

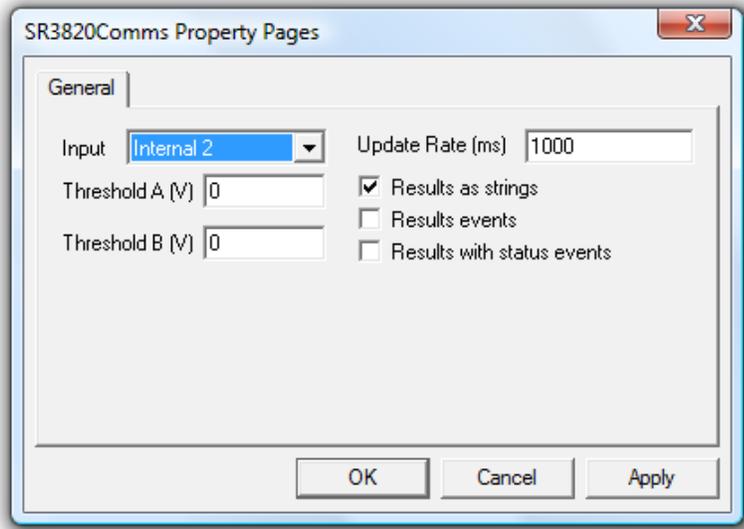


Figure 4-1, SR3820Comms Control Property Page

The method of displaying this property page depends on the control container application.

4.3 Control Methods

The SR3820Comms control provides 37 control methods, listed below, nine properties, listed in section 4.4 and four event notifications, listed in section 4.5

4.3.01 Connect

boolean Connect(BSTR strNetworkPath, BSTR strSerialNumber, BSTR bAskDlg)

Description:

This method attempts to open communications with a model 3820 via the USB interface. It must be the first method called when using the control, and other methods must not be used unless it returns TRUE.

The parameters have the following significance:

- strNetworkPath** This parameter is not used and should be set to a null string.
- strSerialNumber** Use this parameter to force the control to connect to a specific hardware device, determined by the Serial Number. If the specified device is not present on the bus, Connect will return FALSE.
- bAskDlg** When this flag is True and there is more than one 3820 Universal Counter on the bus, a "device picker" dialog, shown below in figure 4-2, opens allowing the user to pick the Universal Counter with which communications will be established. Highlight the required unit and click on Select to choose it as the Universal Counter that is to be used.

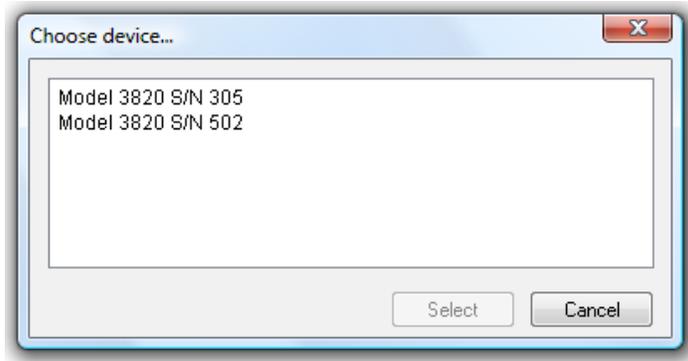


Figure 4-2, Device Picker Dialog

Returns:

TRUE if connection is successful, FALSE otherwise

4.3.02 IsConnected

boolean IsConnected()

Description:

This method determines whether a communications session is currently open with a model 3820 Universal Counter. Use **Connect** to open a session and **Disconnect** to close it.

Returns:

TRUE if connection is present, FALSE otherwise

4.3.03 Disconnect

void Disconnect()

Description:

This method closes a communications session with a model 3820 Universal Counter, and should always be called at the end of a program using the SR3820Comms control. Use **Connect** to open a session and **IsConnected** to determine if a session is currently open.

Returns:

Nothing

4.3.04 Reset

void Reset()

Description:

This method causes a complete reset of the instrument hardware. The operating firmware is reloaded from the internal EEPROM and the firmware is then restarted. All measurement outputs are reinitialized and the event counter and timer are set to zero.

Returns:
Nothing

4.3.05 AutoInputAThreshold

double AutoInputAThreshold()

Description:

If the signal connected to the **A** input is DC or greater than 1 kHz then this method will cause the hardware to automatically determine the optimum voltage threshold for this input.

Returns:

Double value of the new threshold voltage, expressed in volts, in the range -0.2 V to $+0.5$ V.

4.3.06 AutoInputBThreshold

double AutoInputBThreshold()

Description:

If the signal connected to the **B** input is DC or greater than 1 kHz then this method will cause the hardware to automatically determine the optimum voltage threshold for this input.

Returns:

Double value of the new threshold voltage, expressed in volts, in the range -0.2 V to $+0.5$ V.

4.3.07 Start

void Start()

Description:

This method starts the internal update timer of the control running at a rate specified by the **Rate** property, in the range 10 ms to 10,000 ms (10 seconds). This timer causes a new set of output measurements to be computed, and if any of the three boolean properties **ResultsAsStringsEvents**, **ResultsEvents**, or **ResultsWithStatusEvents** are set TRUE then a corresponding event notification is sent to the control container.

Returns:

Nothing.

4.3.08 ResetCounter

void ResetCounter()

Description:

This method resets the time and event counter values to zero.

Returns:

Nothing.

4.3.09 Stop

void Stop()

Description:

This method stops the internal update timer of the control, and so stops the generation of any further sets of output measurements. Note however that the time and event counter values are not reset to zero, so that if a subsequent **Start** method is used then the counter will re-start using the values of these outputs at the time that the **Stop** method was called.

To reset the time and event counter values to zero, use the **ResetCounter** method.

Returns:

Nothing.

4.3.10 GetTimeAsString

BSTR GetTimeAsString()

Description:

This method returns the most recent Time measurement as a formatted string complete with units indicator. This time is the time since the counter was started, either on power-up or on use of the **Reset** method, or since it was reset using the **ResetCounter** method.

Returns:

String containing the most recent Time measurement complete with units (ms or s) indicator.

4.3.11 GetFrequencyAsString

BSTR GetFrequencyAsString()

Description:

This method returns the most recent Frequency measurement as a formatted string complete with units indicator. Note that the units of Hertz are of course equivalent to Counts per Second which is a more commonly used term in some counting applications.

Returns:

String containing the most recent Frequency measurement complete with units (Hz, kHz or MHz) indicator.

4.3.12 GetPulseWidthHighAsString

BSTR GetPulseWidthHighAsString()

Description:

This method returns the most recent Pulse Width High time measurement as a formatted string complete with units indicator.

Returns:

String containing the most recent Pulse Width High measurement complete with units (ns, μ s, ms, s, ks) indicator.

4.3.13 GetPulseWidthLowAsString

BSTR GetPulseWidthLowAsString()

Description:

This method returns the most recent Pulse Width Low time measurement as a formatted string complete with units indicator.

Returns:

String containing the most recent Pulse Width Low measurement complete with units (ns, μ s, ms, s, ks) indicator.

4.3.14 GetPeriodAsString

BSTR GetPeriodAsString()

Description:

This method returns the most recent Period time measurement as a formatted string complete with units indicator.

Returns:

String containing the most recent Period measurement complete with units (ns, μ s, ms, s, ks) indicator.

4.3.15 GetDutyCycleAsString

BSTR GetDutyCycleAsString ()

Description:

This method returns the most recent DutyCycle measurement (essentially the division (Pulse Width High)/(Pulse Period) expressed in terms of a percentage) as a formatted string complete with “%” indicator.

Returns:

String containing the most recent DutyCycle measurement complete with “%” indicator.

4.3.16 GetCountsAsString

BSTR GetCountsAsString()

Description:

This method returns the most recent number of event Counts recorded as a formatted string. The number of counts is the total number recorded since the counter was started, either on power-up or on use of the **Reset** method, or since it was reset using the **ResetCounter** method.

Returns:

String containing the most recent number of event Counts.

4.3.17 GetLogicLevelAsString

BSTR GetLogicLevelAsString()

Description:

This method returns the most recent logic level of the selected input as a formatted string. If the applied voltage is below the set threshold the response is “0”; if it is above the threshold the response is “1”.

Returns:

String containing the most recent logic level of the selected input.

4.3.18 GetTimebaseCorrectionAsString

BSTR GetLogicLevelAsString()

Description:

This method is reserved for future use.

Returns:

Not specified.

4.3.19 GetTimeAsDouble

double GetTimeAsDouble()

Description:

This method returns the most recent Time measurement as a double value. This time is the time in seconds since the counter was started, either on power-up or on use of the **Reset** method, or since it was reset using the **ResetCounter** method.

Returns:

Double value of the most recent Time measurement expressed in seconds.

4.3.20 GetFrequencyAsDouble

double GetFrequencyAsDouble()

Description:

This method returns the most recent Frequency measurement as a double value in units of Hertz. Note that the units of Hertz are of course equivalent to Counts per Second which is a more commonly used term in some counting applications.

Returns:

String containing the most recent Frequency measurement expressed in Hertz.

4.3.21 GetPulseWidthHighAsDouble

double GetPulseWidthHighAsDouble()

Description:

This method returns the most recent Pulse Width High time measurement as a double value.

Returns:

Double value containing the most recent Pulse Width High measurement expressed in seconds.

4.3.22 GetPulseWidthLowAsDouble

double GetPulseWidthLowAsDouble()

Description:

This method returns the most recent Pulse Width Low time measurement as a double value.

Returns:

Double value containing the most recent Pulse Width Low measurement expressed in seconds.

4.3.23 GetPeriodAsDouble

double GetPeriodAsDouble()

Description:

This method returns the most recent Pulse Period measurement (essentially the sum of the Pulse Width Low time and the Pulse Width High time) as a double value, expressed in seconds.

Returns:

Double value containing the most recent Pulse Period measurement expressed in seconds.

4.3.24 GetDutyCycleAsDouble

double GetDutyCycleAsDouble ()

Description:

This method returns the most recent DutyCycle measurement (essentially the division (Pulse Width High)/(Pulse Period)) as a double value.

Returns:

Double value containing the most recent DutyCycle measurement expressed in the range 0 to 1.

4.3.25 GetCountsAsDouble

double GetCountsAsDouble()

Description:

This method returns the most recent number of event Counts recorded as a double value. The number of counts is the total number recorded since the counter was started, either on power-up or on use of the **Reset** method, or since it was reset using the **ResetCounter** method.

Returns:

Double value containing the most recent number of event Counts.

4.3.26 GetLogicLevelAsDouble

double GetLogicLevelAsDouble()

Description:

This method returns the most recent logic level of the selected input as double value. If the applied voltage is below the set threshold the response is “0”; if it is above the threshold the response is “1”.

Returns:

Double value containing the most recent logic level of the selected input.

4.3.27 GetTimebaseCorrectionAsDouble

double GetLogicLevelAsDouble()

Description:

This method is reserved for future use.

Returns:

Not specified.

4.3.28 GetSerialNumber

BSTR GetSerialNumber()

Description:

This method returns the serial number of the connected model 3820 Universal Counter.

Returns:

String containing the instrument's serial number.

4.3.29 GetModelNumber

BSTR GetModelNumber()

Description:

This method returns the model number of the connected instrument. In the case of the model 3820 this is "3820 Universal Counter"

Returns:

String containing the instrument's model number and type.

4.3.30 GetHWRevision

BSTR GetHWRevision()

Description:

This method returns the hardware revision level.

Returns:

String containing the instrument's hardware revision level.

4.3.31 GetHWManufactureDate

BSTR GetHWManufactureDate()

Description:

This method returns the date of manufacture of the hardware.

Returns:

String containing the date of manufacture of the hardware.

4.3.32 GetHWCalibrationDate

BSTR GetHWCalibrationDate()

Description:

This method returns the date of calibration of the hardware.

Returns:

String containing the date of calibration of the hardware.

4.3.33 GetSR3820CommsXVersion

BSTR GetSR3820CommsXVersion()

Description:

This method returns a string containing the version number of the SR3820Comms ActiveX control.

Returns:

String containing the SR3820Comms ActiveX version number.

4.3.34 GetUSBXVersion

BSTR GetUSBXVersion()

Description:

This method is no longer used, but is retained for reasons of compatibility.

4.3.35 AboutBox

void AboutBox()

Description:

Calling this method displays the control's "About" dialog which shows copyright and version information.

Returns:

Nothing

4.3.36 DoDiagnostics

boolean DoDiagnostics(long lTestIdx)

Description:

This method is reserved for future use.

4.3.37 LoadEEProm

void LoadEEProm(BSTR strFilePath)

Description:

This method is reserved for internal use.

Returns:

Nothing.

4.4 Control Properties

4.4.01 Input

long Input

Description:

This property sets or retrieves the input that is selected as input to the Universal Counter. It has the following values:

Value	Selected Input
0	A
1	B
2	Internal test source, nominally 24 MHz
3	Internal test source, nominally 190 MHz
4	Internal test source, nominally 272 MHz

4.4.02 InputAThreshold

long InputAThreshold

Description:

This property sets or retrieves the threshold voltage for the **A** input in the range -0.2 V to +0.5 V. Note that if a x10 external probe is used then these thresholds translate to -2.0 V to +5.0 V at the probe tip. Note that the **AutoInputAThreshold** method can be used to automatically select a suitable threshold as long as the applied signal is changing.

For reference, the typical threshold voltages and logic voltage swings for common logic families are:

Family/Signal	Threshold	Voltage Swing
TTL	1.4 V	0 V to +3.5 V
CMOS	2.5 V	0 V to +5 V
LV-CMOS (3.3V)	1.4 V	0 V to 3.3 V
AC-coupled Sine	0.0 V	N/A
ECL: (-5.2v)	-1.35 V	-1.8 V to -0.9 V
PECL (+5V)	3.65 V	+3.2 V to +4.2 V
LVPECL (+3.3V)	2 V	+1.5 V to +2.5 V
Fast Negative NIM	-0.4 V	0 V to -0.8 V

4.4.03 InputBThreshold

long InputBThreshold

Description:

This property sets or retrieves the threshold voltage for the **B** input in the range -0.2 V to +0.5 V. Note that if a x10 external probe is used then these thresholds translate to -2.0 V to +5.0 V at the probe tip.

Note that the **AutoInputBThreshold** method can be used to automatically select a suitable threshold as long as the applied signal is changing.

For reference, the typical threshold voltages and logic voltage swings for common logic families are:

Family/Signal	Threshold	Voltage Swing
TTL	1.4 V	0 V to +3.5 V
CMOS	2.5 V	0 V to +5 V
LV-CMOS (3.3V)	1.4 V	0 V to 3.3 V
AC-coupled Sine	0.0 V	N/A
ECL: (-5.2v)	-1.35 V	-1.8 V to -0.9 V
PECL (+5V)	3.65 V	+3.2 V to +4.2 V
LVPECL (+3.3V)	2 V	+1.5 V to +2.5 V
Fast Negative NIM	-0.4 V	0 V to -0.8 V

4.4.04 Rate

double Rate

Description:

This property sets the rate of the internal timer for the control in the range 10 ms to 10,000 ms (10 seconds). The timer causes a new set of output measurements to be computed, and if any of the three boolean properties **ResultsAsStringsEvents**, **ResultsEvents**, or **ResultsWithStatusEvents** are set TRUE then a corresponding event notification is sent to the control container.

4.4.05 ResultsAsStrings

boolean ResultsAsStrings

Description:

This property sets whether the control will fire a notification to the container each time the internal timer, as set by the **Rate** property, times out and supply the container with a complete set of measurements, each formatted into a string complete with units indicator. If the property is set TRUE then these notifications are generated; if it is set FALSE then they are not.

4.4.06 ResultsEvents

boolean ResultsEvents

Description:

This property sets whether the control will fire a notification to the container each time the internal timer, as set by the **Rate** property, times out and supply the container with a complete set of measurements, each expressed as a double value. If the property is set TRUE then these notifications are generated; if it is set FALSE then they are not.

4.4.07 ResultsWithStautusEvents

boolean ResultsWithStautusEvents

Description:

This property sets whether the control will fire a notification to the container each time the internal timer, as set by the **Rate** property, times out and supply the container with a complete set of measurements, each expressed as a double value, together with corresponding status values, one for each measurement output. If the property is set TRUE then these notifications are generated; if it is set FALSE then they are not.

4.4.08 Simulate

boolean Simulate

Description:

This property has no effect, but is retained for reasons of compatibility.

4.4.09 Test

long Test

Description:

This property is reserved for internal use.

4.4.10 bMonitorPNP

boolean bMonitorPNP

Description:

When this property is set TRUE, the control will notify the container if communications are lost between the multiplexer and the computer, by firing the ConnectionLost event.

4.5 Control Events

4.5.01 Results

void Results(double dTime, double dFrequency, double dPulseWidthHigh, double dPulseWidthLow, double dDutyCycle, double dPeriod, double dEvent, double dLogic, double dTimeBaseCorr)

Description:

This event is fired when the **ResultsEvents** property is set TRUE and the counter has been started using **Start**. It returns a full set of measurements, each expressed as a double value.

4.5.02 ResultsAsStrings

void ResultsAsStrings(BSTR strTime, BSTR strFrequency, BSTR strPulseWidthH, BSTR strPulseWidthL, BSTR strDutyCycle, BSTR strPeriod, BSTR strEvent, BSTR strLogic, BSTR strTimeBaseCorr)

Description:

This event is fired when the **ResultsAsStringsEvents** property is set TRUE and the counter has been started using **Start**. It returns a full set of measurements, each expressed as a string complete with units. There is always a space character between the value and the units, which can be useful when parsing the response to separate it into parts.

4.5.03 ResultsWithStatus

void ResultsWithStatus(double dTime, double dFrequency, double dPulseWidthHigh, double dPulseWidthLow, double dDutyCycle, double dPeriod, double dEvent, double dLogic, double dTimeBaseCorr, long lStatusTime, long lStatusFrequency, long lStatusPulseWidthHigh, long lStatusPulseWidthLow, long lStatusDutyCycle, long lStatusPeriod, long lStatusEvent, long lStatusLogic)

Description:

This event is fired when the **ResultsWithStatusEvents** property is set TRUE and the counter has been started using **Start**. It returns a full set of measurements, each expressed as a double value, together with a status value for each measurement.

The status values are bitmasks indicating current measurement conditions and have the following bit-significance:

Bit	Description if TRUE
0	Stale measurement (not able to produce valid result on the most recent measurement).
1	Overflow; Data is out of representable range.
2	Invalid; No result could be meaningfully computed.
3	Error; Error occurred during the measurement process.

4.5.04 ConnectionLost()

void ConnectionLost()

Description:

The control fires this message to the container application if for any reason communications are lost between it and the hardware. Typically the application will then display a dialog box to warn the user that this has happened.

4.6 Demonstration Files

The SR3820Comms control is supplied with demonstration files showing how it can be used in with Visual C++, Visual Basic, VBScript, LabVIEW 8.2 and Excel 2003. These files are installed by default in subfolders of the c:\Program Files\Signal Recovery\SR3820Universal Counter\Demonstration Files folder.

4.6.01 Visual C++

The files supplied form a workspace which may be opened in Visual C++ version 6.0 When compiled, the result is a simple dialog box application allowing the counter input to be selected, the input thresholds to be adjusted and any one of the eight measurement outputs to be displayed when the **Read Output** button is clicked.

Figure 4-2 shows the main dialog box of this application. Click the **About** button to show the control's About dialog box

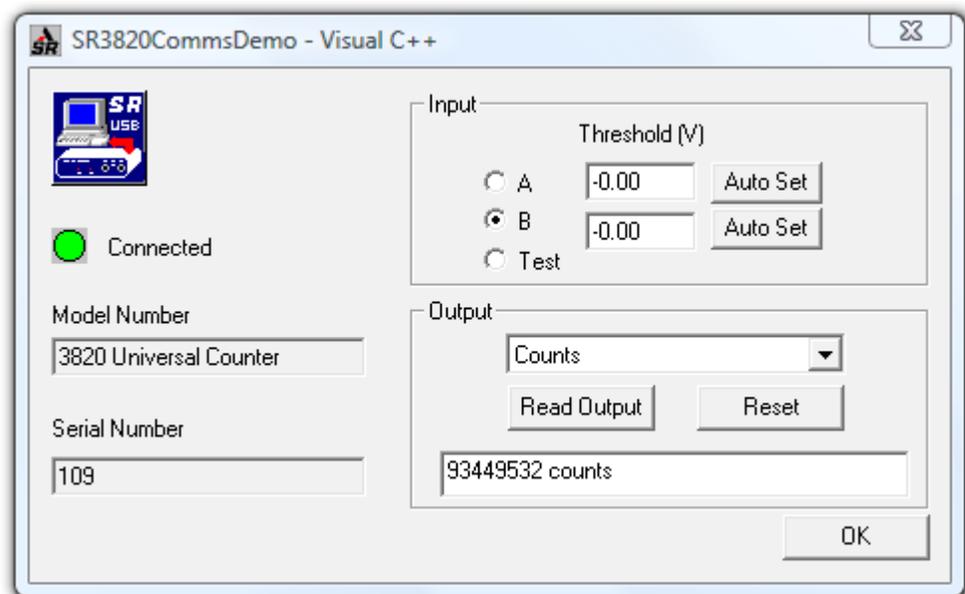


Figure 4-2 SR3820CommsDemo.exe C++ Main Dialog Box

4.6.02 Visual Basic

The files supplied form a workspace which may be opened in Visual Basic version 6.0. When compiled, the result is similar to the application generated by the Visual C++ demonstration files. A compiled version of this project, SR3820Comms.exe, is also included in the demonstration folder as an executable file that can be run “as supplied”.

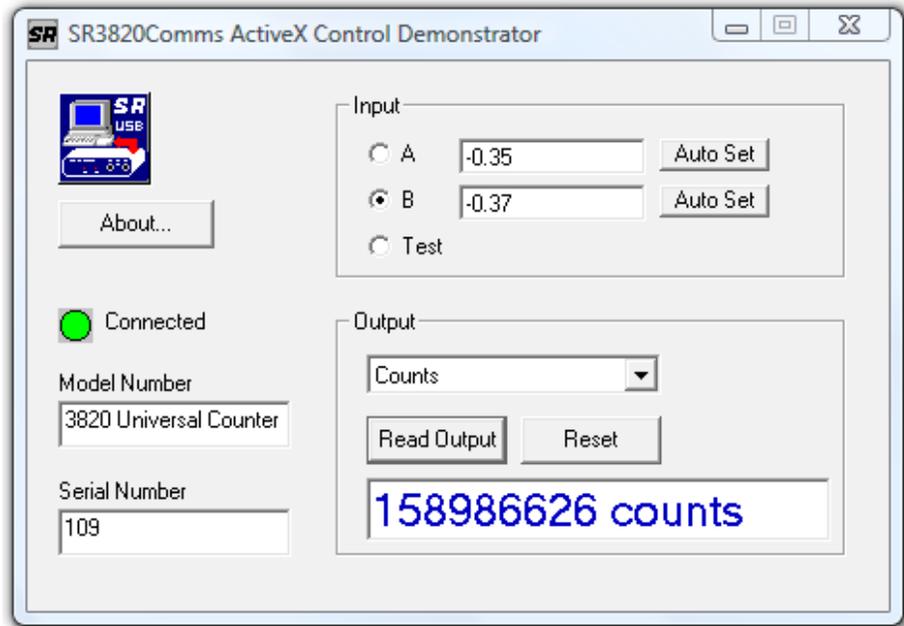


Figure 4-3 SR3820Comms.exe Visual Basic Main Dialog Box

Click the **About** button to show the control’s About dialog box.

4.6.03 VBScript (HTML)

The supplied file can be run under Internet Explorer and provides a simple interface to operate the instrument.

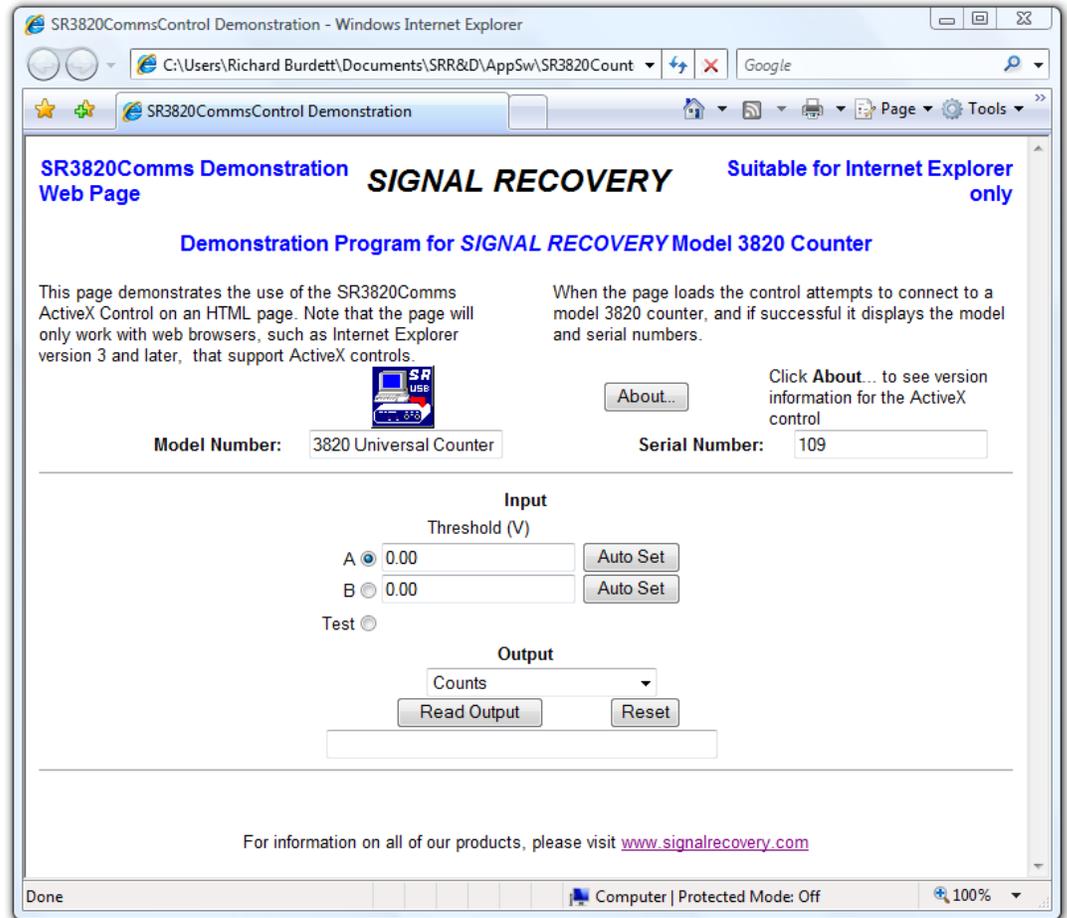


Figure 4-4 SR3820Comms HTML Demonstration File

Click the **About** button to show the control's About dialog box.

4.6.04 Excel 2003

The demonstration workbook contains one worksheet. The sheet has similar capabilities to the other example projects, but also shows how the counter can be programmed to output a series of measurements, which are then shown graphically.

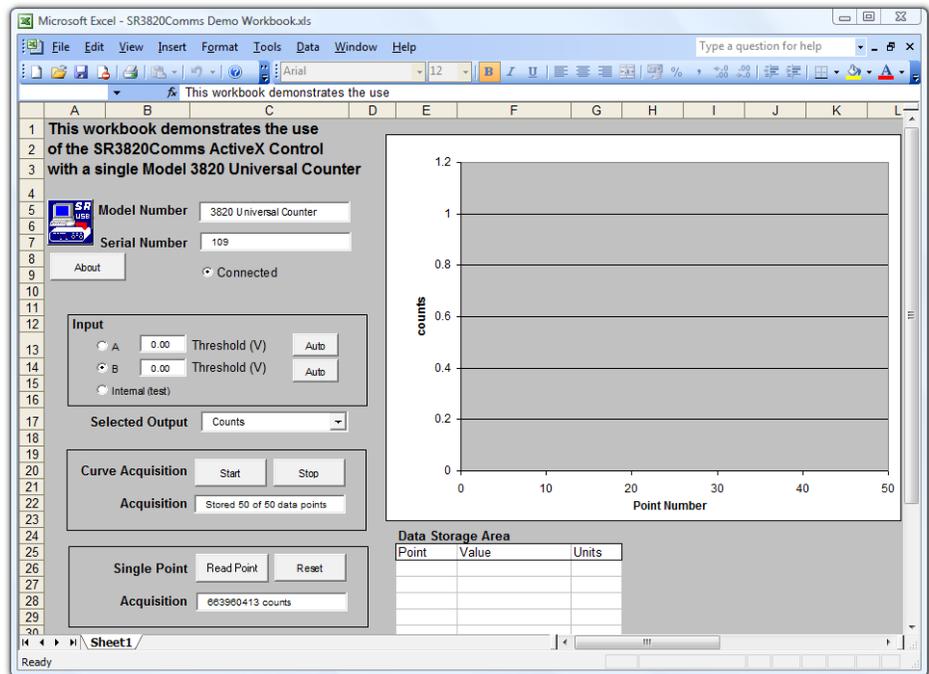


Figure 4-5 SR3820Comms Demo Workbook.xls Excel 2003 Workbook

Click the **About** button to show the control’s About dialog box.

4.6.05 LabVIEW 8.2

The demonstration VI offers similar functionality to the VisualBasic and VisualC++ projects.

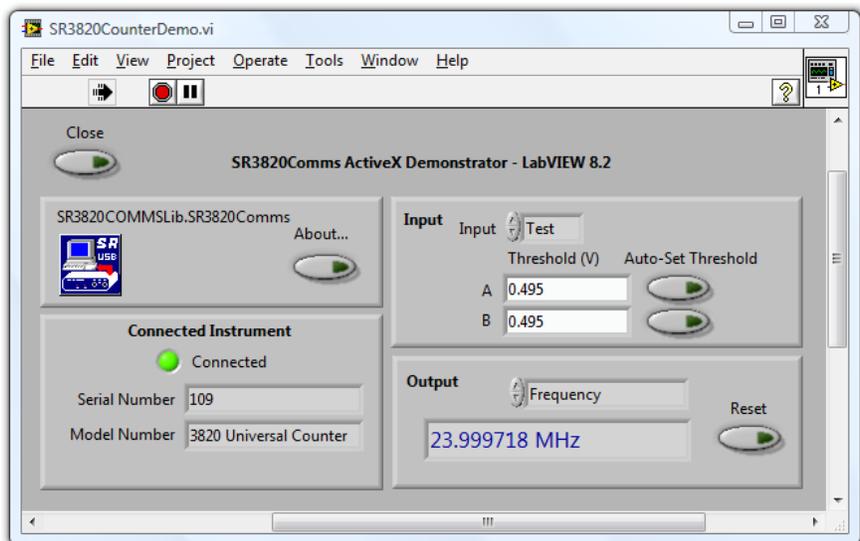


Figure 4-6 SR3820CommsDemo.vi LabVIEW VI - Front Panel

Click the **About** button to show the control’s About dialog box

Specifications

A.1 Signal Inputs

Impedance	1 M Ω , DC coupled
Threshold	
Direct	-0.2 V to +0.5 V in 1 mV steps
With x10 probe	-2 V to +5 V in 10 mV steps
Polarity	Event counter triggers on rising edge of signal
Sensitivity	
Direct	-15 dBm/50 Ω (23 mV rms) at 100 MHz, -10 dBm/50 Ω (0.7 V rms) at 120 MHz
With x10 probe	1 V pk-pk sinewave at 125 MHz
Absolute max input	\leq 50 V DC

A.2 Measurements

Frequency Range	DC to 125 MHz min, 160 MHz typ
Timebase accuracy	\leq 5 ppm, 0 to 50 $^{\circ}$ C
Reporting Intervals	100, 50, 25, 10 and 5 ms
Functions	
Frequency	0 to 125 MHz
Average Period	\geq 8 ns
Average Duty Cycle	0 to 100%. Measured by sampling with a 65 MHz clock.
Average Pulse High or Low Time	\geq 5ns. Computed from Duty Cycle and Frequency
Events	0 to 9,999,999,999 counts
Logic Level	0 = input voltage below threshold, 1 = input voltage above threshold.

A.3 Indicators

USB	Front-panel LED turns on during USB communications
EVENT	Front-panel LED turns on when the input signal is above the threshold

A.3 Computer Interface

USB Connector	Rear-panel, female USB connector for connection to the PC or a USB hub.
System Requirements	Windows XP SP2 or Windows Vista (32-bit) USB port available CD-ROM drive if installing from CD Minimum 16-bit color display (32,000 colors)

A.4 General

Power Requirements	
Voltage	<500 mA @ +5 V DC, supplied via USB
Dimensions	
Width	5½" (134 mm)
Depth	4½" (114 mm)
Height	1¼" (32 mm)
Weight	12 oz (350 g)
EMC	Meets EU standards (CE) for emissions and susceptibility.

B.1 Introduction

This appendix provides details of the software license and support policy which applies to the SR3820 Universal Counter software and SR3820Comms Active X control and discusses how you may obtain help in using it from us. It also discusses how you may help us by reporting any errors you discover and making suggestions for new features.

SIGNAL RECOVERY is granting you a non-exclusive license to use the software programs and associated documentation (collectively, the "Software") for your internal use in accordance with the following terms:

B.2 License

You may:

- a) use the Software (including any subsequent updates) on a single computer system;
- b) copy the Software into any machine readable or printed form for backup or modification purposes in support of your use of the Software on a single computer, provide that you make only one such copy for such purposes;
- c) modify the Software and/or merge it into another program for **your personal use only on a single machine** (Any portion of the Software merged into another program will continue to be subject to the terms and conditions of this Agreement);
- d) transfer the Software and this license to a third party if the other party agrees to accept all the terms and conditions of this Agreement. If you transfer the Software, you must at the same time notify the name and address of the third party to **SIGNAL RECOVERY** and either transfer all copies, whether in printed or machine readable form, to the same party or destroy all copies not transferred; this includes all modifications and all portions of the Software contained in or merged into other programs.

You must reproduce and include the copyright notice on any copy, modification or portion merged into another program.

B.2.01 Updates and Upgrades

If this Software is an update or upgrade, then the purchaser must have the original Software and license Agreement which becomes part of this Agreement. The original Software may not be transferred to another computer or party except as part of this product.

You may not use, copy, modify, or transfer this Software or any copy, modification or merged portion of it, in whole or in part except as expressly provided for in this

license.

If you transfer possession of any copy, modification or merged portion of this Software to another party, your license is automatically terminated.

B.2.02 Term

The license is effective until terminated. You may terminate it at any time by destroying the Software together with all copies, modifications and merged portions in any form. It will also terminate upon conditions set forth elsewhere in this Agreement. You agree upon such termination to destroy the Software together with all copies, modifications and merged portions in any form.

B.2.03 General

You may not sublicense, assign or transfer the license or the Software except as expressly provided in this Agreement. Any attempt otherwise to sublicense, assign or transfer any of the rights, duties or obligations hereunder is void.

If you wish to incorporate the SR3820Comms control into your own application software (with or without the SR3820 Universal Counter program) for sale, distribution or use by anyone other than yourself you must obtain a separate license from us to cover this distribution.

This Agreement will be governed by the laws of the state of Tennessee, USA.

B.3 Software Registration

SIGNAL RECOVERY will provide technical support in the use of the SR3820 Universal Counter program and/or SR3820Comms Active X control to one registered, named, user for each copy purchased. In order to allow this to operate you must therefore complete the product registration form located on our website at www.signalrecovery.com/register

Once registered, you may then telephone, fax or e-mail your query to the **SIGNAL RECOVERY** office in Wokingham, UK. You must give the model 3820 serial number (located on the rear panel), together with the registered user's name, on each occasion you contact us, or support will be denied.

If at all possible, when telephoning, please be in front of the computer and lock-in amplifier you are using, so that you may immediately try our suggestions. You must also be able to tell us the type of computer and computer interface you are using, the versions of the SR3820 Universal Counter and SR3820Comms Active X control software and the version of Windows.

Registered users of the software will also be informed of future product upgrades and enhancements.

B.4 Software Errors

The SR3820 Universal Counter program and SR3820Comms Active X control have been extensively tested prior to release, but it is possible that some errors remain which were not revealed by the testing process. **SIGNAL RECOVERY** would very much appreciate receiving, by e-mail to info@signalrecovery.com or by post or fax to the Wokingham, UK, office, written notification of any problems you discover.

When reporting problems please try to be as specific as possible. Include details of the type of instrument used, the revision level of the instrument firmware and the type of interface and computer used.

If you feel there is a specific feature that would add to the software's functionality then please let us know. Naturally we cannot commit ourselves to adding any given new feature, but even so such suggestions are useful in identifying those areas in which the product may be improved.

Uninstalling Earlier Software Versions

C.1 Overview

If it is intended to install version 3.0 the SR3820 Universal Counter software on a computer that already has an earlier version of the software installed, then we recommend that the older software first be uninstalled. This section describes how to do this, and how to remove references to the device driver files used by the older software.

C.2 Uninstalling the Operating Software

These instructions apply only to systems running Windows XP. Version 3.0 of the software is not suitable for systems running older operating systems, so there will be no reason to remove the original software from such systems. In addition, since the older software cannot be installed on systems running Vista, there will be no need to remove it from such systems before installing version 3.0

Note that these procedures require that you be logged onto the PC with full administrator rights, and that model 3820 Universal Counter be plugged into the computer and capable of working when the Universal Counter software is running.

Click on **Start - Settings - Control Panel - Add/Remove Programs** and scroll down the list of available programs until you find the entry “**SR3820 Universal Counter**”. Click on the entry to select it, as shown below in figure C-1

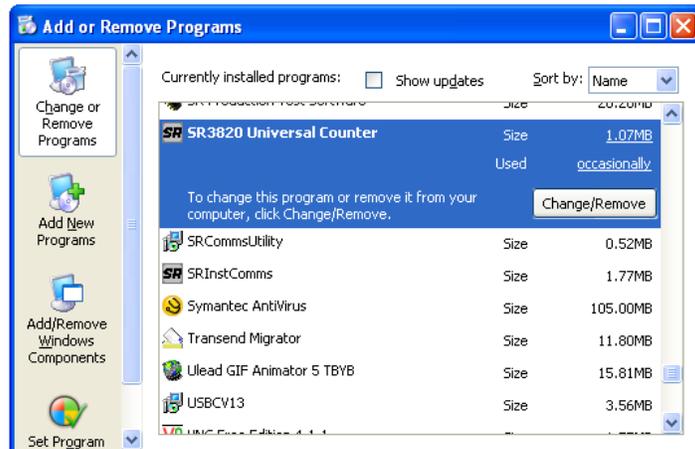


Figure C-1, Locating SR3820 Universal Counter Software

Click on **Change/Remove** and then click **Yes** in the next dialog, shown below in figure C-2.

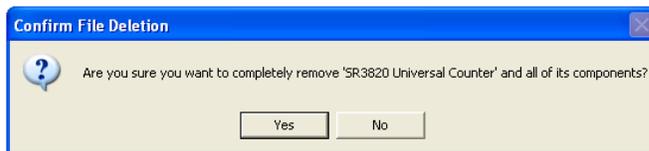


Figure C-2, Confirm File Deletion

Follow any further prompts to completely remove the SR3820 Universal Counter software version 2.0 from the computer. If prompted as to whether to remove the shared components **SR3820Comms.ocx** and **IUSBX.DLL** then you may accept their deletion.

This procedure has now completely removed the operating software, demonstration files, and underlying ActiveX control. However, the system will still have references to device driver files in its system .inf file cache. Follow the procedure in section A-3 to remove these references.

C.3 Removing References to old Driver Files

Click on **Start - Settings - Control Panel - System**. The System Properties dialog will be shown. Navigate to the **Hardware** tab, and click on **Device Manager**, to show all installed hardware. Scroll down the tree until you reach the **SIGNAL RECOVERY** item; expand the tree, and select the item **3820 Universal Counter**. Left-click on the entry to display the pop-up menu, shown below in figure C-3.



Figure C-3, Device Manager Dialog, 3820 Universal Counter Selected

Select **Uninstall**, and click **Yes** at the confirmation dialog, shown below in figure C-4



Figure C-4, Confirm Device Removal

This process removes the original driver files.

Finally, it is necessary to remove the copy of the original .inf file that the operating system placed in the **c:\windows\inf** folder on the computer when the original software was installed. This file will have a name in the format **oemnnnn.inf**, where **nnnn** is an integer, and on a typical machine there may be twenty or more such files.

In order to make it simple to locate and remove the correct file, the distribution CD or downloaded zip file contains a folder **Uninstall38203830**, which in turn contains a simple utility **Uninstall38203830.exe**.

Browse to this folder and start the utility by double-clicking the **Uninstall38203830.exe** item. The dialog shown in figure C-5 will be displayed.



Figure C-5, Uninstall 3820 3830 Main Dialog

Click on the **Uninstall V2.0 Drivers** button. The program searches for the correct .inf file and deletes it, and any associated compiled .pnf file as well. If the file is found the response is as shown in figure C-6, whereas if it is not found then the response is as in figure C-7.

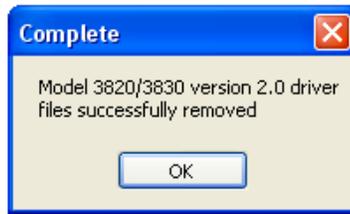


Figure C-6, Uninstall 3820 3830 - File Removal Success

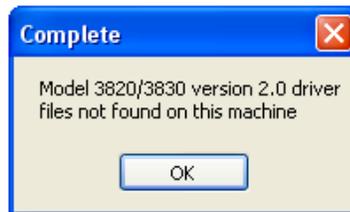


Figure C-7, Uninstall 3820 3830 - File Removal Failure

Click **OK** to close the information box, and **Close** to close the **Uninstall38203830** program. Finally, unplug the model 3820.

The system is now ready to install the updated version 3.0 software

- Compatible instruments 2-1
- Computer requirements 2-1
- Examples supplied 4-18
- Excel demonstration workbook 4-20
- Initial Checks procedure 2-2
- Installation 2-1
- Installation procedure 2-2
- LabVIEW demonstration VI 4-21
- Media 2-1
- Product Registration 2-1
- SENSE-ations model 3820, changes from 1-2
- Software errors, reporting B-3
- Software license B-1
- Software registration 2-1, B-2
- Software support B-1
- SR3820Comms events
 - ConnectionLost 4-17
 - Results 4-3, 4-16
 - ResultsAsStrings 4-3, 4-17
 - ResultsWithStatus 4-3, 4-17
- SR3820Comms methods
 - AboutBox 4-13
 - AutoInputAThreshold 4-2, 4-3, 4-6
 - AutoInputBThreshold 4-2, 4-3, 4-6
 - Connect 4-2, 4-3, 4-4
 - Disconnect 4-2, 4-3, 4-5
 - DoDiagnostics 4-13
 - GetCountasDouble 4-2
 - GetCountAsString 4-2
 - GetCountsAsDouble 4-11
 - GetCountsAsString 4-9
 - GetDutyCycleAsDouble 4-11
 - GetDutyCycleAsString 4-8
 - GetDutyCycleAsString 4-2
 - GetFrequencyAsDouble 4-10
 - GetFrequencyAsString 4-2, 4-7
 - GetHWCalibrationDate 4-12
 - GetHWManufactureDate 4-12
 - GetHWRevision 4-12
 - GetLogicLevelAsDouble 4-11
 - GetLogicLevelAsString 4-9
 - GetLogicLevelAsString 4-2
 - GetModelNumber 4-12
 - GetPeriodAsDouble 4-10
 - GetPeriodAsString 4-8
 - GetPeriodAsString 4-2
 - GetPulseWidthHighAsDouble 4-10
 - GetPulseWidthHighAsString 4-8
 - GetPulseWidthHighAsString 4-2
 - GetPulseWidthLowAsDouble 4-10
 - GetPulseWidthLowAsString 4-8
 - GetPulseWidthLowAsString 4-2
 - GetSerialNumber 4-12
 - GetSR3820CommsXVersion 4-13
 - GetTimeAsDouble 4-9
 - GetTimeAsString 4-2, 4-7
 - GetTimebaseCorrectionAsDouble 4-11
 - GetTimebaseCorrectionAsString 4-9
 - GetUSBXVersion 4-13
 - IsConnected 4-5
 - LoadEEProm 4-14
 - Reset 4-5
 - ResetCounter 4-2, 4-3, 4-6
 - Start 4-2, 4-3, 4-6
 - Stop 4-2, 4-3, 4-7
- SR3820Comms properties
 - Input 4-2, 4-3, 4-14
 - InputAThreshold 4-2, 4-3, 4-14
 - InputBThreshold 4-2, 4-3, 4-15
 - Rate 4-3, 4-15
 - ResultsEvents 4-16
 - ResultsAsStrings 4-15
 - ResultsAsStringsEvents 4-3
 - ResultsEvents 4-3
 - ResultsWithStatusEvents 4-3
 - ResultsWithStautusEvents 4-16
 - Simulate 4-16
 - Test 4-16
- Starting the software 3-1
- USB computer port 2-1, 2-2
- VBScript demonstration 4-20
- Visual Basic demonstration workspace 4-19
- Visual C++ demonstration workspace 4-18

WARRANTY

SIGNAL RECOVERY, a part of AMETEK Advanced Measurement Technology, Inc, warrants each instrument of its own manufacture to be free of defects in material and workmanship for a period of ONE year from the date of delivery to the original purchaser. Obligations under this Warranty shall be limited to replacing, repairing or giving credit for the purchase, at our option, of any instruments returned, shipment prepaid, to our Service Department for that purpose, provided prior authorization for such return has been given by an authorized representative of AMETEK Advanced Measurement Technology, Inc.

This Warranty shall not apply to any instrument, which our inspection shall disclose to our satisfaction, to have become defective or unusable due to abuse, mishandling, misuse, accident, alteration, negligence, improper installation, or other causes beyond our control. This Warranty shall not apply to any instrument or component not manufactured by AMETEK Advanced Measurement Technology, Inc. When products manufactured by others are included AMETEK Advanced Measurement Technology, Inc equipment, the original manufacturers Warranty is extended to AMETEK Advanced Measurement Technology, Inc customers. AMETEK Advanced Measurement Technology, Inc reserves the right to make changes in design at any time without incurring any obligation to install same on units previously purchased.

THERE ARE NO WARRANTIES WHICH EXTEND BEYOND THE DESCRIPTION ON THE FACE HEREOF. THIS WARRANTY IS IN LIEU OF, AND EXCLUDES ANY AND ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESSED, IMPLIED OR STATUTORY, INCLUDING MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AS WELL AS ANY AND ALL OTHER OBLIGATIONS OR LIABILITIES OF AMETEK ADVANCED MEASUREMENT TECHNOLOGY, INC, INCLUDING, BUT NOT LIMITED TO, SPECIAL OR CONSEQUENTIAL DAMAGES. NO PERSON, FIRM OR CORPORATION IS AUTHORIZED TO ASSUME FOR AMETEK ADVANCED MEASUREMENT TECHNOLOGY, INC ANY ADDITIONAL OBLIGATION OR LIABILITY NOT EXPRESSLY PROVIDED FOR HEREIN EXCEPT IN WRITING DULY EXECUTED BY AN OFFICER OF AMETEK ADVANCED MEASUREMENT TECHNOLOGY, INC.

SHOULD YOUR EQUIPMENT REQUIRE SERVICE

- A. Contact your local **SIGNAL RECOVERY** office, agent, representative or distributor to discuss the problem. In many cases it may be possible to expedite servicing by localizing the problem to a particular unit or cable.
- B. We will need the following information, a copy of which should also be attached to any equipment which is returned for service.
- | | |
|---|---|
| 1. Model number and serial number of instrument | 6. Symptoms (in detail, including control settings) |
| 2. Your name (instrument user) | 7. Your purchase order number for repair charges (does not apply to repairs in warranty) |
| 3. Your address | 8. Shipping instructions (if you wish to authorize shipment by any method other than normal surface transportation) |
| 4. Address to which the instrument should be returned | |
| 5. Your telephone number and extension | |
- C. If you experience any difficulties in obtaining service please contact:

SIGNAL RECOVERY Service

AMETEK Advanced Measurement Technology, Inc
801 South Illinois Avenue
Oak Ridge
TN 37831-2011, USA

Phone: +1 865 482 4411
Fax: +1 865 483 2119
E-mail: info@signalrecovery.com

or

SIGNAL RECOVERY Service

AMETEK Advanced Measurement Technology
Spectrum House, 1 Fishponds Close
Wokingham,
RG41 2TZ, UNITED KINGDOM

Phone: +44 (0)118 936 1210
Fax: +44 (0)118 936 1211
E-mail: info@signalrecovery.com