DASY*Lab*®

Data Acquisition System Laboratory

Version 7.0

32-bit version for
Windows 98
Windows NT V4
Windows 2000
Windows XP Pro

Including

DASY*Lab*-Net
&
Analysis Toolkit Option

Book 3:
Hands On Guide

MANUAL NO **ME 3-7.0-1**

License Agreement

By opening the sealed license envelope, you are agreeing to the following license:

NATIONAL INSTRUMENTS AND DASYTEC USA grants you a non-exclusive license to use this Software on one computer at a time. You do not obtain title to the Software or any copyrights or proprietary rights in the Licensed Software. You may not transfer, sub-license, rent, lease, convey, copy, modify, translate, convert to another programming language, decompile or disassemble the Licensed Software for any purpose, except as expressly provided for in this license. You may not copy the Documentation.

You may copy the Licensed Software for backup purposes only, in support of your use of the Software in accordance with the terms and conditions of this license.

NATIONAL INSTRUMENTS AND DASYTEC USA warrant, for a period of ninety days after your receipt of the product, 1) the disks on which the Software is distributed to be free from defects in materials and workmanship, and 2) that the software will perform substantially in accordance with the Documentation. If the product fails to comply with the warranty set forth above, NATIONAL INSTRUMENTS AND DASYTEC USA's entire liability and your exclusive remedy will be replacement of the disks or NATIONAL INSTRUMENTS AND DASYTEC USA's reasonable effort to make the product meet the warranty set forth above. If NATIONAL INSTRUMENTS AND DASYTEC USA is unable to make the Product conform to the above warranty, then you may return the complete package to NATIONAL INSTRUMENTS AND DASYTEC USA or its dealer, and NATIONAL INSTRUMENTS AND DASYTEC USA, at its option, may refund all or a fair portion of the price you paid for this package.

Although NATIONAL INSTRUMENTS AND DASYTEC USA has tested the Software and reviewed the Documentation, NATIONAL INSTRUMENTS AND DASYTEC USA MAKE NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS SOFTWARE OR DOCUMENTATION, THEIR QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS SOFTWARE AND DOCU-MENTA-TION ARE LICENSED "AS IS," AND YOU, THE LICENSEE, ARE ASSUMING THE ENTIRE RISK AS TO THEIR QUALITY AND PERFORMANCE.

IN NO EVENT WILL NATIONAL INSTRUMENTS AND DASYTEC USA BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE OR DOCUMENTATION, even if advised of the possibility of such damages. In particular, NATIONAL INSTRUMENTS AND DASYTEC USA shall have no liability for any programs or data stored or used with the Software, including the costs of recovering such programs or data.

DASYTEC USA
A National Instruments Company
PO Box 748
Amherst, NH 03031-0748 USA

National Instruments
Postfach 401264,
D - 41182 Mönchengladbach, Germany

The DASY*Lab*® software and information in this document are subject to change without notice.

Year 2000 Compliance Statement

DASYLab 7.0 is Year-2000-compliant. It conforms to the British Standards Institution BSI DISC PD-2000-1 'A Definition of Year 2000 Conformity Requirements‘ document. Specifically, DASYLab will:

- Work correctly for the following the transition from December 31, 1999 to January 1, 2000

- Recognize that 2000 is a leap year;

- Work correctly for all dates in the range January 1, 1980 through December 31, 2037, but not before, and not after

- Read and store dates correctly using its private (.DDF) data format, and store appropriately for other date formats

Risky Applications

Warning:
(1) National Instruments products are not designed with components and testing for a level of reliability suitable for use in or in connection with surgical implants or as critical components in any life support systems whose failure to perform can reasonably be expected to cause significant injury to a human.
(2) In any application, including the above, reliability of operation of the software products can be impaired by adverse factors, including but not limited to fluctuations in electrical power supply, computer hardware malfunctions, computer operating system software fitness, fitness of compilers and development software used to develop an application, installation errors, software and hardware compatibility problems, malfunctions or failures of electronic monitoring or control devices, transient failures of electronic systems (hardware and/or software), unanticipated uses or misuses, or errors on the part of the user or applications designer (adverse factors such as these are hereafter collectively termed "system failures").
Any application where a system failure would create a risk of harm to property or persons (including the risk of bodily injury and death) should not be reliant solely upon one form of electronic system due to the risk of system failure.
To avoid damage, injury, or death, the user or application designer must take reasonably prudent steps to protect against system failures, including but not limited to back-up or shut down mechanisms.
Because each end-user system is customized and differs from NATIONAL INSTRUMENTS' testing platforms and because a user or application designer may use NATIONAL INSTRUMENTS products in combination with other products in a manner not evaluated or contemplated by NATIONAL INSTRUMENTS, the user or application designer is ultimately responsible for verifying and validating the suitability of NATIONAL INSTRUMENTS products whenever NATIONAL INSTRUMENTS products are incorporated in a system or application, including, without limitation, the appropriate design, process and safety level of such system or application.

TABLE OF CONTENTS

CHAPTER 1: DATA ACQUISITION BASICS

Welcome to the world of data acquisition. This book will focus on the software side of a data acquisition project and stay clear of the field of sensors and wiring. Detailed information on wiring is typical provided with your hardware and sensor.

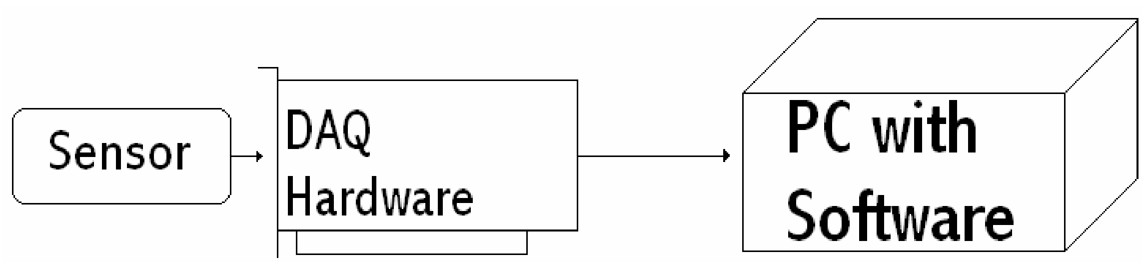
The demonstrations used in this book will use a very basic and widely available data acquisition hardware device, the standard windows compatible sound card. The driver for using the sound card is available on your **DASY*Lab*** CD or from the DASYTEC USA worldwide web page (www.dasylab.net). Some drivers are only available from the data acquisition device vendor.

Before we get started with actually acquiring data into the computer there are a few fundamentals to review. First of all, what is computer based data acquisition? Secondly, how do we best apply computer based data acquisition?

Computer based data acquisition is the application of computer systems to acquire, monitor, and record real world events. In this system, the computer has taken the place of a physical piece of equipment such as an oscilloscope or DMM (digital multi-meter). A computer biased data acquisition system typically consists of a PC, data acquisition hardware and an acquisition and analysis software package. Many systems include sensors and signal conditioning to prepare the sensor signal for the data acquisition hardware.

When you create a software application on the computer system for data acquisition you are creating a ***virtual instrument***. A virtual instrument performs the function of a physical instrument such as a DMM or oscilloscope. The major benefit of a virtual instrument is the flexibility in creating and modifying the instrument to fit your needs and desires.

The computer-based instrument acquires data in a rather straightforward system. A sensor, for example a thermocouple, picks up the "event" or real world property, the temperature. This analog information is received by the data acquisition hardware and is digitized; this is known as "Analog to Digital conversion". The digitized value is sent to the computer and received by the data acquisition software. The software then interprets the value, conditions, scales, displays and stores the data.

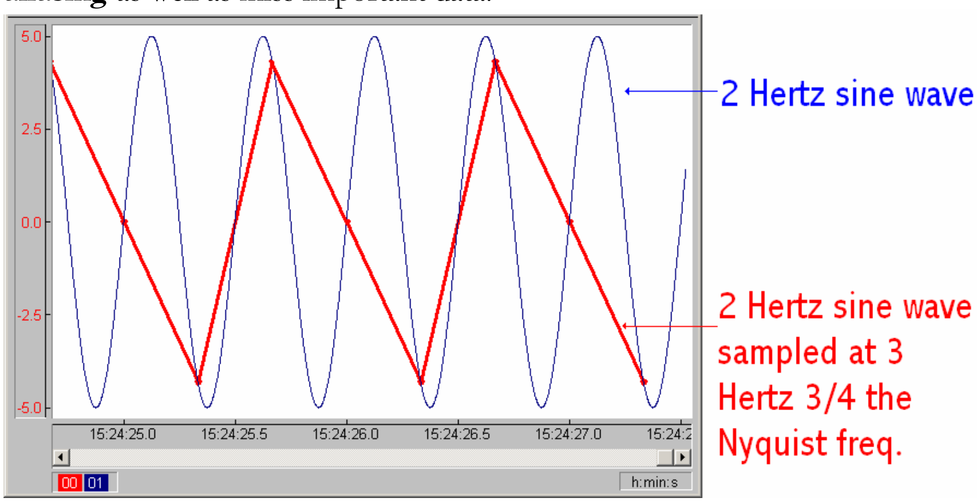


In **DASY*Lab*,** the job of communicating with the hardware is taken care of by a driver. **DASY*Lab*** also takes care of the sample blocking and time stamp, features we will discuss at a later time. In essence, **DASY*Lab*** creates the basic virtual instrument, allowing the users to design what **DASY*Lab*** will do with the data it acquires from the hardware.

Hardware and sensors are always advancing and changing; therefore, selecting the appropriate hardware or sensor for your data acquisition application is a complicated process and is best left to the sales team at your local distributor. Before contacting your hardware distributor it is best to know a little about the data you wish to acquire. The following chart should help you and your sales representative select the best hardware for your application.

Computer Type	Desktop, Laptop, Compact-Backplane, PXI
Available PC slot	ISA, PCI, PXI, PCMCIA, PC104, IEEE, USB, FireWire, Serial, EtherNet
Length of event to be captured	
Maximum amplitude of signal	
Minimum amplitude of signal	
Number of signals to acquire	
Type of signal	
Type of sensors	

Before we get into the creation of virtual instruments and the actual acquiring of data we should cover one of the major rules surrounding data acquisition, the ***Nyquist theorem***. The Nyquist theorem sets the rules for sampling waveforms or capturing transient events. Nyquist states that the sampling rate must be at least two times the maximum frequency component of a given wave form. For example to capture a 2-hertz waveform we would have to sample at a minimum of 4-hertz. If the Nyquist theorem is broken we may introduce ***aliasing*** as well as miss important data.



Aliasing is the appearance of waveforms that may not exist with in the signal. These are created by under-sampling waveforms higher than one half the ***sampling rate***. There may be signals higher than the sampling rate, however these are not of interest to us. To remove these extraneous signals we employ ***filtering*** to block out signals with a frequency above the cut off value. We will discuss the application of filters in a later chapter.

CHAPTER 2: DASYLAB BASICS

The **DASY*Lab*** software package is available in several different levels to suit the varying needs of different users. The major levels of **DASY*Lab*** are Lite, Basic, Full and Pro. Each level of the software provides different options and abilities; it is very important that you have the appropriate version of the software to suit the needs of your application.

DASY*Lab* Lite provides the user with the ability to easily create a ***data logger*** or simple oscilloscope application. Lite has a limited number of channels that can be used, which limits the user to smaller ***worksheets***. This is the lowest level of design available and logically the cheapest. Lite allows the user to create a program that will display the data and save it to the PC hard drive.

The next level in ability and complexity is **DASY*Lab*** Basic. Basic may be used to create a "smart data logger", which has the ability to reduce the amount of data and perform straightforward calculations. **DASY*Lab*** basic also has the ability to control alarms via analog and digital output. Basic allows the user to create a program that does most data acquisition, some analysis, display and saving. It is suited to environments where the operator is present and can react to events.

The second highest level is **DASY*Lab*** Full. This package has the ability to perform ***FFTs*** as well as automated tasks. These tasks include setting alarms; generating file names; post trigger-based data acquisition; and other automation-based tasks. The Full package is suited to most applications, allowing unattended use, automatic operation, as well as frequency analysis functions.

DASY*Lab* Pro contains the same features as Full with the addition of more signal analysis, a Sequence Generator, high-end or complex frequency analysis, additional filtering and other tools. Pro is most useful when creating "stand alone" test stands or automated applications.

If you feel that your level of **DASY*Lab*** may be inadequate for your needs please feel free to contact your distributor, they will be glad to assist you in reassessing your software needs.

CHAPTER 3: SETUP AND INSTALLATION

We are now ready to install our hardware and software. The hardware should be installed first, following the instructions provided by your hardware provider. The hardware should be tested with any software provided by the manufacturer to ensure that it has been installed and is functioning correctly.

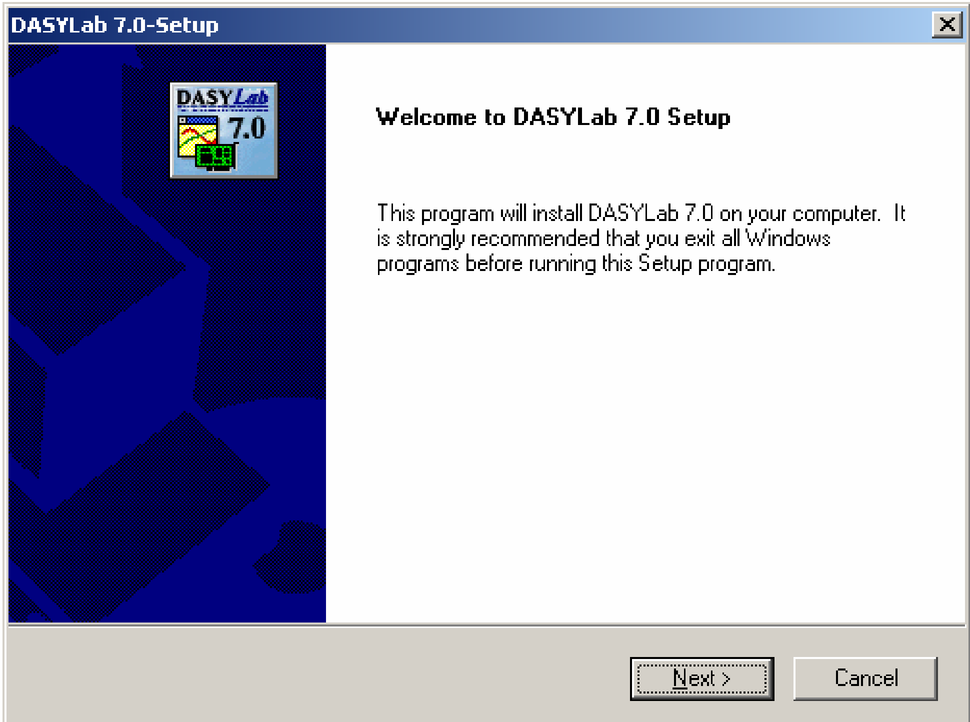
Also the hardware should be checked to be sure that its ***IRQ*** is not being "shared" by any device. This can be accomplished through the "*Hardware Manager*" in Microsoft® Windows 95,98, 2000 and XP. A problem may occur if the IRQ is shared with other devices. The hardware vendor should be contacted on the best method of changing the devices IRQ address.

If all is well with the hardware we are ready to install the software and get to work. Remove the **DASY*Lab*** CD from its cover and insert it into the CD Rom drive on your computer. In the first startup screen click Continue to get into the next window.

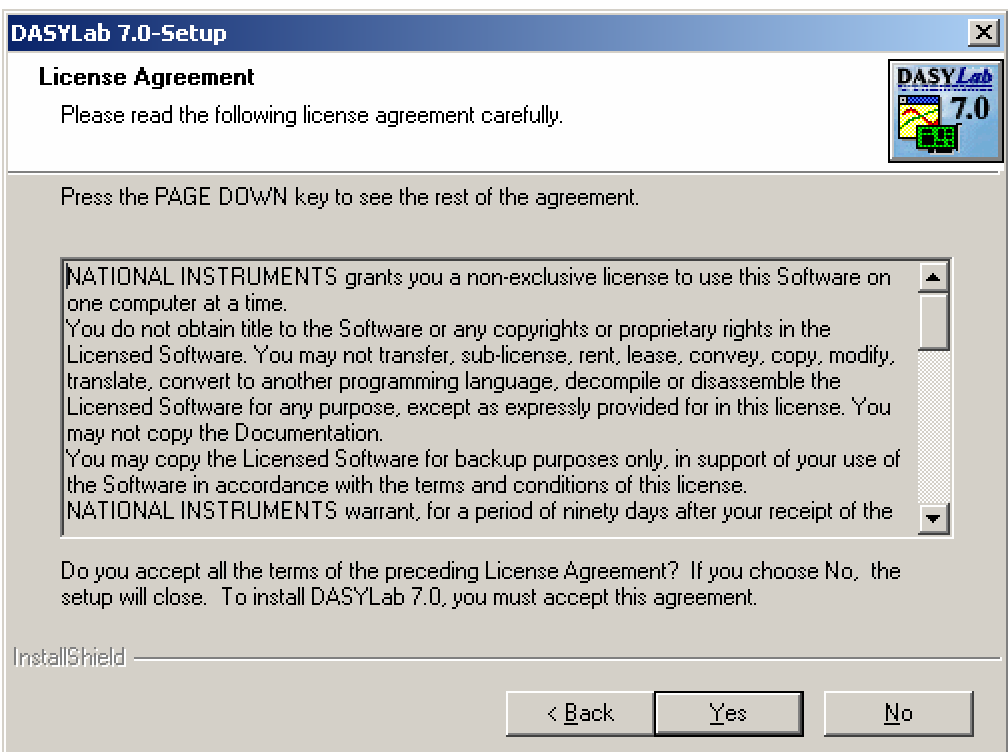
Click on **Continue** to move on.

Click on **Installation** to move on.

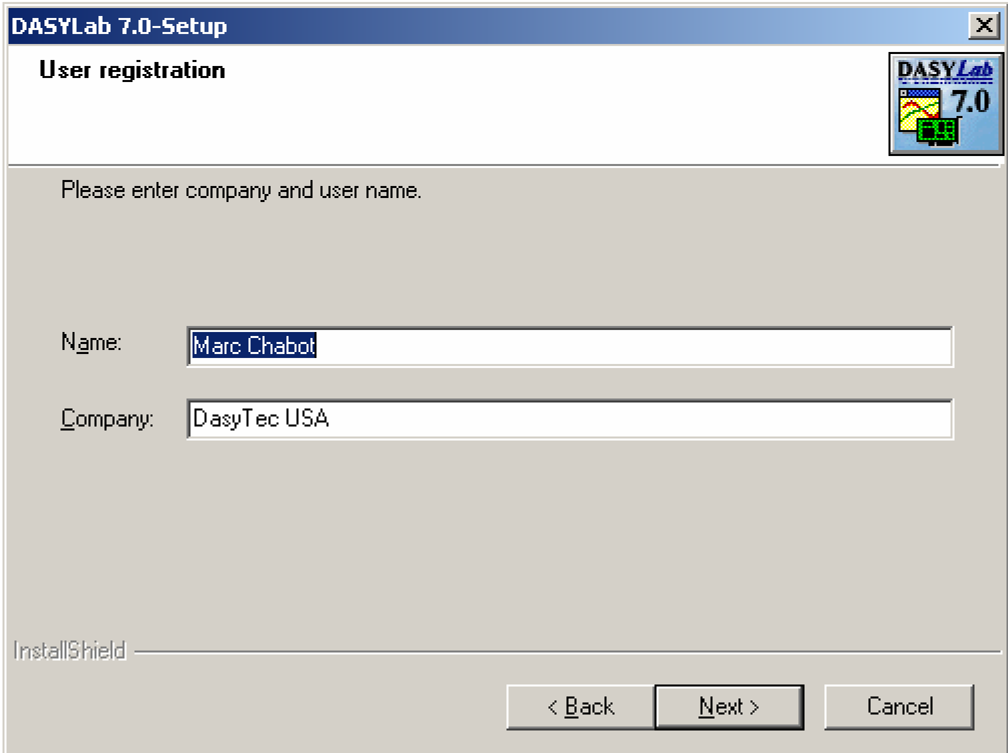
Click on **DASY*Lab*** Full Version to install the software or **DASY*Lab*** Demo Version to try the software. The Demo version will not be capable of actually acquiring data. After clicking on Full Version a new window should launch.



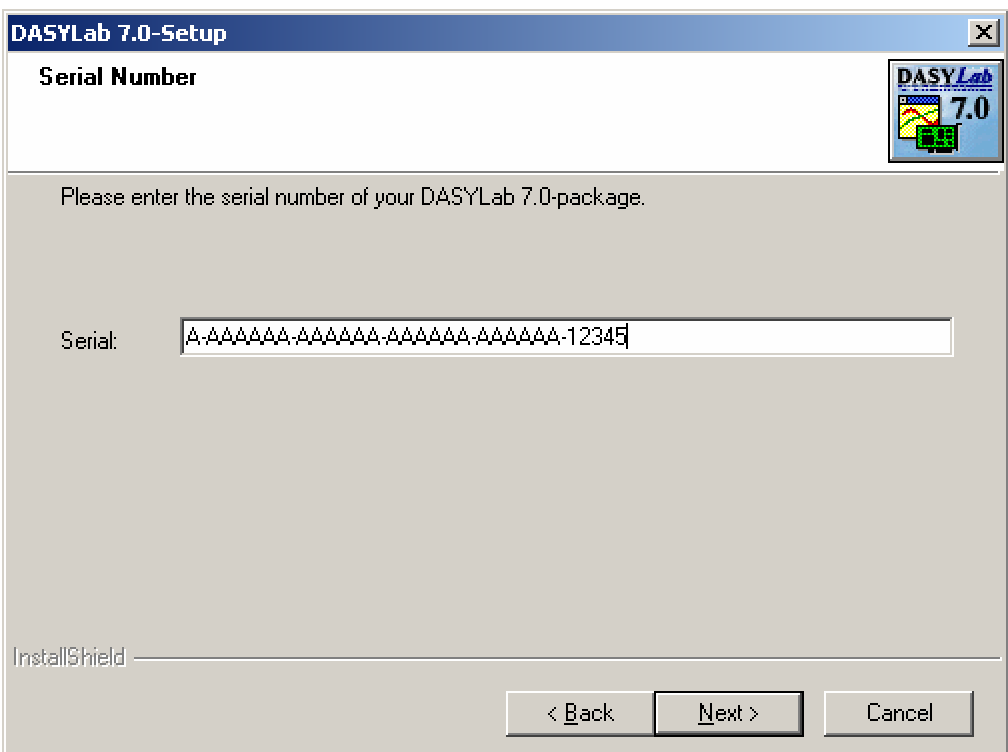
Click "Next"



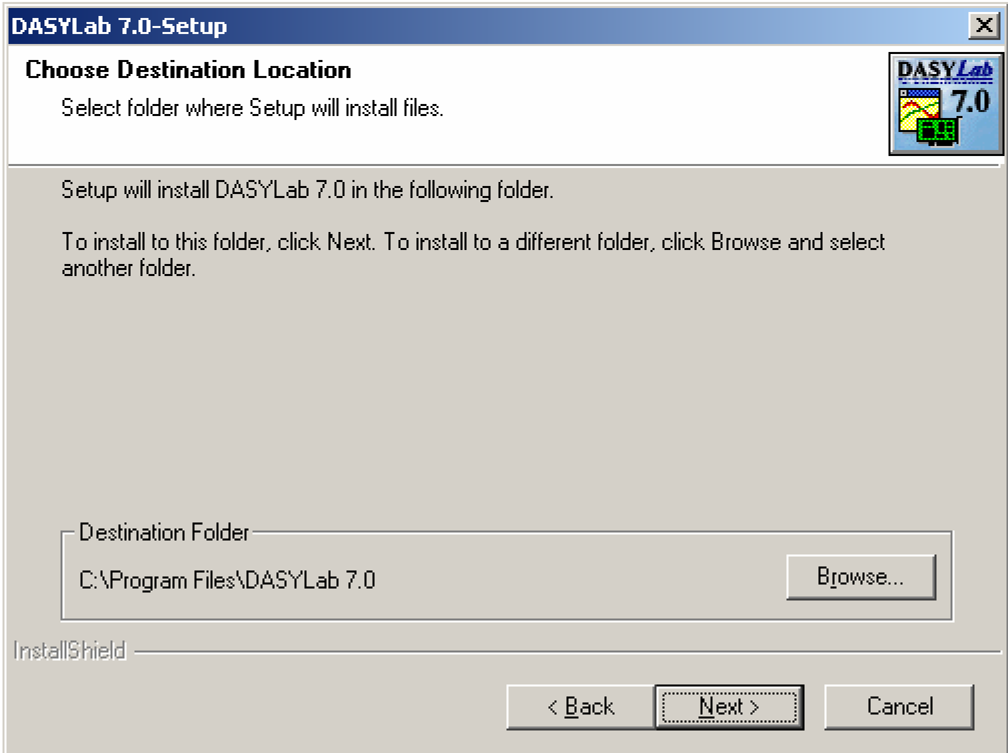
Please read the License Agreement and, if you agree, click "Yes".



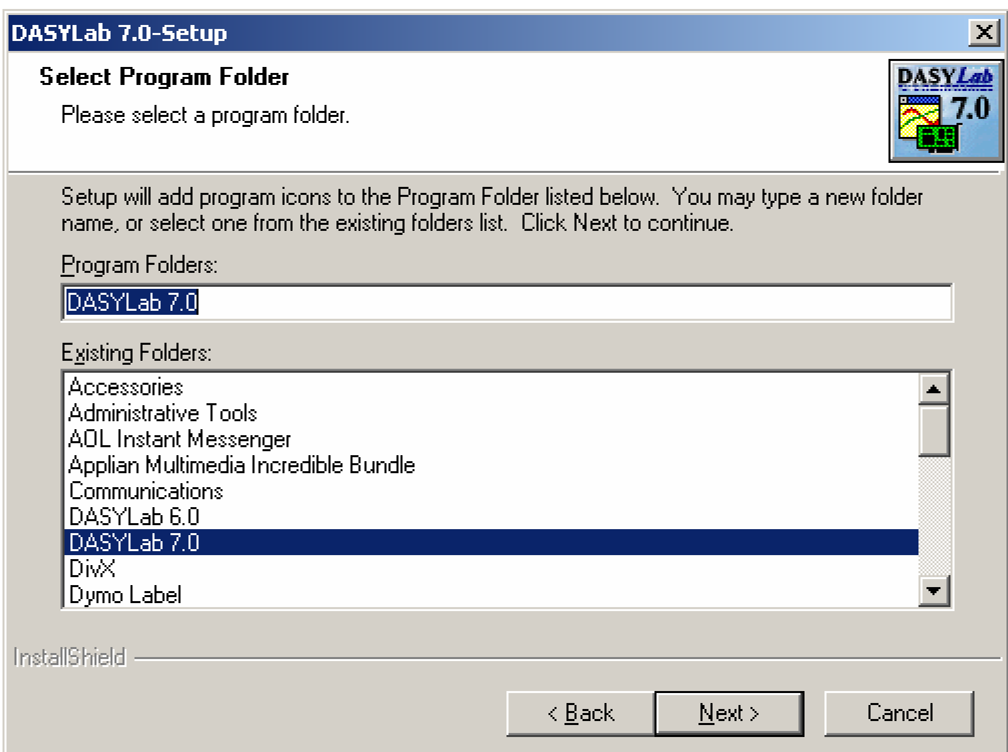
Enter your name and the company name, and then click “Next”



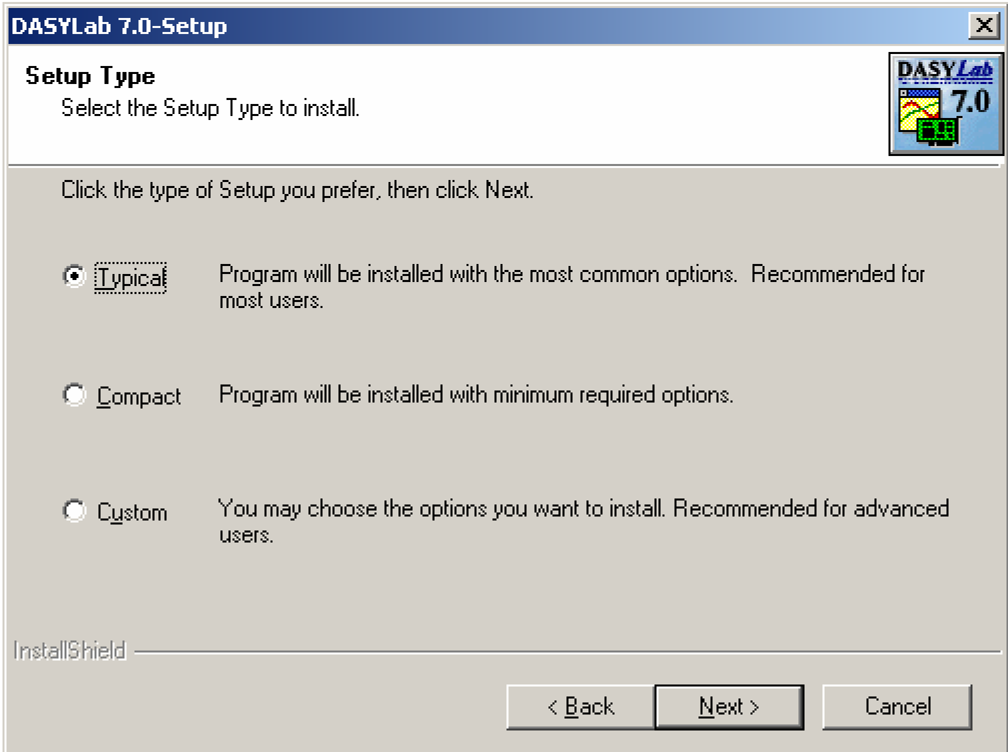
Enter the serial number provided by your distributor and then click “Next”



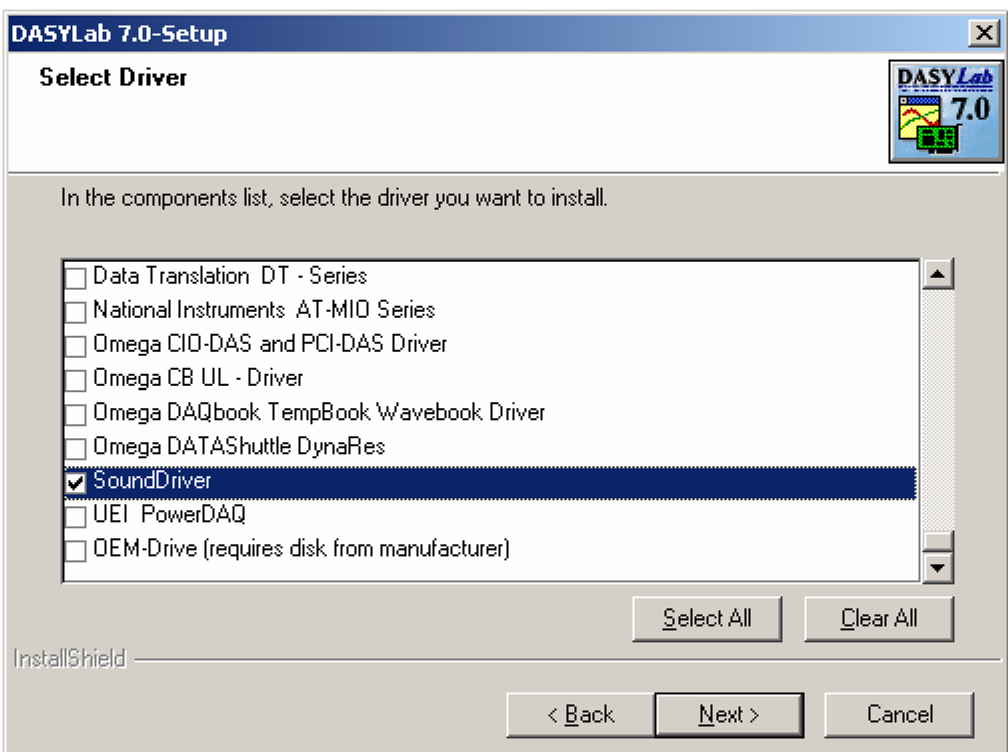
Select the location you wish to have the **DASY*Lab*** software installed and then click "Next"



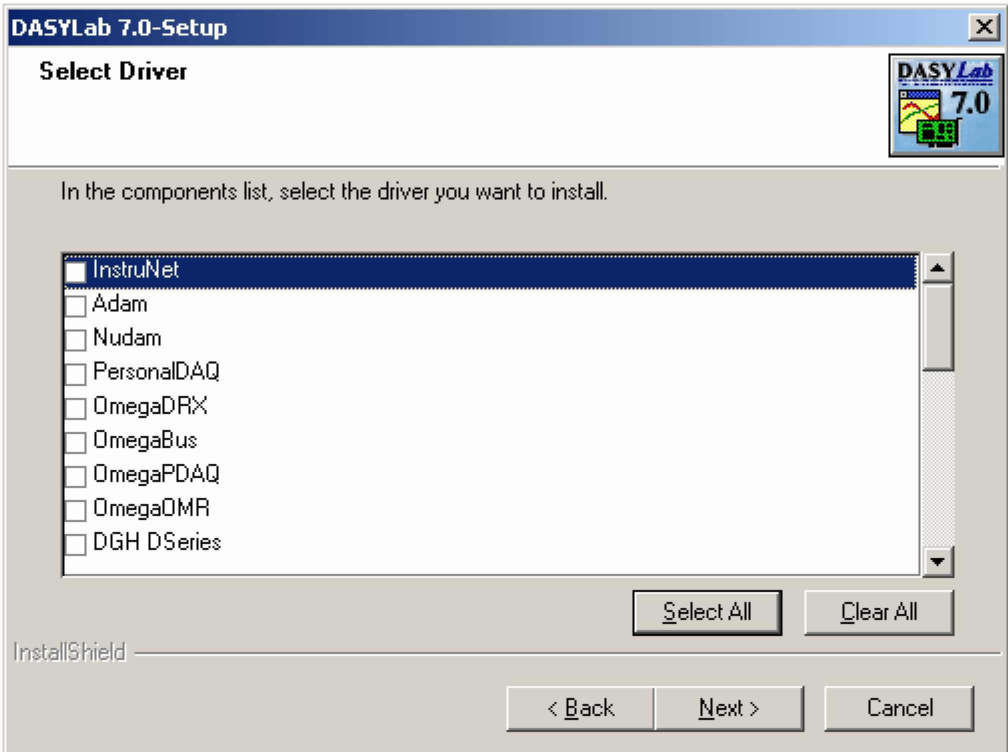
Select where you want the Icons placed for **DASY*Lab***, then click "Next"



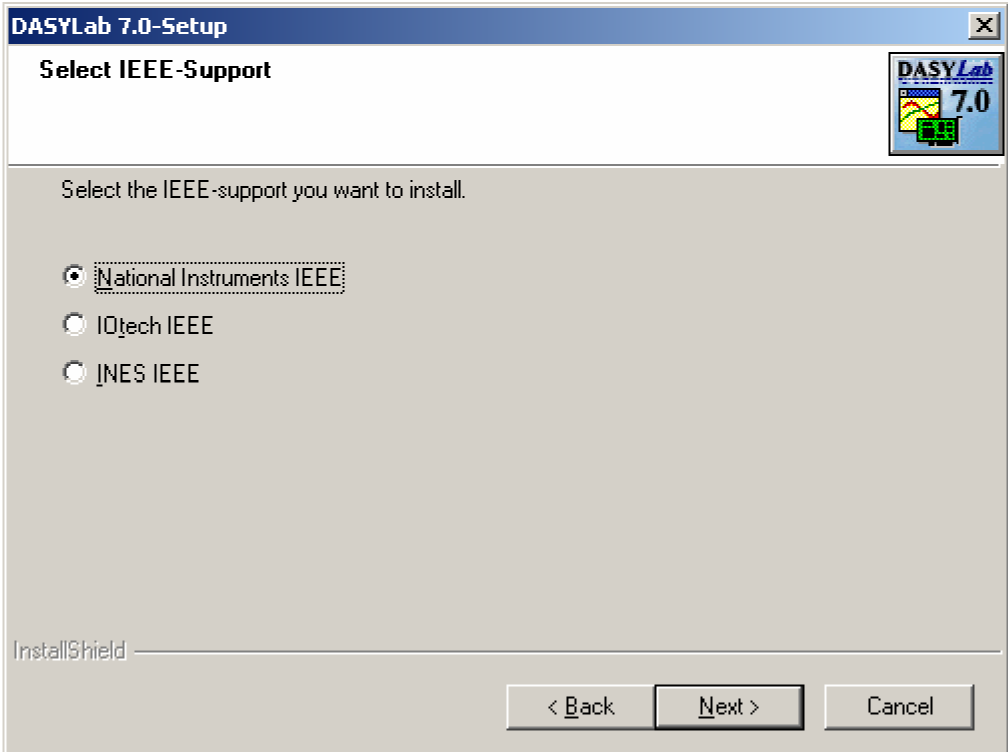
Here we can select how much of the optional software is installed. We are going to use the recommended installation. Check that "Typical" is selected and then click "Next"



Here is where you select the appropriate driver for your hardware. Your hardware vendor should have recommendations on which **DASY*Lab*** driver would best suit your current hardware. Some hardware devices do **not** require a driver to be selected in this menu, these devices include National Instruments®, instruNet®, and some IOTech® hardware. If you do not see a driver that is appropriate for your supported device it may be supported by **DASY*Lab*** and not require additional driver in this menu. For our demonstrations you should find and select the "SoundDriver". Once you have made your selection click "Next".



Here is a list of additional hardware devices supported by **DASY*Lab***. These devices, once installed, will appear in a separate dropdown menu next to the "Modules" dropdown. You should also find all the important configuration setting under this new menu.



Select the brand of IEEE you have. If you don't have any IEEE hardware installed then just leave it as default and click "Next". Many IEEE boards have a National Instruments emulation mode and may work with the National Instruments IEEE driver.

The last screen provides an overview of your installation and the last chance to click "Back" and make changes to the installation before the software is installed. If everything looks correct then click "Next" and the installation should commence.

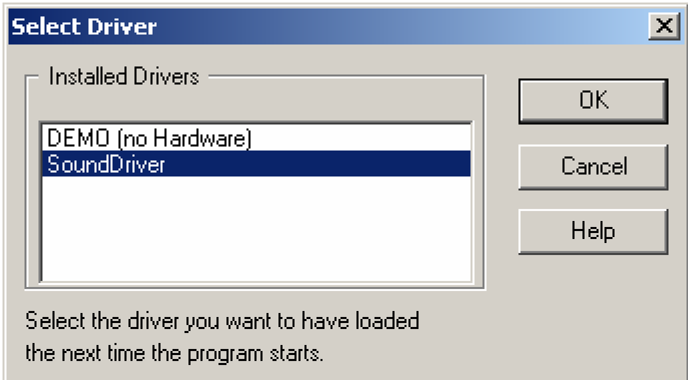
When the installation has completed please take a moment to register your software. Registration allows you to receive technical assistance and product upgrades.

When everything is completed you should restart your computer.

Once your computer has been started we can configure **DASY*Lab*** to communicate with your device.

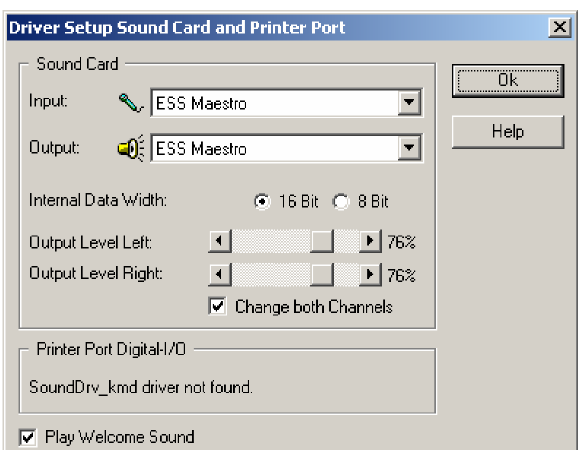
We must now set up **DASY*Lab*** to communicate with our hardware. In this case we are assuming that we are using internal hardware NOT manufactured by National Instruments, IOTech or GW Instruments. Each of these hardware setup ups may be found in APPENDIX A. As mentioned before we will be creating these applications using the sound card driver.

To setup our hardware, first start up **DASY*Lab***, click on "Experiment " and select "Select Driver…". In this menu select the appropriate driver for your installed hardware. If you have hardware from National Instruments, IOTech, DAP, Personal DAQ or GW Instruments the DEMO driver should be selected. In our case we select the "SoundDriver". Then click OK. Exit and restart **DASY*Lab***.



Now we must configure the selected driver to communicate with our hardware. This is where we enter the information necessary for **DASY*Lab*** to communicate with the hardware. The parameters entered here are dependent on the installed hardware and the selected driver. If you have any questions about your specific hardware you should contact your hardware vendor.

To access the hardware setup menu click on "Experiment" and select "Hardware setup". A menu will appear and have options dependent on your hardware. Our menu will look as follows:



There are no appropriate changes to be made in this menu, so we click OK.

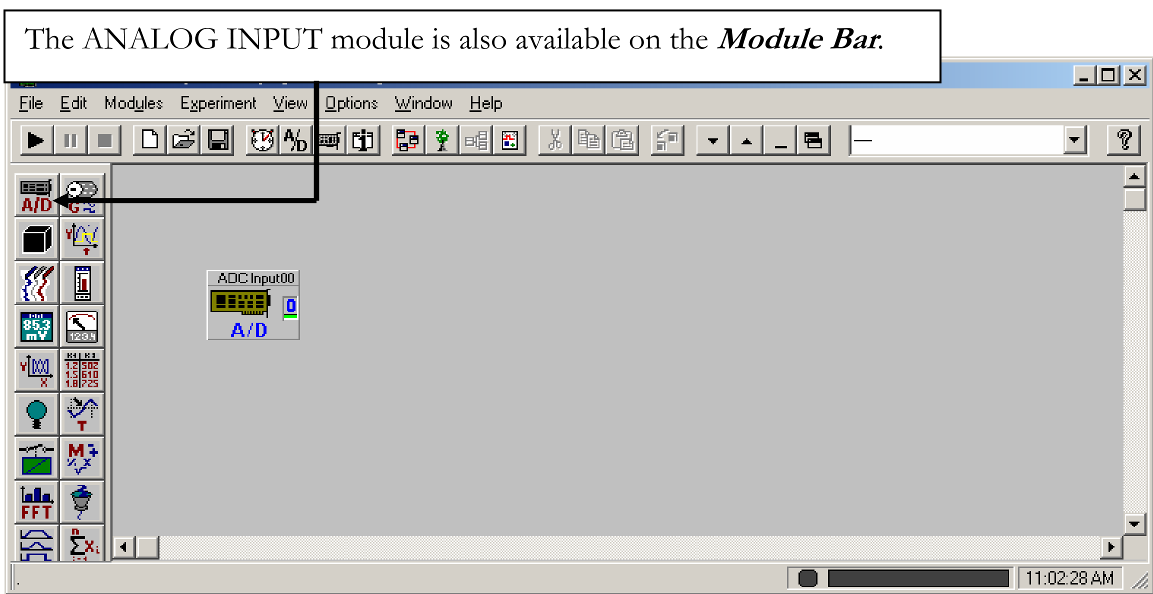
We have now completed our **DASY*Lab*** hardware configuration and are ready to create our first **DASY*Lab*** worksheet.

CHAPTER 4: BASIC DASYLAB APPLICATIONS

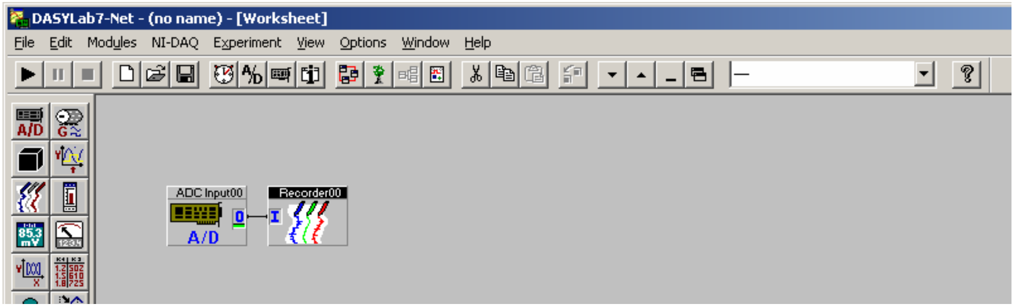
BASIC DATA LOGGER

We will now start **DASY*Lab*** and create our first application, a simple data logger. This data logger will read data from one channel, display the data in a strip chart type display, and then store the data on the hard drive. We will accomplish these tasks using three **DASY*Lab*** modules.

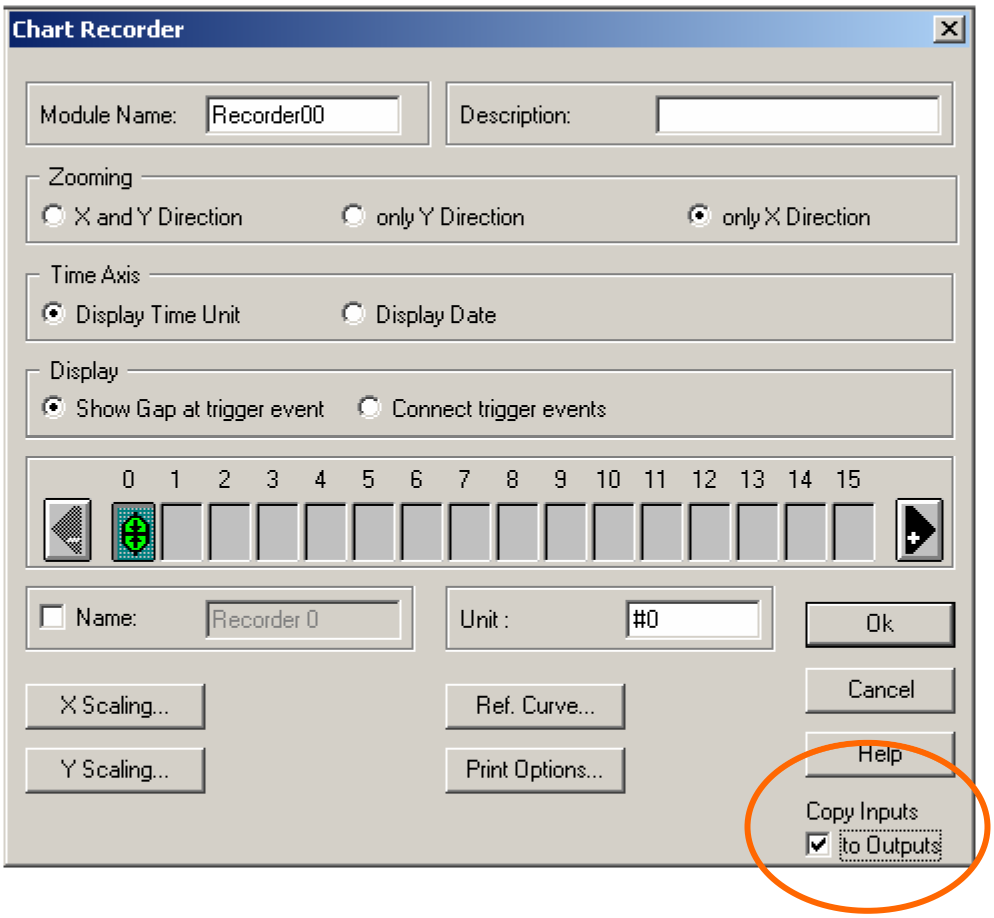
Start up **DASY*Lab***. We will first bring the data into **DASY*Lab***; to do this we use an ANALOG INPUT module. The ANALOG INPUT module is located under the Modules menu, Input/Output, Analog Input. Once selected your pointer should change to the "place" pointer, click anywhere on the worksheet to place the Analog Input Icon. After placing the icon our worksheet should look like this:



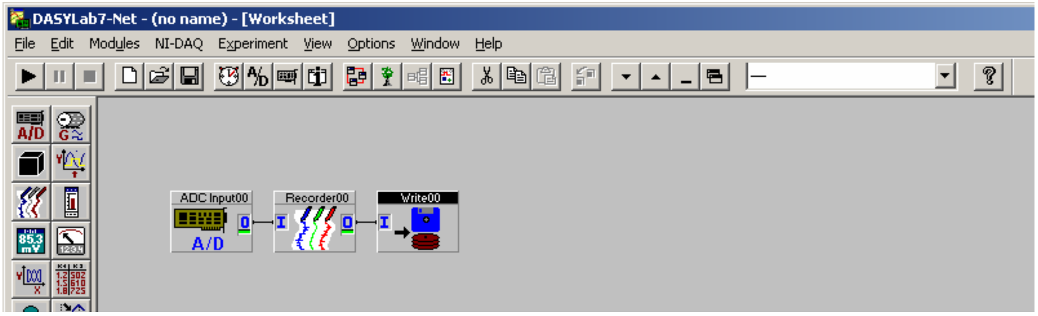
Now we can connect the analog input to our display. We have chosen to display the data on a "strip chart" type display. In **DASY*Lab*** we use a Chart Recorder, found under Modules, Display, Chart Recorder. Once again the icon should change from a pointer to the place icon. Click on the worksheet close to the Analog input we placed earlier. The two modules must now be connected together. To accomplish this, we can click and drag the chart recorder module close to the analog input module so that the 0 on the analog input touches the "I" on the chart recorder icon, then release. If done properly a "wire" should connect the two modules.



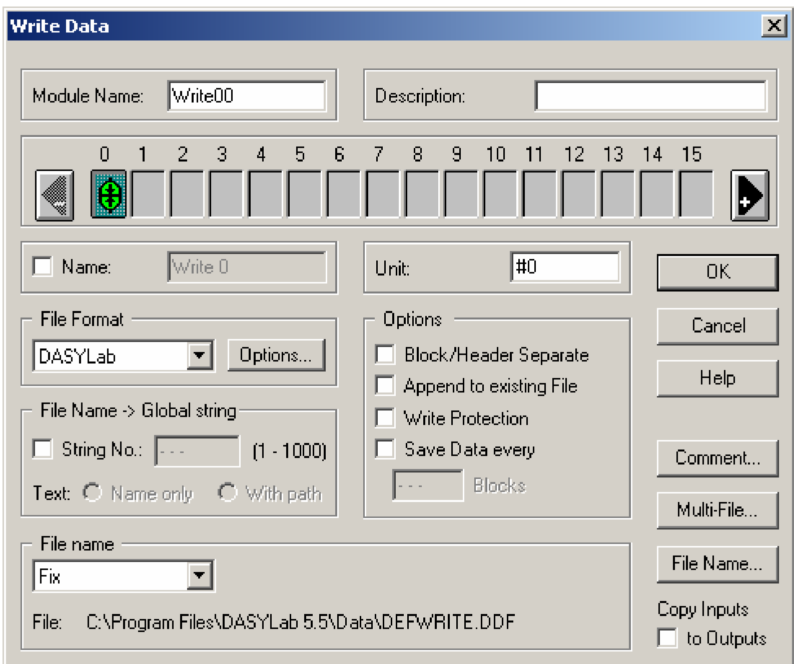
A feature of many modules is the ability to allow data to “pass through”. This allows us to connect many modules in a chain and keep our worksheet neat and uncluttered. To enable this pass through double click on the chart recorder module and click the “copy inputs to outputs” in the lower right hand corner. Then click OK.



We have almost completed our worksheet and only a few steps remain. We should now save the data to the hard drive. The module required to save data to the hard drive is found under Modules, Files, Write Data. Place this module on the worksheet next to the chart recorder. Now we connect the chart recorder to the write module. We can use the method discussed earlier by “bumping” them together or we can click on the output of the chart recorder, which should change the pointer into the wiring control, then click the input of the write module. When completed the worksheet should look as follows:

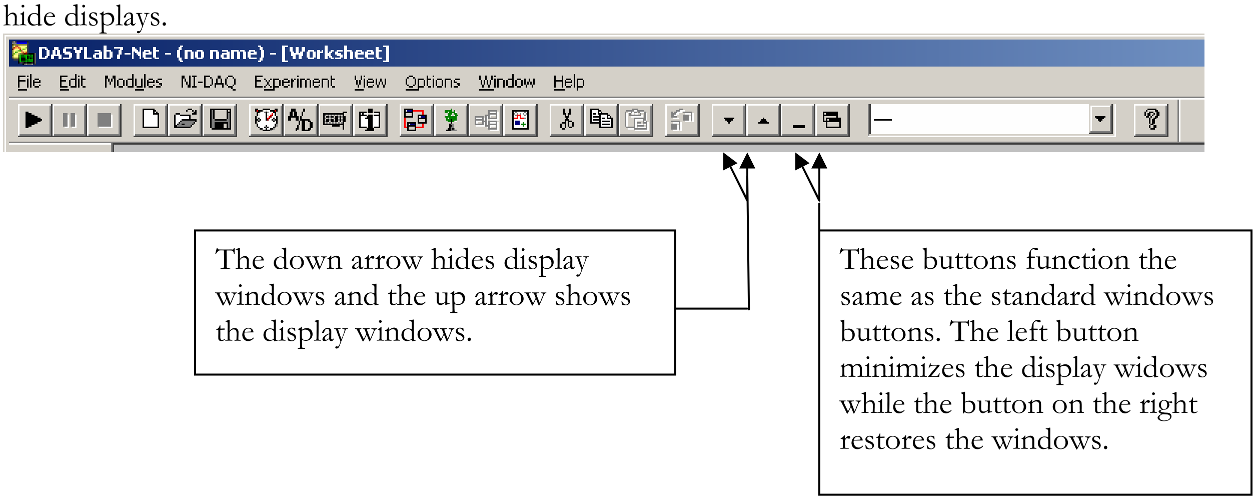


By double clicking on the write module we can access the properties. Here we can change the file type, name and path. The property menu looks as follows:



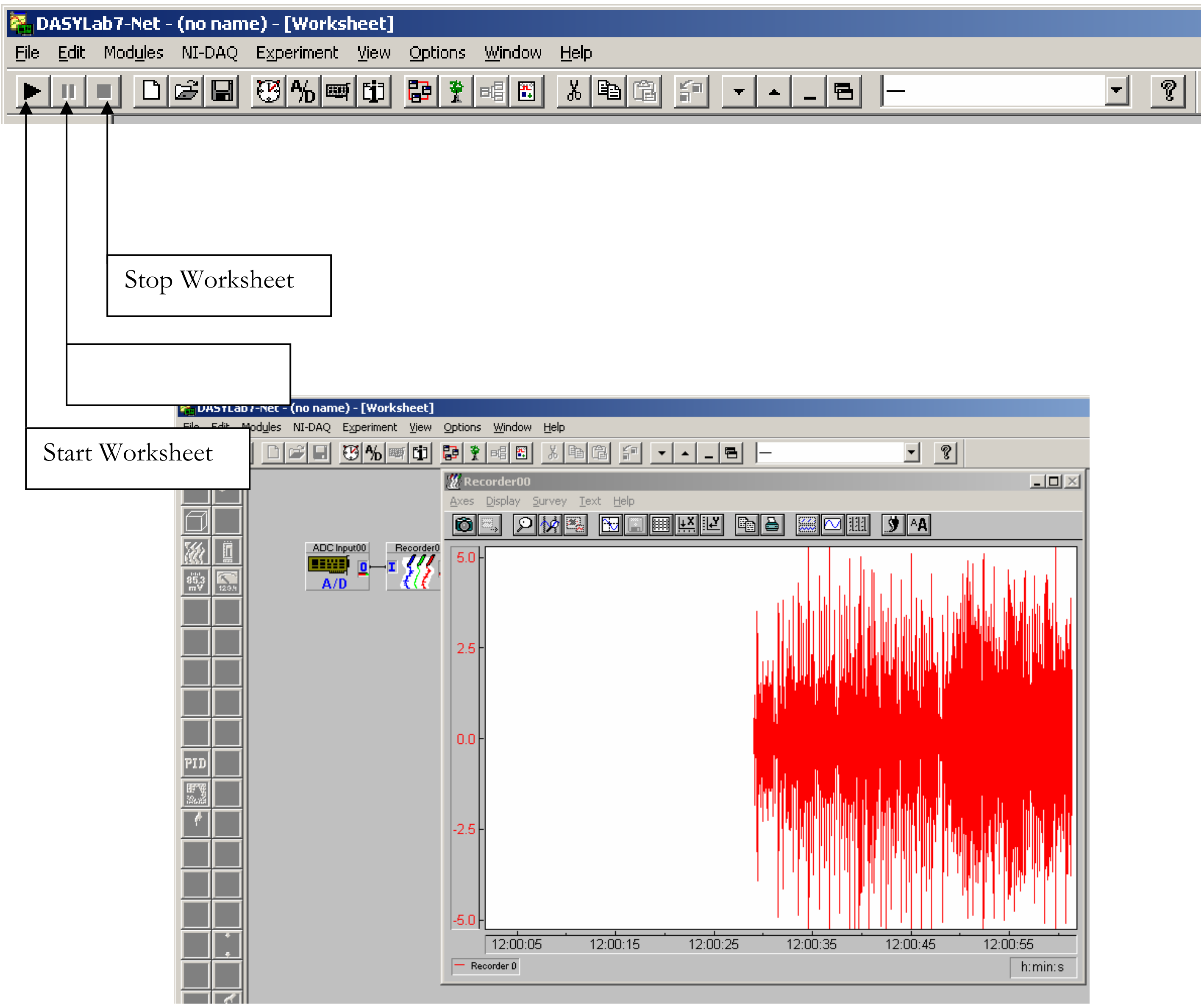
Click “File Name…” and enter a file name for the stored data. Click OK.

Now let’s bring our displays up to visibility. There are four buttons designed to show and hide displays.



To bring the chart recorder display up and make it visible first click the "Show displays" button (Up arrow), then click the restore button.

We are now ready to start our worksheet. Click the start Worksheet button.



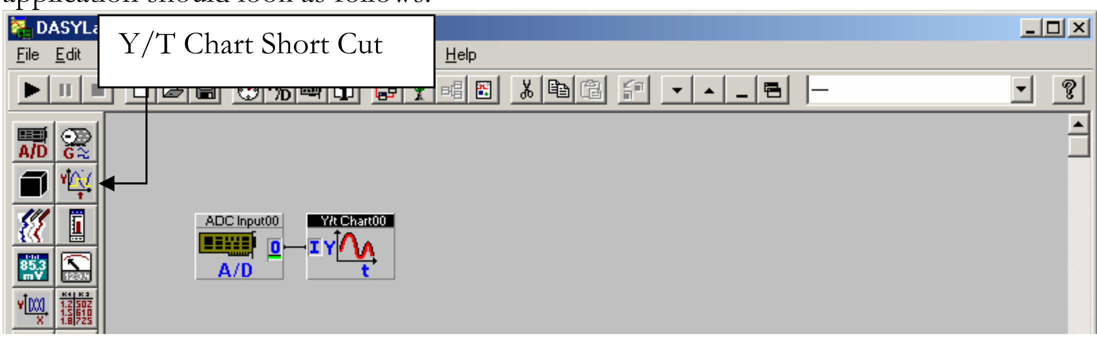
The running worksheet should look like this; I played some music to generate this signal. The red trace shows the result of the data on channel 0 of the sound card, typically this is the microphone. While the worksheet is running data is being displayed, it is also being saved to the hard drive in the file we selected.

As we move on we will cover how to change the sampling rate and adjust the block size for responsiveness, for now let's save this worksheet so that we can use it later. Click on File, then Save As… Select a location and a name for our file.

BASIC OSCILLOSCOPE

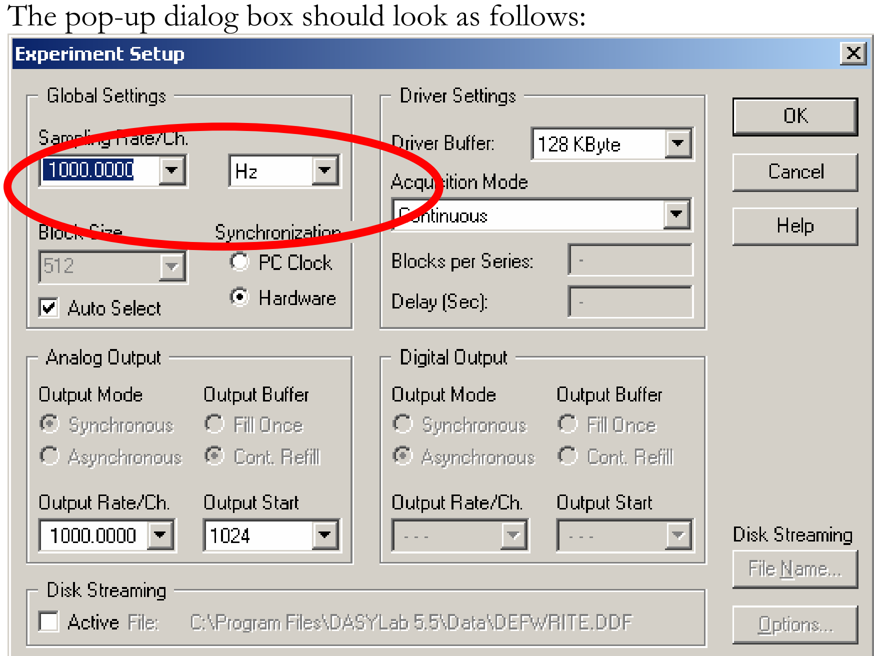
The next common application is the oscilloscope. The main difference between this application and the previous one is the optional **Write Module** and the replacement of the **Chart Recorder** with an **Y/t Chart Module**. The advantage of the **Y/t Chart Module** is that it can display data sampled at a higher rate then the **Chart Recorder Module**. We will also take the opportunity to change the sampling rate and block size.

We start by creating our application. We connect an **Analog Input Module** to a **Y/t Chart Module**. The **Y/t Chart** is found under Modules, Display, Y/t Chart. The completed application should look as follows:



You may now start the worksheet. You should see the data appear as it would on an oscilloscope screen. By default **DASY*Lab*** samples at a rate of 1000 samples a second or 1000 Hertz, which is equal to 1 Kilohertz. As we discussed before this is only adequate when we are sampling a waveform with a frequency of 250 Hertz. In order to sample a faster waveform we must increase **DASY*Lab's*** sampling rate.

In order to change the sampling rate we must access the "Experiment Setup" menu. This dialog box is located under the Experiment drop down menu, Experiment Setup… option. The pop-up dialog box should look as follows:



We can enter any desired sampling rate into the "Sampling Rate/Ch." Box, or select a common sampling rate from its dropdown box. This rate will be the frequency at which each data channel is sampled and data is passed into **DASY*Lab***.

If we change the sampling rate from 1000Hz to 5000Hz and chick OK, we should be able to run the application. With the application running we should be able to see that the trace is much smoother and more defined then at the 1000Hz rate. This is because we are now receiving 5 times the number of samples in the same amount of time. This will also allow us to sample a waveform of 1.25 kHz (1250 Hz).

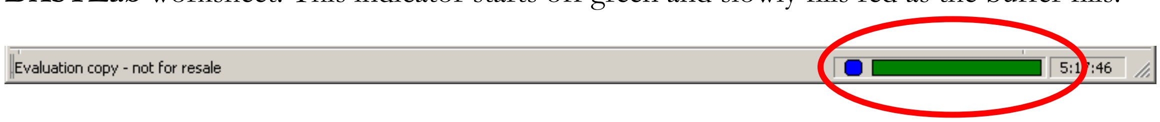
As you may have noticed there is a slight amount of "lag" or delay in your signal. In our case with a microphone connected to the microphone jack, when we tap on the microphone the resultant waveform doesn't appear on the Y/t Chart immediately; it takes a moment. This is due to the "blocking" of the data.

When data is sent to **DASY*Lab*** the data is placed, point-by-point, into "blocks" or groups. These groups are then "time stamped" for accuracy and documentation then sent from module to module. **DASY*Lab*** waits for an entire block to fill before sending it out the **Analog Input** module and to the **Y/t Chart**.

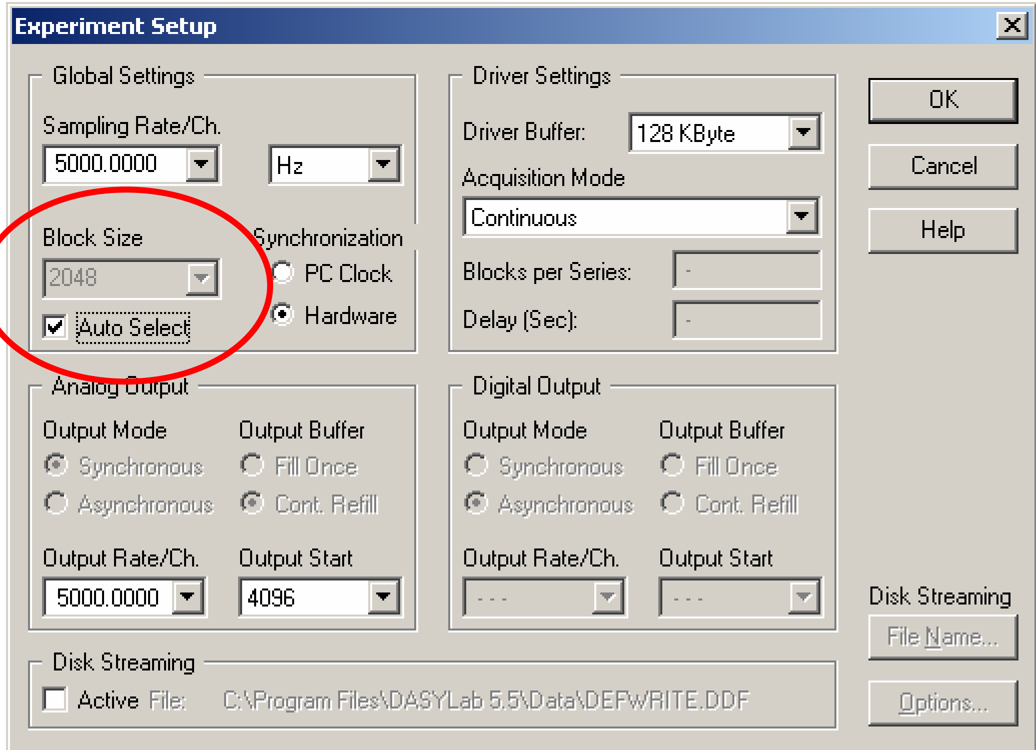
To better explain the sampling rate / block size relationship I will draw an analogy. Imagine a marble packaging machine. This machine has one unit that spits out marbles, these marbles land in boxes of a predetermined marble count; these boxes are on a conveyor belt which may move the boxes to painters or counters exc.

The rate that the machine spits out the marbles is the sampling rate, the size of the box is the block size and the speed of the conveyor belt is the load on the computers processor. As the boxes get bigger it takes longer for the machine to fill each box, however the conveyor belt doesn't have to move as fast. As we decrease the box size the conveyor belt must speed up. If the conveyor belt can't move the boxes as fast as the machine is spitting them out then we have an ***overflow***.

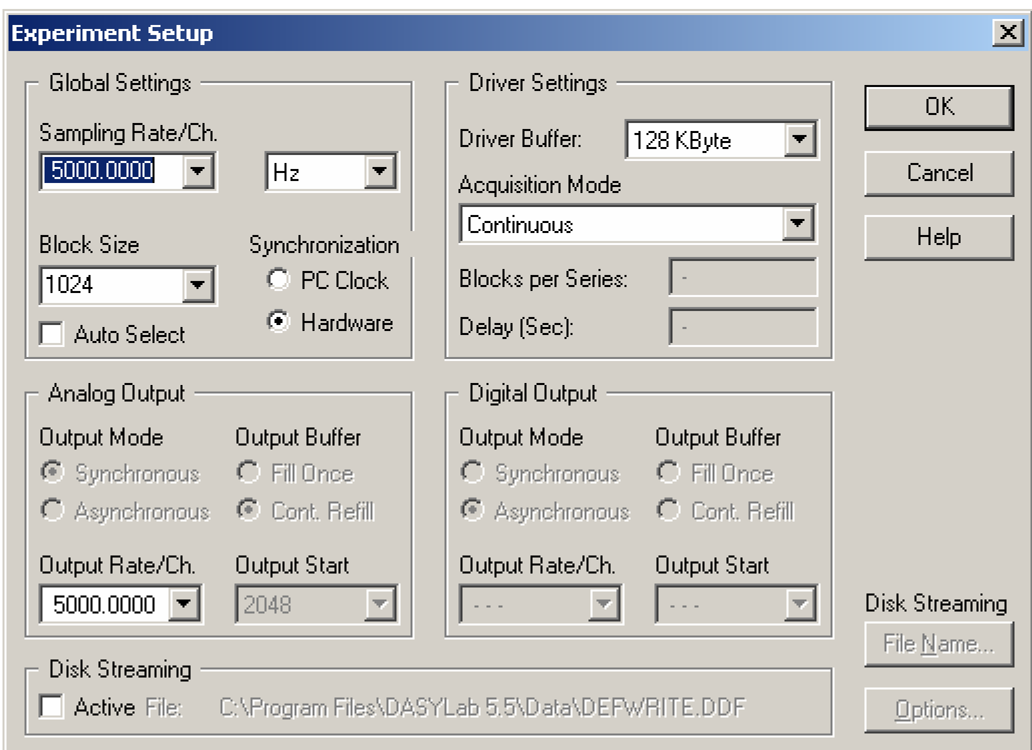
In the real world data is acquired from the data acquisition device and placed into a buffer inside **DASY*Lab***, this buffer then empties into the blocks. This internal buffer is where the overflow actually occurs. An indicator of this buffers status appears at the bottom of the **DASY*Lab*** worksheet. This indicator starts off green and slowly fills red as the buffer fills.



With all this in mind we can now start to adjust the sampling rate and decrease the lag in our oscilloscope worksheet. To modify the block size open the "Experiment Setup" menu, the same way as to set the sampling rate. By default the block size is set to auto select, uncheck the auto select box.

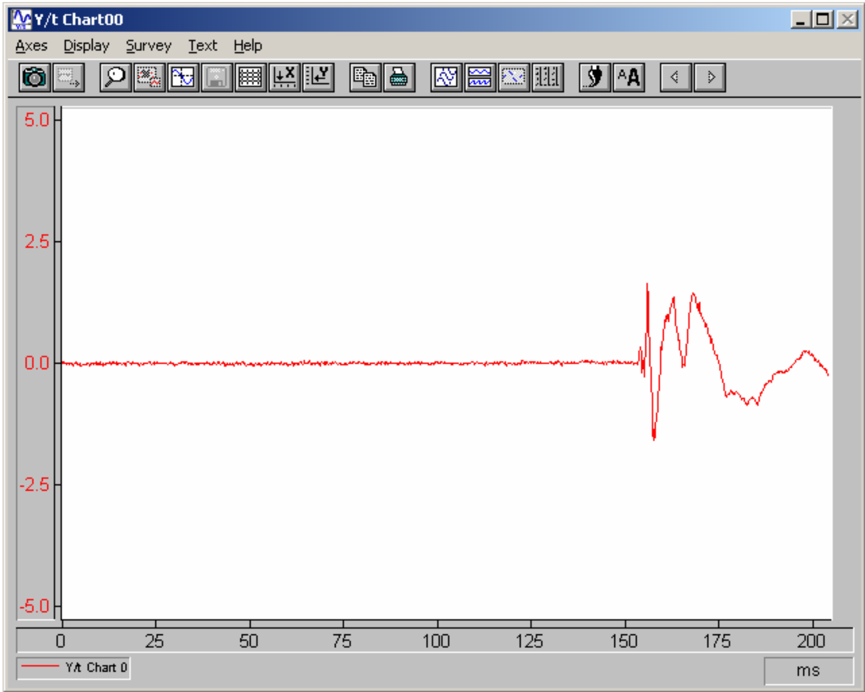


Let's set the block size to half the recommended rate for a 5000Hz-sampling rate, 1024.

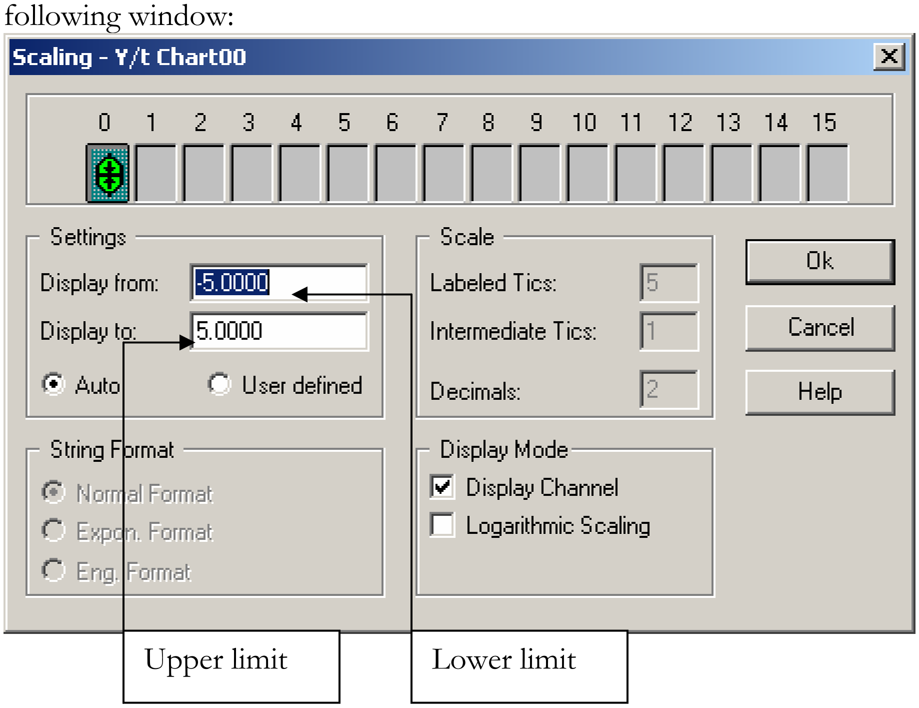


When you run **DASY*Lab*** we notice that the lag had decreased. Another change has occurred; the display on our Y/t Chart has decreased from about 500ms to about 200ms.

This happens to be the amount of samples stored in one block of data. Your display should look as follows:

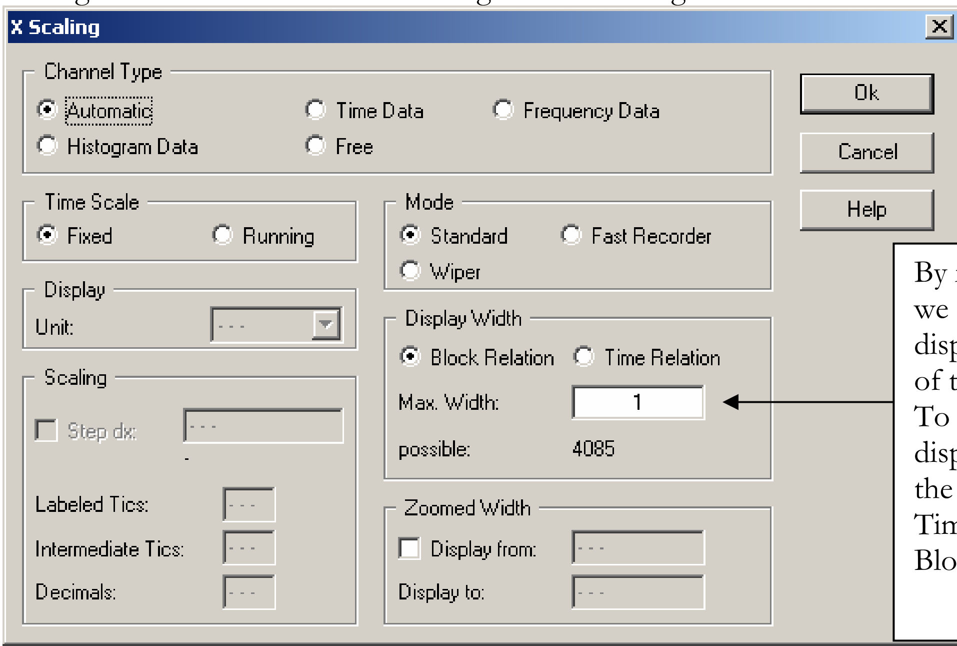


The scale on the **Y/t chart** can be changed by double clicking on either the scale on the left (Y axis) or the scale on the bottom (X axis). When we double click on the Y axis we get the following window:



Changing these values and then clicking on OK will change the scale of the display. However changing these settings will not change the scale of your data. To change the data we use a **Scaling Module**.

If you wish to change the amount of data displayed in the chart we will need to adjust the X Scaling. Double click on the X axis to get the following screen.



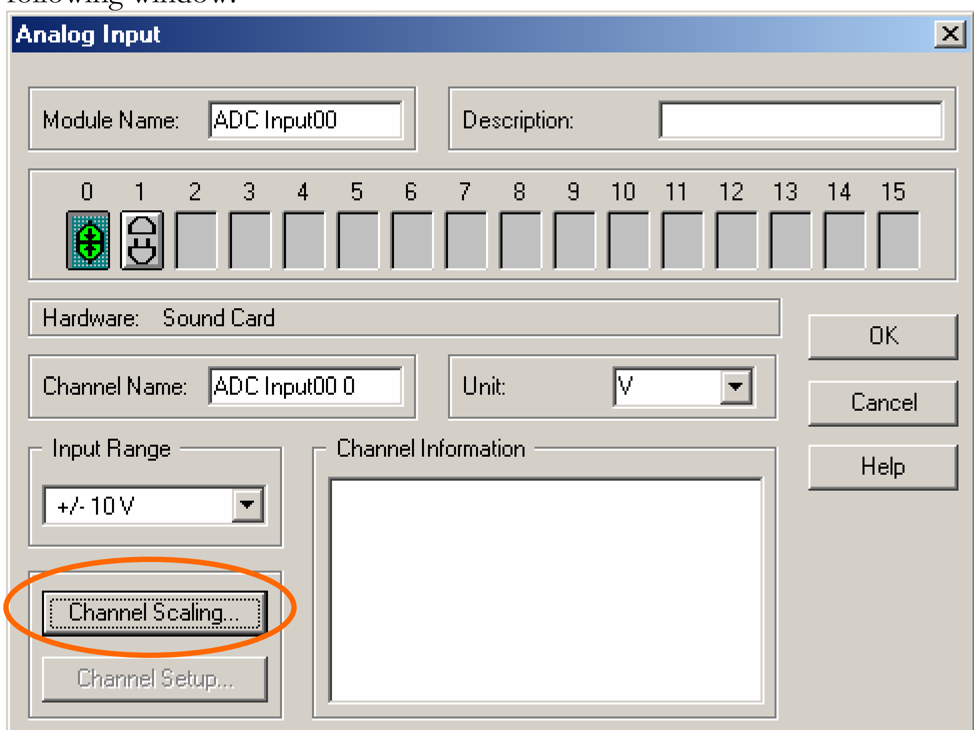
By increasing this value we can increase the display size by a multiple of the block size.
To find the length of this display in time we use the equation:
Time (Seconds)= BlockSize/SampleRate

BASIC SCALING

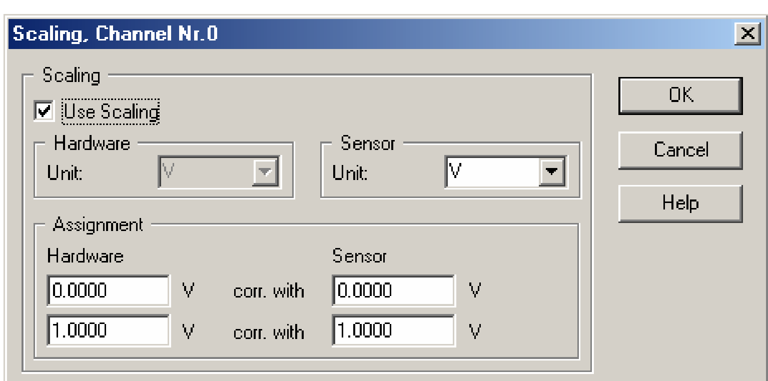
Many forms of analog data need to be scaled into units and magnitudes that are usable and visible to us. For example, thermocouples read temperature and send an analog signal in millivolts to the DAC card. The data then needs to be *linearized* into a temperature. The **DASY*Lab* Analog Input Module** and the **Scaling Module** take care of these calculations.

Other data may need to be scaled also. A signal of ±10mV is difficult to calculate with or may have no meaning in the real world. However if 1mV were equal to 10lbs then we can scale this data to real world values.

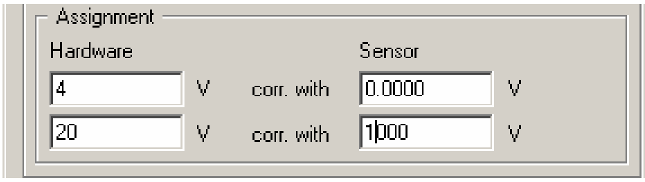
There are two ways to scale data in **DASY*Lab***. The first way is to use the Scaling found inside the **Analog Input** module. Double clicking on the **Analog Input** module displays the following window:



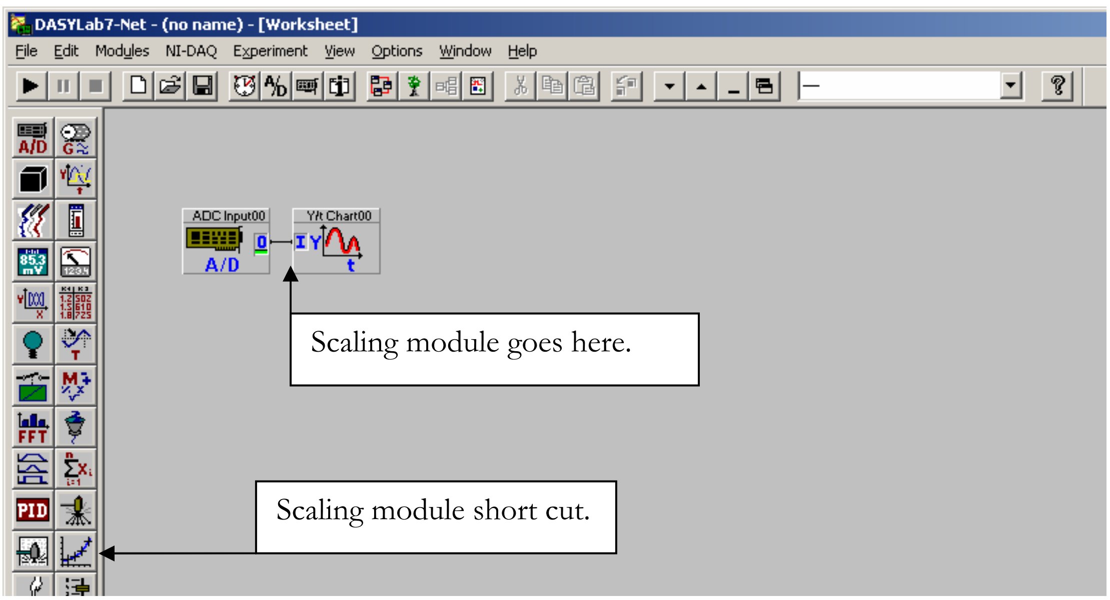
When we click on the "Channel Scaling" button we see:



Once the "Use Scaling" option is selected we can define a Two-Point Scaling scale. The way we enter a Two-Point scale is draw a correlation of a reading at the hardware with a real world value at the sensor. For example if we have a pressure transducer that outputs 4mA at 0 psi and 20mA at 1000 psi we would complete the Two-Point scaling as follows:

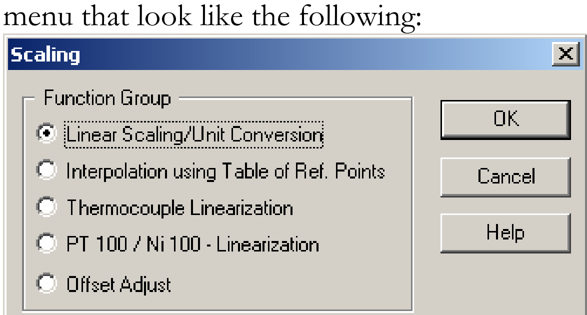


Two-Point scaling only works for linear relationships. For other scaling relationships we use the **Scaling** module found in **DASY*Lab***. The **DASY*Lab* Scaling** Module needs to be placed into the data line. The most common place for this calculation to take place is directly after the **Analog Input** module.

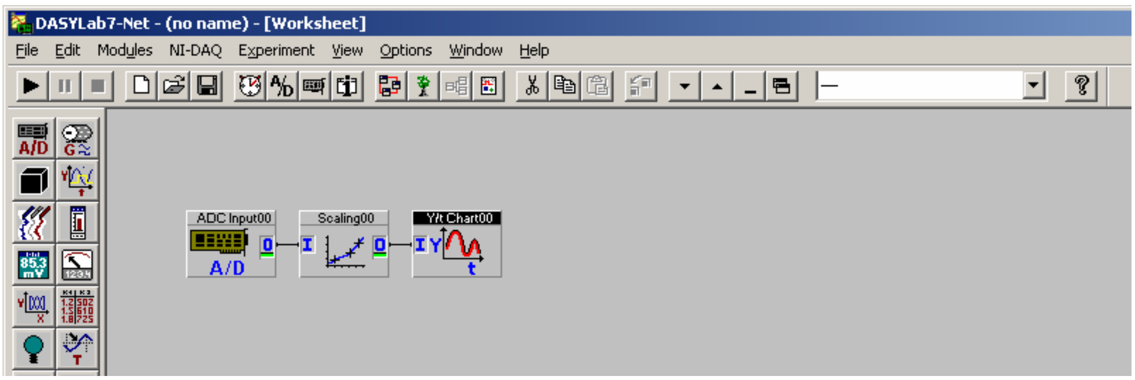


There are two way of disconnecting the **Y/t Chart** and the **Analog Input** module. First you can "right click" twice, not double click, on the wire, or you can right click on the **Y/t Chart** and select "Delete Input Channels". Move the two modules apart so that we have plenty of room to place the **Scaling Module**.

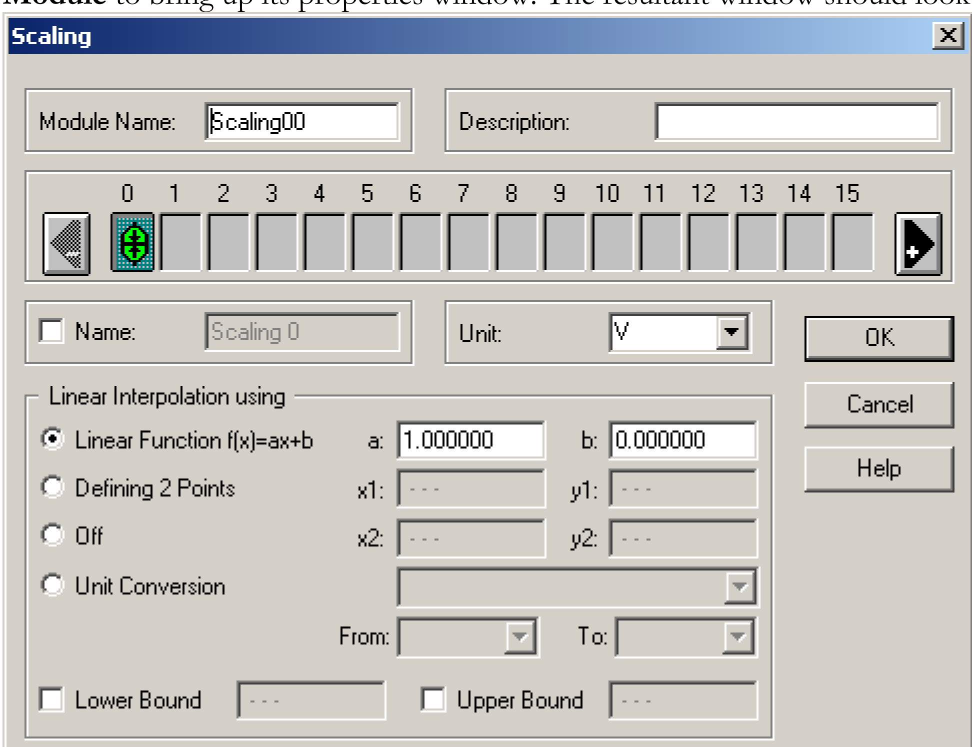
Select a **Scaling Module** from the module bar on the left or from the menu; Modules: Mathematics: Scaling. Once placed you will be confronted with a series of selections in a menu that look like the following:



Each of these options customizes the module for a single purpose. For example the third option, "Thermocouple Linearization" adjusts the millivolt reading from a thermocouple into a temperature. For our example we will be using the "Linear Scaling/Unit Conversion" option. Select this option and click OK.

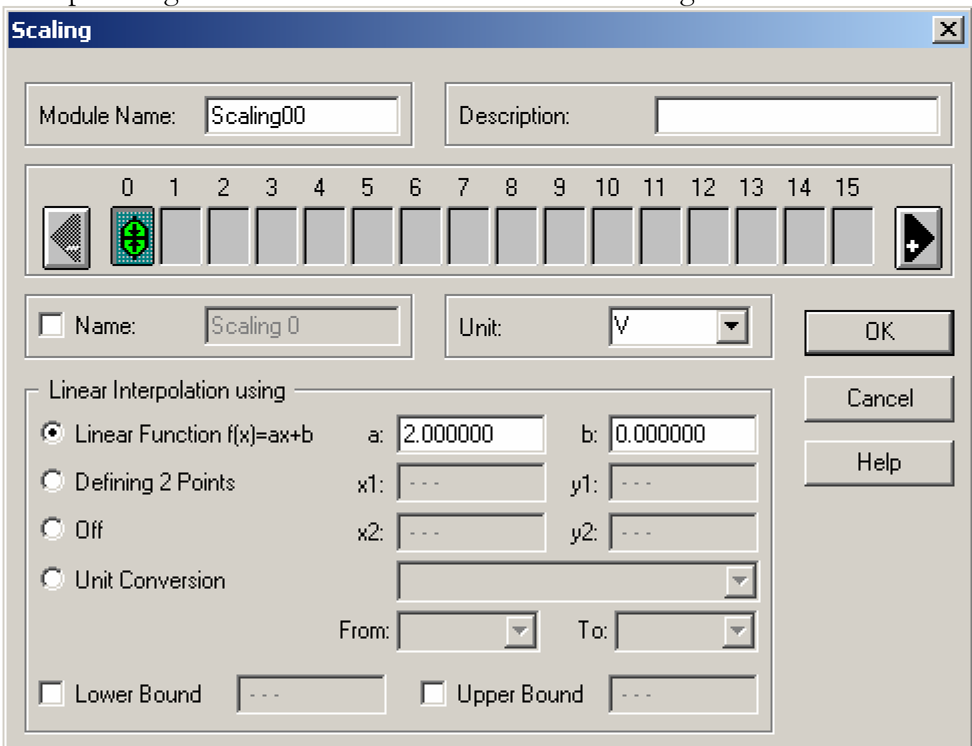


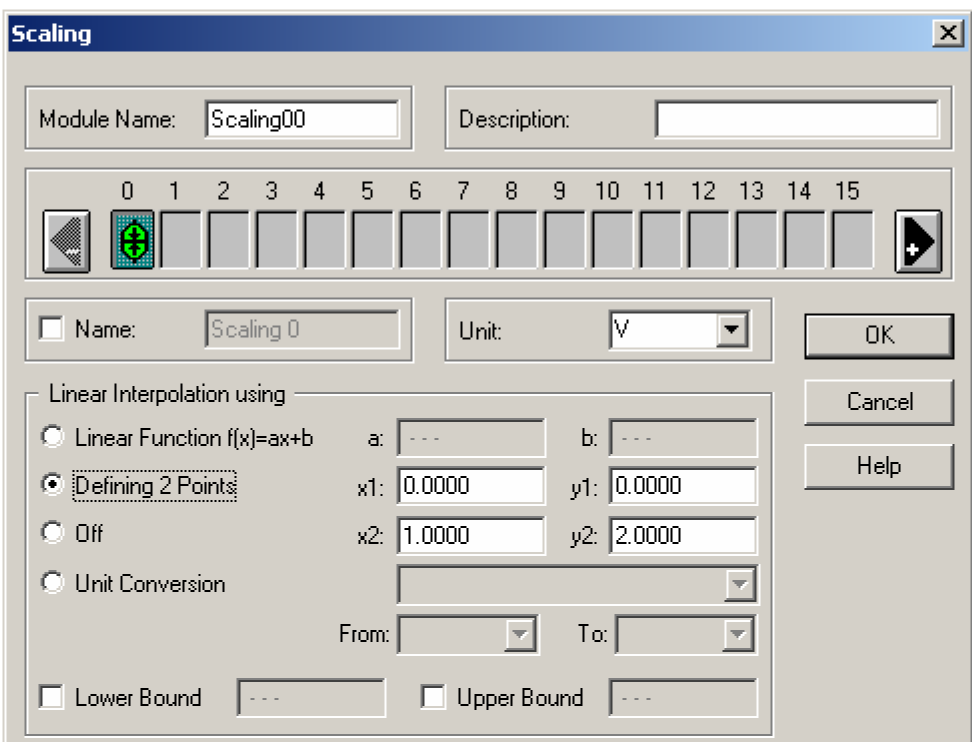
Once wired the worksheet should look like the example above. Double click on the **Scaling Module** to bring up its properties window. The resultant window should look as follows:



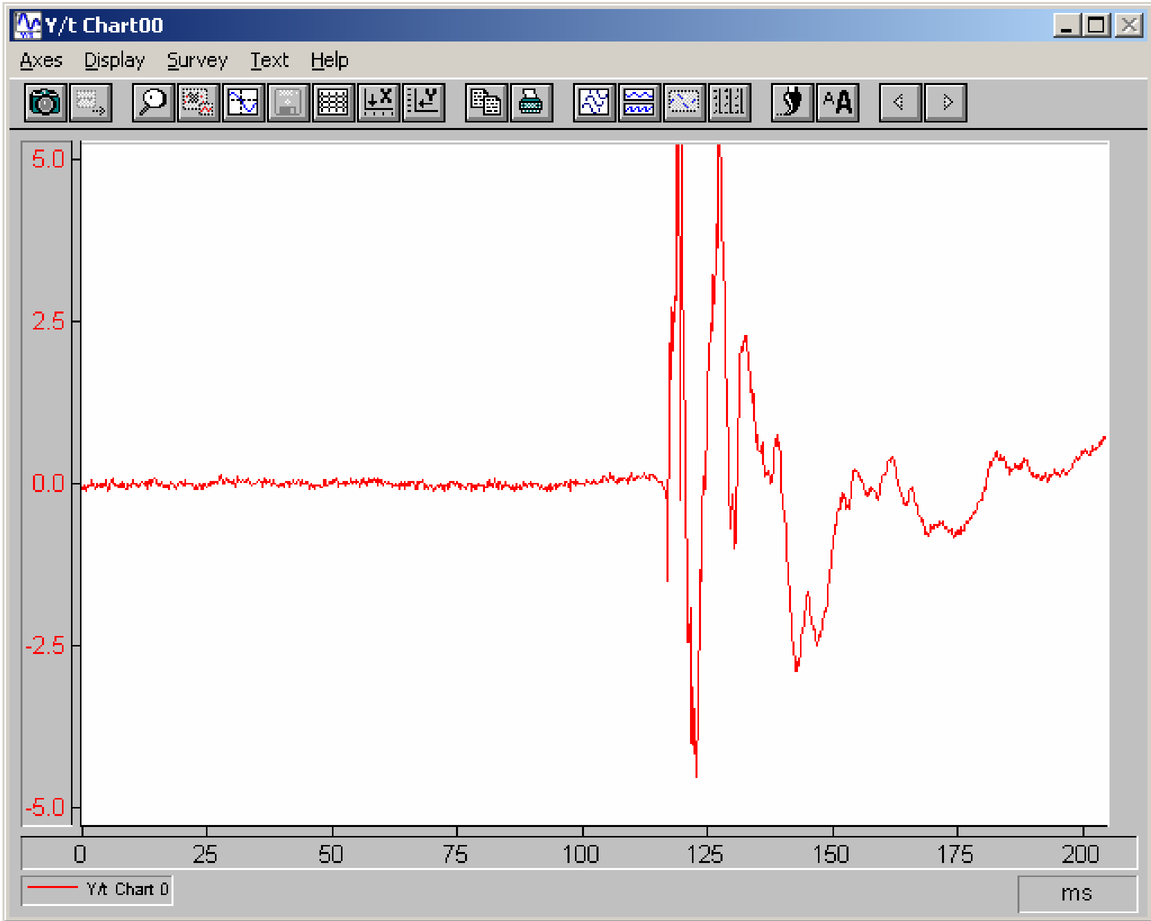
This dialog box will allow you to define the type of linearization we wish to implement. The first option adjusts for both magnification (scaling) and offset. The second option, defining 2 points, interpolates the data based on two received data points and their known real world equivalents. The third, off, can be used to limit data to defined lower and upper bounds, also configurable inside this module. The last option provides an easy way to convert from one known unit to another.

In this example we will be using the "Linear Function" option. In our previous examples you may have noticed that our signal was about ±2.5volts. We may, however, want a visual resolution of ±5 volts. Therefore we can scale this value by a factor of 2. Here are two example using the "Linear Function" and the "Defining 2 Points" method:





When we run the application with these setting we see a display similar to this:



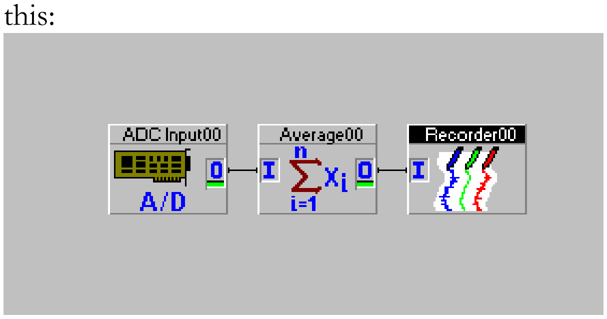
We will save this worksheet for use later.

BASIC DATA REDUCTION

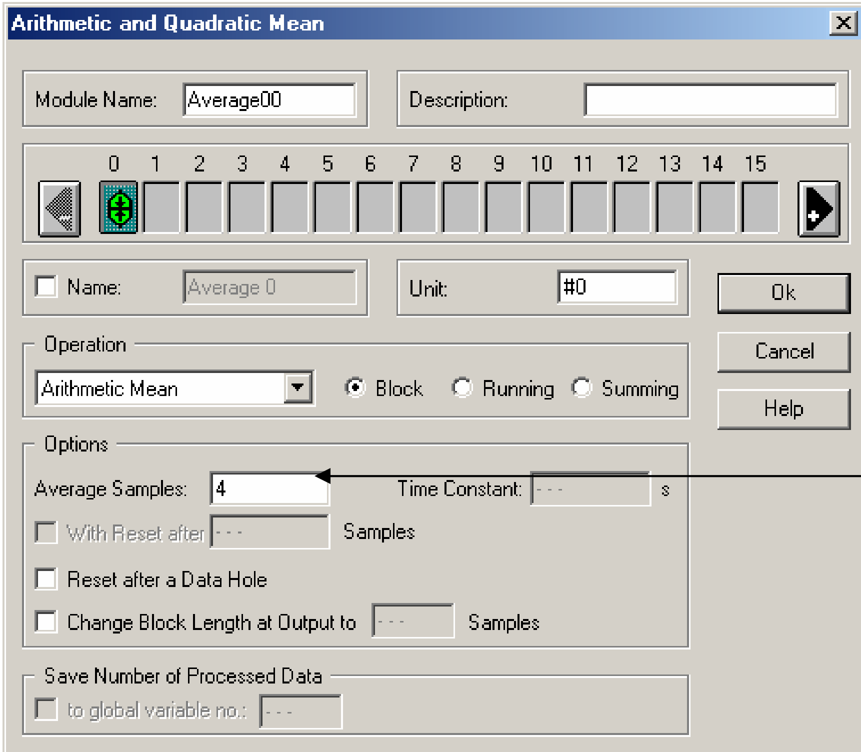
When recording data and saving it to the hard drive at a higher speed we can consume a huge amount of hard drive space in a short period of time. However we do not wish to slow down our sampling rate, so what can we do? We can "cut out" or reduce the number of samples sent to the hard drive, we can then increase the rate of data saved to the drive if necessary.

There are several different ways to perform a data reduction. The one way is to average the data by producing one sample which is the average of several others. By averaging the data we reduce the data points while still keeping some information about any events that occurred.

To demonstrate this we will create a new worksheet. This one will be similar to our oscilloscope worksheet, with the exception that a Chart Recorder Module will replace the Y/t Chart. Between the **Analog Input** and the **Chart Recorder** we will place an **Average** Module, found under Module, Data Reduction. The completed worksheet should look like this:

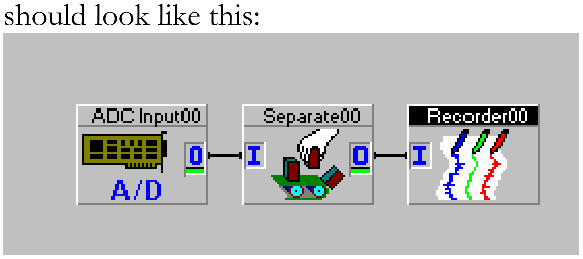


By double clicking on the **Average** module we get into its properties window:

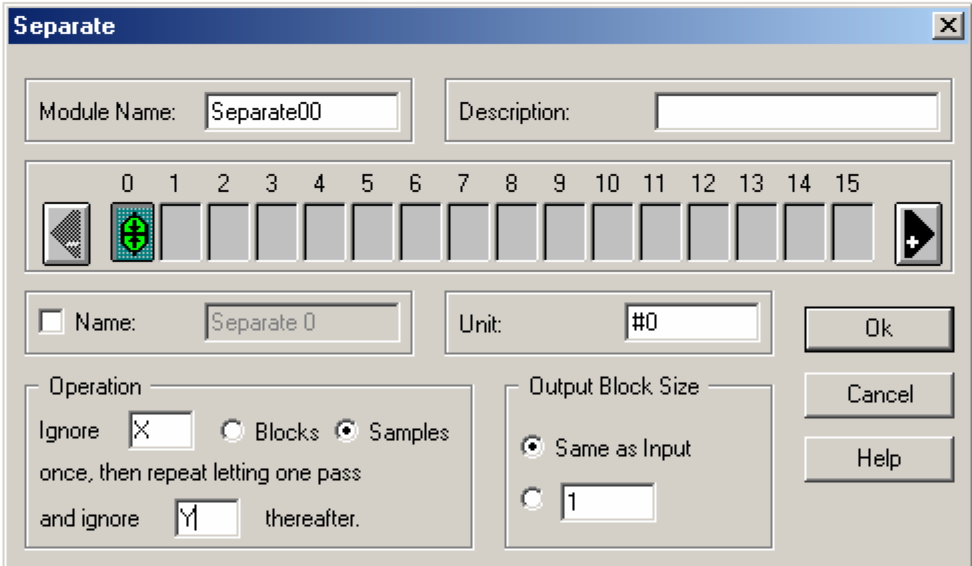


By increasing or decreasing this value we increase or decrease the number of samples that are averaged together into one sample. By default the module averages 4 samples into 1, thereby decreasing your data by 1/4th.

Another way to reduce data is to eliminate the extraneous samples completely. To accomplish this we can use the **Separate** module in place of the **Average** module is our last example (**Separate** is found under Module, Data Reduction). The complete worksheet should look like this:



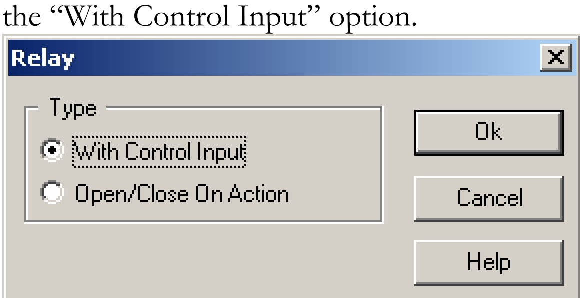
When we double click on the separate module we see its properties window:



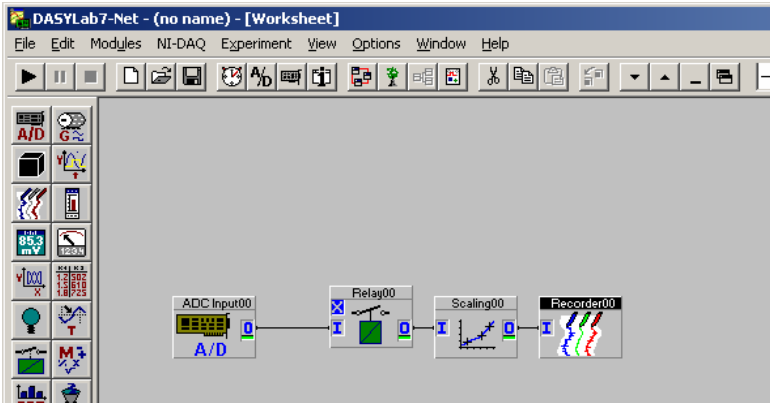
This module can be configured to ignore samples or to ignore entire blocks of samples. The module ignores X and then lets one pass through then ignores Y letting 1 pass, then again ignoring Y again and letting 1 pass and so on. X and Y both default to 9.

Another form of data reduction allows us to acquire data at one frequency and save it at another, ignoring the extra samples. Using the worksheet we have already used for the last two examples we delete the **Separate** Module and we will need to add a **Slider Module, Generator Module,** and a **Relay Module**. The idea is that the data is acquired at a fast rate; the data enters **DASY*Lab*** at the **Analog Input Module** and immediately hits the **Relay Module**. Here data either passes through or falls into the "Bit Bucket" and disappears. The relay is being opened and closed at a rate that we define and can change "On the Fly" as we see fit.

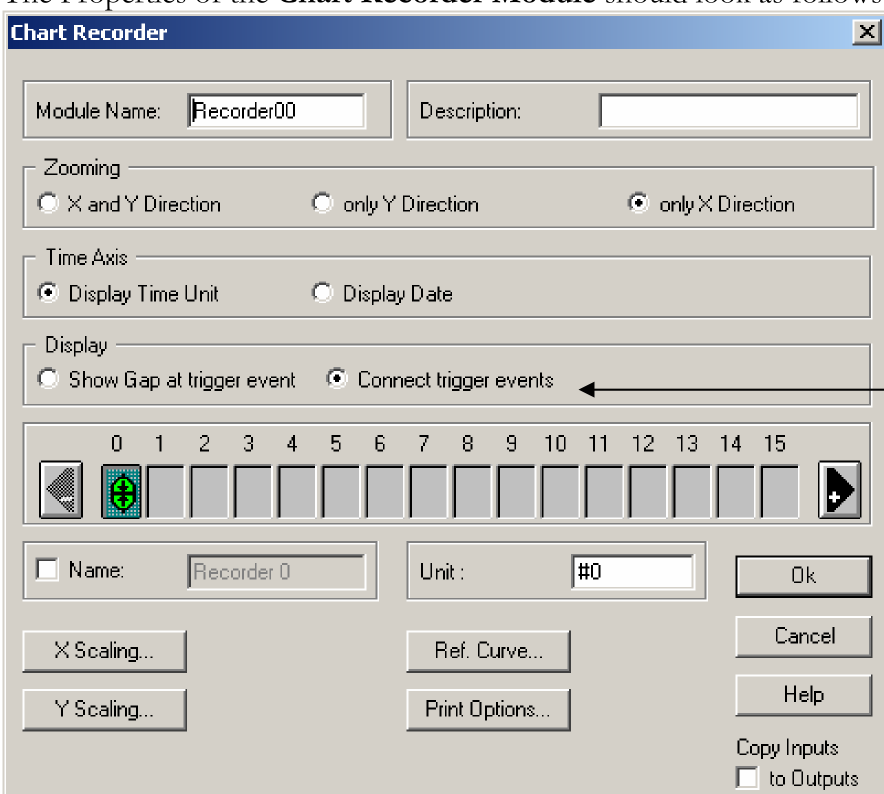
Here is how the pieces fit together. First let's disconnect the **Analog Input** from the **Scaling Module**. Move everything apart so that we can fit another module in between the analog input and the scaling. Place a **Relay Module** onto the worksheet, Module, Triggers, Relay. You will have a choice of two options for which type of relay you want, we will use the "With Control Input" option.



Note that this **Relay Module** has an input (I) and output (O) and a control (X). Connect the output of the **Analog Input** to the input of the **Relay Module** and the output of the **Relay Module** to the input of the **Scaling Module**. So far the worksheet should look like this:



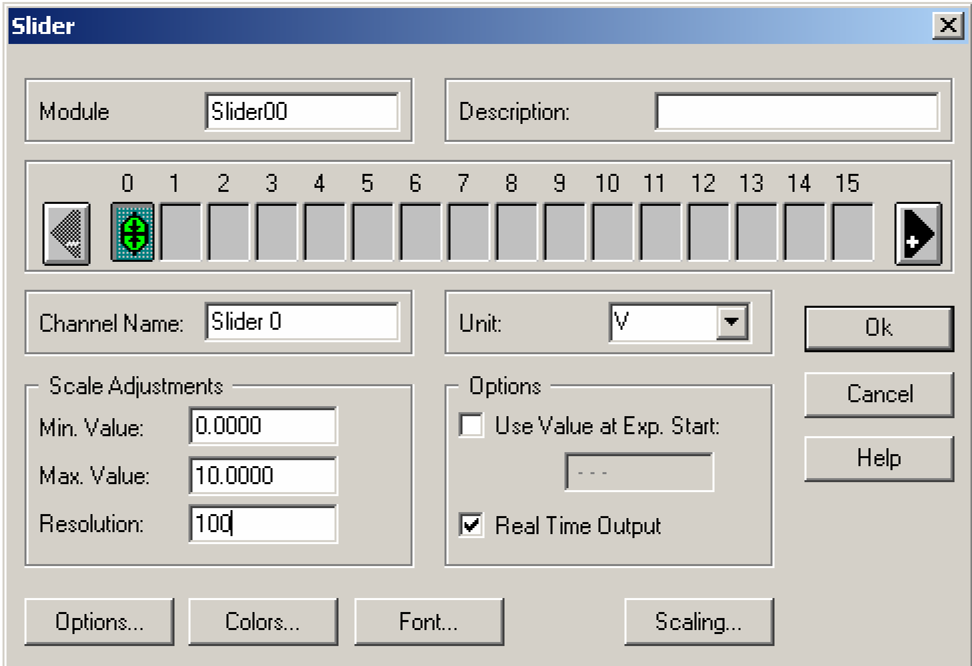
The Properties of the **Chart Recorder Module** should look as follows:



NOTE:
The connect trigger events option is selected.

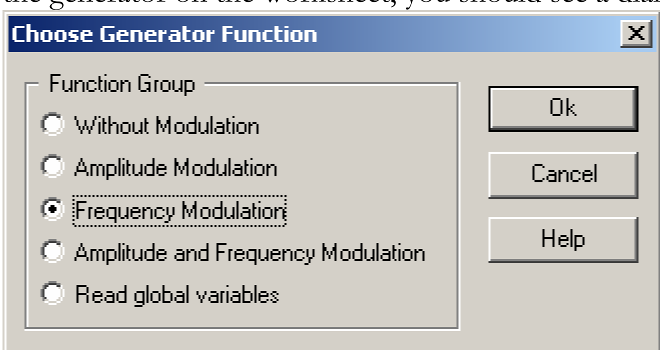
We now need a form of control to open and close the relay at our desired rate. We will use a **Generator Module** to generate a "pulse train" to close the relay at a regular interval. We will also use a **Slider Module** to gain a graphical method of modifying the rate of the pulse train.

First we will add the **Slider Module** to the worksheet. These are located under Modules, Control, Slider. Double click on the **Slider Module** to access its properties.



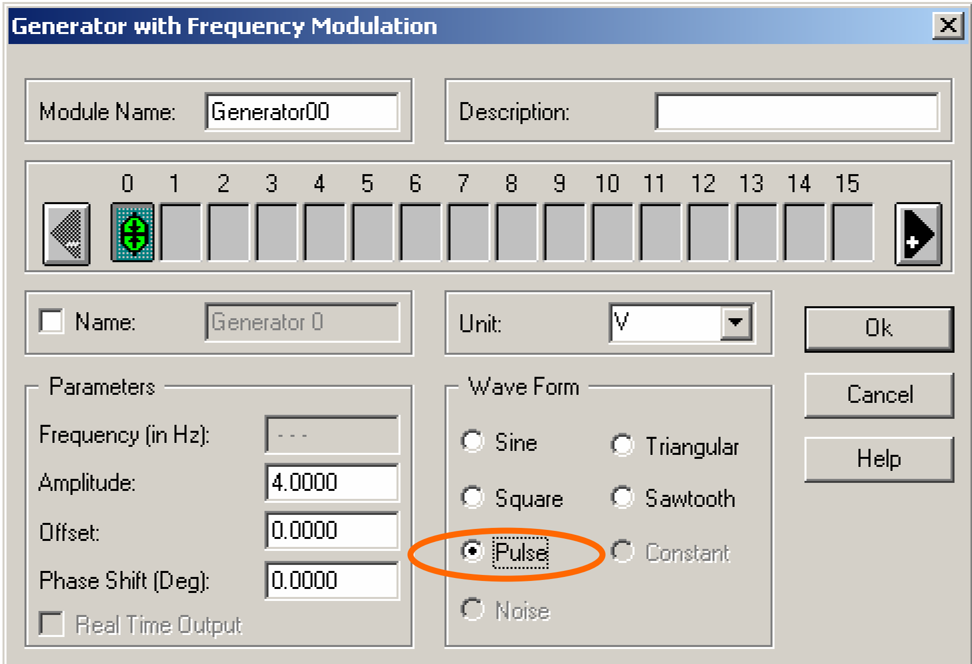
For this example we are going to set the "Max. Value" property to 10. Click OK.

We can leave that module floating there for the moment while we setup a **Generator Module.** To find the Generator go to the Modules Menu, Control, Generator. After placing the generator on the worksheet, you should see a dialog box like this:

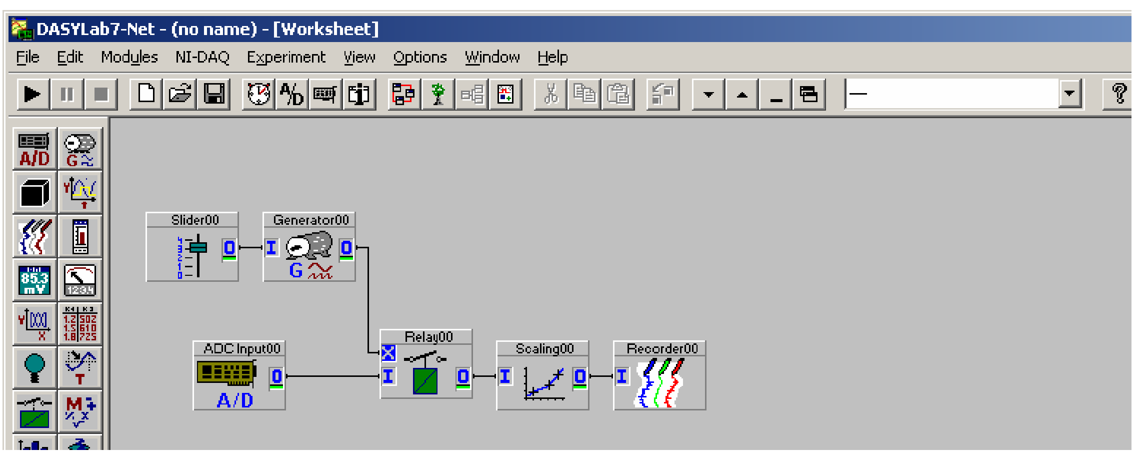


This dialog box allows you to select the function and available variables for the generator. For this example we want the third option "Frequency Modulation". Select this and click OK.

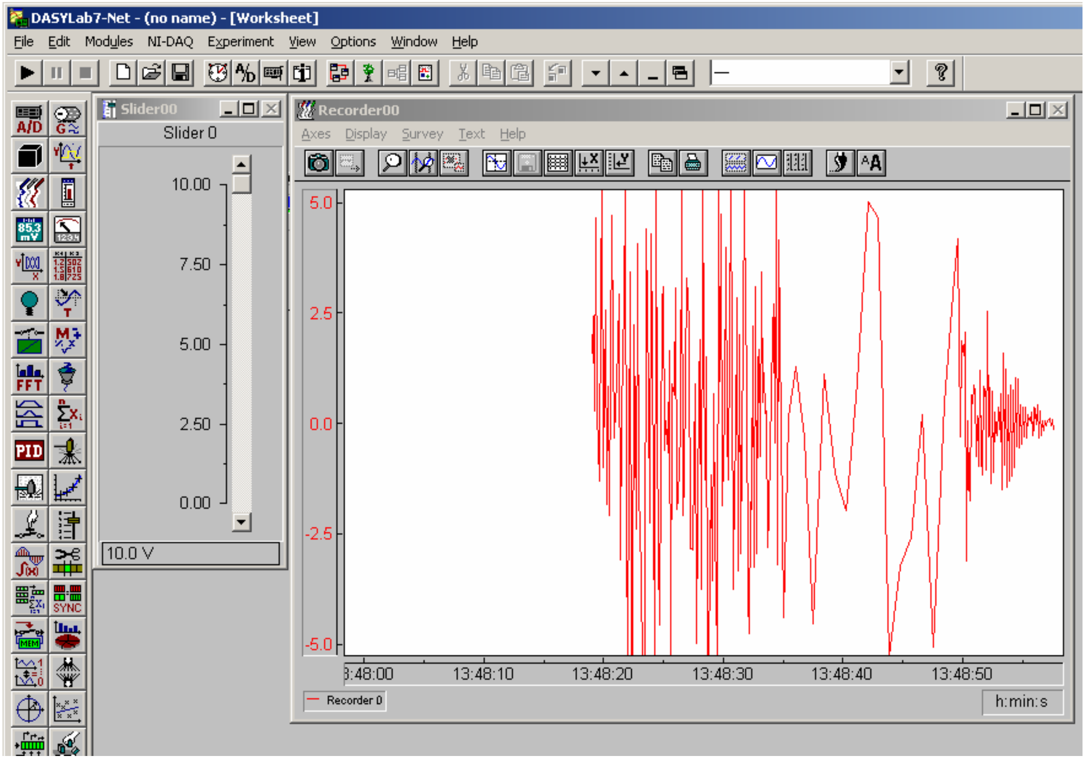
Double click on the **Generator** and change the wave form to pulse:



Now we can connect the output on the **Slider** to the input on the **Generator**. Lastly we connect the output of the **Generator** to the control (X) input on the **Relay**. The finished worksheet should look like this:



by bringing up the displays you should see a slider with the option of selecting a value from 0 to 10 and your Chart Recorder display, as follows:



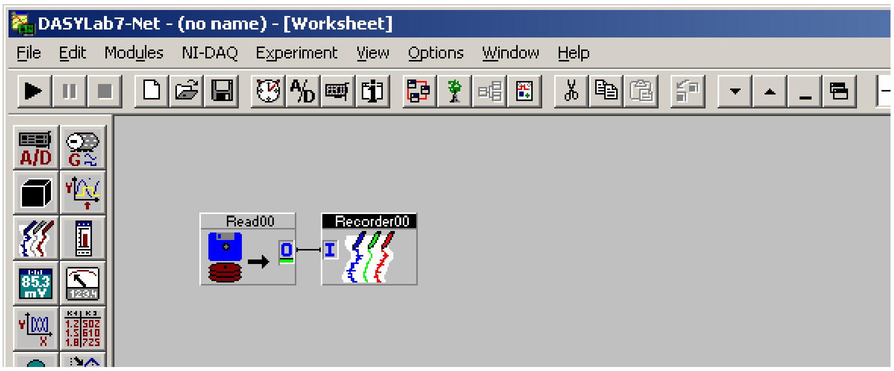
As we decrease the slider value we will decrease the number of samples sent to the display and, if connected, the **Write Module** and the hard drive. We will also see that the signal quality on the graph will dwindle as the number of samples it receives drops.

Other variations of this include the ability to record data only after an event, pre-triggering to record only an event, and recording only for a period of the day. These types of data reduction are covered in the intermediate sections.

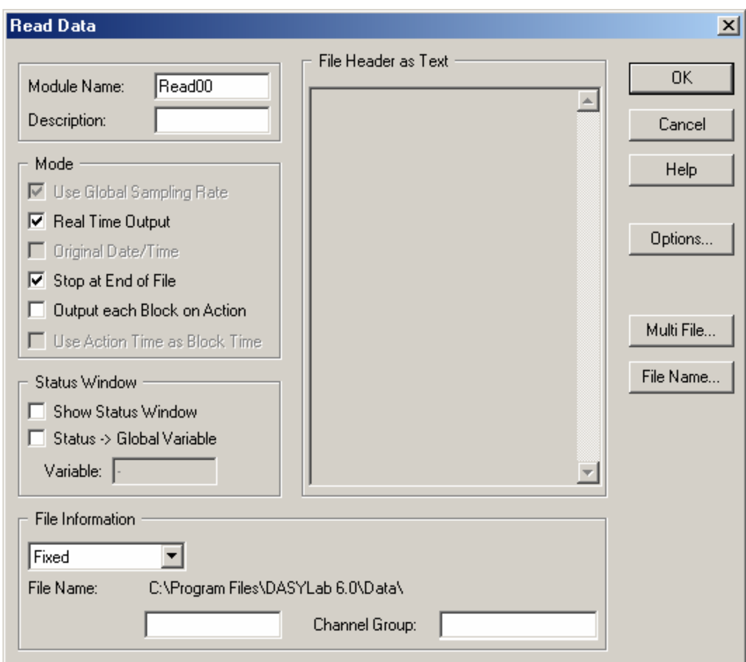
BASIC DATA PLAYBACK

Once we have saved data to the hard drive we need a way of playing the data back. **DASY*Lab*** has the **Read File Module** that performs the function of reading data back into **DASY*Lab***. Once data is read back into **DASY*Lab*** it appears the same as our **Analog Input** data.

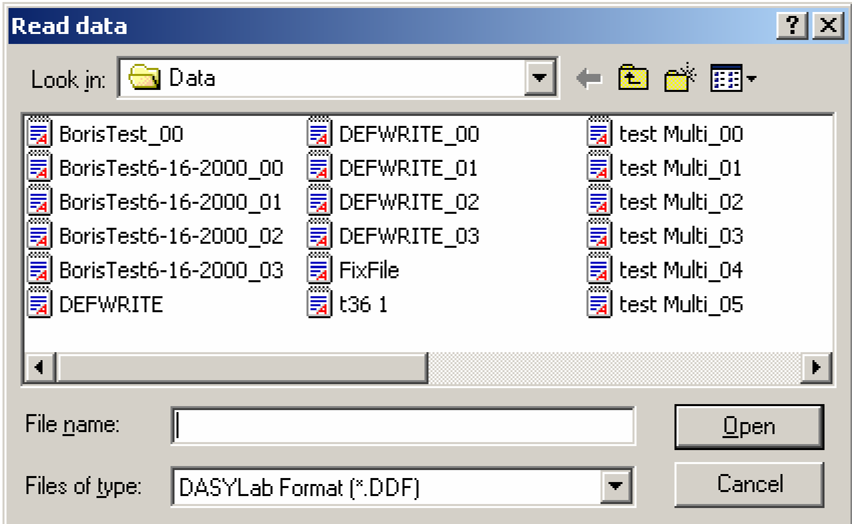
Open a black worksheet and place a **Read Module** onto the worksheet, found under Modules, Files, Read Data. Connect this module to a **Chart Recorder** module. The worksheet should look as follows:



Double click on the **Read Module** to access its properties.



Click on the "File Name…" button to bring up the files selection dialog box.



Click on the file of your choice and click "Open"

Click OK to close the **Read Module** configuration dialog box. Start the worksheet. We now see the data we recorded earlier being played back and displayed.

This application concludes the "Basic" examples. We will now move on to the slightly more complicated "Intermediate" examples. Here we will work on writing multi files, creating alarms, and using the **Action Module**.

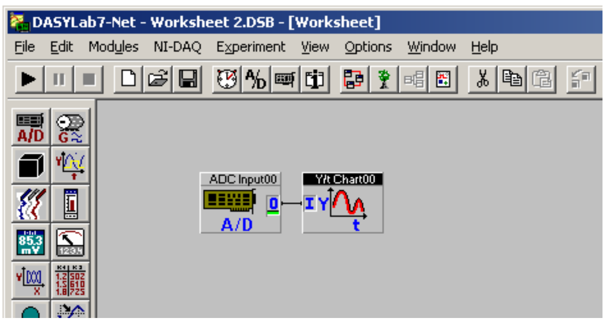
CHAPTER 5: INTERMEDIATE DASYLAB APPLICATIONS

ADVANCED DATA LOGGER WITH ALARMS

We will now re-visit our data logger and add more functionality to it. First we will add *Alarms* to it. The alarm will indicate if the signal as surpassed a minimum or maximum value.

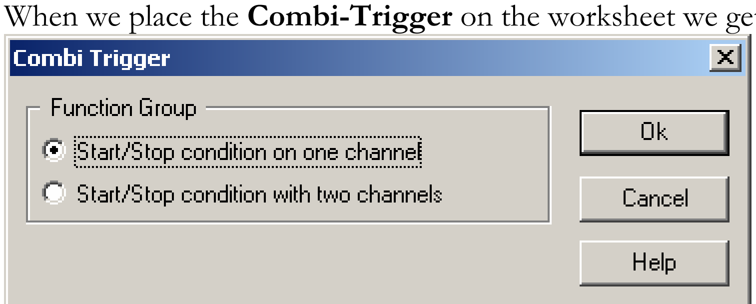
There are two types of alarms available, the momentary and the latch. A momentary alarm is active only while the alarm condition is true, for example if we wish to alarm when the signal exceeds 10 volts then the alarm will trigger as long as the signal is greater then 10 and will stop sounding when the signal drops below this value. The latch alarm triggers the same however when the signal drop below the latch value the alarm continues to sound until the operator resets it.

First we will create a *Latch* alarm first. Let's recall our oscilloscope from our previous worksheets. The worksheet should look as follows:



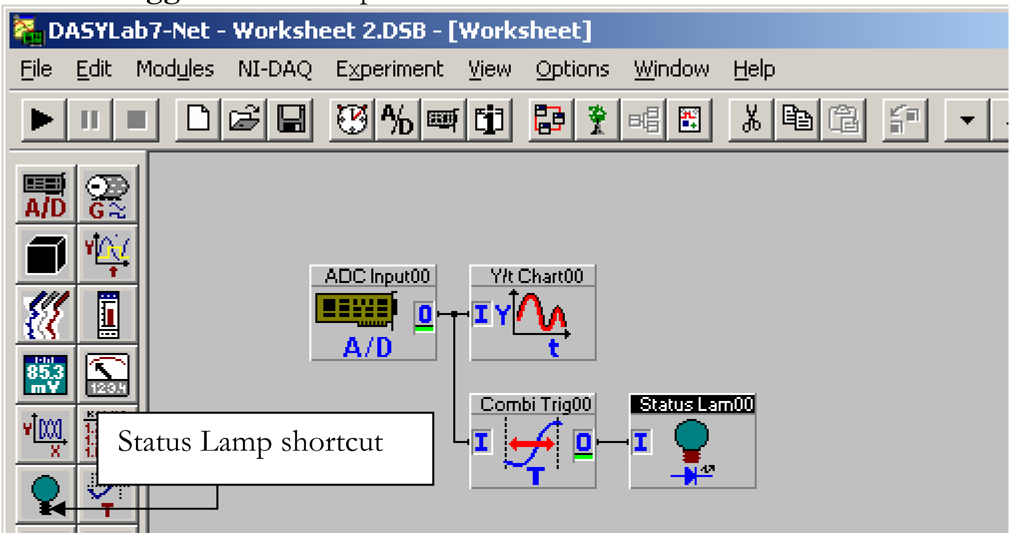
We will be adding a **Combi-Trigger** and a **Status Lamp** to this worksheet. The **Combi-Trigger** can be found under Modules, Trigger Functions, Combi-Trigger and the **Status Lamp** under Modules, Display, Status Lamp.

When we place the **Combi-Trigger** on the worksheet we get a dialog box like this:

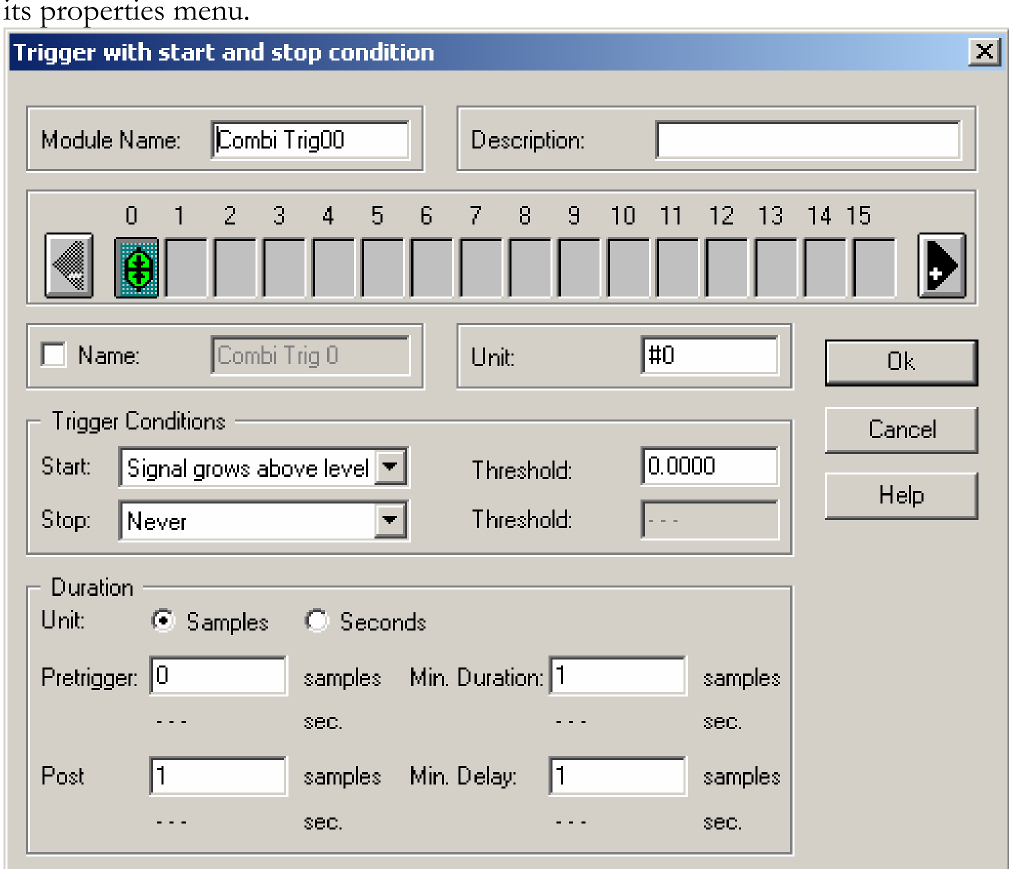


We will use the "Start/Stop condition on one channel" option for our worksheet. To connect this module into our data left click on the wire between the **Analog Input** and the **Y/t Chart**, your cursor should change to the wiring cursor, then click on the input of the **Combi-Trigger**.

We can now place a **Status Lamp** on to our worksheet and connect it to the output of the **Combi-Trigger.** When completed the worksheet should look like this:



We should now set up our alarm conditions. Double click on the **Combi-Trigger** to access its properties menu.

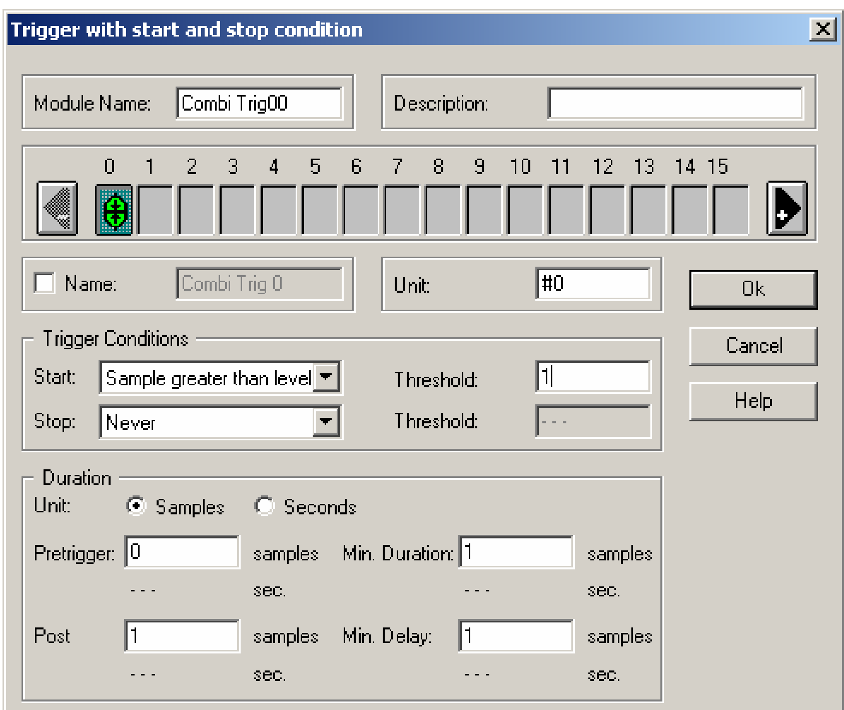


We can now select the conditions that will cause the trigger to go high (Start) and the conditions that will cause the trigger to go low (Stop). We have many options here, this is a breakdown of the meaning of each option.

Trigger Condition	Explanation
Signal Greater than Level	If the signal is greater then the threshold value then the event is true
Signal Less than Level	If the signal is less then the threshold value then the event is true
Signal Grows Above Level	If the signal grows above the threshold value then the event is true. In order to show valid growth there must be a sample lower then the threshold first.
Signal Falls Below Level	If the signal falls below the threshold value then the event is true. In order to show a valid fall there must be a sample higher then the threshold first.
Rising TTL Edge	The signal changes from <1.5 to >1.5 in 1 sample
Falling TTL Edge	The signal changes from >1.5 to <1.5 in 1 sample
Never	The condition is never true
Direct	The condition is true as soon as the initial condition is false.

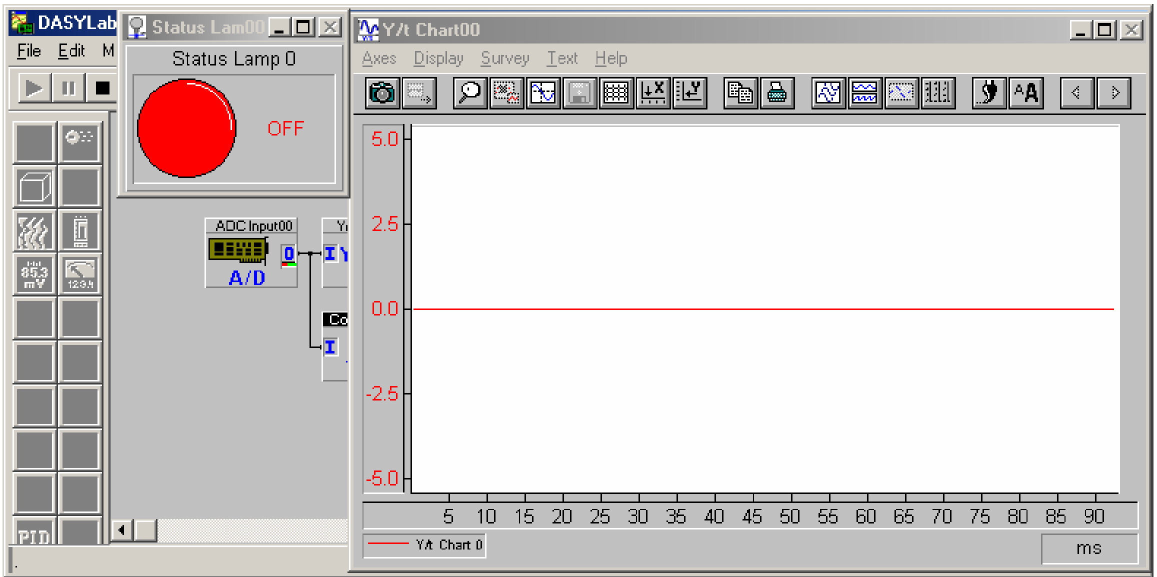
We will select the following trigger conditions:

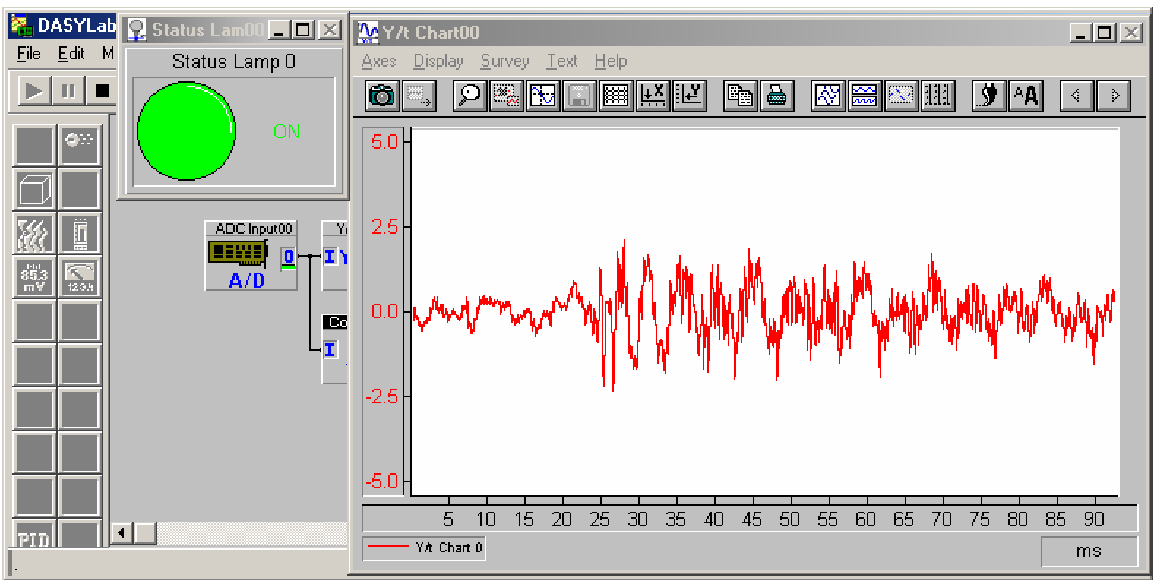
TRIGGER	EVENT	THRESHOLD
START	Signal Greater then Level	1.00
STOP	NEVER	



Click OK, bring up the display windows and click start.

With the application running, when I tap the microphone I get a spike above 1. This causes the **Combi-Trigger** to go *High*. Because there is no stop condition the signal will stay high until we stop ***DASYLab,*** this is a condition known as "latched".





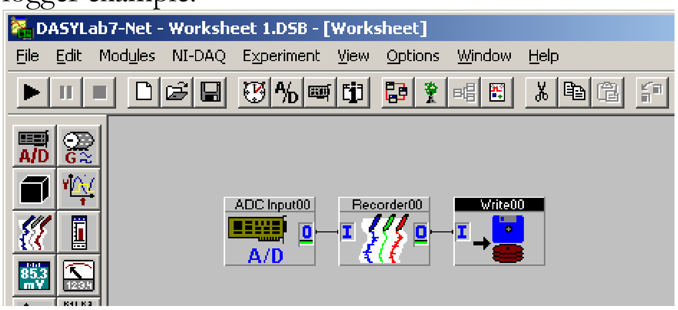
Changing the Stop condition to direct can easily change this Latch alarm into a momentary alarm. However, because we are working with a sound card it is difficult to sustain a sound level about 1 without disturbing everyone in the office, therefore you way wish to change the threshold value to .005. Still sound waves modulate and will pass through the alarm values quickly and due to the graphics nature of Windows® the screen may not update the **Status Lamp**. To better test this, replace the **Analog Input** with a **Slider Module** and experiment with that setup.

Save your worksheet for later.

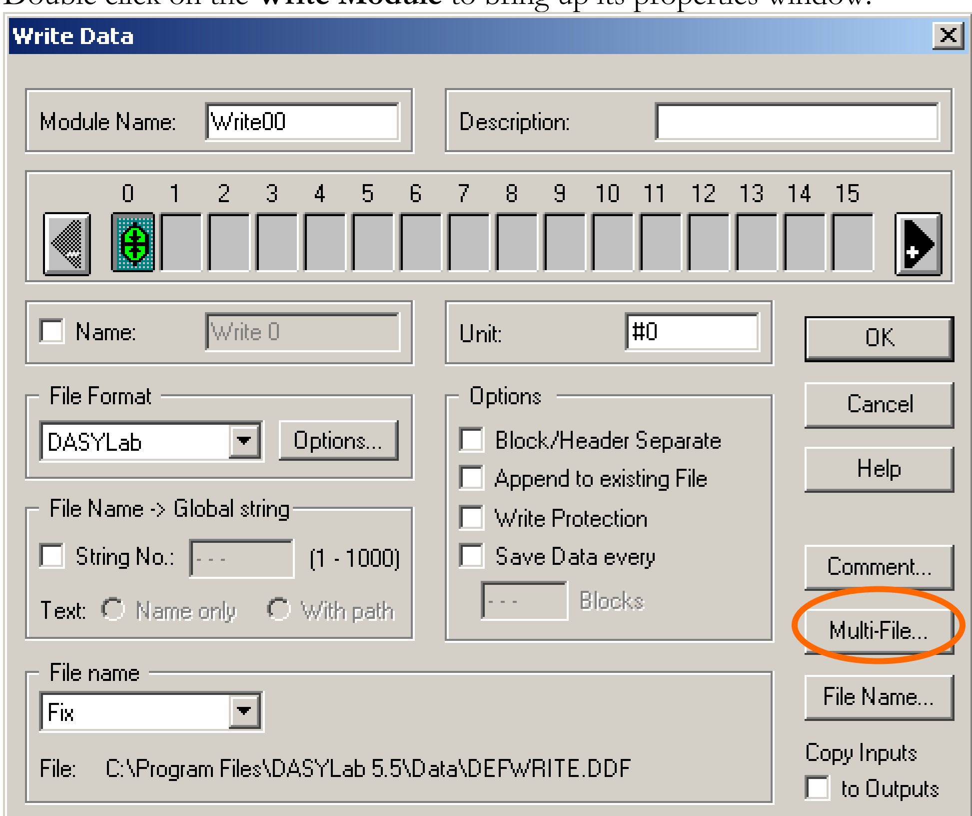
DATA LOGGER WITH MULTI-FILE

Despite our best efforts we may need to acquire a large amount of data and save it to the hard drive. It is rather cumbersome and hazardous to handle extremely large files. **DASY*Lab*** has the ability to write a file chain with the same amount of data in each file to split it up into several smaller files. These files follow a naming convention, FILENAMEXX where XX is an incremental number from 00 to 99 or higher. The file changes can occur at regular intervals or by interacting with the ACTION module.

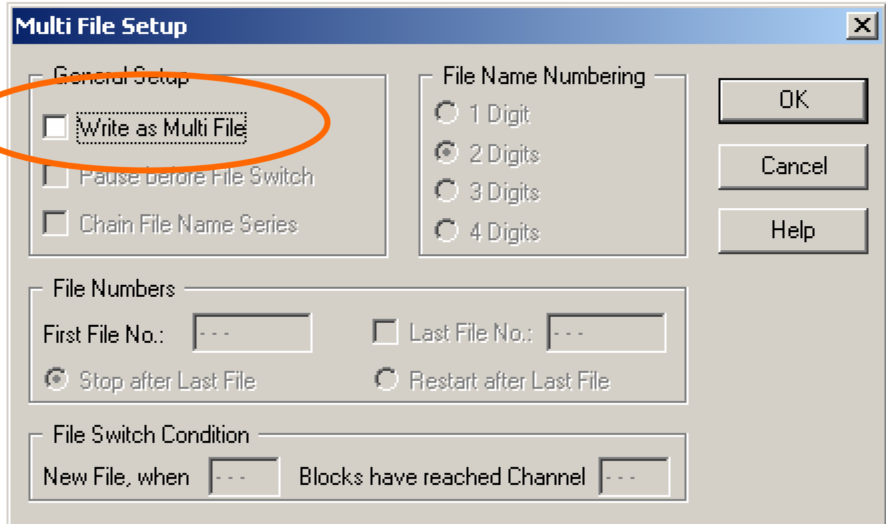
We will configure **DASY*Lab*** to change files at a regular intervals. Start by loading our data logger example.



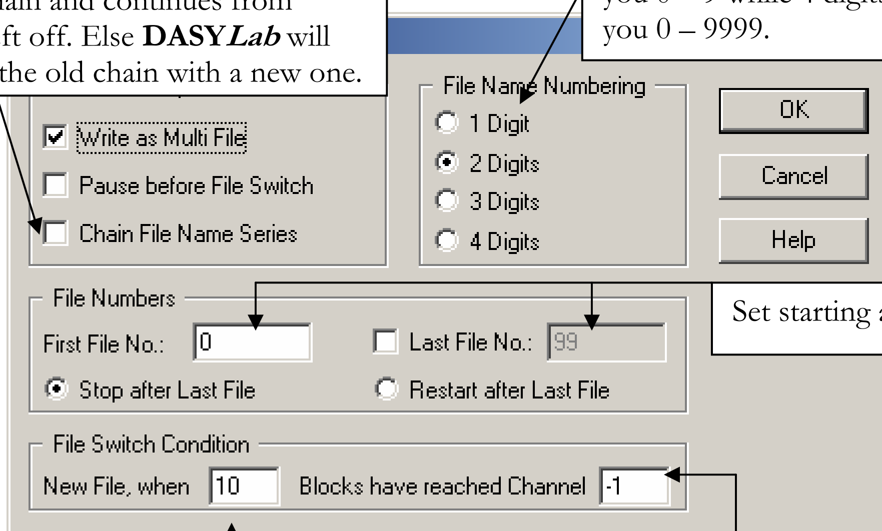
Double click on the **Write Module** to bring up its properties window.



Notice the *Multi-File* button on the left side of the screen, click it.



Click the "Write as Multi File" check box.
We may now set the parameters to fit our needs:



Will cause **DASY*Lab*** to search for an existing chain and continues from where it left off. Else **DASY*Lab*** will overwrite the old chain with a new one.

Select the number of files you expect to use. 1 digit will give you 0 – 9 while 4 digits will give you 0 – 9999.

Set starting and ending file number.

This selects how many blocks of data **DASY*Lab*** will write to the file before creating a new file. To calculate this into samples take this number and multiply it by the block size. To calculate this into seconds take the number of samples and divide by the sampling rate.

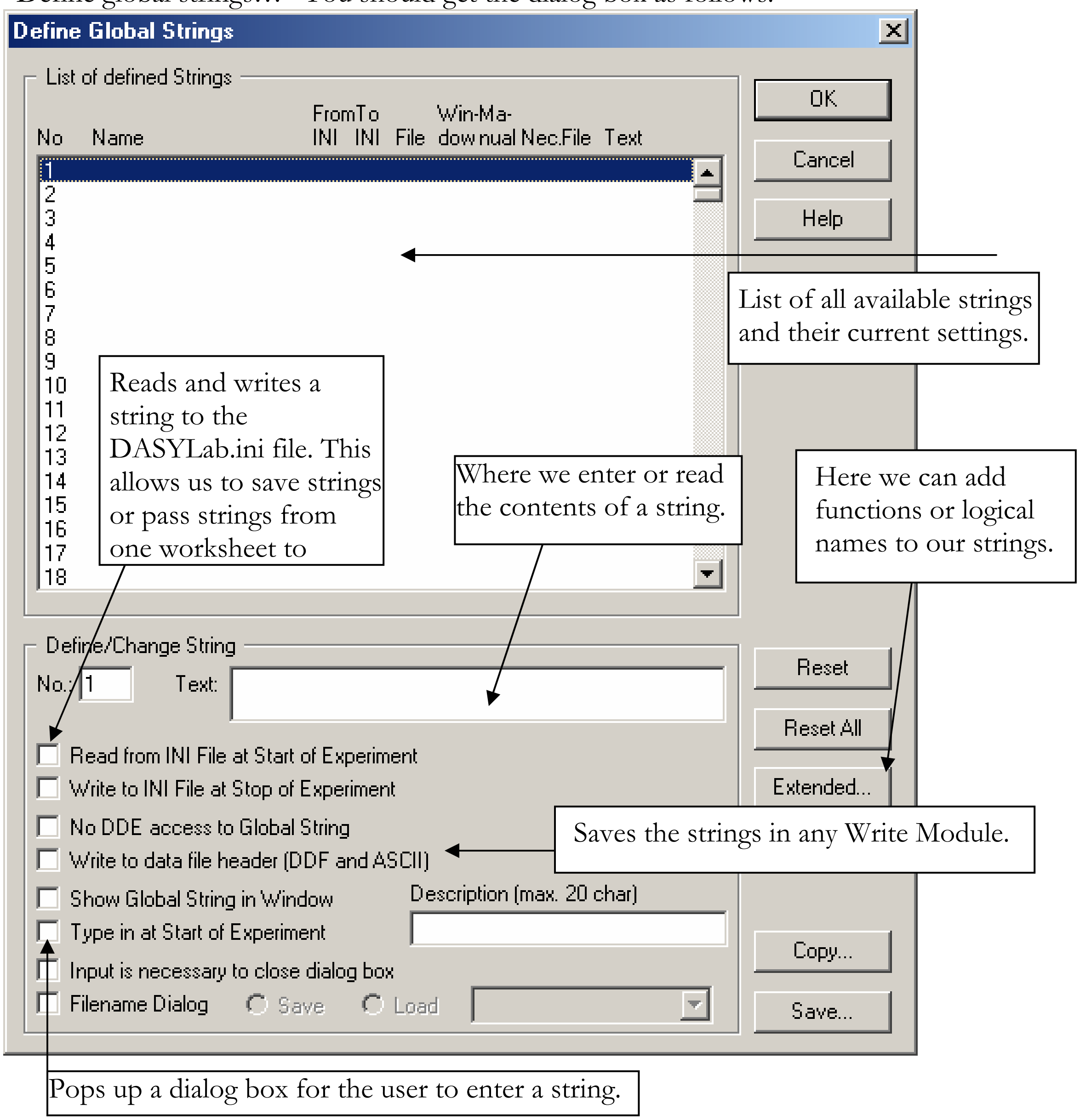
Which channel should **DASY*Lab*** monitor for the file switch condition? –1 indicates any channel.

After setting the specifications to our desire click OK, then OK again. The worksheet should not have changed, however if we run our worksheet, and let enough time pass, we will find that **DASY*Lab*** has created several files in the file chain system.

DATA LOGGER WITH AUTOMATIC FILE NAMING

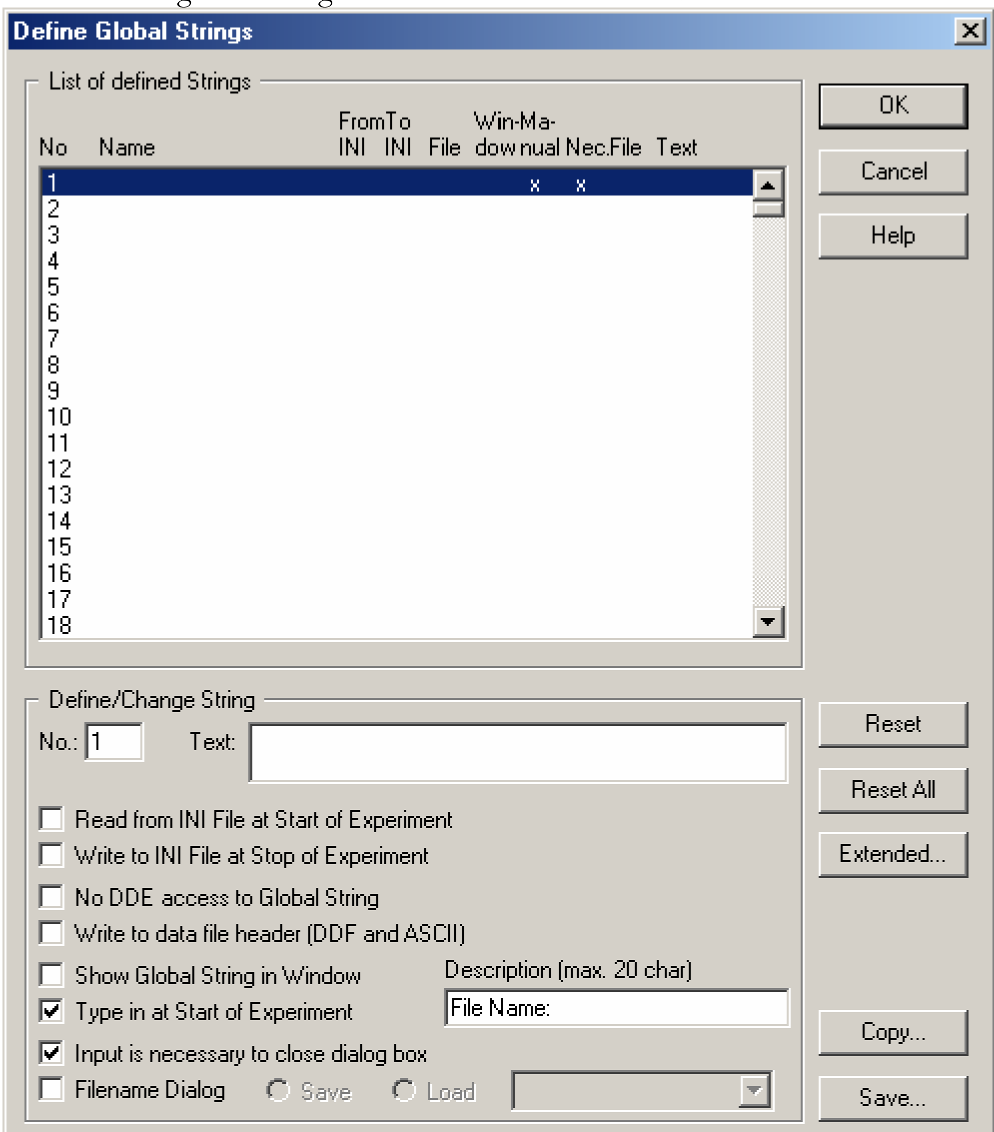
A simple change to any write module will enable it to change the file name based on user input or to use the time and/or date. This will require us to start using the String functions inside **DASY*Lab***. There are 999 user definable strings as well as several system-based strings available inside **DASY*Lab***.

To demonstrate this we will use our Write as Multi-File worksheet. As you may have noticed there is an options dropdown menu available on the menu bar. Click on "Options" then "Define global strings…" You should get the dialog box as follows:



We will go through three different configurations; the first will be using a User Entered string, second; using both a User Entered string and a System String; and lastly; Using the Filename dialog option.

The first dialog box configuration should look as follows:

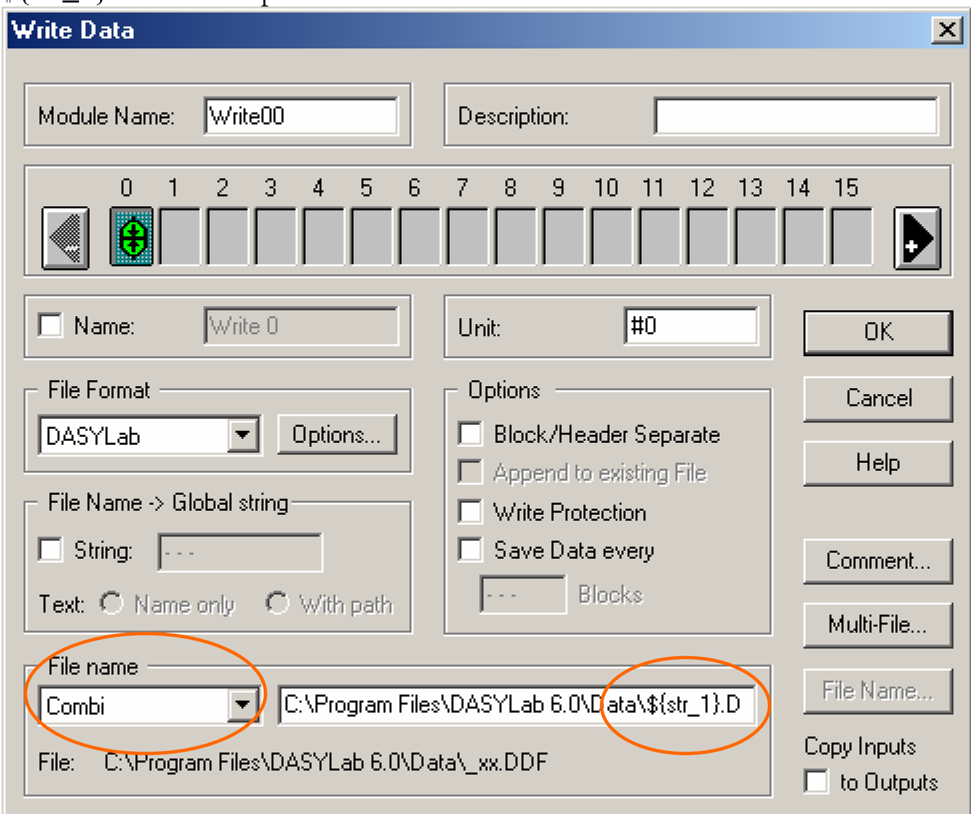


This configuration will pop up a dialog box when the experiment starts and will require the user to enter information before the window can be closed. It will prompt the user with "File Name:" and will store the string in location 1.

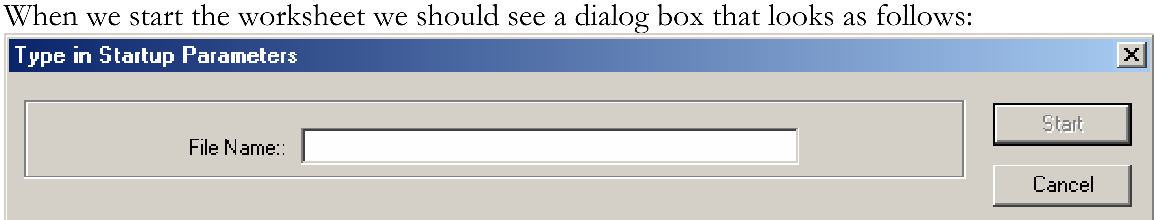
We can now click "OK" and exit this dialog box and double click on the **Write Module** to "interpret" this variable.

Interpreting a string requires us to place a representation of the string where we want **DASY*Lab*** to place the data from inside that string. We use the following format to interpreting a string inside **DASY*Lab***: ${str_#}, where # is the number of the string we wish to interpret. In this example we will be de-referencing using ${str_1}.

If we edit the properties of the **Write Module** we can add the string to our existing file name. We first change the name to "Combi" in the drop down menu to the left of the file name. This will allow us to combine **DASY*Lab*** strings with constant characters. We can now edit the file name to include our string. I replaced the old name "Defwrite" with ${str_1}. When completed the **Write Module** should look as follows:

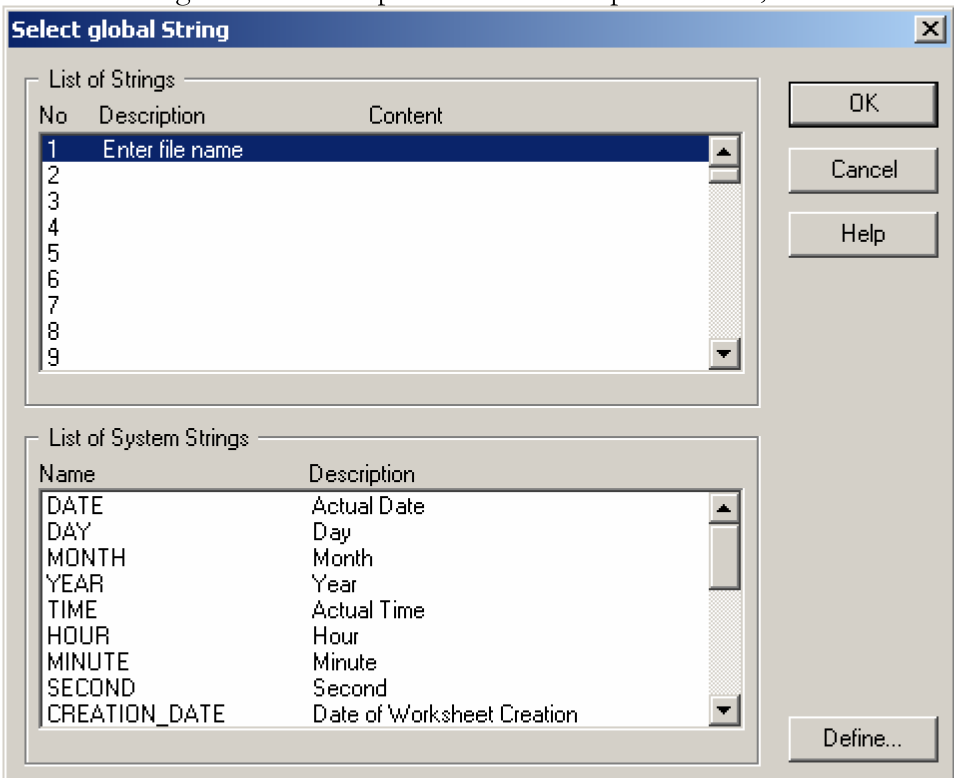


When we start the worksheet we should see a dialog box that looks as follows:



Once we enter a file name and hit start **DASY*Lab*** will start acquiring data and saving it to the hard drive with the file name we specified.

We can also use System strings in our filenames. Double click on the **Write Module** to modify its properties. Using the mouse, "right click" after the "}" and before the "." in the filename dialog area. This will produce a new drop down box, select "Global String…"



This new dialog box allows us to select from any of the 999 user strings as well as select from the available system strings.

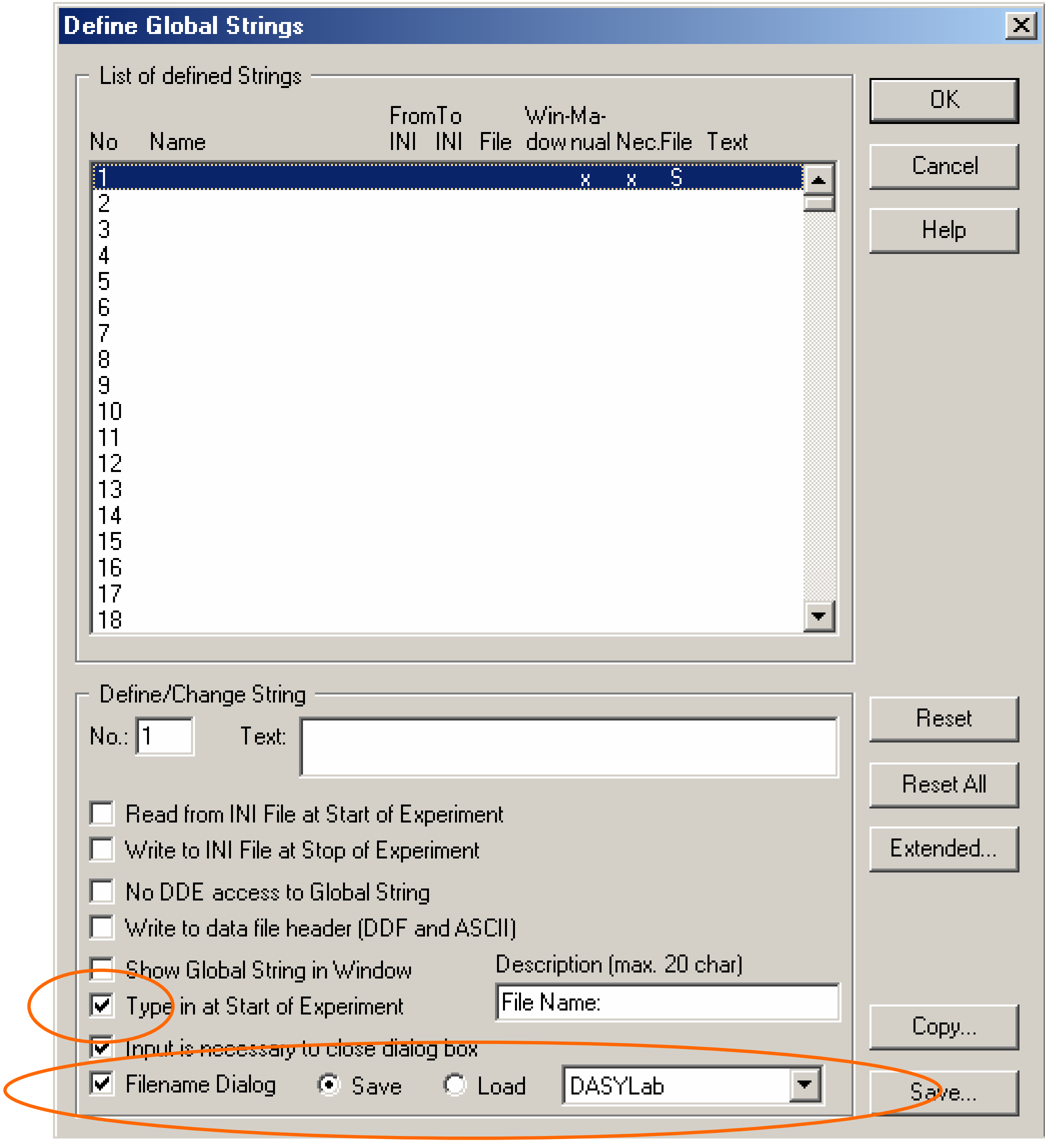
There are a few things to keep in mind when using system strings as file names. In the USA the common date convention is mm/dd/yyyy, however the "/" is used as a directory delimiters in windows, therefore using it as a file name is inappropriate.

However we can change the date format to mm-dd-yyyy in Windows, thus allowing us to use it as a part of the file name. The easiest way to get around this is to use the month string followed by a – then the day and so on.

In this example I will use the date as a part of the file name. When completed the file name should be:
C:\Program Files\DASYLab 60\Data\${str_1}${MONTH}-${DAY}-${YEAR}.DDF

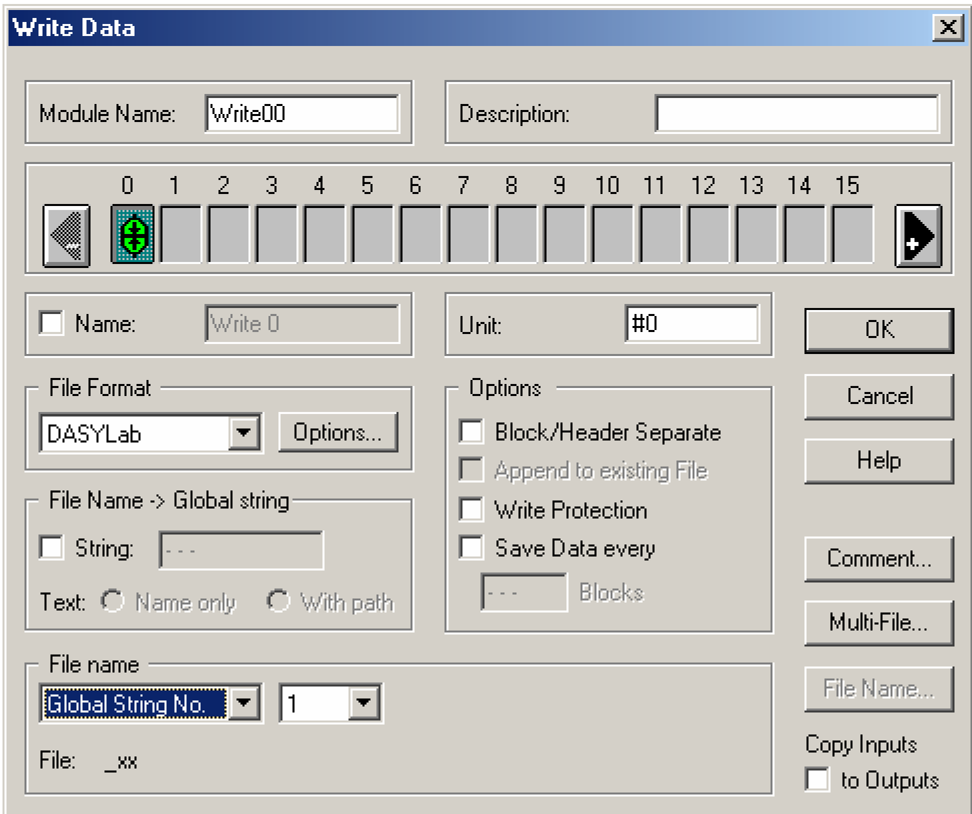
The last method of using a global string in a file name is to use the "File Name Dialog" option. To access this option click on the Options menu and select "Global Strings…". De-select all the check boxes for string 1 and the select the "File Name Dialog" check box.

There are a few options in reference to the "File Name Dialog" options. First is whether you want a load file or save file dialog box. We also must select whether which file format we are using. After selecting all the appropriate options our dialog box should look like this:

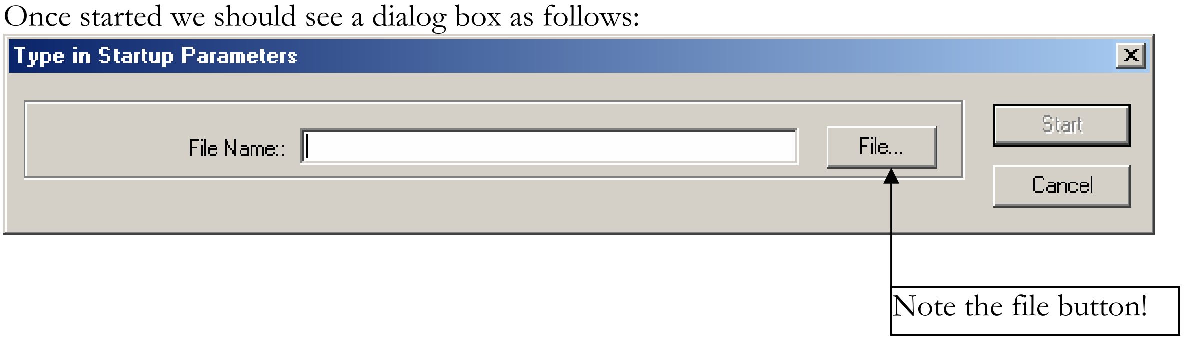


We must now make one change to the **Write Module** in order to take advantage of this option. Click ok to close this dialog box and double click on the **Write Module** to change its properties.

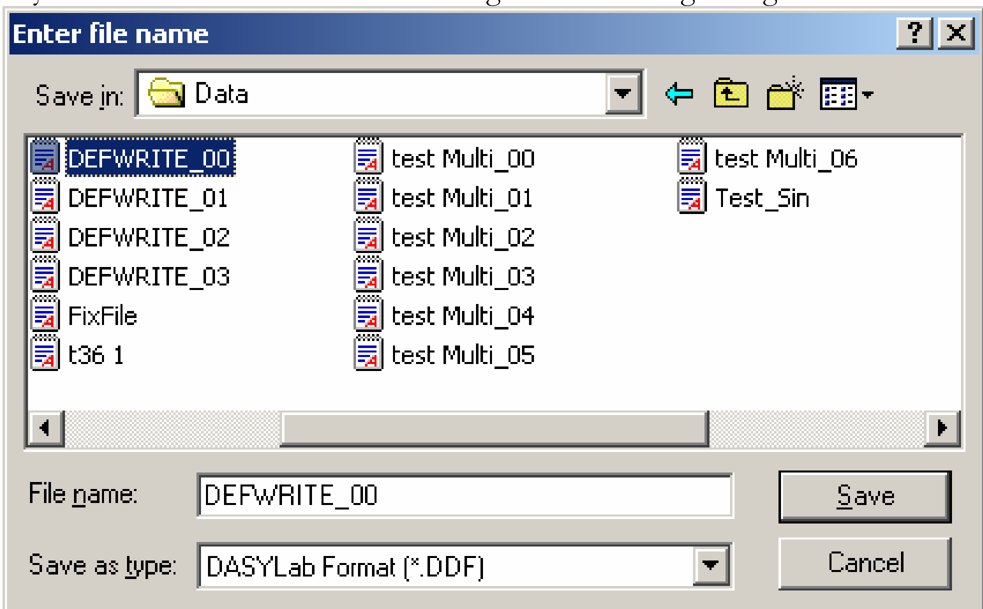
In the properties of the **Write Module** we can under "File Name" we select "Global String No.". We then get another drop down menu where we can select a number between 1 and 999. These numbers refer to our global strings; select number 1. Click "OK" and start the worksheet.



Once started we should see a dialog box as follows:



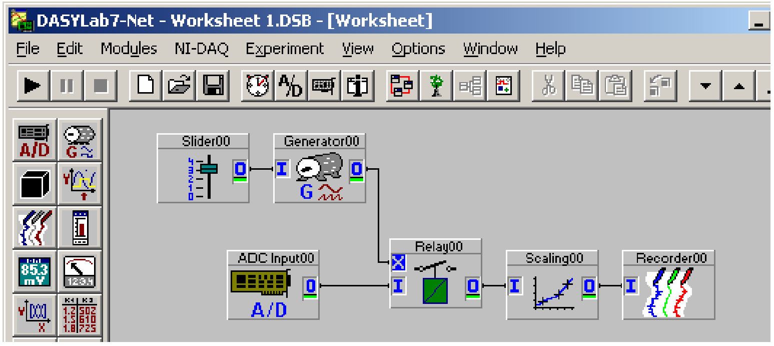
If you click on the "File…" button we get the following dialog box:



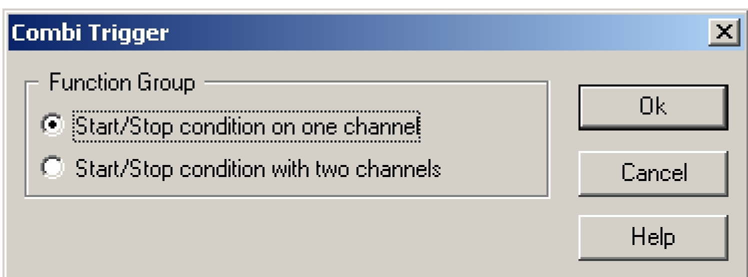
Here the user can enter a new file name or select an existing file name. When the user clicks "Save" the file name and path will be entered into the global string and used in the write module.

ADVANCED DATA REDUCTION

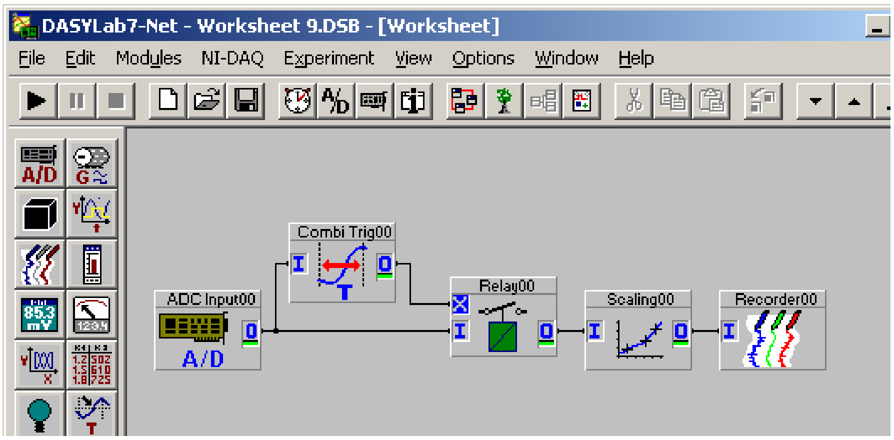
For some applications it is only important to capture an event. In these cases we care only about the event data and not the data before it or after it. For this application we will need the "Basic Data Reduction" worksheet, which looked as follows:



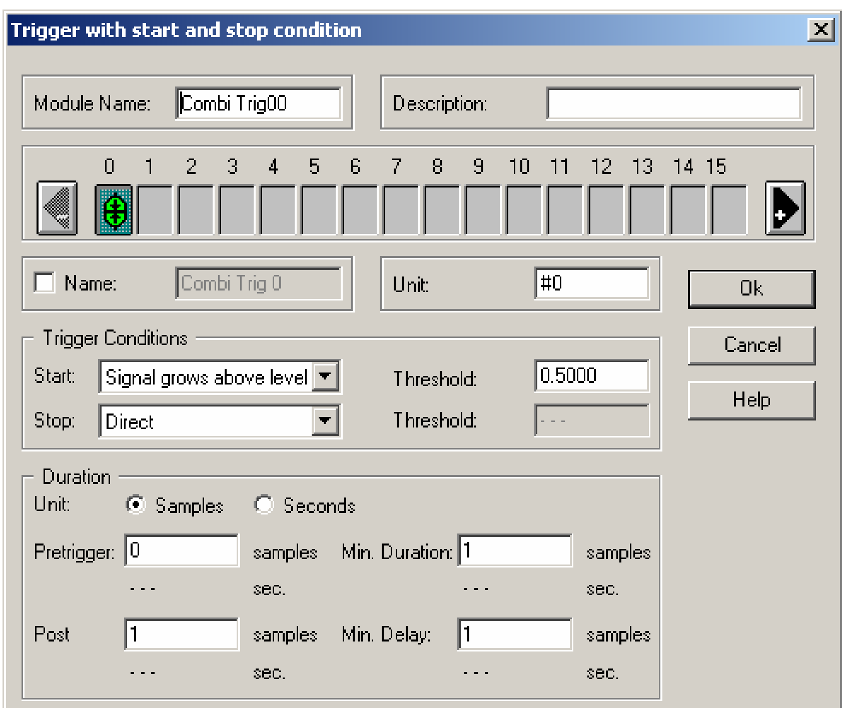
We will need to delete the **Slider00** and **Generator00** modules and replace them with a **Combi Trigger Module**. The **Combi Trigger** can be found under the Modules Menu, Triggers, Combi Trigger. Once placed on the worksheet we will need to select which type of Trigger we want "Start/Stop condition on once channel".



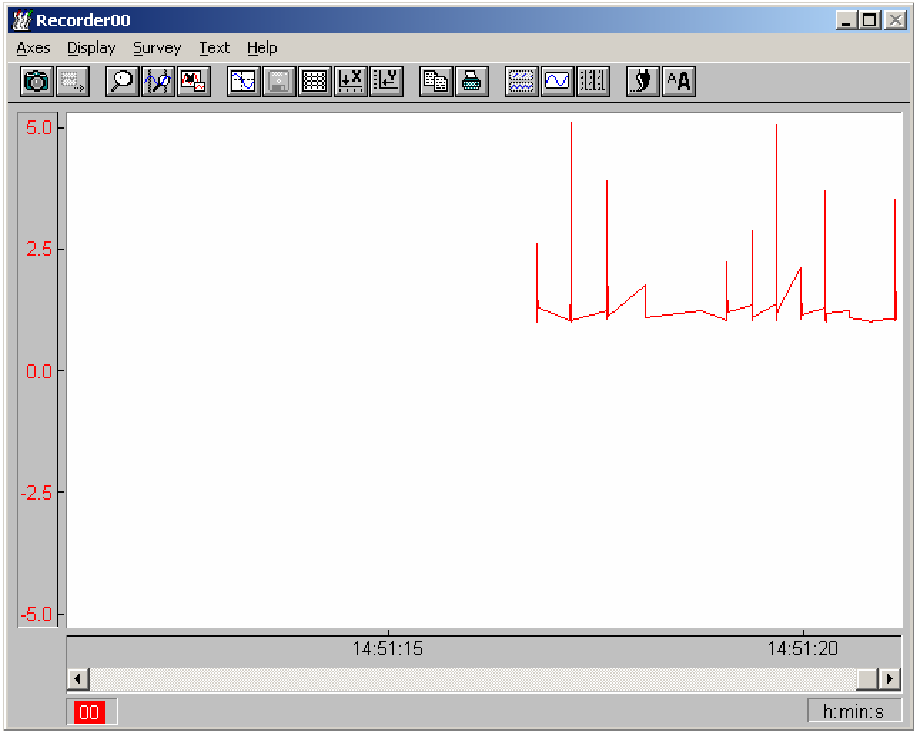
The **Combi Trigger** has both an input and an output. The trigger inputs data and based on user settings outputs either a high (5) or a low (0) indicating a true or false respectively as we discussed in "Advanced data logger with alarms". We will connect data from our analog input to the input of the trigger and the output to the control (X) input of our **Relay Module**. The completed worksheet should look as follows:



In this example I wish to view only the event of the data rising above 0.5. The properties of the **Combi Trigger** should look as follows:

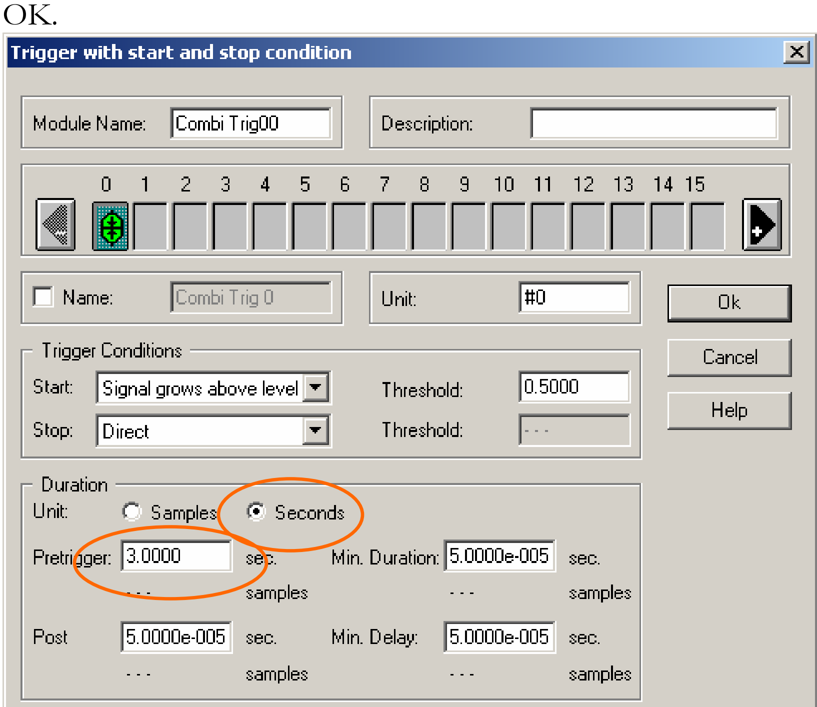


Now when the signal grows above lever 0.5 then the **Combi Trigger** will output a High, which will close the relay and record the event. When running you should see the value of any event that was greater then 0.5. The data my look like this:

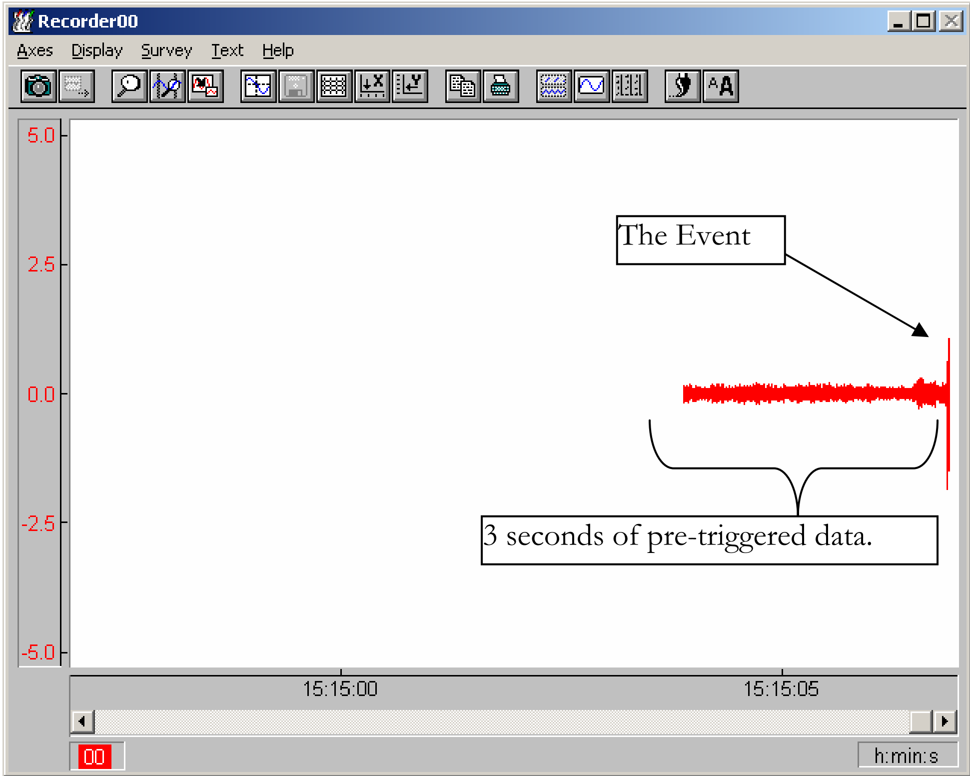


If we wanted to capture a set amount of data before the event we can specify a "Pre-Trigger" in the **Combi-Trigger** module. The Pre-Trigger allows you to "Look into the Future" and cause a trigger to occur before the event. **DASY*Lab*** accomplishes this by looking into the buffer to see the event, then trigger before that point in the buffer is read.

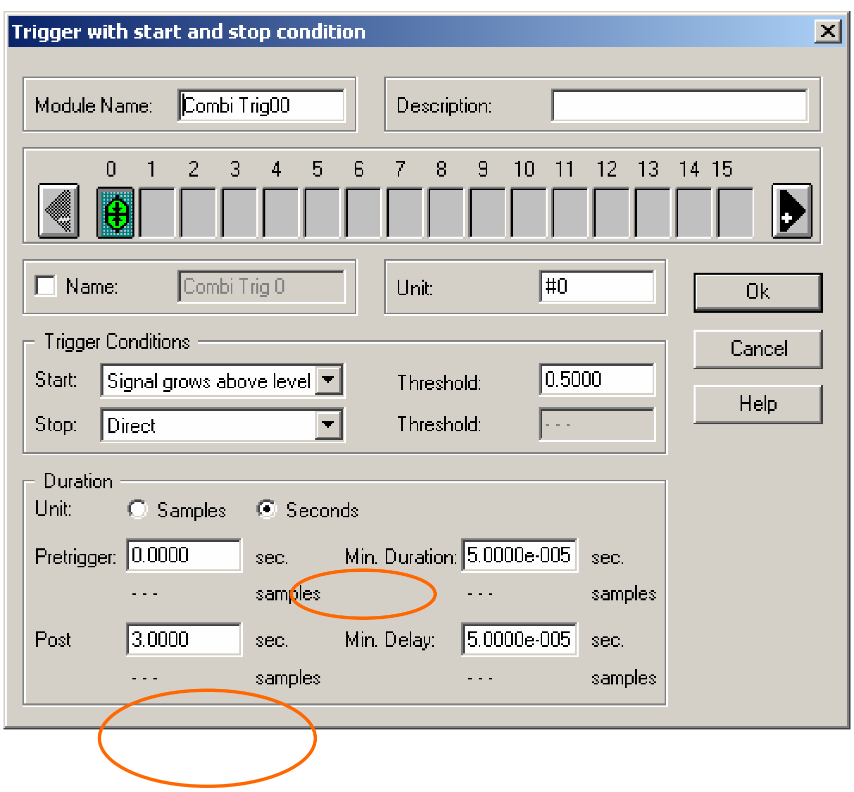
In this example I wish to capture the data 3 seconds before the event as well as the event. In the properties of the **Combi-Trigger** module I set the pre-trigger to 3 seconds and click OK.



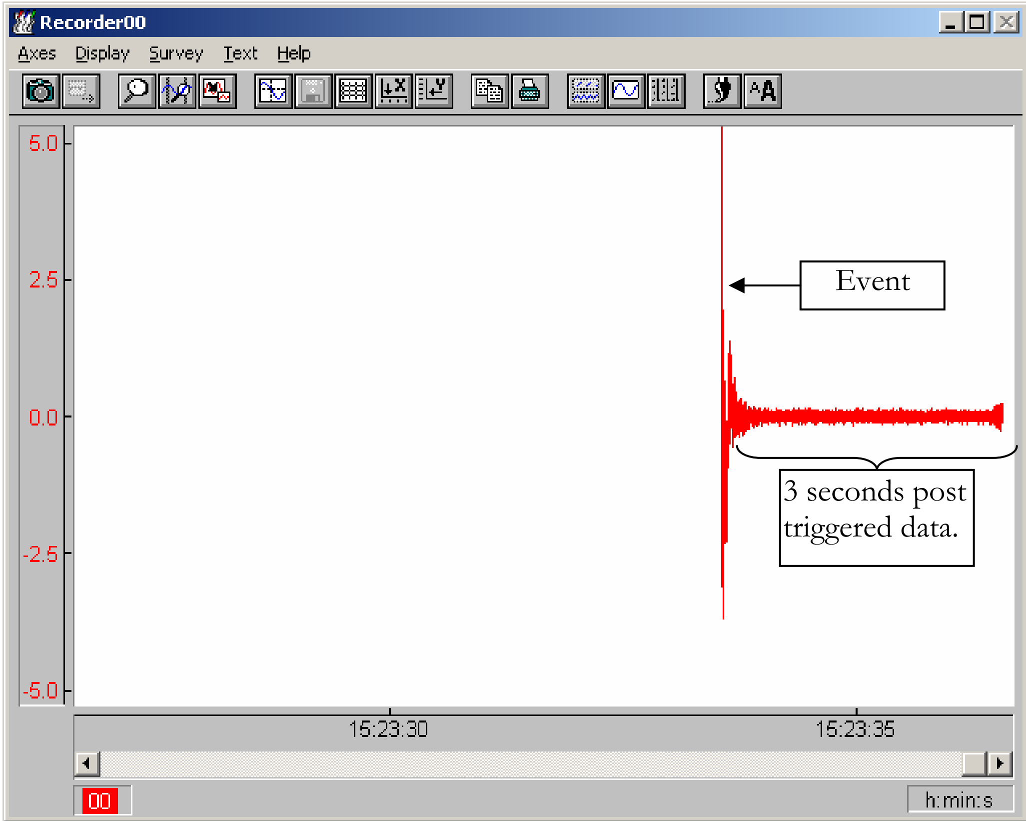
The resultant data will look as follows:



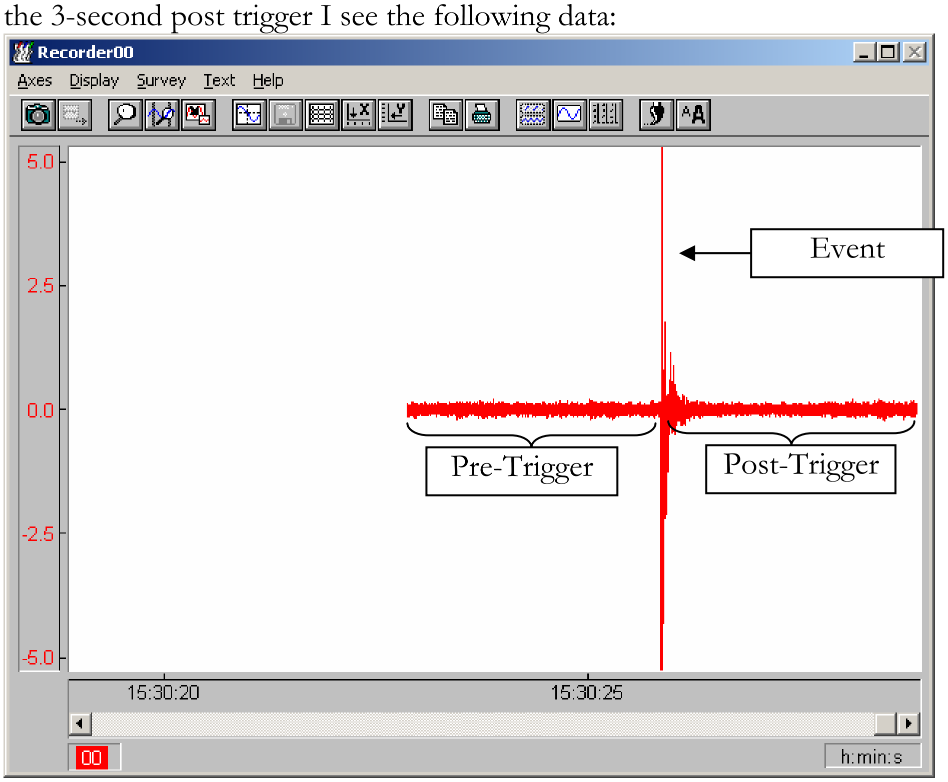
We can also look for an event and capture data for a pre-determined amount of time after. This is known as a "post trigger". For this type of application we would specify how many samples or seconds we wish to trigger after the event has occurred. For this example I will capture 3 seconds after the event. The settings for this trigger will look as follows:



The resultant data should look like this:



We can then combine pre and post triggered data to form a picture of what happened before and after an event. If I combine my settings for the 3-second pre-trigger and the settings for the 3-second post trigger I see the following data:

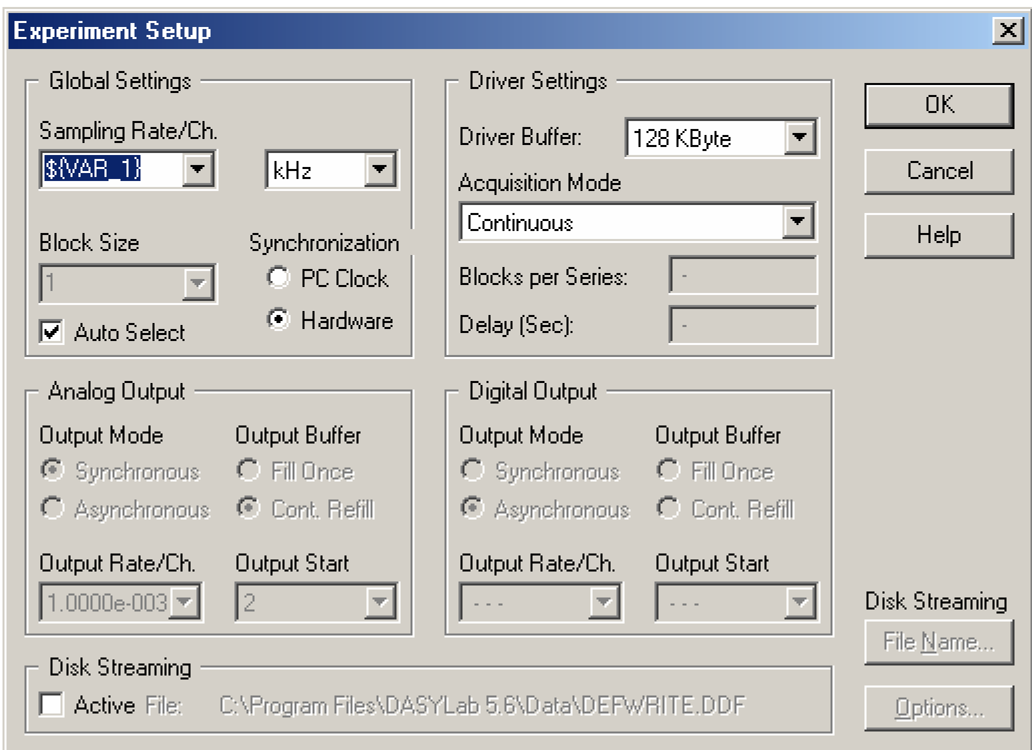


AUTOMATIC RATE VARYING DATA REDUCTION

In some applications we may want to acquire data at a slow rate, then based upon an event start acquiring at a faster rate. There are two ways to change the rate automatically. The first way is to change the speed of the worksheet or we can use the data reduction worksheet we created a while ago.

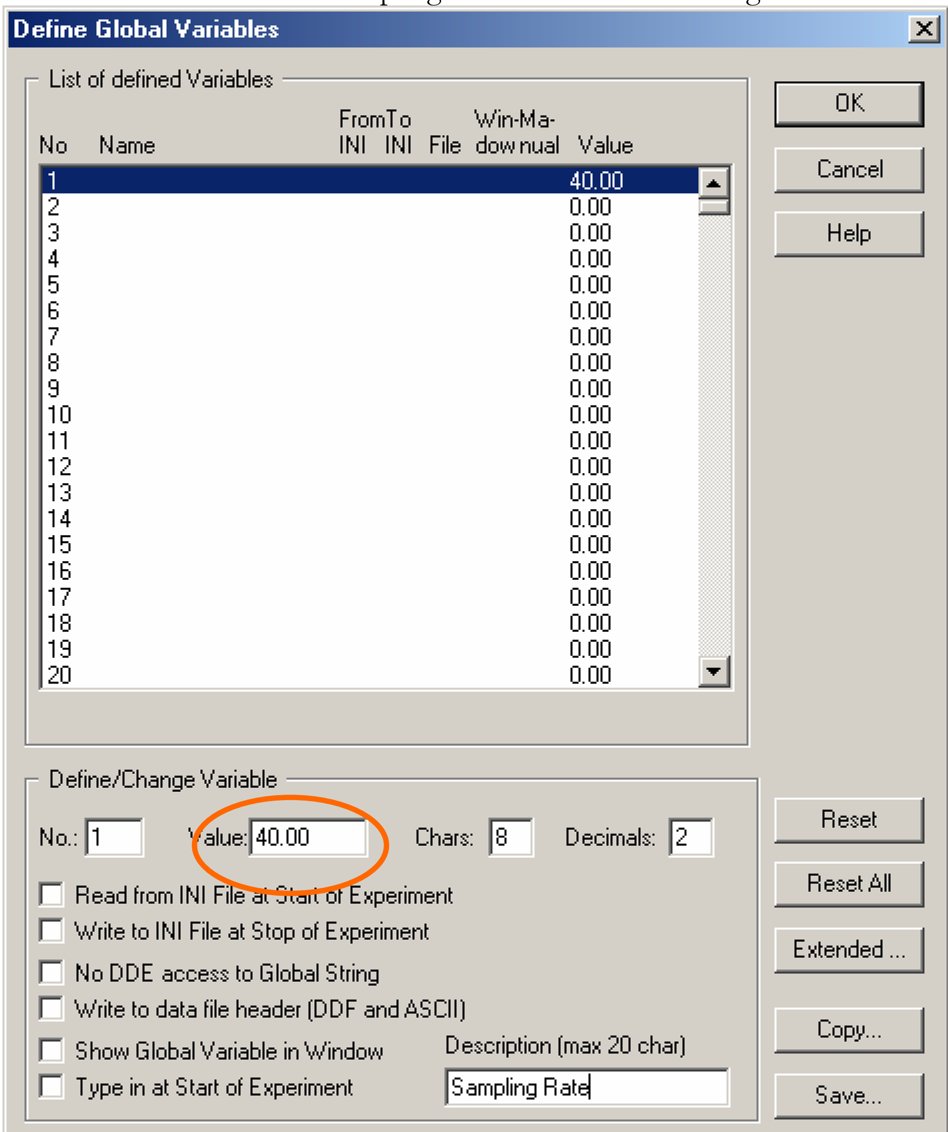
The global sampling rate of a worksheet cannot be changed during runtime. However we can use a global variable for the sampling rate, by changing the variable then stopping and starting the worksheet we can effectively change the sampling rate. The down side of this is that there is a short time where no data was acquired; therefore this should not be used for fast changes. Also when the worksheet stops and restarts all the data that is currently on the screen is lost. If a **Write** Module is being used the "Append to File" option should be selected as to not have the file over written when the worksheet restarts.

First access the Experiment Setup window found under the Experiment menu. Set the "Sampling Rate" to Global Variable 1 (or any other global variable you choose). The settings should look like this:

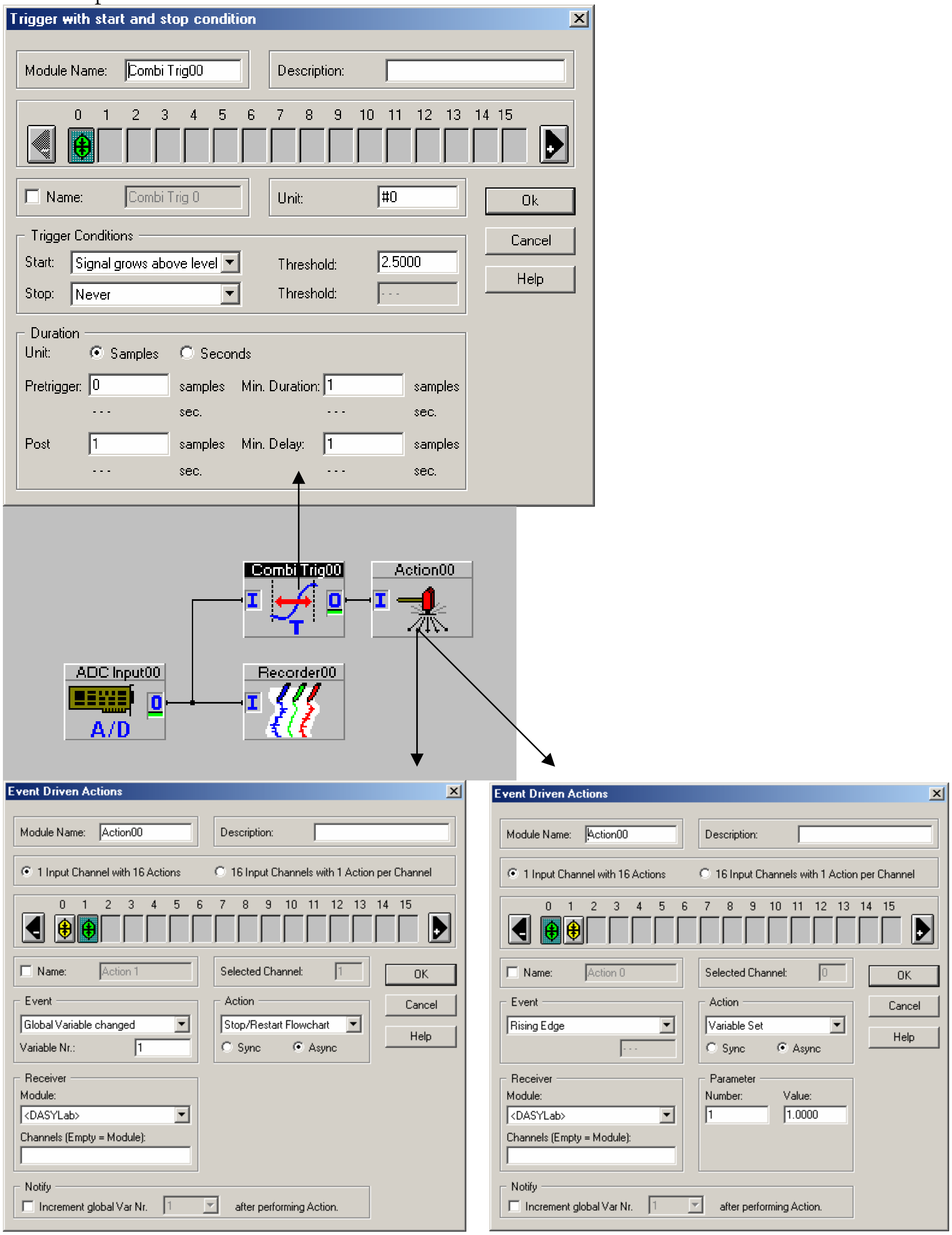


Notice how the unit is kHz, any value I enter into the global variable will be interrupted as kHz.

Next we have to enter in our starting value. Click on Options, Define Global Variables. Set the Value of Variable 1 to sampling rate we want. The setting should look like:



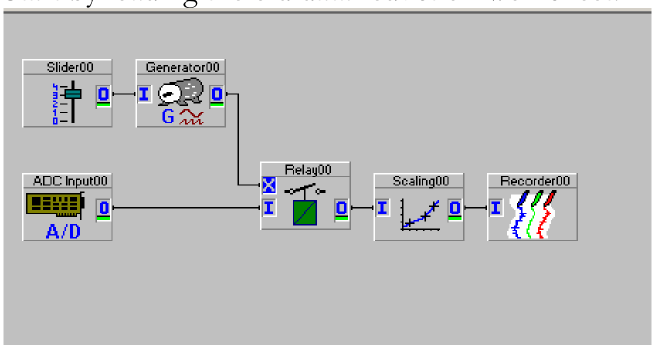
The completed worksheet should look like this:



This setting will trigger when the signal grows above 2.5 a rising edge is created by the Combi-Trigger. The Action Module receives the rising edge and channel 0 changes the variable to the new rate, Channel 1 is triggered by the change in the variable, and causes the worksheet to stop and start again.

The other method is a little more complex, however it can change the effective sampling rate without stopping the worksheet. We will start this worksheet by loading an earlier worksheet. We had created an application earlier where by use of a slider we could change the sampling rate (Basic data reduction). In that example we used a **Slider Generator** to control the frequency of a **Generator Module** to create a pulse train. That pulse train closed a relay to allow one data point to pass the desired rate. We will now automate that process.

Start by loading the old data reduction worksheet.



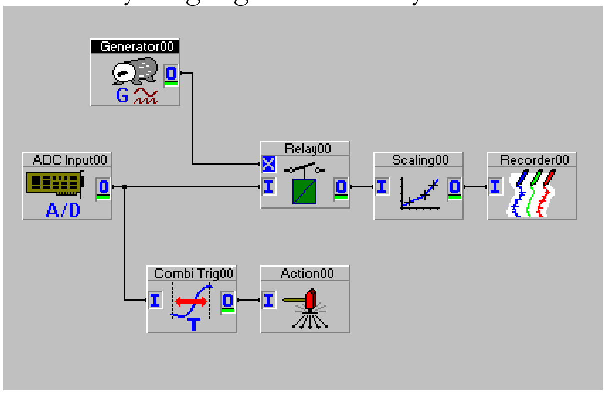
Delete the slider and the generator modules.

We will now add an **Action Module**, **Combi Trigger,** and a **Generator Module** to the worksheet. The end result will be a worksheet that will acquire data at one rate and when an event occurs sample faster for a period of time. For this example we will sample at 1000 Hz and increate to 40000 Hz on an event. Keep in mind that the Global acquisition rate must be at least 40000 Hz for this to work appropriately.

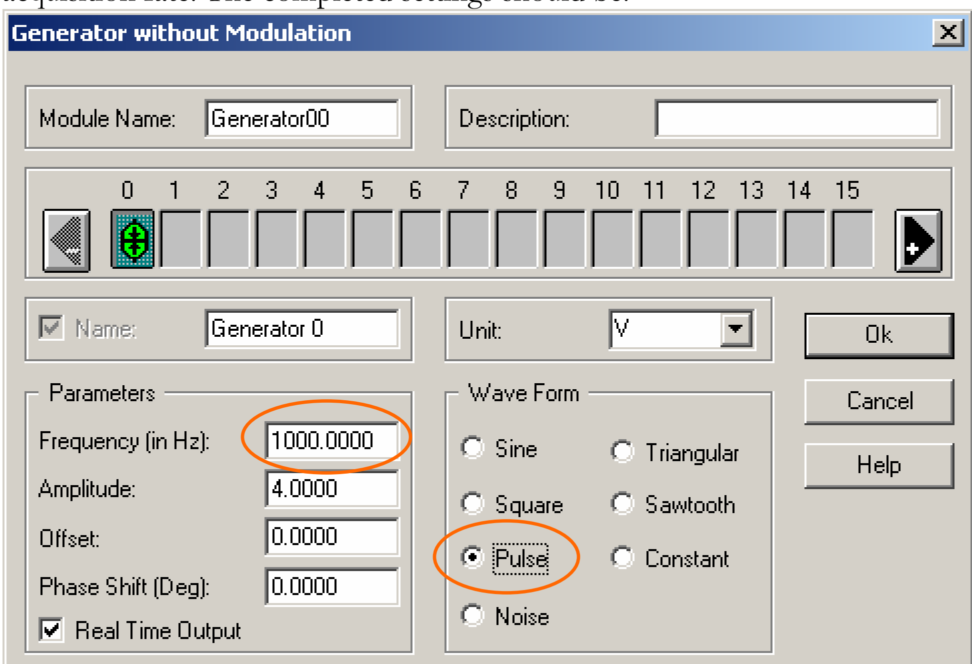
First we add the **Combi Trigger**, we want the "Start/Stop Condition on one channel" type. For this example I wish to start acquiring faster when the data exceeds 0.5. I would therefore configure the **Combi Trigger** to have a *Start condition* of "Signal Grows Above level" and a threshold of 0.5, the rest of the settings may remain default for the time being.

We will now place an **Action Module** on the worksheet. The **Action Module** is located under Modules, Special, Action. Lastly we must place a **Generator Module**, select the "Without Modulation" option.

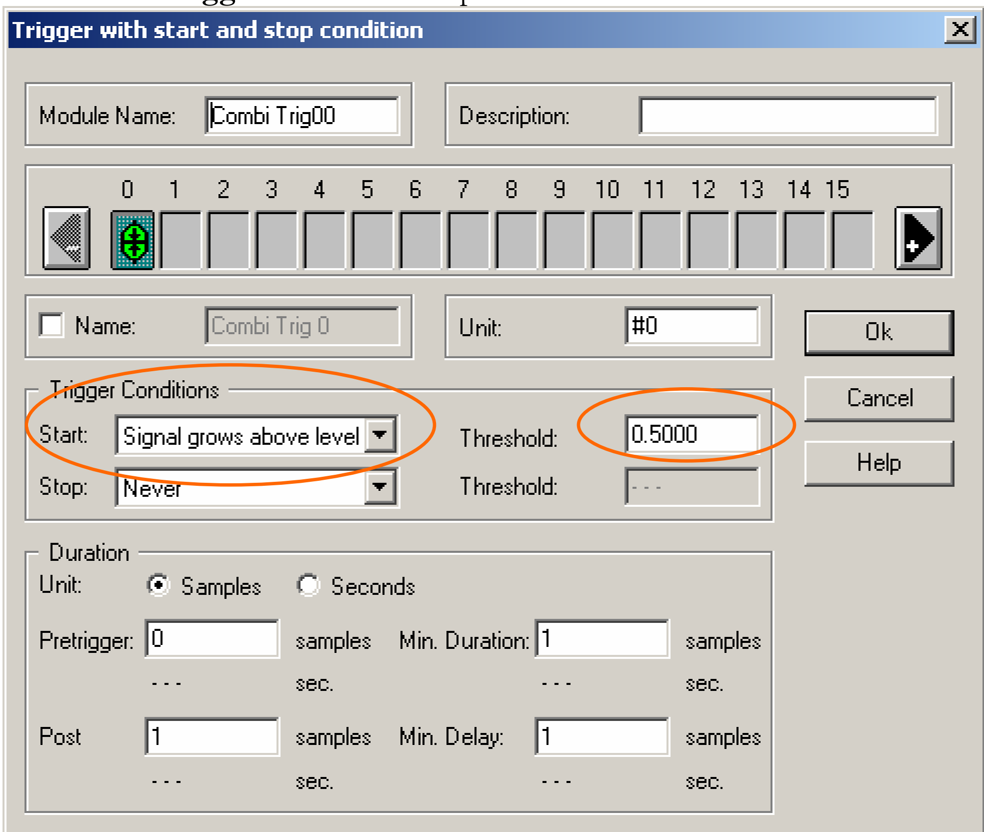
Wire everything together so that your worksheet looks as follows:



We should now configure the **Generator Module** to output a pulse train at our slow acquisition rate. The completed settings should be:

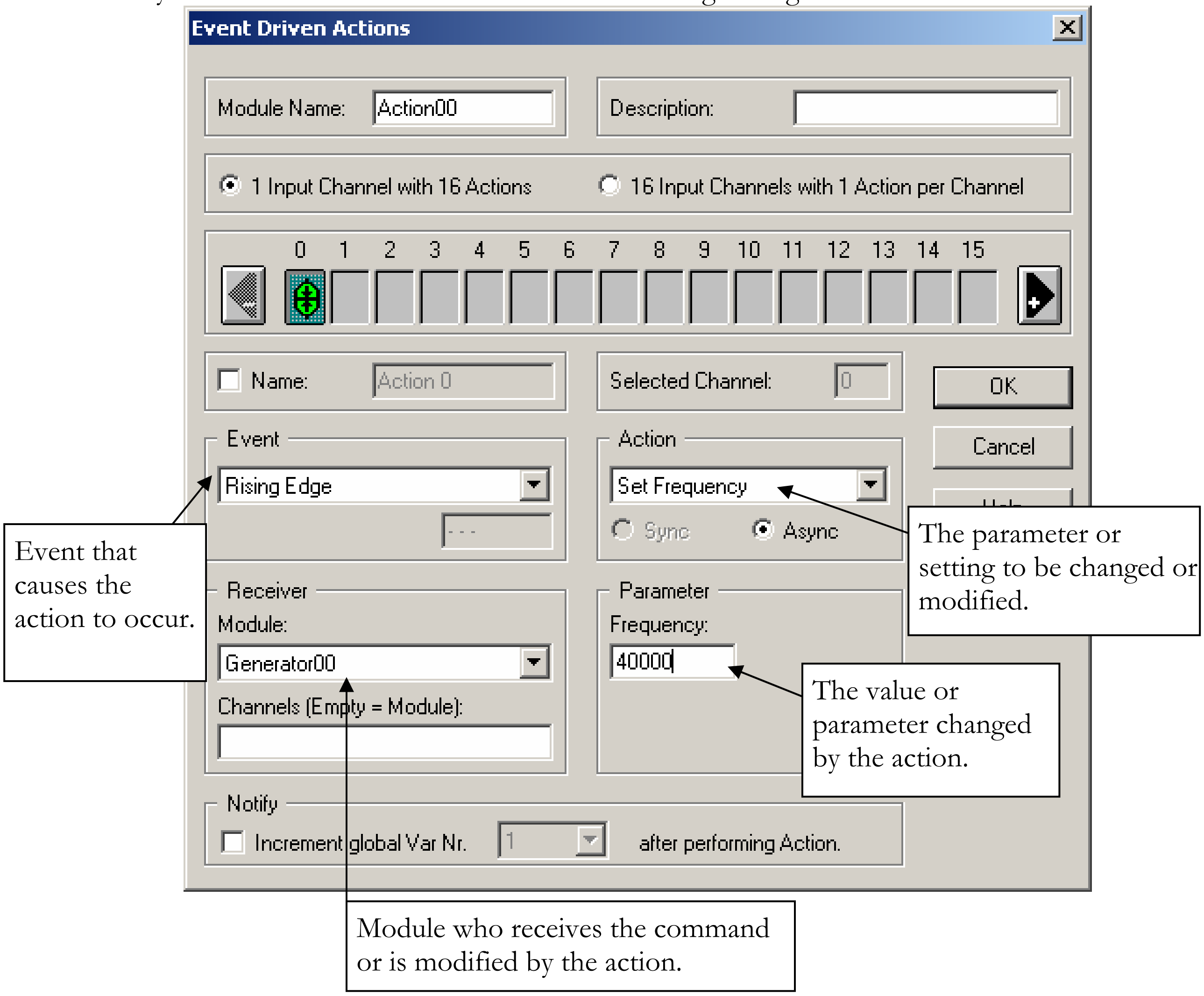


The **Combi Trigger** should be set-up as follows:



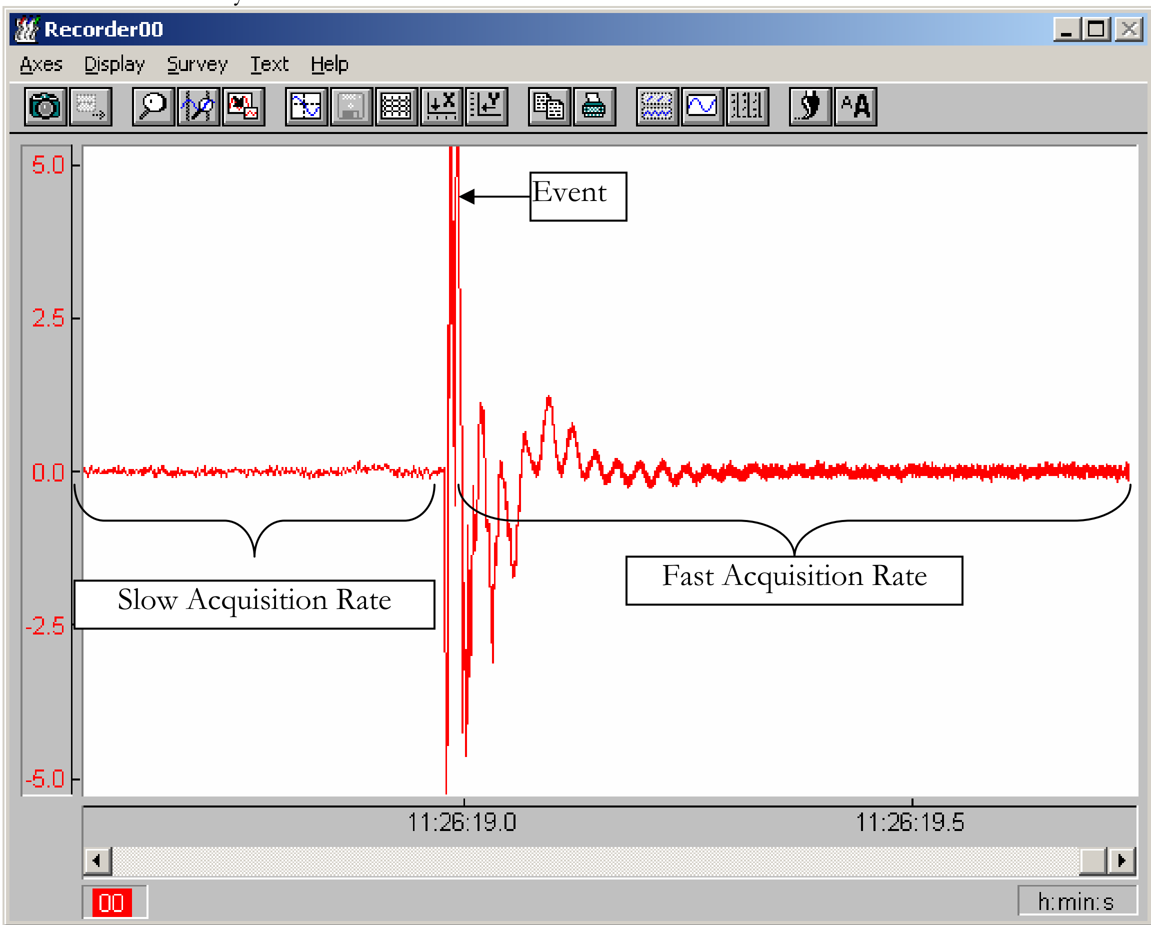
This trigger will output a High signal when the input grows above 0.5. At this point in time the trigger will **NOT** reset and will stay high until we stop the worksheet. We could add a stop condition or a pre and post trigger to this, however we will keep this simple.

Lastly the **Action Module** should have the following settings:



Here is what is happening here; when the signal grows about 0.5 the **Combi Trigger** will go from a Low to a High generating a "Rising Edge". The **Action Module** will see this rising edge and will modify the **Generator Module** Generator00 and "Set Frequency" to 40000. The **Generator Module** will then go from generating a pulse train at 1000 Hz to a pulse train at 40000 Hz.

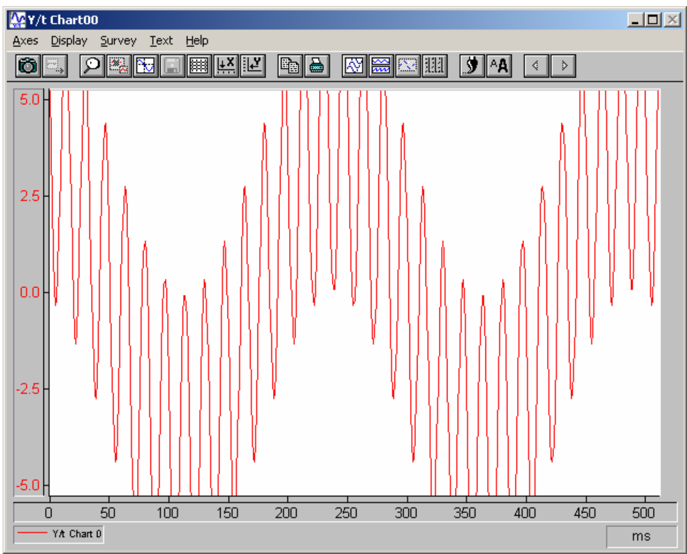
The result data may look like this:



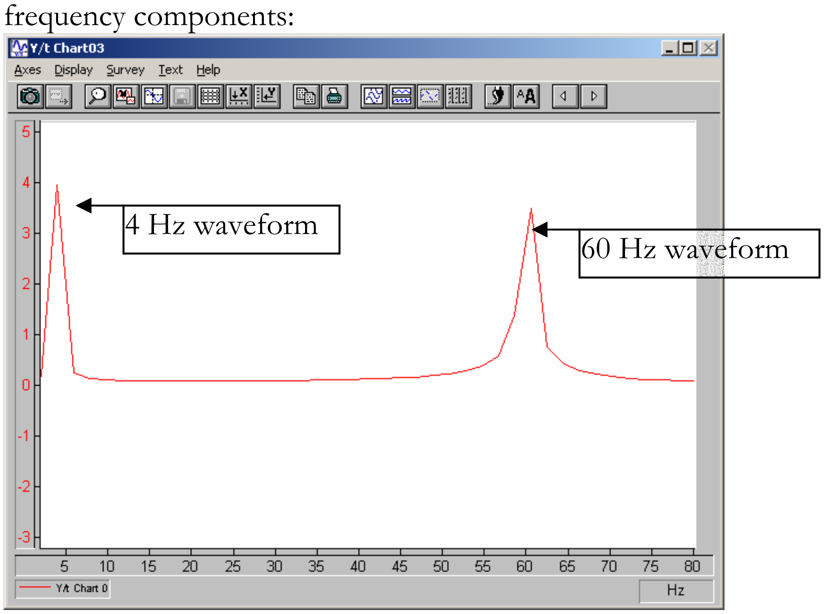
FILTERING

Filtering is very important when there are frequencies greater then our sampling rate or if there is simply noise in our signal. As we discussed, the Nyquist theorem that if we do not sample at least twice as fast as our waveform we get an alias wave form that may not exist in our signal. However if we are monitoring a 4 Hz waveform and there is a 60 Hz noise component we would like to remove the 60 Hz components to have a clean signal without any interference.

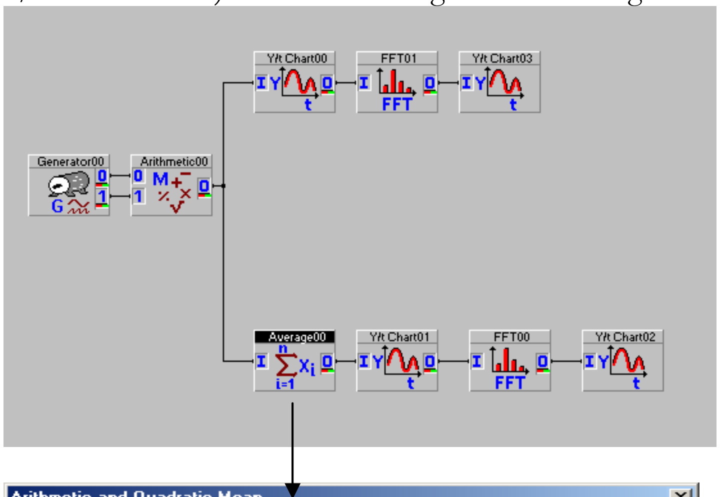
For example, here is a waveform with a 4 Hz signal and an additional 60 Hz signal added to it:

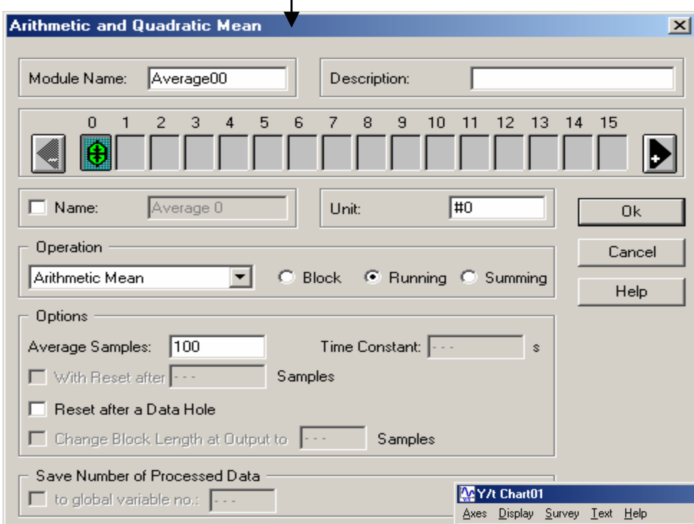


To further illustrate the point I have included an FFT of the waveform to show the frequency components:

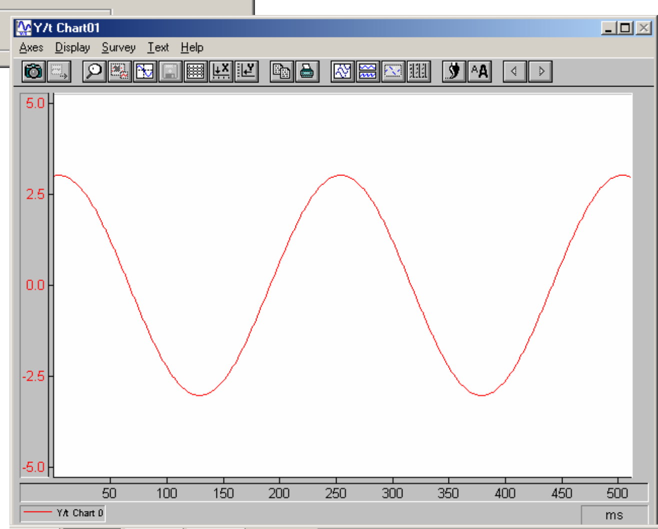


There are two ways to remove the noise from these signals, the first is to use averaging via the **Average** Module. By adding the **Average** Module and performing a "Running" Average of 100 Samples (our sampling rate in 1000 Hz in this example, therefore we are averaging 1/10 of a second). The resultant signal is a clean sign wave. The worksheet looks as follows:



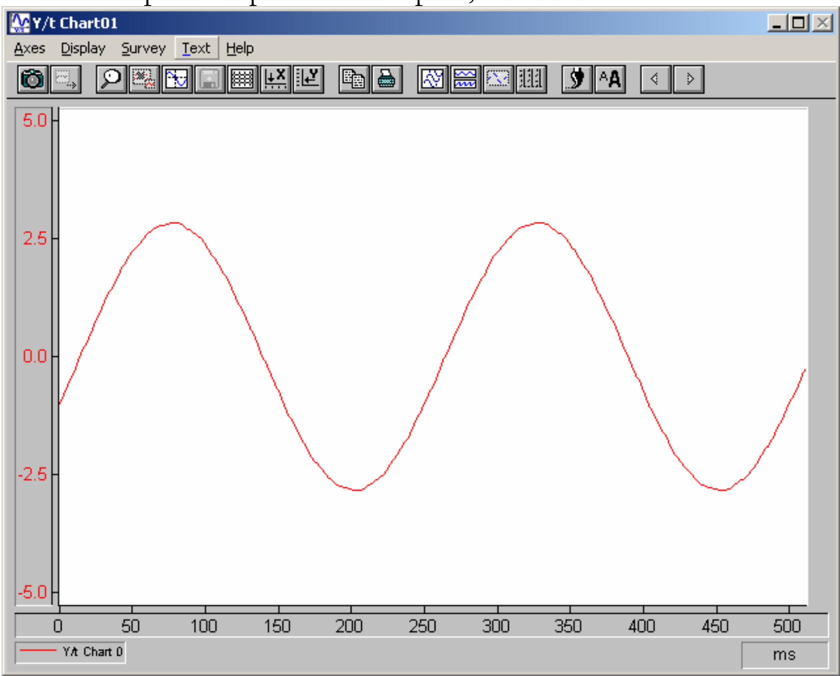


The processed signal looks like:

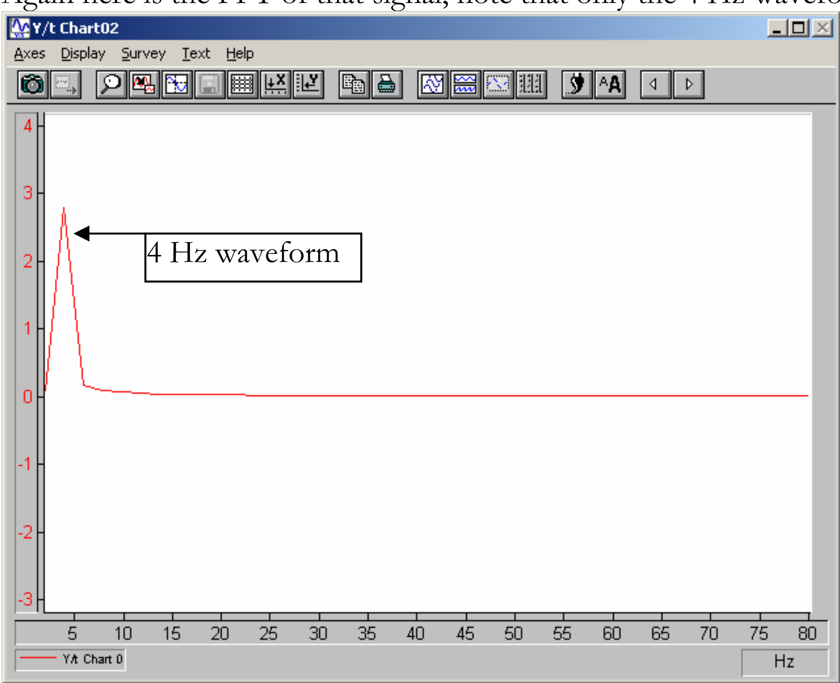


The other method of cleaning out noise is by applying a filter to the data. Using a filter we can, for example, remove every component above 4 Hz. Keep in mind that a filter cut off frequency is not exact and is more of a slope and starts cutting off at 4Hz.

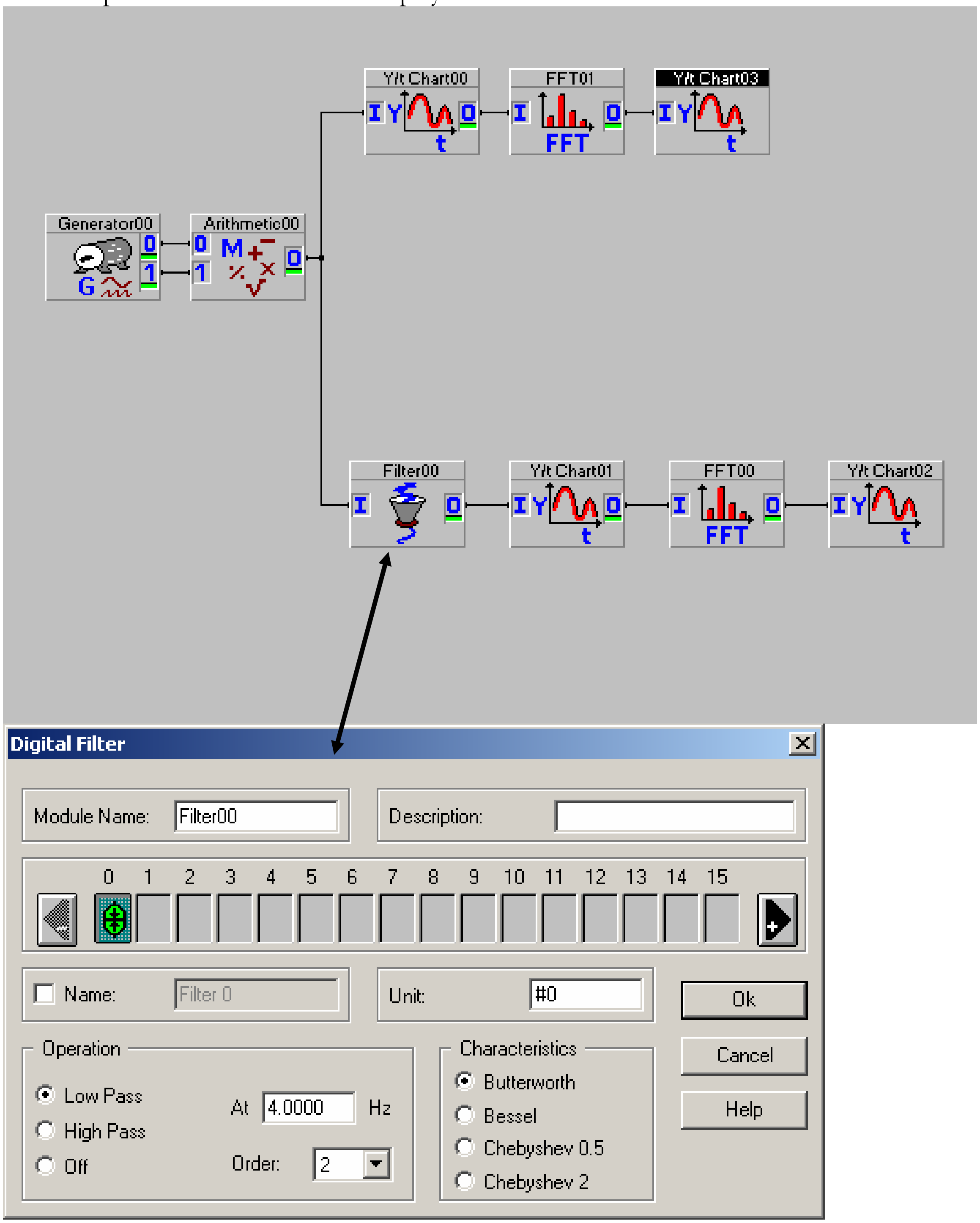
The result of filtering our signal with a "Low Pass" filter, which allows only frequencies under our specified parameter to pass, of 4 Hz look like this:



Again here is the FFT of that signal, note that only the 4 Hz waveform remains:



The implementation of a filter is quite simple. To use a filter insert a **Filter Module** into your worksheet so that the data enters the filter before it is displayed or calculated upon. In this example I have shown how to display filtered and non-filtered data:



CHAPTER 6: COMMUNICATING WITH OTHER PROGRAMS

WHAT IS DASYLAB NET?

DASY*Lab* NET is an additional add-on to the **DASY*Lab*** PLUS package. **DASY*Lab*** NET allows **DASY*Lab*** to communicate to other **DASY*Lab*** software programs with the NET ability.

DASY*Lab* NET offers advantages over other network based communication protocols, mainly it includes the transmission of “time stamped data”. Due to the fact that as data packets are sent through the Internet they may arrive at the receiving computer at different times. This may cause the data to appear asynchronously sampled and inaccurate. With the transmission of the time stamp **DASY*Lab*** can interpret this data and display as well as calculate based on the time stamp and ensure accurate data calculations and display.

DASY*Lab* Net has the ability to communicate “Coupled” or “Uncoupled”. In coupled mode the receiving **DASY*Lab*** sends a confirmation each time it receives a data packet. This enables the sending **DASY*Lab*** to resend un-received data packets as well as know the status of the receiving computer. The downfall is that if the sending or receiving computers stop then the data acquisition stops. In uncoupled mode if the sending receiving computer fails to receive then the sending computer will continue acquiring data. When the receiving computer is available again then it can reestablish the communication with the serving computer.

In order to create a **DASY*Lab*** NET communication you must have two copies of **DASY*Lab*** NET or **DASY*Lab*** NET RUNTIME, these must have independent serial numbers, and a TCP/IP connection between them. You must also be able to “ping” the two computers. To ping a computer, Click on the “START” button and select “Run…” and type “command”, click OK, at the prompt on the “DOS SCREEN” type “ping <<IP ADDRESS>>” (replace << IP ADDRESS>> with the IP address of the other computer) and press enter.

If the computers are communicating correctly then the results should look like this:

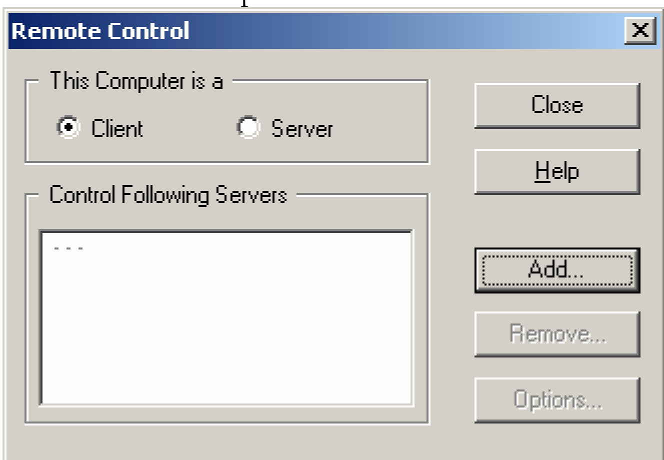
```
Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-1999.

C:\>ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:

Reply from 192.168.0.1: bytes=32 time<10ms TTL=64
Reply from 192.168.0.1: bytes=32 time<10ms TTL=64
Reply from 192.168.0.1: bytes=32 time<10ms TTL=64
Reply from 192.168.0.1: bytes=32 time<10ms TTL=64

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms

C:\>
```

If the computers are not communicating then the result may look like this:

```
Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-1999.

C:\>ping 192.168.0.21

Pinging 192.168.0.21 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.21:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms

C:\>
```

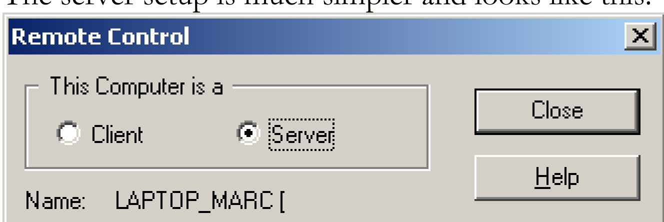
Once we have two computers communicating then we can setup and communicate via **DASY*Lab*** NET. First we will configure the serving computer, this is the computer with the data that we wish to transmit.

One computer must be selected to be the server and the other as the client. To select these options click on Experiment, Remote Control… A client must specify which server it will use. The client setup windows look as follows:

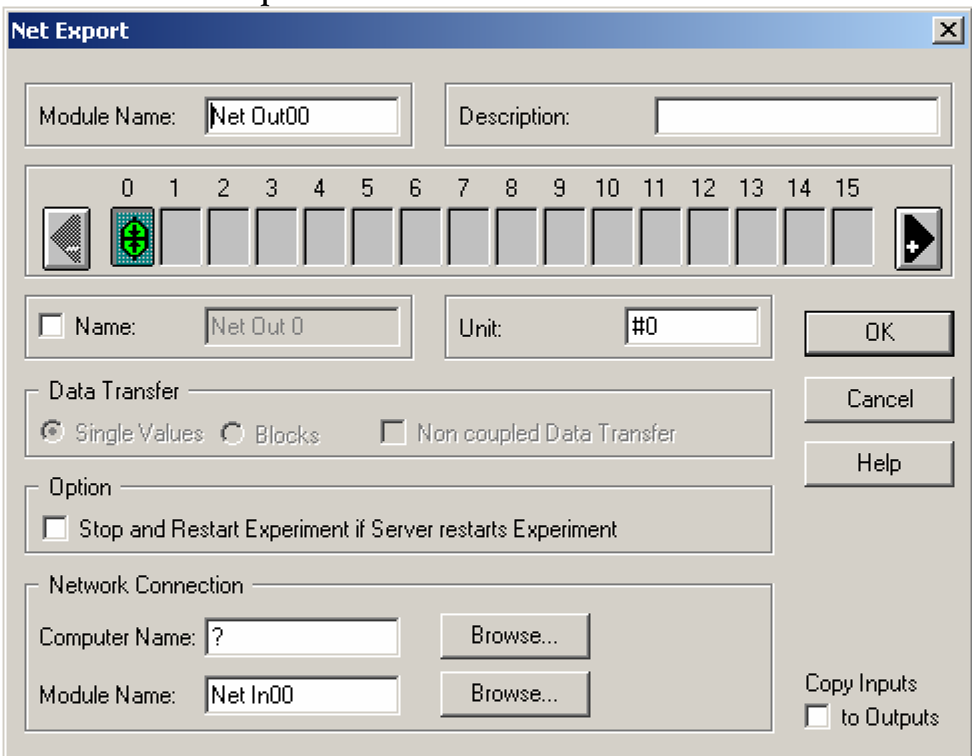


By clicking on the "Add…" button servers may be added to the list.

The server setup is much simpler and looks like this:



Place a **Net Export Module** on the worksheet, found under Modules, Network, Net Export. This module takes data much the same as our display modules, however as opposed to sending data to the screen it sends it via a network to another computer. The properties menu of the **Net Export Module** looks like this:



Computer Name:
Must be the name of the PC on a local area network or the IP address of the computer if over the Internet. The “Browse…” button only works to find computer on a local area network.

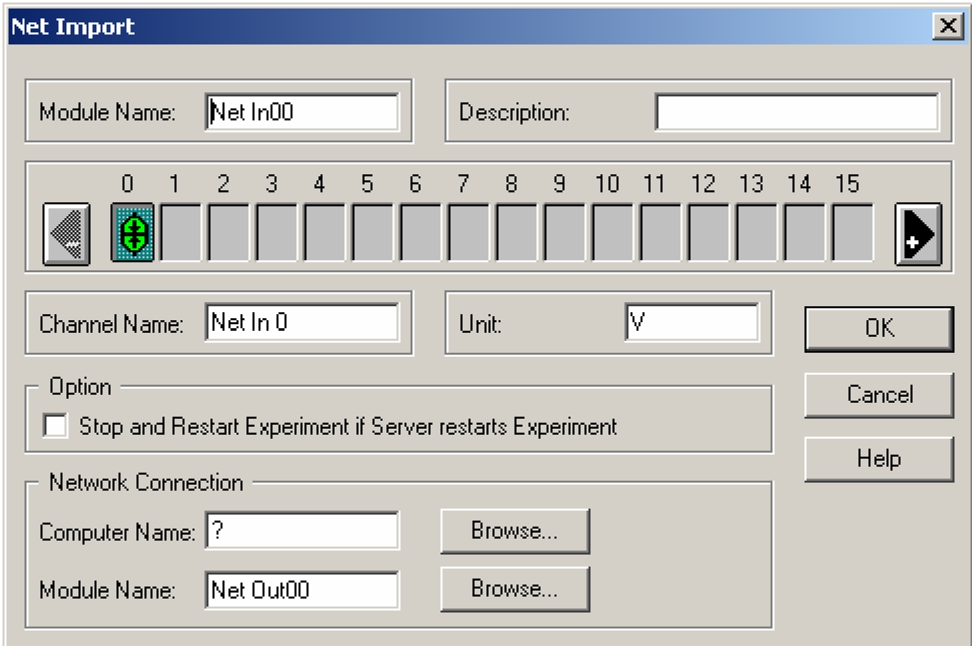
Module Name:
This is the name of the module to receive the data from the **Net Export Module.**

Data Transfer:
You can send “Single Values” or “Blocks”. Blocks take up more bandwidth however are more processor efficient. Single Values “stream” the data however causes the processor to work harder.

Non Coupled Data Transfer:
When checked the server will send the data regardless of the status of the other computer. This will also allow for faster communication.

Stop and Restart Experiment if Server restarts Experiment:
If checked when the server restarts so will the client.

In order to receive the data on the other computer place a **Net Import Module** on the worksheet. The properties of this module will look like this:



Computer Name:
Must be the name of the PC on a local area network or the IP address of the computer if over the Internet. The "Browse…" button only works to find computer on a local area network.

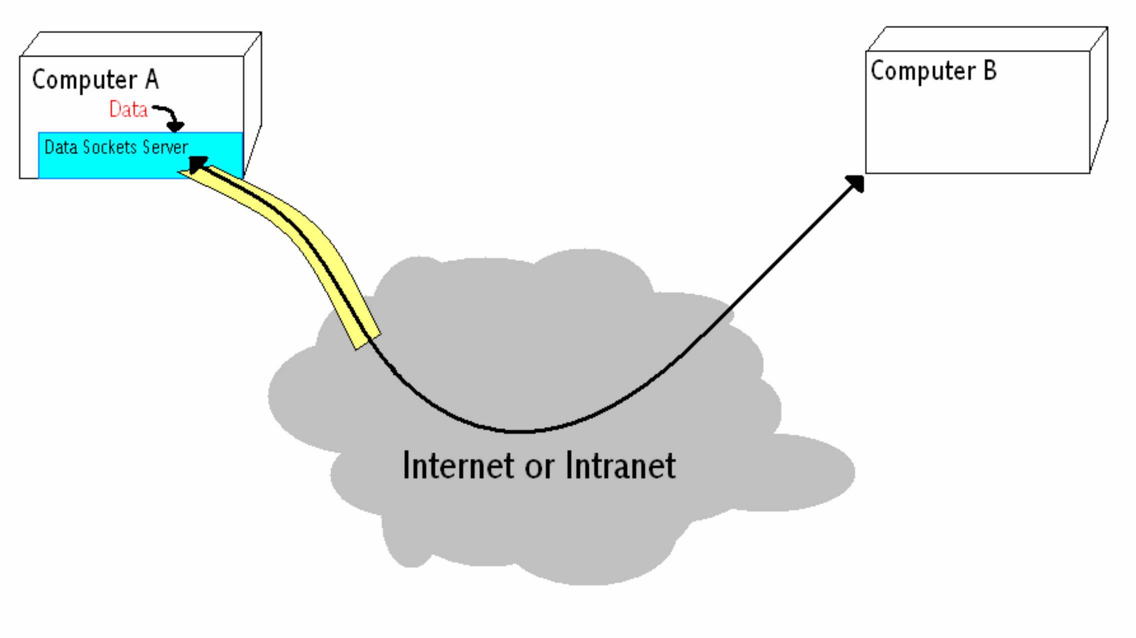
Module Name:
This is the name of the module to send the data from the **Net Import Module.**

Stop and Restart Experiment if Server restarts Experiment:
If checked when the server restarts so will the client.

Once configured and running the worksheet should now be sending data from **DASY*Lab*** to **DASY*Lab***.

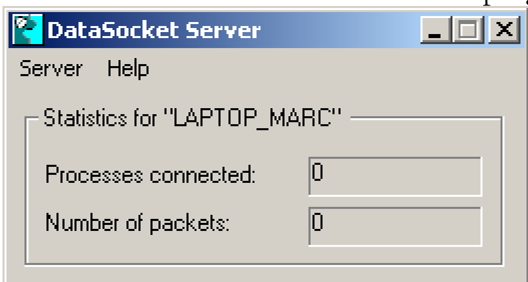
WHAT IS DATA SOCKETS?

Data Sockets protocol was developed by National Instruments as a connectionless TCP/IP protocol for sharing information over the Internet as well as an Intranet. Data Sockets is capable of sending and receiving data from any Data Sockets enabled device, including **DASY*Lab***, Lab View and Component Works enabled Web pages. We will be communicating via **DASY*Lab*** to **DASY*Lab*** communication.



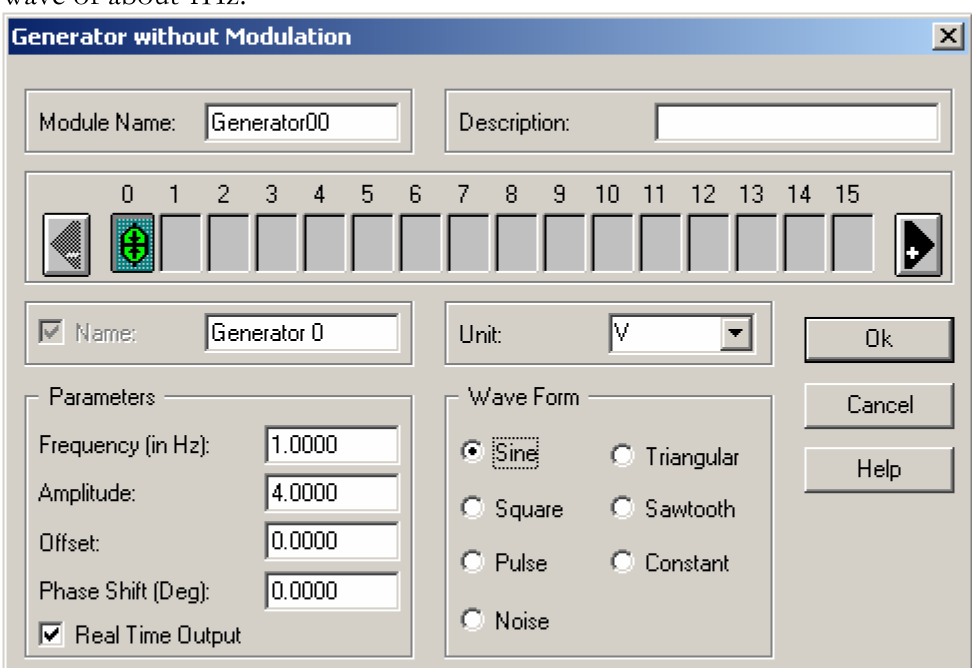
In this diagram Computer A is running the data sockets server and **DASY*Lab***, Computer B is running **DASY*Lab***. Both computer have access to the Internet and have known IP addresses. Note that the Data Sockets Server can be running on either Computer A or Computer B, however for this example the server will be running on the computer acquiring the data.

A Data Sockets server was installed with your version of **DASY*Lab*** 5.5 and later. Typically this program is located in the Start Menu, Programs, National Instruments Data Sockets, Data Socket Server. When we run this program we get a window as follows:

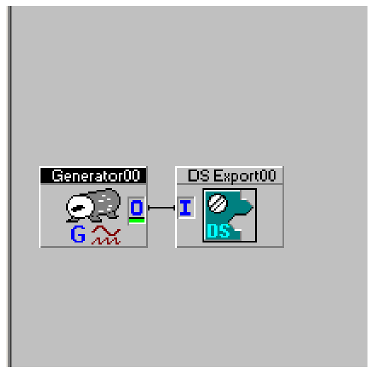


The server is now running. There should be an icon in the "Tray" next to the clock also. This is all the preparation needed to start the Data Sockets server. We can now start **DASY*Lab*** and create our sending worksheet.

For this example I will send a slow sine wave over the Internet to another copy of **DASY*Lab***. Therefore I will create a Generator Module and connect it to a Data Sockets Export Module. The Generator will be set to "Without Modulation" and produce a sine wave of about 1Hz.

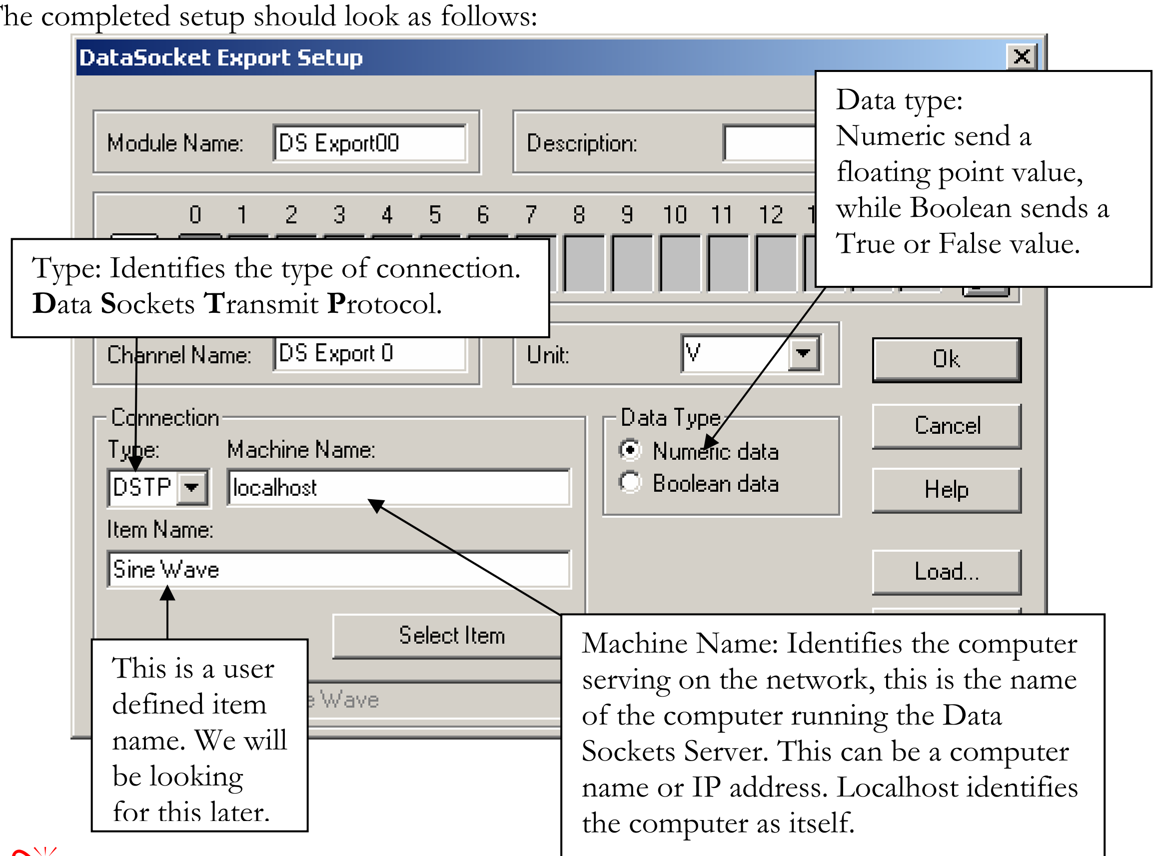


This will be connected directly to a Data Sockets Export Module, which is located under Modules, Network, Data Sockets Export. Once connected the worksheet should look like this:



There are very few changes to be made to the Data Sockets Export Module and most of them optional. We can add names to the data channels to make them easier to identify on the receiving computer. I have chosen to name our data channel "Sine Wave", logically.

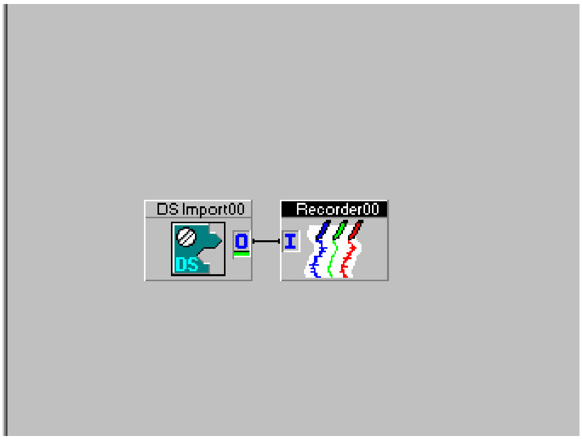
The completed setup should look as follows:



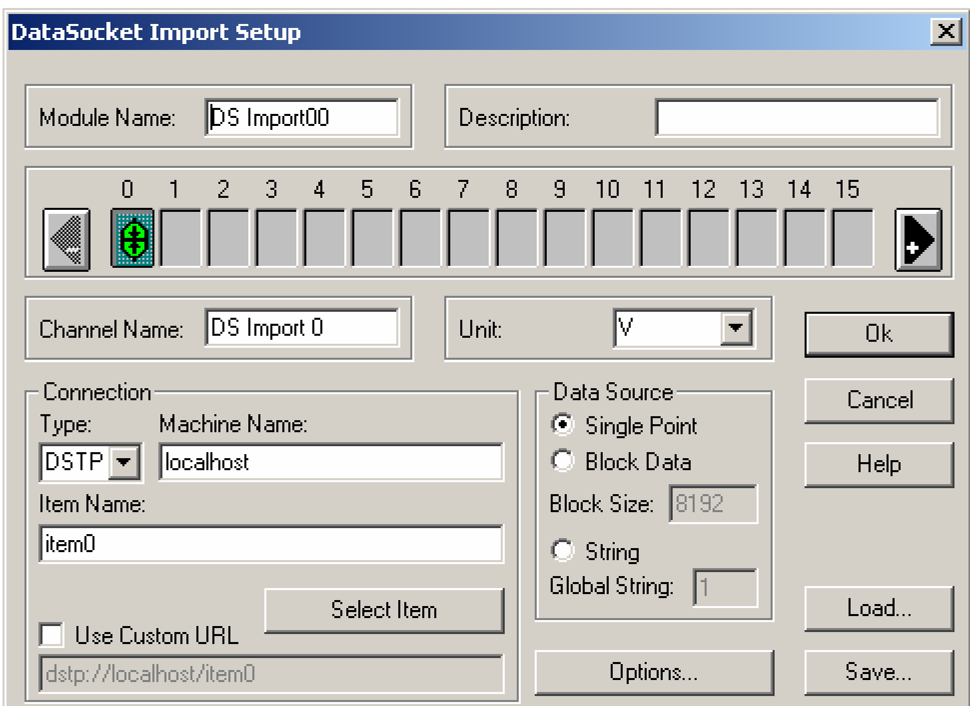
NOTE: Check and record the block size on the serving worksheet, under Experiment Setup. We will need this information in the receiving worksheet.

We can now start this worksheet and move on to create the receiving worksheet. We will need to perform this on another computer, due to the fact that **DASY*Lab*** can not be run more than once on a computer system.

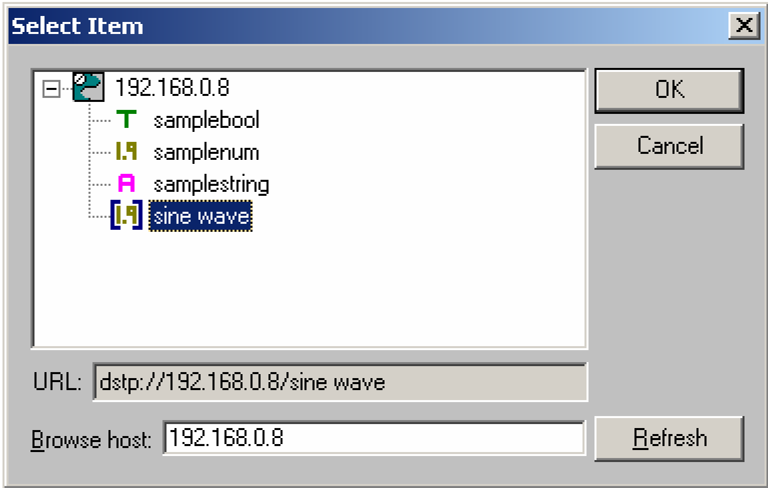
Open a blank worksheet and place a Data Sockets Import Module located in the same menu as the Data Sockets Export Module. Connect a Chart Recorder Module to this module. The completed worksheet should look as follows:



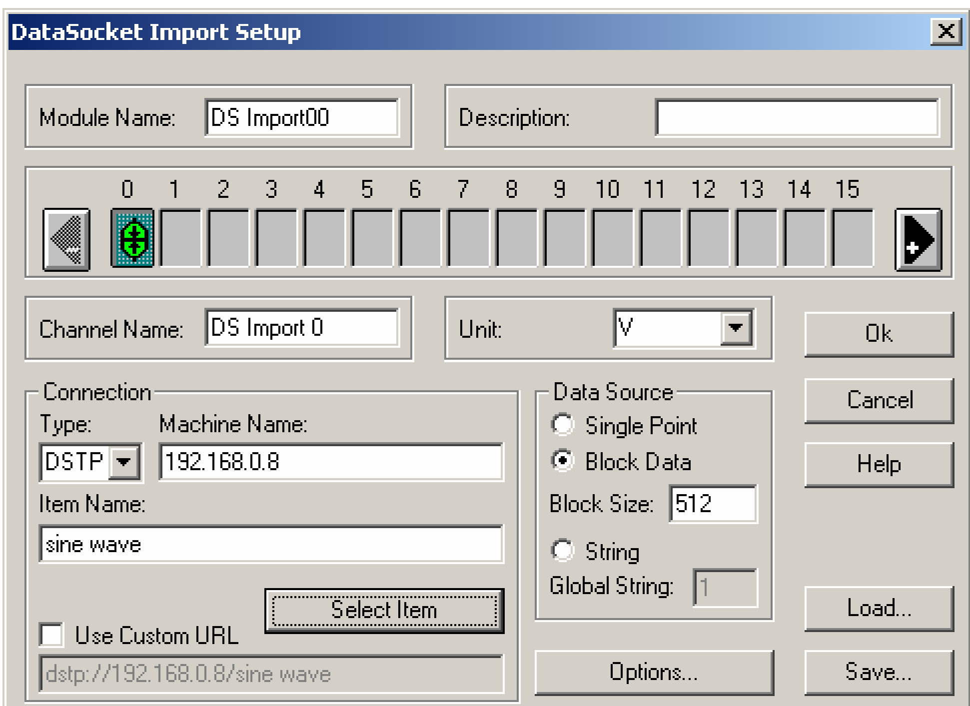
We have a few properties to set inside the Data Sockets Import Module. Double click on the Data Sockets Import Module to bring up its properties. The properties windows should look like this:



Under Machine Name enter the IP address or computer name of the computer running the Data Sockets Server. With this information entered we can click on the "Select Item" button to retrieve a list of data available on the Data Sockets Server.



Notice our "sine wave" data is available. Double Click on "sine wave" or select it and click OK. Once last change to make is to set the "Data Source" to "Block Data" and enter the block size from the serving computer. When Completed the Data Sockets Import Module should look as follows:

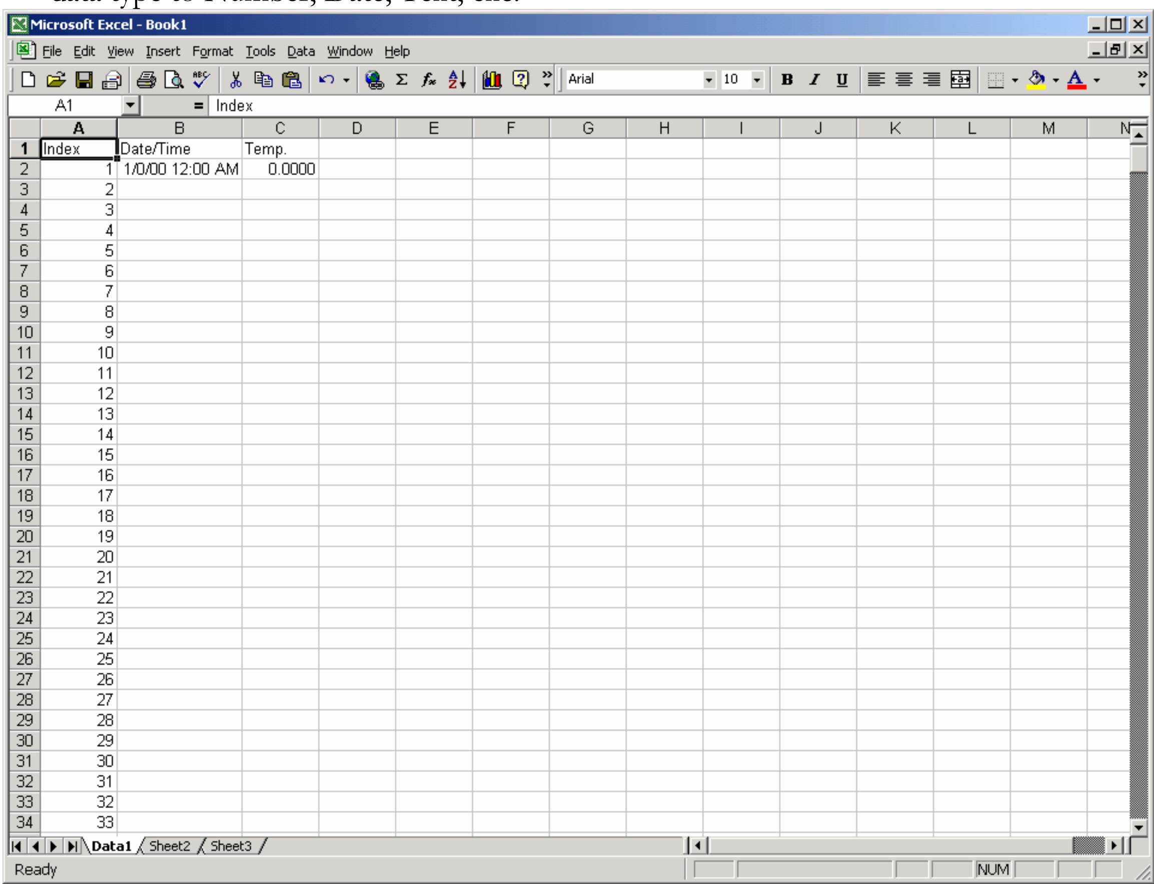


We are now able to bring up our display and start our worksheet. You should be able to see a 1Hz sine wave being displayed on the **Chart Recorder Module.** If need be more channels can be added to the Export Module and Import Module to send and receive more channels.

ODBC SETUP

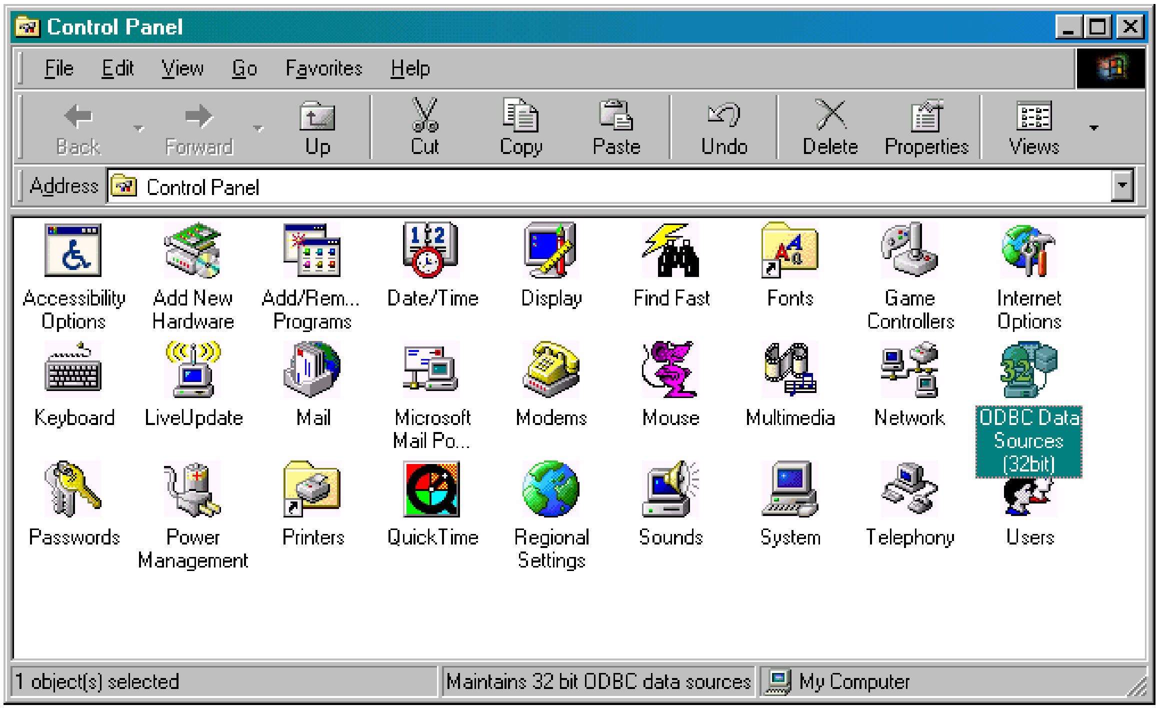
ODBC Input and Output can be used to receive and send global variables and strings to databases. In addition to products like MS Access, you can use MS Excel if you have installed the ODBC manager software available with most Microsoft products.

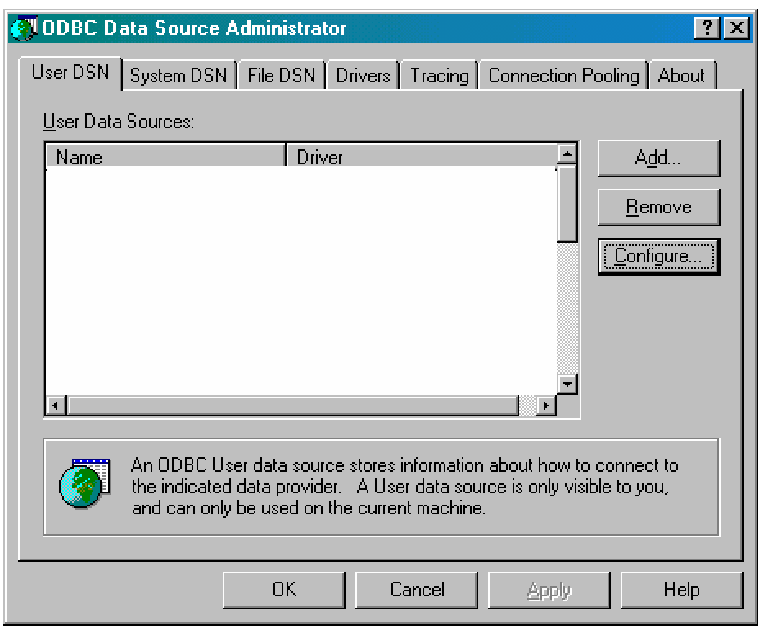
1) Create an Excel worksheet
2) Write click on the Name Tags to create a new name for the sheets you wish to send the data to. (Fig 1)
3) Enter "Titles" at the top of each column you wish to place data into.
4) One Column should be titled Index, Count, or some other such indicator.
5) Fill the new index column with numbers. We will need these numbers to direct **DASY*Lab*** on where to place the data.
6) Now is also a good time to place a value in the first row of each column and set the data type to Number, Date, Text, exc.



7) Save and Close the File.

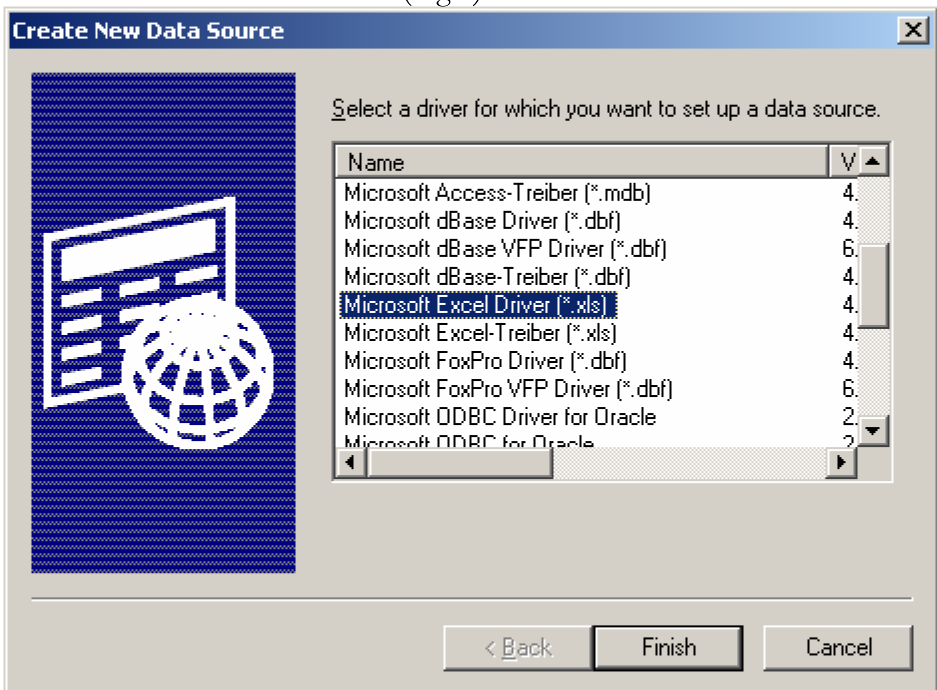
8) Enter the Windows Control Panel; open the Data Sources (ODBC).



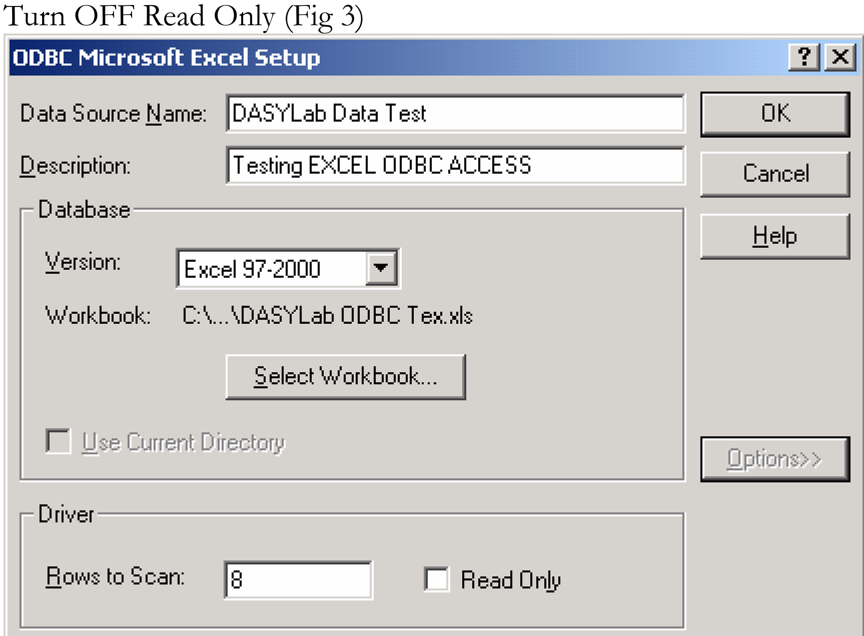


9) Click the ADD button

10) Select a Microsoft Excel Driver (Fig 2)

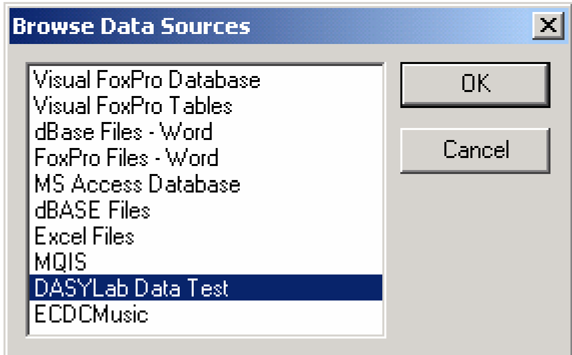


11) Click Finish.
12) Click "Select WorkBook" and select the previously saved Excel workbook.
13) Type in a user description.
14) Click on "Options"
15) Turn OFF Read Only (Fig 3)



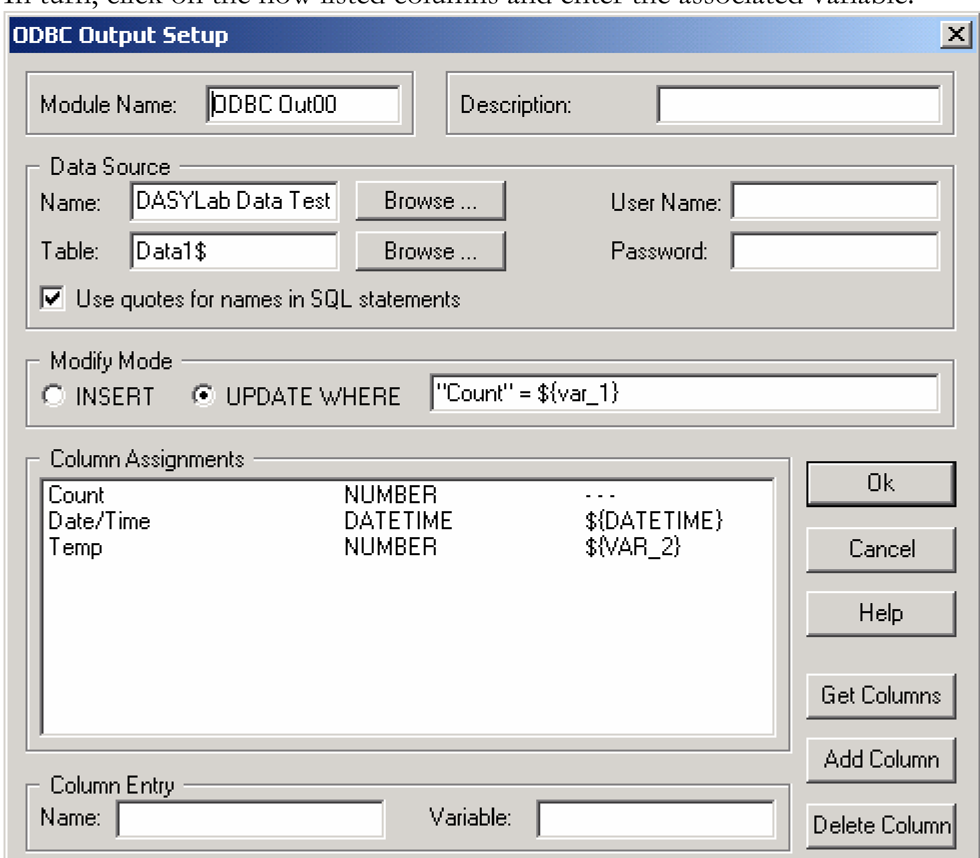
16) Click OK until you are back at the control panel. Close the Control Panel.

17) Start **DASY*Lab***.
18) Place an ODBC output module on your worksheet.
19) Access the properties of the ODBC output module.
20) Next to "Name" click on Browse and select the name of the Excel then click OK



(fig 4)

21) Next to "Table" click on Browse and select the name of the Excel Table you wish to place data into and then click OK.
22) Click the "Get Columns" button to retrieve the list of columns you entered in at step 3.
23) In turn, click on the now listed columns and enter the associated variable.



24) Start **DASY*Lab***.

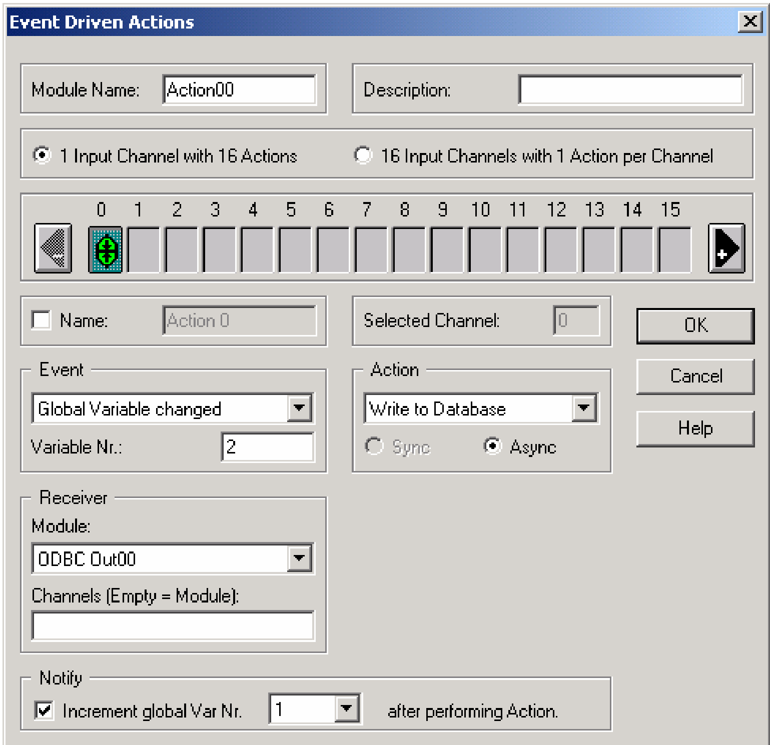
To send data to MS Excel you must select the "Use quotes for names in SQL statements" and fill in the "UPDATE WHERE" field. We need to tell **DASY*Lab*** where to place the data in the Excel Spread Sheet; here is where the "Count" column comes in. I have told **DASY*Lab*** to update where Count is equal to some variable number. If we increment, decrement or set this number we can select where **DASY*Lab*** places this data. For this example I will assume that we want to start at 0 and increment up.

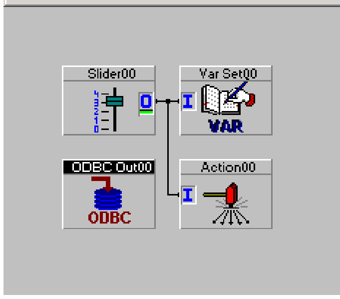
The data must be placed into a *Global Variable* in order for the ODBC to send the data. In order to place the data into a *Global Variable* you must use the **Global Variable Set Module** or the **Latch Module** with the latch to global variable option. Within these modules you may set which global variable you wish to use.

To reference this variable use the format ${VAR_XX} where XX is the number of the variable you wish to reference. Global string may also be used, to reference these use the ${STR_XX}, where XX is the number of the string you wish to use.

There are also predefined global string, such as time and data, ${DATETIME}.

An **Action** module must also be included in order to cause the ODBC to transfer the variables or strings. The action module should have the following settings; Event: Global Variable Changed, Variable Nr: The number of the global variable you wish to write, Module: ODBC Out, Action: Write to database. The action module may be connected to any line in your worksheet.
I have turned on the "Notify" option to increment global variable 1 after this action is performed. I have also used global variable 1 in my UPDATE WHERE statement. Using these two features together **DASY*Lab*** automatically increments where it will place the data each time. We can use other actions to set this variable back to zero or to any other value we like.





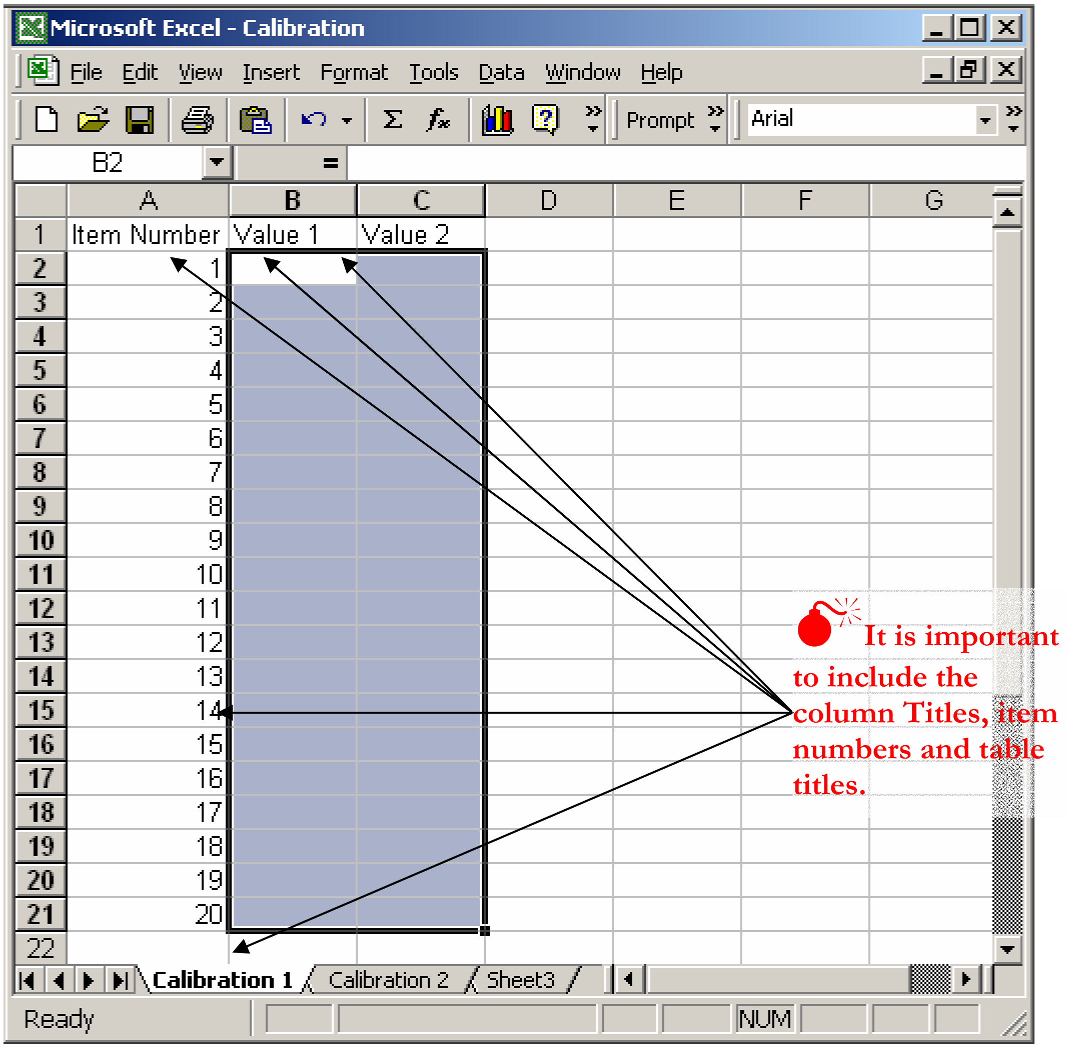
USING DASYLAB TO GENERATE AND READ A CALIBRATION FILE VIA ODBC.

It's possible to use **DASY*Lab*** to save calibration data to a file, and to read it back again for use in various modules, including the Scaling module.

As parts age or are exchanged calibration data needs to be updated to reflect these changes. The easiest way to maintain a list of calibration values is via a Microsoft Excel spreadsheet or via a database (such as Microsoft Access). **DASY*Lab*** can modify and read this spreadsheet using the ODBC input and output modules and use the saved values in calibration calculations.

This example demonstrates how to read and write to an Excel spreadsheet containing 20 sets of 2 values used for calibration. **DASY*Lab***, however, has the ability to read more values than this as well as read values from Excel-performed calculations.

First create an Excel spreadsheet to serve as the template for the ODBC read and write. The worksheet has a column for part number as well as a column for value 1 and value 2. This spreadsheet must then be registered with the Microsoft® ODBC database (see the chapter on ODBC Setup). The worksheet looks as follows:

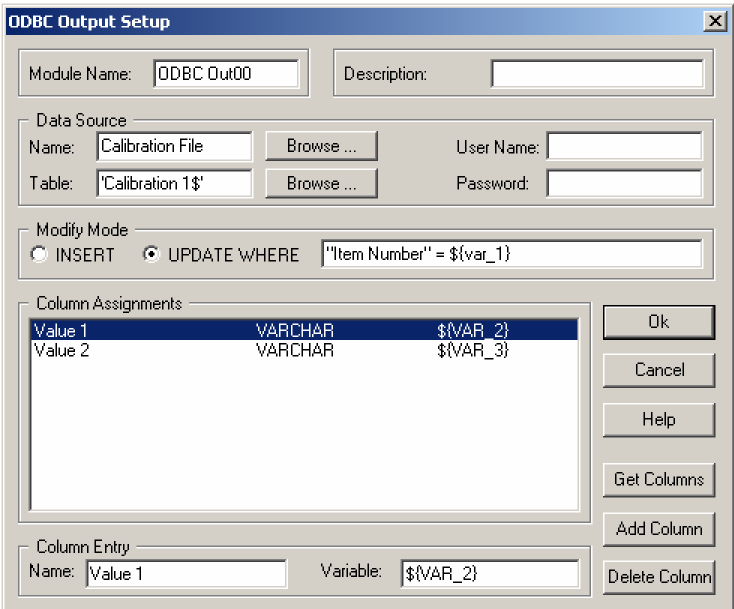


Once you've registered the spreadsheet with Microsoft® ODBC you are ready to start setting the values with **DASY*Lab***. Create a new worksheet and place an ODBC OUTPUT module from the Modules: Files menu. Click the Browse button after the Name box and select your Excel Spreadsheet from the menu. Next click the Browse button after the Table box and select the table containing your values. Then click on the "Get Columns" button: the list of columns should now be visible.

The next couple of steps rely on the arbitrary selection of a few "Global Variables". **DASY*Lab***'s ODBC Module is only able to send and receive data via global variables. When using Global Variables, it is imperative that you keep the variables separate and remember their assignments for ease in debugging and testing. This example uses variable 1 for the item, 2 for the value 1 and 3 for the value 2.

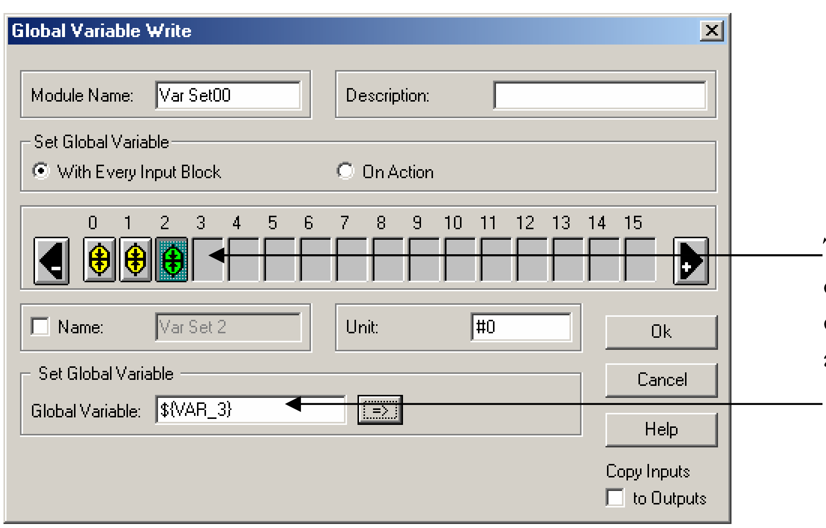
The "Modify Mode" needs to be set to "UPDATE WHERE" and a search string entered. In this case you will be searching the "Item Number" column for the number of where you wish to enter the data. For this example, you will use variable location 1 for the index number - therefore you will test for the "Item Number" to be equal to ${var_1}, which is set by **DASY*Lab***.

Click on the column assignment for Value 1. In the variable location enter the reference for variable 2; ${VAR_2}. Do the same for Value 2, however, use variable 3, ${VAR_3}. This completes the setup for the ODBC portion. The ODBC output setup window should look like the example window below.



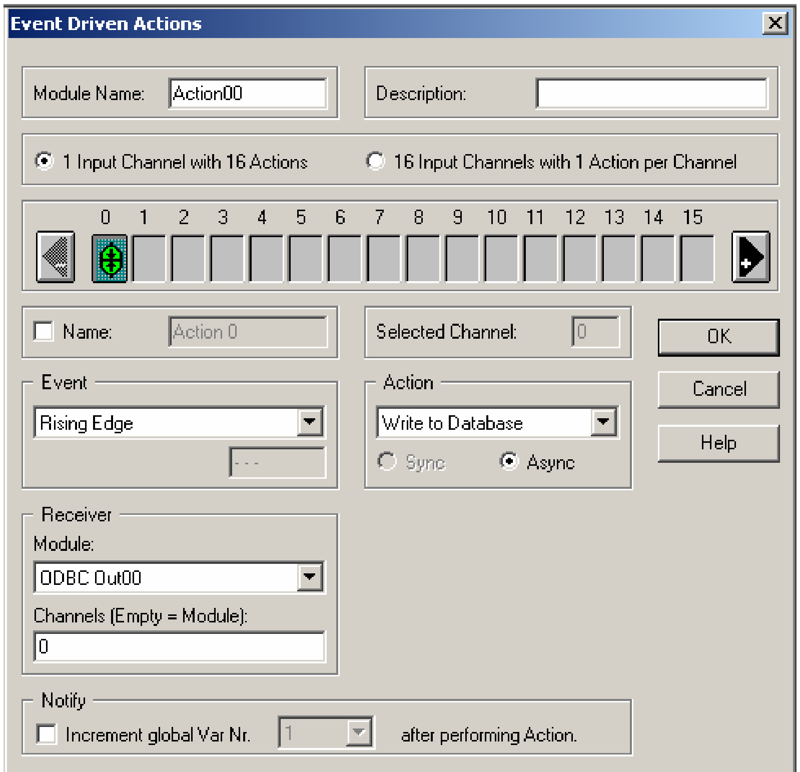
The next step is to enter the value into the global variables and send them to the ODBC database. To accomplish this, use Slider modules to set the location and the two values, Set Global Variable modules to send the data to the global variables and an Action module with a Switch to transmit the data.

The Slider module in this example uses three channels. The first channel selects the location or "Item Number" referenced in the ODBC database, the second channel is the first value and the third channel is the second value. The resultant output from the slider goes to a Set Global Variables module. The Set Global Variables Module has three inputs, each defined to set variable 1, 2 or 3 respectively.

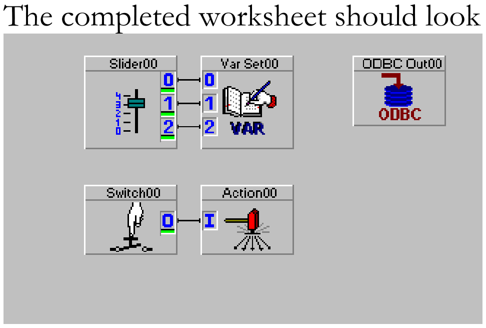


Three channels: channel one sets variable one, channel two sets variable 2 and so on.

Lastly we create a Switch module and set it to "One Shot". An Action module is placed onto the worksheet. The action module should be set as follows:
EVENT: rising edge
RECEIVER: ODBC Out 00
ACTION: Write to database



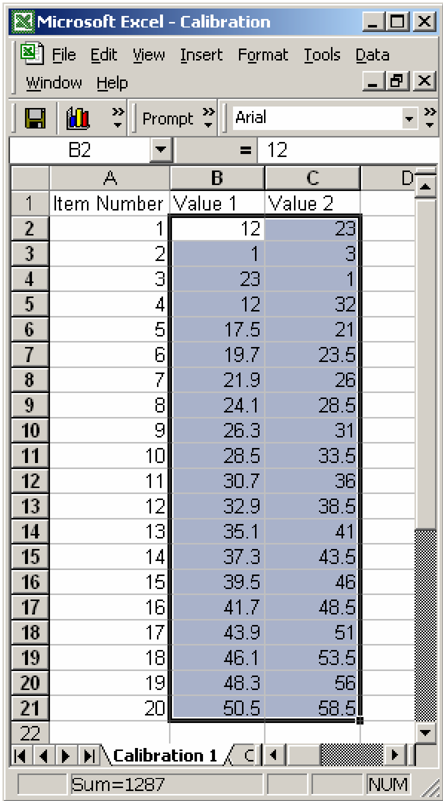
The completed worksheet should look as follows:



By setting the slider values and clicking on the switch the database will be updated with new values. The data from the slider (Channel 1 and 2) can be replaced with data from an Analog input source to use real world values in your calculations.

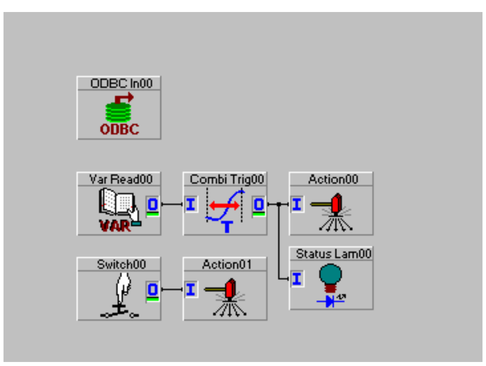
Note: Excel can only hold 65535 data rows and therefore only 65535 data samples per column.

With the database successfully created, you may now start the task of reading the values back into **DASY*Lab*** for use in the calculations. For this example the database is the same as the one created previously, with twenty sets of values. Read these values into **DASY*Lab*** and store them in the Global variables for use in calculations. The Excel database looks as follows:



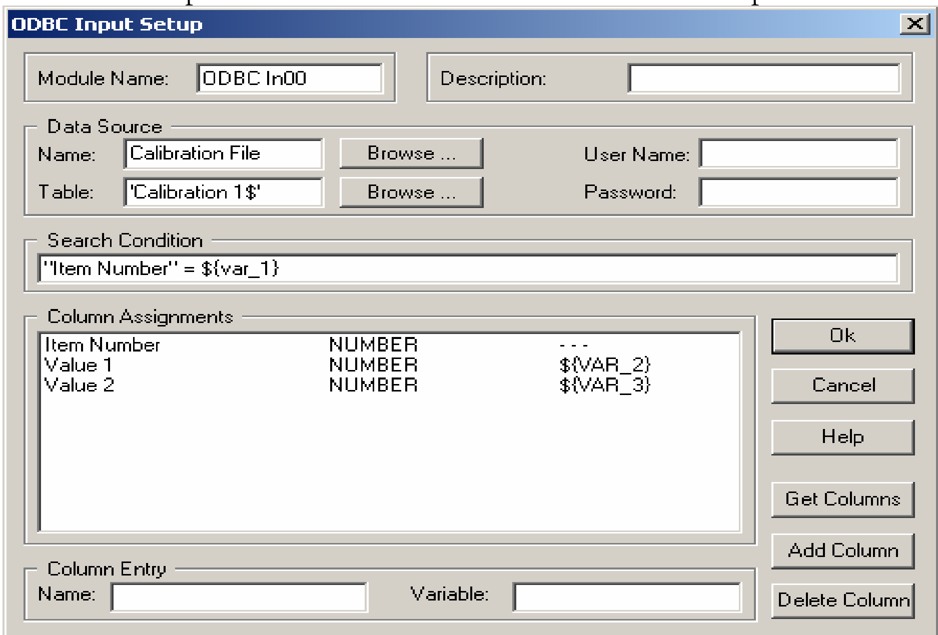
	A	B	C
1	Item Number	Value 1	Value 2
2	1	12	23
3	2	1	3
4	3	23	1
5	4	12	32
6	5	17.5	21
7	6	19.7	23.5
8	7	21.9	26
9	8	24.1	28.5
10	9	26.3	31
11	10	28.5	33.5
12	11	30.7	36
13	12	32.9	38.5
14	13	35.1	41
15	14	37.3	43.5
16	15	39.5	46
17	16	41.7	48.5
18	17	43.9	51
19	18	46.1	53.5
20	19	48.3	56
21	20	50.5	58.5
22			

It is more complicated to read data from an ODBC database than to write it. This example starts with the finished worksheet explains the function of each module.



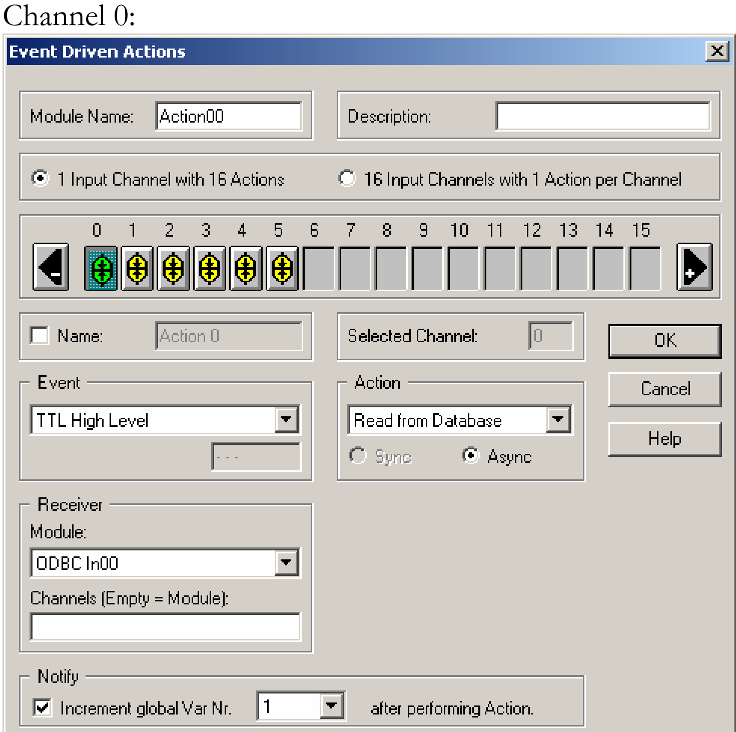
Several things remain unchanged. For example the variable used for the index in the ODBC input will be variable 1 and the two values will be variable 2 and 3 respectively. To use all of the values, they need to be saved within **DASY*Lab***. This example stores the values for Value 1 in variable 100-119 and the values for Value 2 in 200-219. There is also another index used for the location to store the values. I have used 99 and 199 for Values 1 and 2.

The ODBC Input module should look as follows when completed:

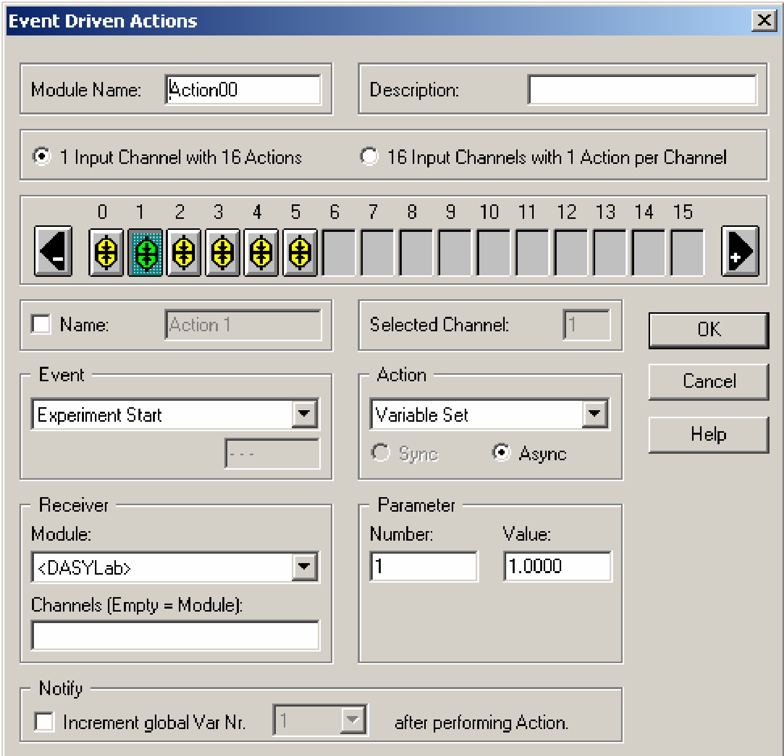


Module Action00 forms the core of this application. This module will cause the ODBC In00 module to read in data and the Action00 will then place that data into appropriate variable locations. Channel by channel descriptions are below.

Channel 0:

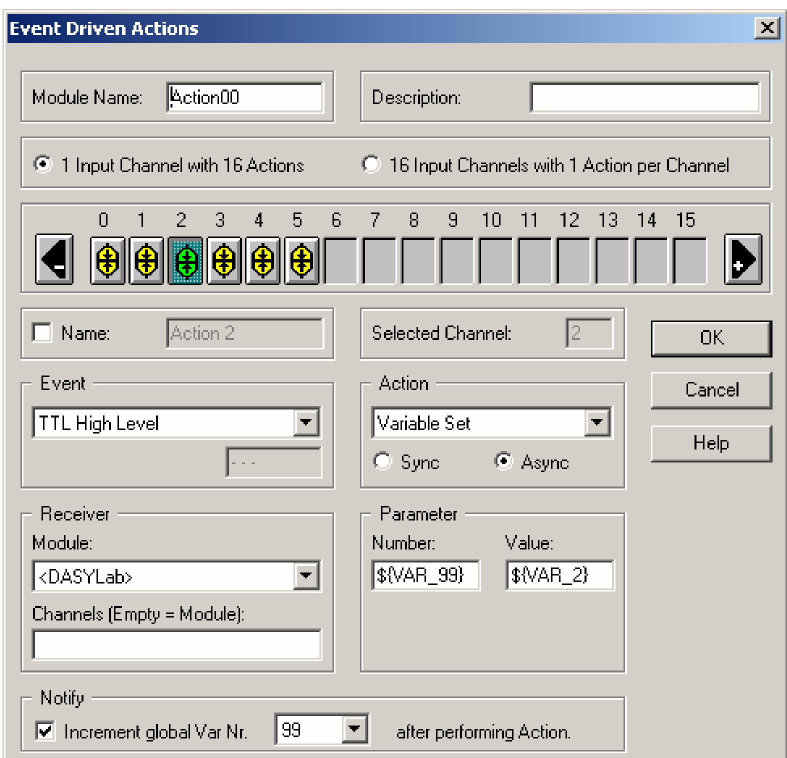


While Action00 receives a TTL high signal it will read data from the ODBC database. The "Notify" option has been enabled, which will increment the select variable each time this action is performed. Notice that the variable incremented is the same as the variable which is used as the index in the ODBC input module.



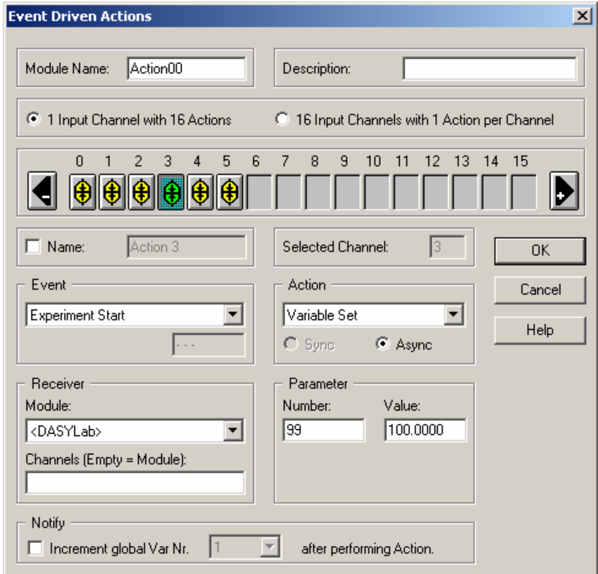
Channel 1:

This action performs something similar to a "RESET" when the application starts. This action could easily be placed anywhere in the series of channels. This action sets the variable 1 to 1; notice that 1 is the first index number in our database of values.



Channel 2:

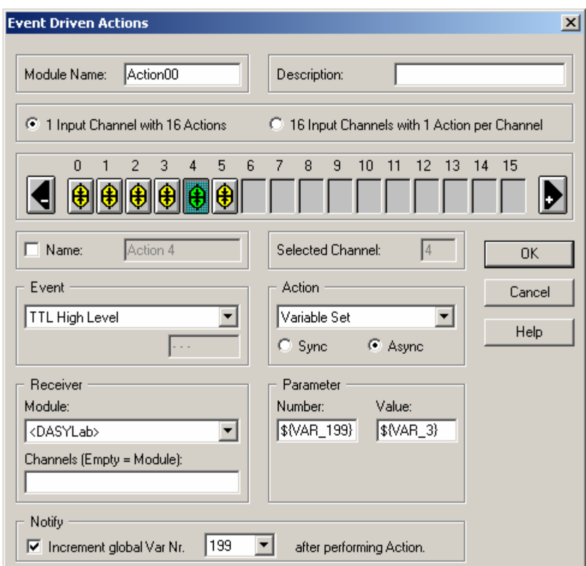
While Action00 receives a TTL high signal it will copy data from Variable 2 into the location pointed to by Variable 99. After each action is performed Variable 99 is incremented (notice the "Notify" option) and points to the next location. For example, when the application starts, variable 99 contains the value 100, therefore the value from variable 2 is copied into variable 100. When this is done, variable 99 increments by one and points at location 101. Channel 0, by similar action, increments variable 1 and the ODBC reads in a new value into variable 2.

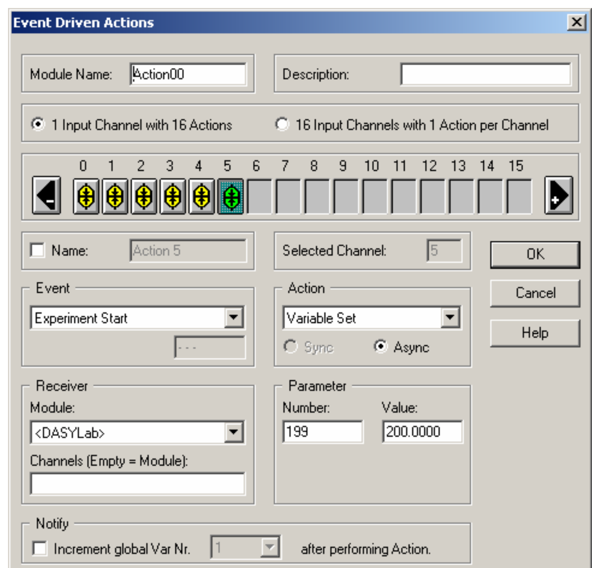


Channel 3:

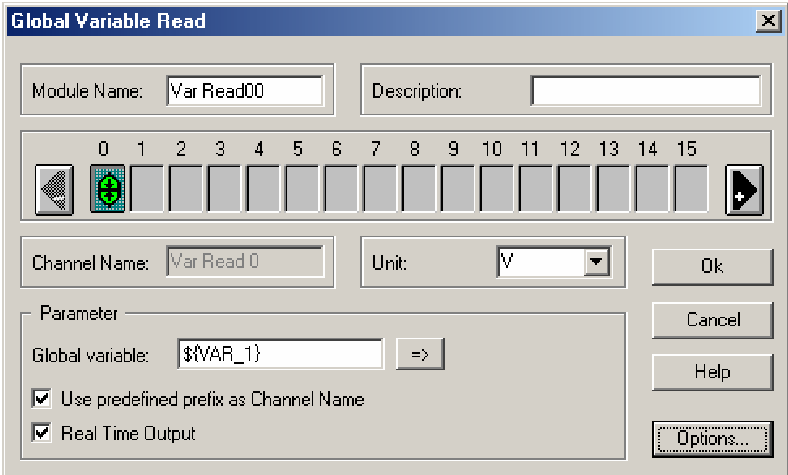
This channel functions similar to channel 1 where as it is setting the default value for variable 99, as noted in the explanation of channel 2.

Channel 4 and 5 are the same as channel 2 and 3 however the variable values and references have been changed.





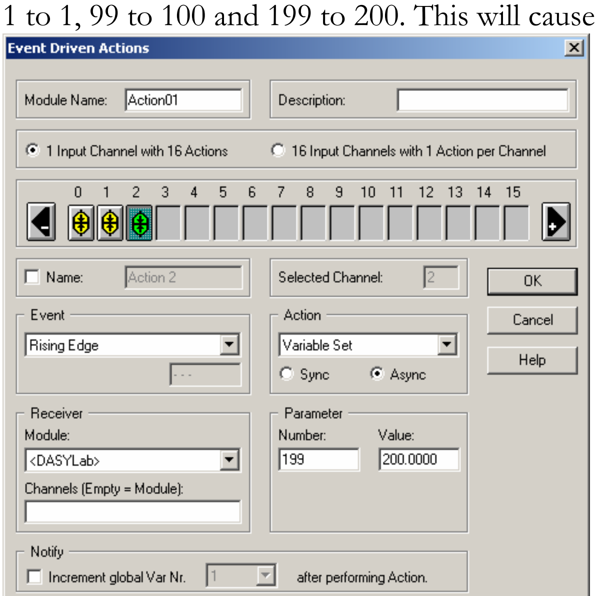
Var Read00:
The Global Variable Read will read variable 1, which is the index to read from the ODBC and output its value. This monitors Action00s progress through the database. The output from the Global Variable Read goes to a Combi Trigger.



Combi trig00:
The Combi Trigger generates a TTL High signal as long as the value in is less than 21 (>=20). Notice that we have 20 elements in our database. For more elements, increase this value.

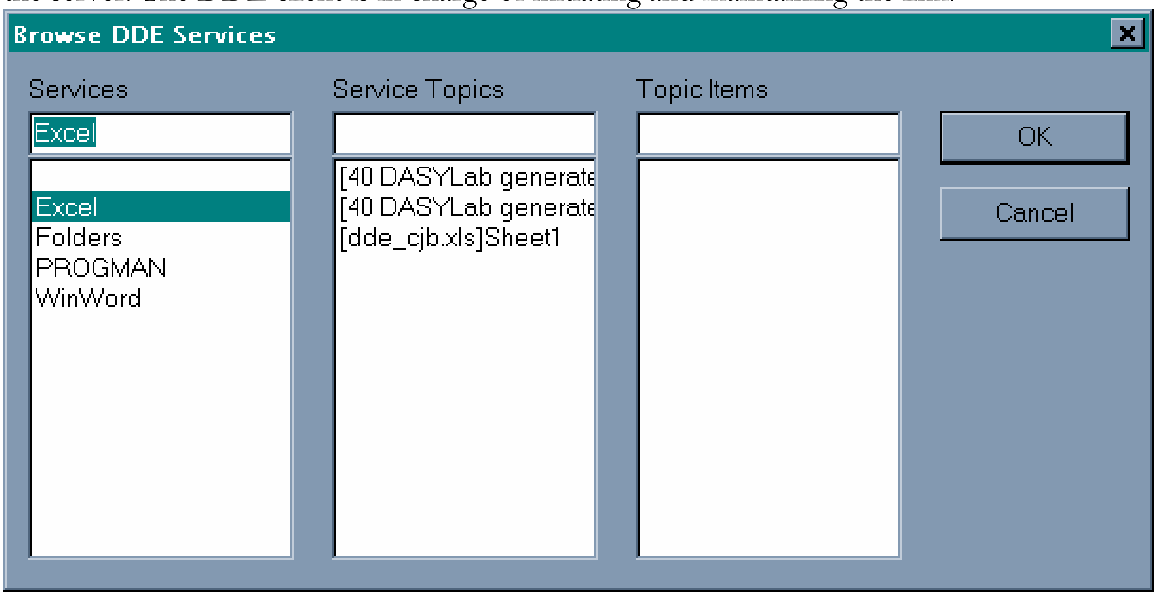
Tying it all together:
The Global Variable Read will output the current element being read from the ODBC database. The Combi Trigger will output a high signal as long as the index is less than or equal to 20. As long as the action module receives a high signal it will cause the ODBC module to read in a value and copy the values into other locations inside **DASY*Lab***.

Extras:
To re-read the database, possibly after you have manually changed it, use a one shot switch and an action module. The action module needs three channels; each with an event of Rising Edge and each channel will set a variable back to default. For this example it will set variable 1 to 1, 99 to 100 and 199 to 200. This will cause the system to re-read in all the values.



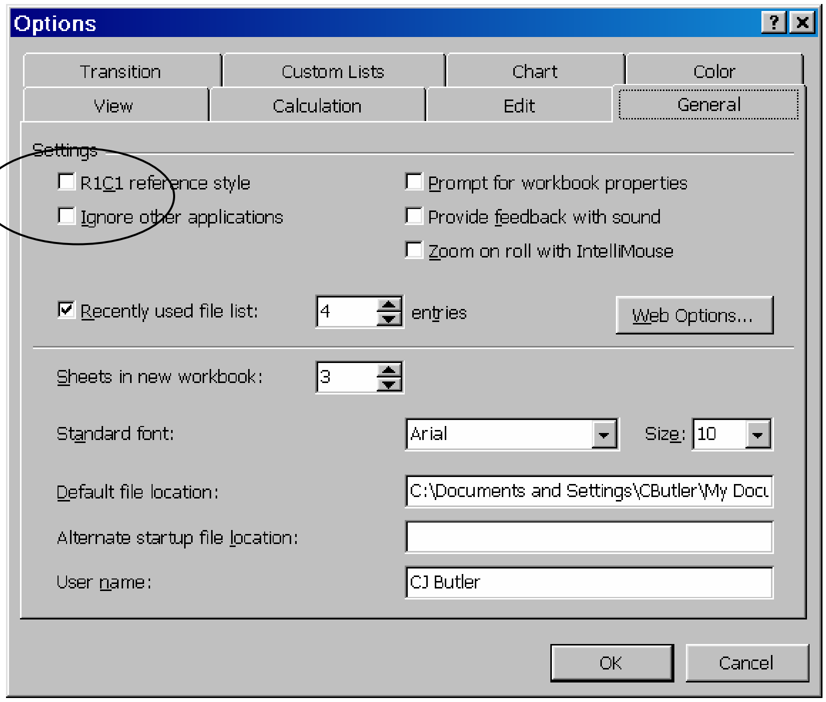
DYNAMIC DATA EXCHANGE (DDE)

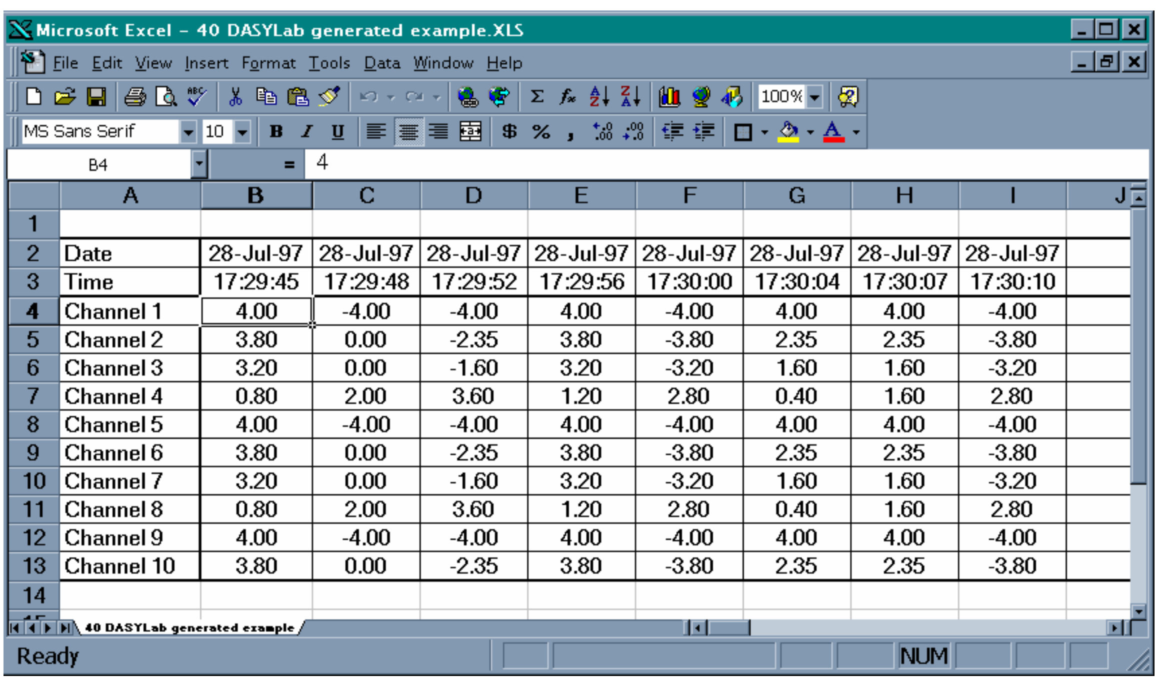
The DDE Output Module has the ability to generate a DDE Topic Item, or you can simply send data to a fixed set of cells, which are overwritten by each new block of data. You can use the generate Item feature to send successive blocks of data to another application, continually iterating the target range of the receiving application.
To use this feature, set up the DDE link with Excel (or another DDE capable spreadsheet product) by selecting the Application and Topic. Use **Browse** to select an active Application and Topic. Note that Excel must be running; **DASY*Lab*** will not start the receiving application. For the purposes of this example, **DASY*Lab*** is the client and Excel is the server. The DDE client is in charge of initiating and maintaining the link.



This example will send 10 channels of data to Microsoft Excel, and will include the timestamp. For display purposes, the time in Row 3 is copied to Row 2. Row 2 is then formatted to display the date, and Row 3 the time. **DASY*Lab*** will start filling in the date/time and values at cell B3 (Row 3, Column 2) and it will fill down the column.

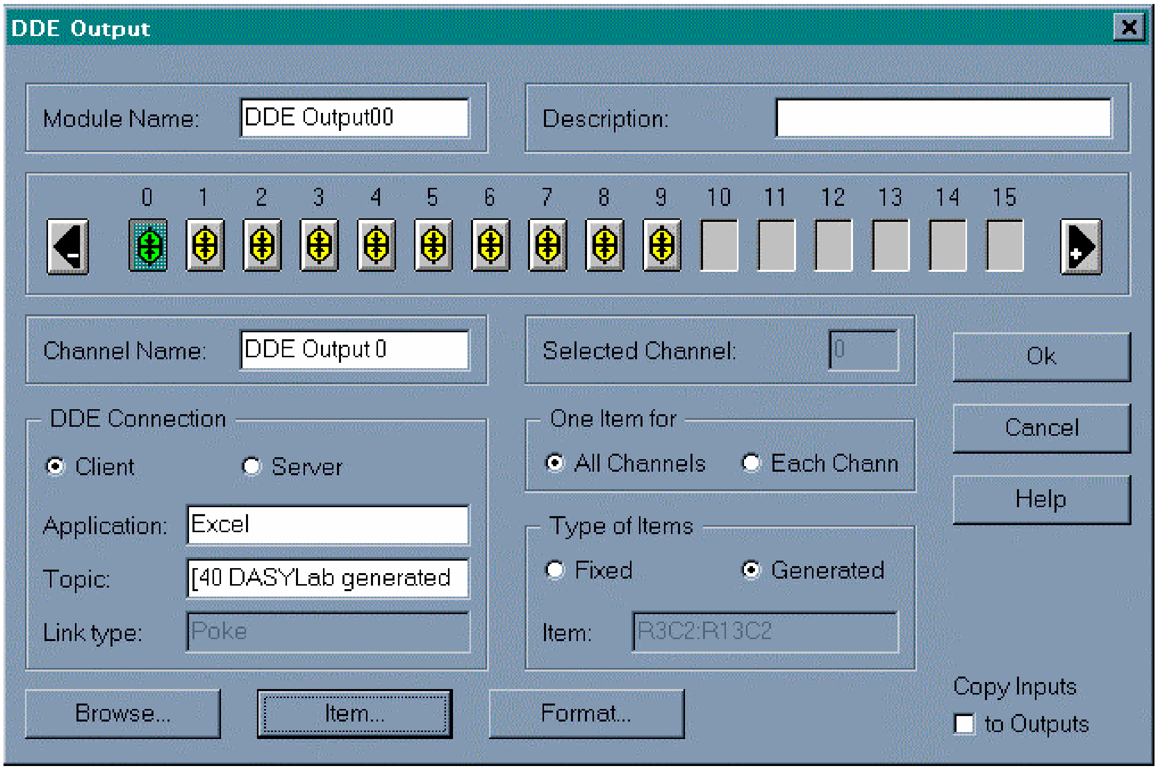
Note: the row/column notation varies by local language for Excel. For example, in English the first cell (A1) is R1C1, in French, it is L1C1. If you are using a version of Excel that is not English, verify the row/column notation for your package by looking at the Excel Options Dialog, the General tab. Replace your correct R / C notation wherever we refer to R and C.





	A	B	C	D	E	F	G	H	I
1									
2	Date	28-Jul-97	28-Jul-97	28-Jul-97	28-Jul-97	28-Jul-97	28-Jul-97	28-Jul-97	28-Jul-97
3	Time	17:29:45	17:29:48	17:29:52	17:29:56	17:30:00	17:30:04	17:30:07	17:30:10
4	Channel 1	4.00	-4.00	-4.00	4.00	-4.00	4.00	4.00	-4.00
5	Channel 2	3.80	0.00	-2.35	3.80	-3.80	2.35	2.35	-3.80
6	Channel 3	3.20	0.00	-1.60	3.20	-3.20	1.60	1.60	-3.20
7	Channel 4	0.80	2.00	3.60	1.20	2.80	0.40	1.60	2.80
8	Channel 5	4.00	-4.00	-4.00	4.00	-4.00	4.00	4.00	-4.00
9	Channel 6	3.80	0.00	-2.35	3.80	-3.80	2.35	2.35	-3.80
10	Channel 7	3.20	0.00	-1.60	3.20	-3.20	1.60	1.60	-3.20
11	Channel 8	0.80	2.00	3.60	1.20	2.80	0.40	1.60	2.80
12	Channel 9	4.00	-4.00	-4.00	4.00	-4.00	4.00	4.00	-4.00
13	Channel 10	3.80	0.00	-2.35	3.80	-3.80	2.35	2.35	-3.80
14									

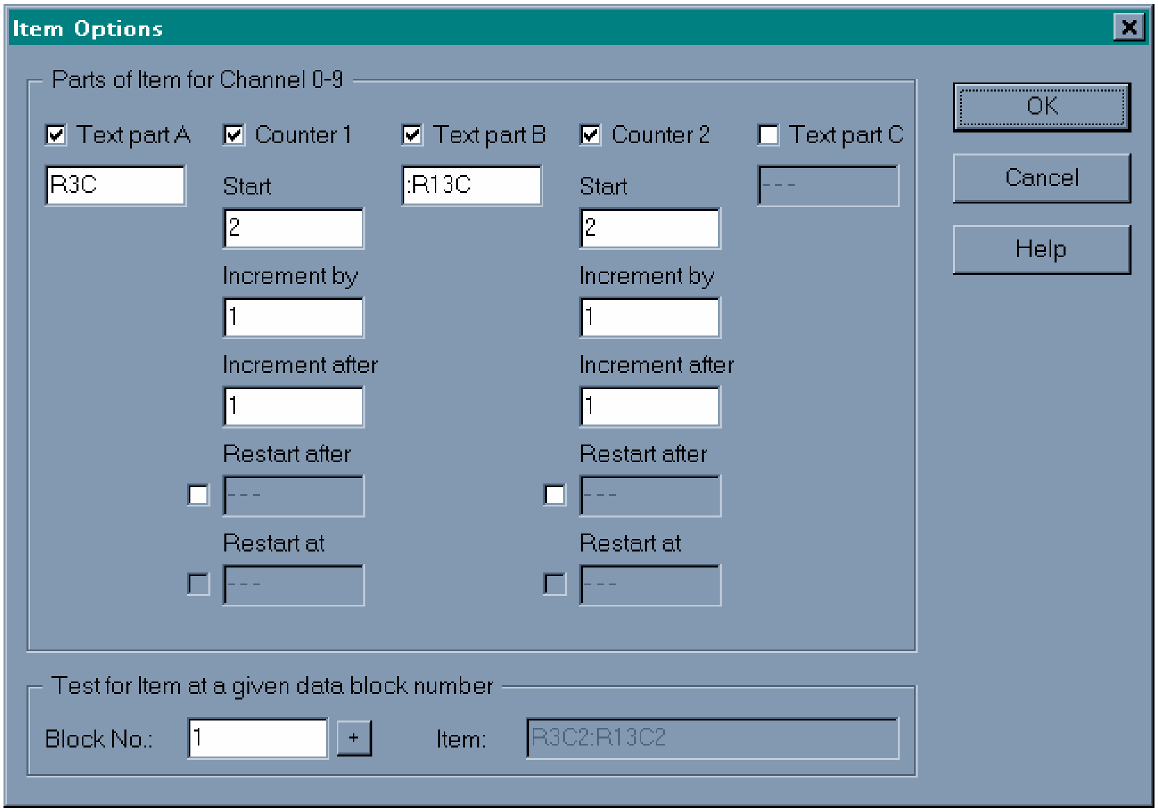
To set up the Item range in Excel, you must first specify the **Type of Items** as **Generated**. Then, click on the **Item** button to open the next dialog box.



To specify the desired range, enter the fixed text part into the Text Part boxes. In this case, we will be filling columns of data from row 3 through row 13.

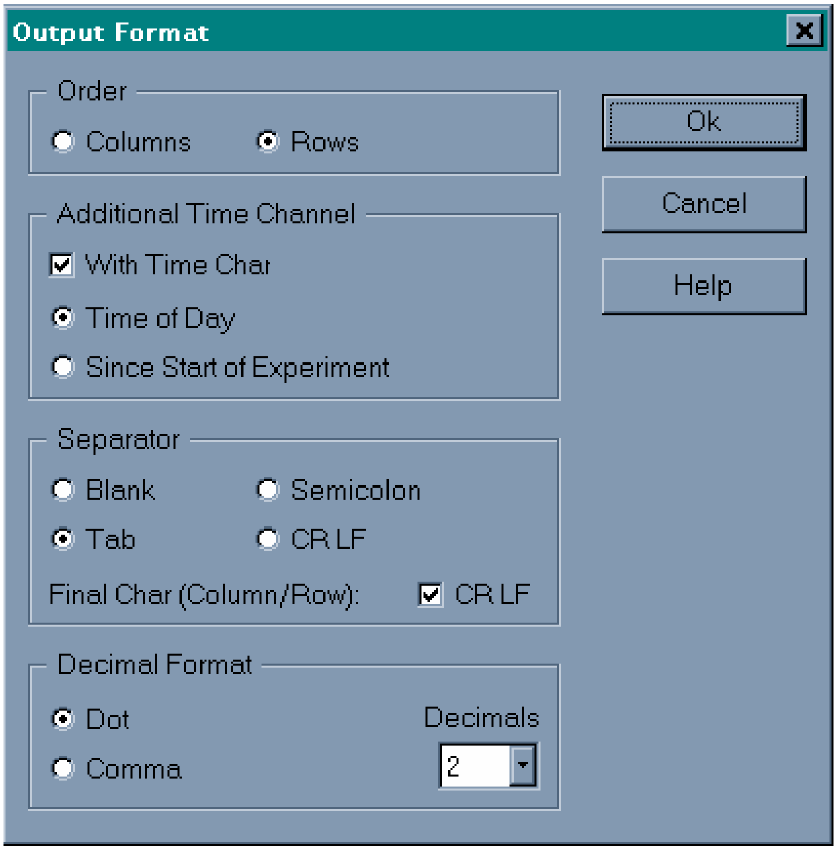
Box A	R3C
Box B	:R13C

Note that the text in box B must start with a colon ":".



The counter field specifies how **DASY*Lab*** will increment through the specified range. This example starts at cell B3, or R3C2. The starting number for counter 1 is 2, incremented by 1 for each value. The range ends at cell B13, or R13C2. The starting number for counter 2 is also 2, also incrementing by 1 for each value. This example does not use Text part C.

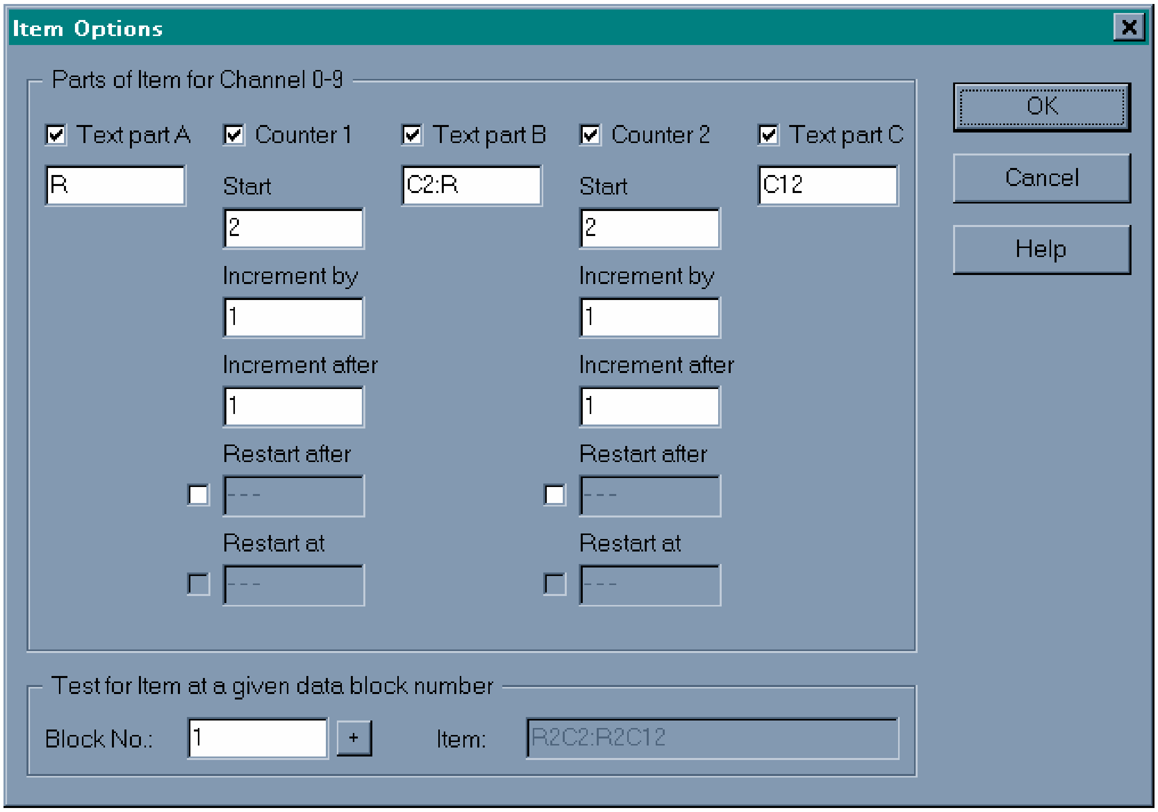
To complete the setup, it's necessary to define the Format.



The order is Rows, and the time channel is selected as Time of Day. For US versions of Excel, the separator is Tab and the Decimal Format is Dot (period).

At this point, **DASY*Lab*** should successfully communicate with Microsoft Excel, sending columns of data. Remember that DDE is relatively slow; don't expect high data rates.

To send data by Rows, change the settings as follows:

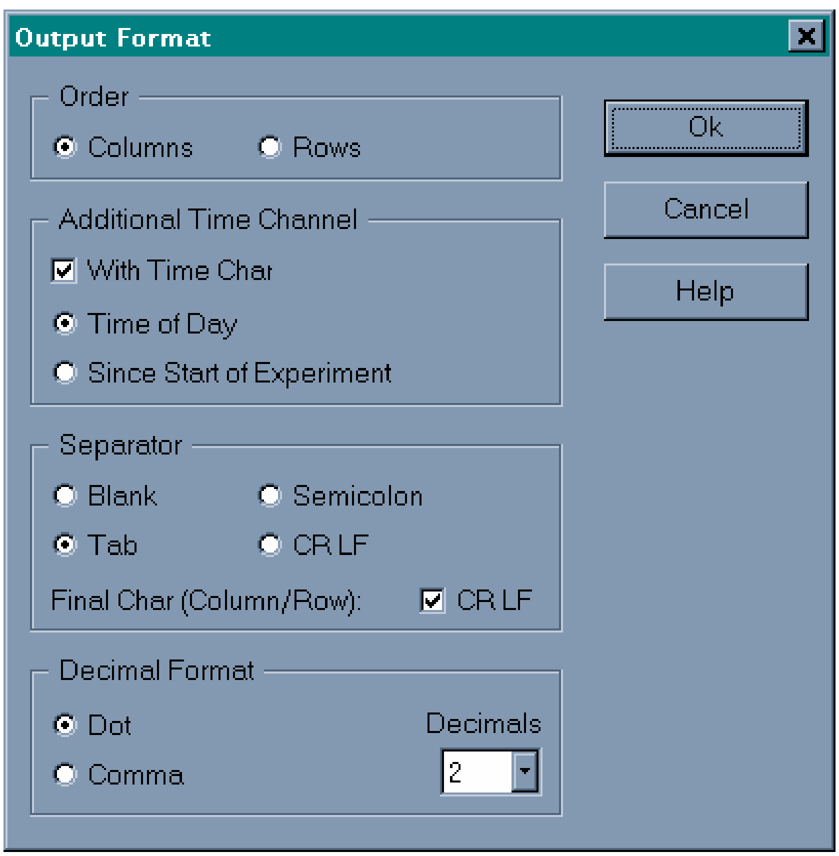

Item Options
Parts of Item for Channel 0-9
Text part A
Counter 1
Text part B
Counter 2
Text part C
R
Start
2
C2:R
Start
2
C12
Increment by
1
Increment by
1
Increment after
1
Increment after
1
Restart after

Restart after

Restart at

Restart at

OK
Cancel
Help
Test for Item at a given data block number
Block No.:
1
+
Item:
R2C2:R2C12


Output Format
Order
Columns
Rows
Additional Time Channel
With Time Char
Time of Day
Since Start of Experiment
Separator
Blank
Semicolon
Tab
CR LF
Final Char (Column/Row):
CR LF
Decimal Format
Dot
Decimals
Comma
2
Ok
Cancel
Help

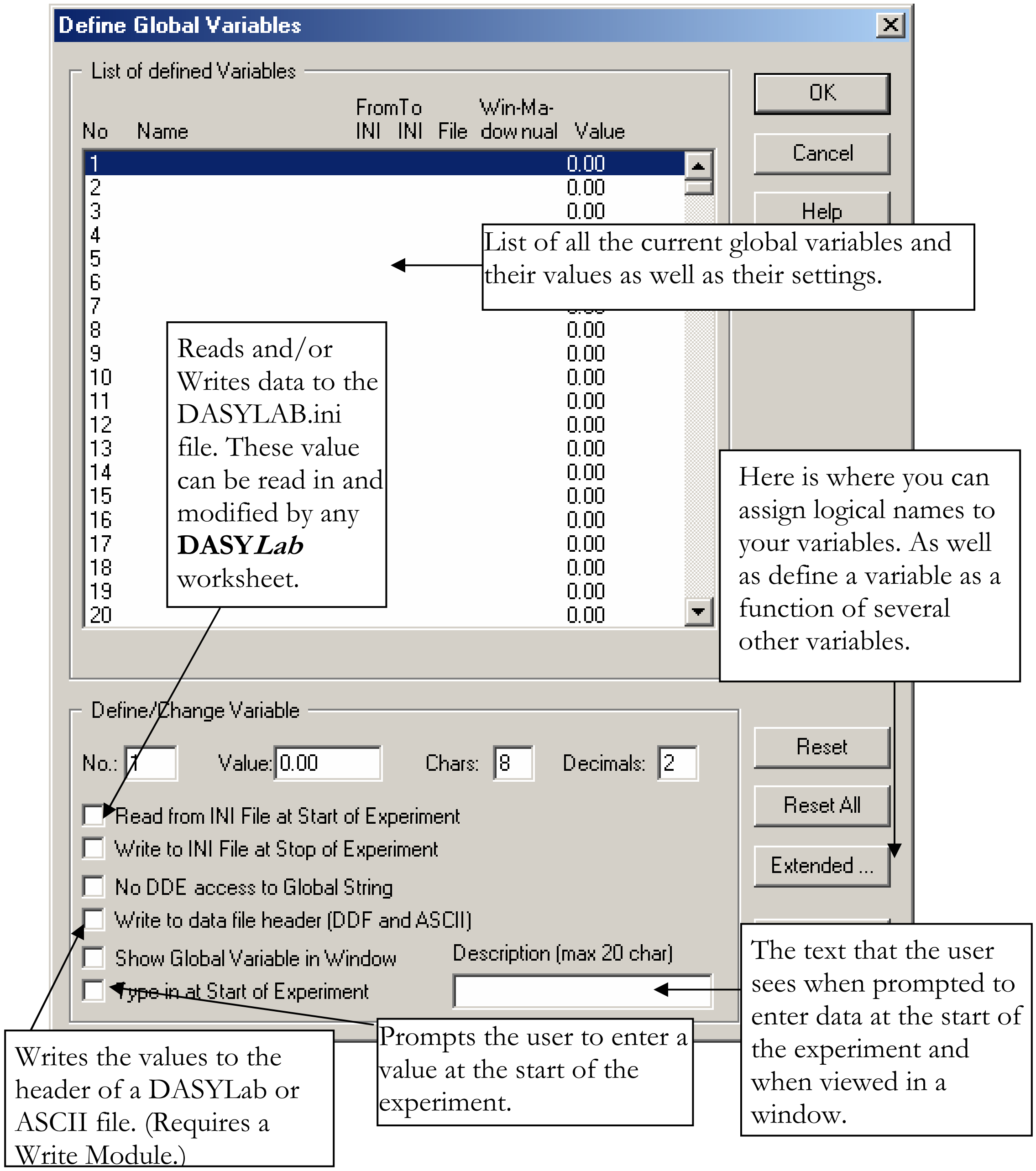
CHAPTER 7: ADVANCED DASYLAB FUNDAMENTALS

GLOBAL VARIABLES

DASY*Lab* allows users to save numbers in specified memory locations within the **DASY*Lab*** program. These locations are known as Global Variables, there are 999 variables available within **DASY*Lab***. These variables can be written to, read from, modified and monitored by **DASY*Lab*** and a select few of its modules.

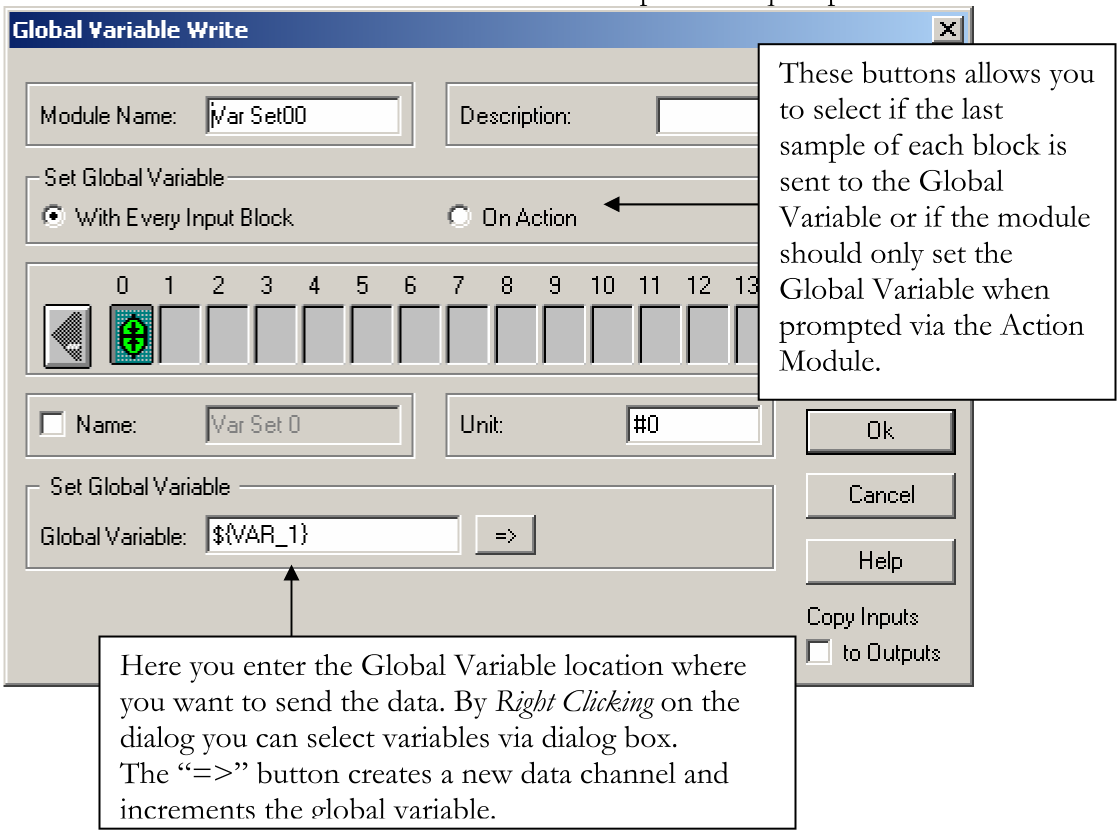
In some of the more advanced applications with **DASY*Lab*** Global Variables are used to transfer data into and out of the **DASY*Lab*** environment. Global Variables are also used as a semaphore by Action Modules to indicate when an action has been completed.

To access the Global Variable menu click on Options, Define Global Variables… The resultant dialog box should look like this:

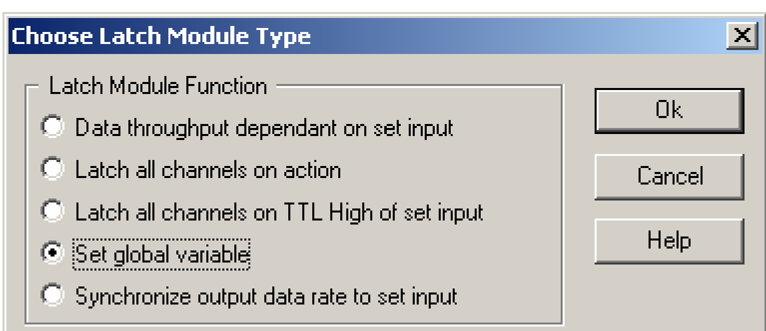


In order to place data into a Global Variable once **DASY*Lab*** has started we need to use a Global Variable Set Module, Latch Module or an Action Module. The Global Variable Set and Latch Modules are best suited to place dynamic data into a variable, such as data from an analog input or slider control, while the Action Module is best for setting static or conditional values.

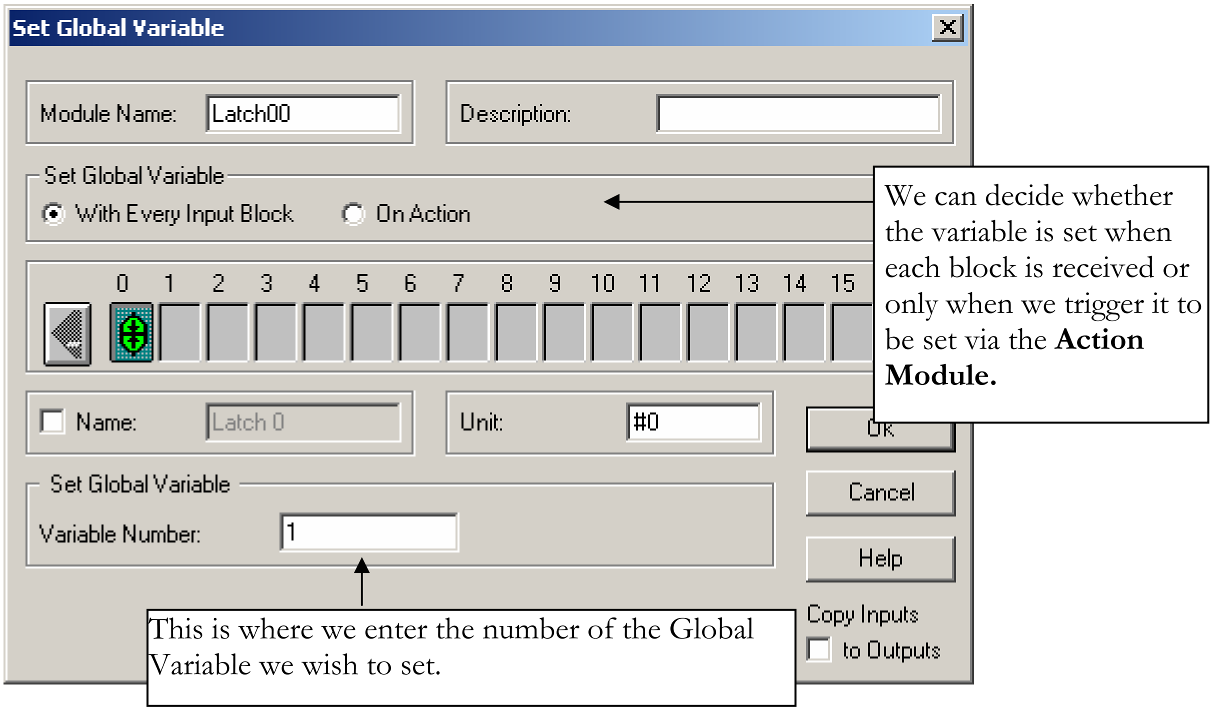
The easiest method of setting data into a global variable is using the Set Global Variable Module. The Set Global Variable Module has a couple of unique options:



When using a Latch Module we are given several options. Upon placing a Latch Module on the worksheet we get a dialog like this:



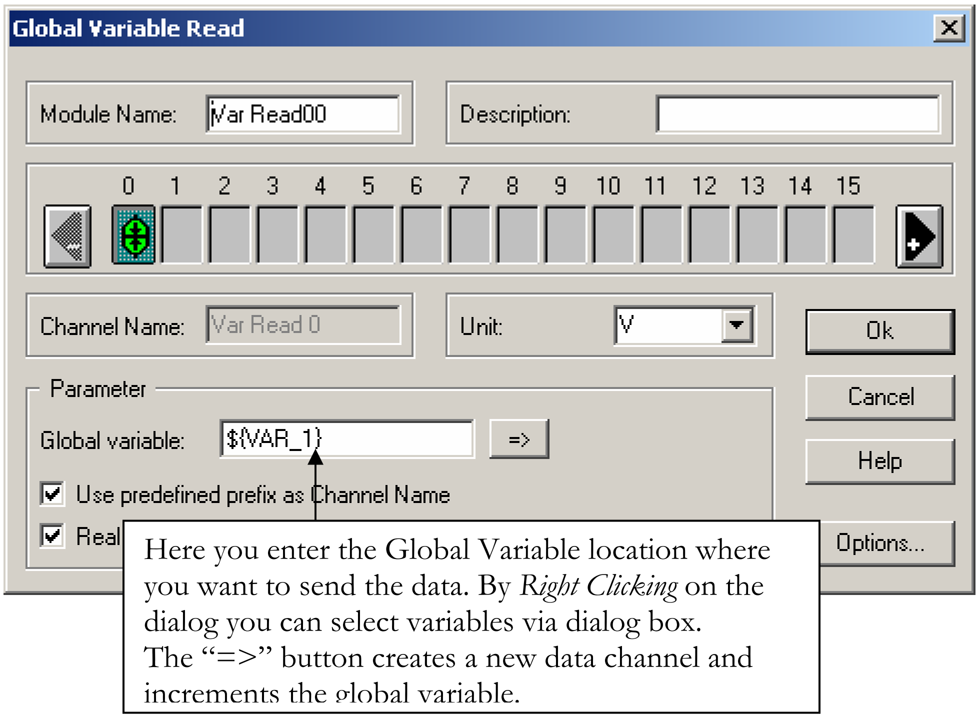
By selecting the "Set global variable" option the inputs will be written into the Global Variables. The properties of the Latch Module look like this:


Set Global Variable
Module Name:
Latch00
Description:
Set Global Variable
With Every Input Block
On Action
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Name:
Latch 0
Unit:
#0
Ok
Cancel
Help
Set Global Variable
Variable Number:
1
Copy Inputs
to Outputs
We can decide whether the variable is set when each block is received or only when we trigger it to be set via the **Action Module.**
This is where we enter the number of the Global Variable we wish to set.

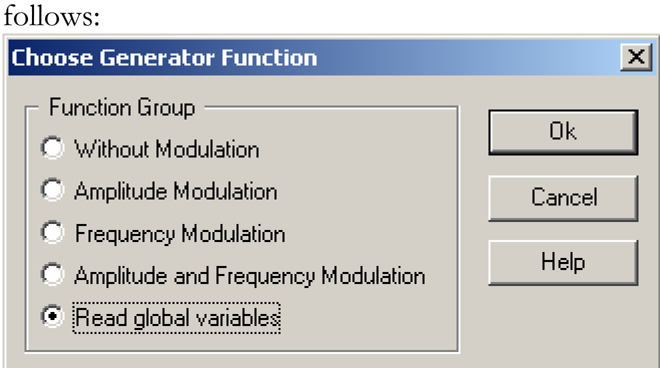
If we wanted to read a Global Variable back into our worksheet we can either interpreting it or generate it. To interpreting a Global Variable we use ${var_#}, where # is the number of the variable we wish to interpreting or ${Logical Name} which I will cover in the next section. For example to read variable 1 we would use ${var_1}. These interpreted variables can be used as set points in Scaling Modules, parameters in Generator Modules and factors in equations through **DASY*Lab*** as well as in many other modules and **DASY*Lab*** functions.

If we chose to generate the global variable we can use a Global Variable Read Module or a Generator Module.

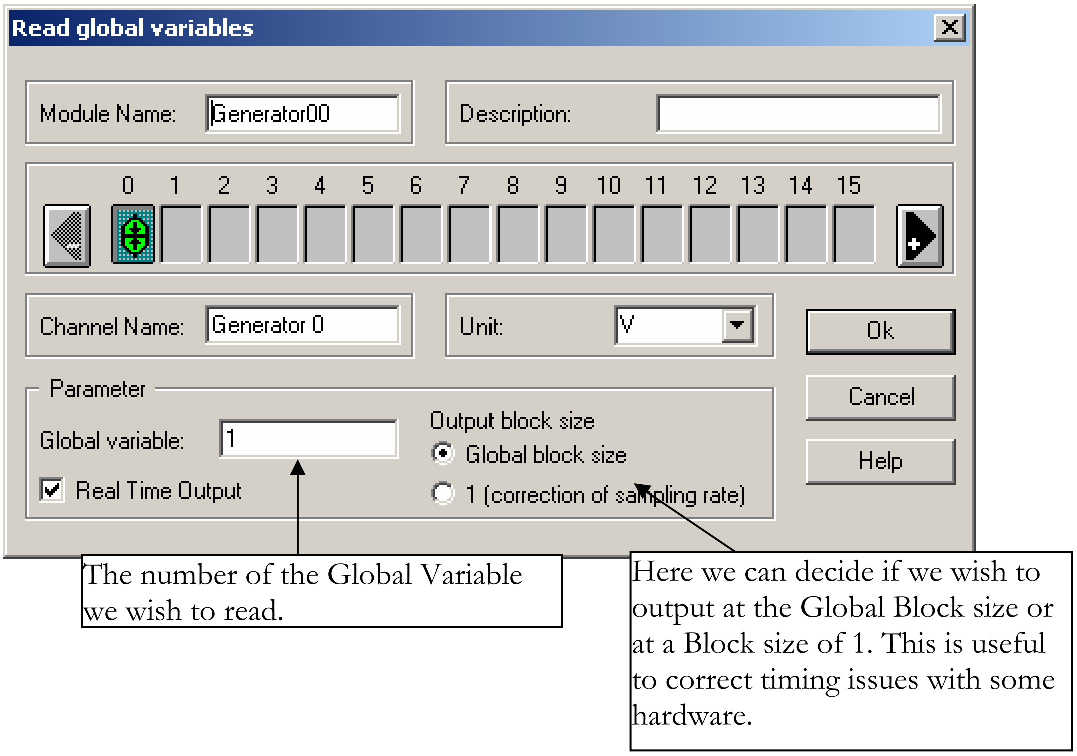
The Global Variable Read Module looks like this:



By selecting the Generator Module and placing it onto the worksheet we see a dialog box as follows:



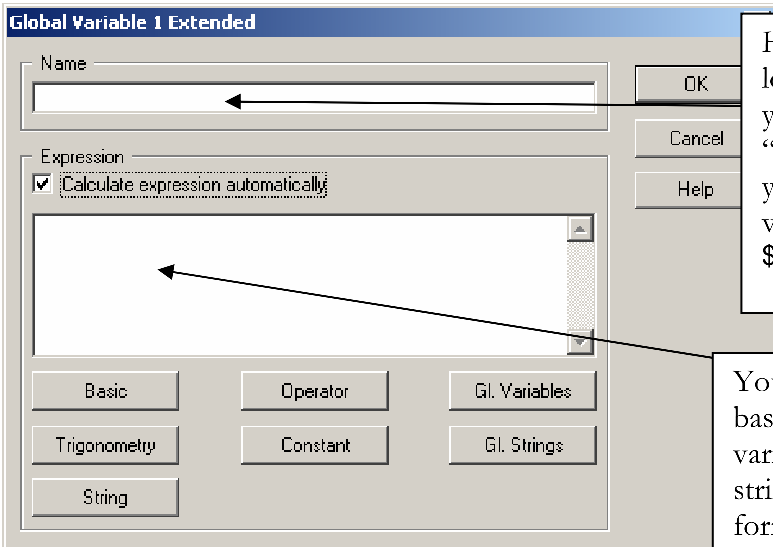
By selecting the “Read global variables” option and viewing the properties of the Generator Module we get the following dialog box:



The output of the Global Variable Read Module or the Generator Module will be the value of the Global Variable referenced within the module.

There are other functions we can perform with Global Variables. For example we can assign logical names to the variables to make them easy to remember. We can also define a Global Variable as a function of another global variable.

To access these functions click on Options; Define Global Variables, select the variable you wish to define and click the "Extend" button.

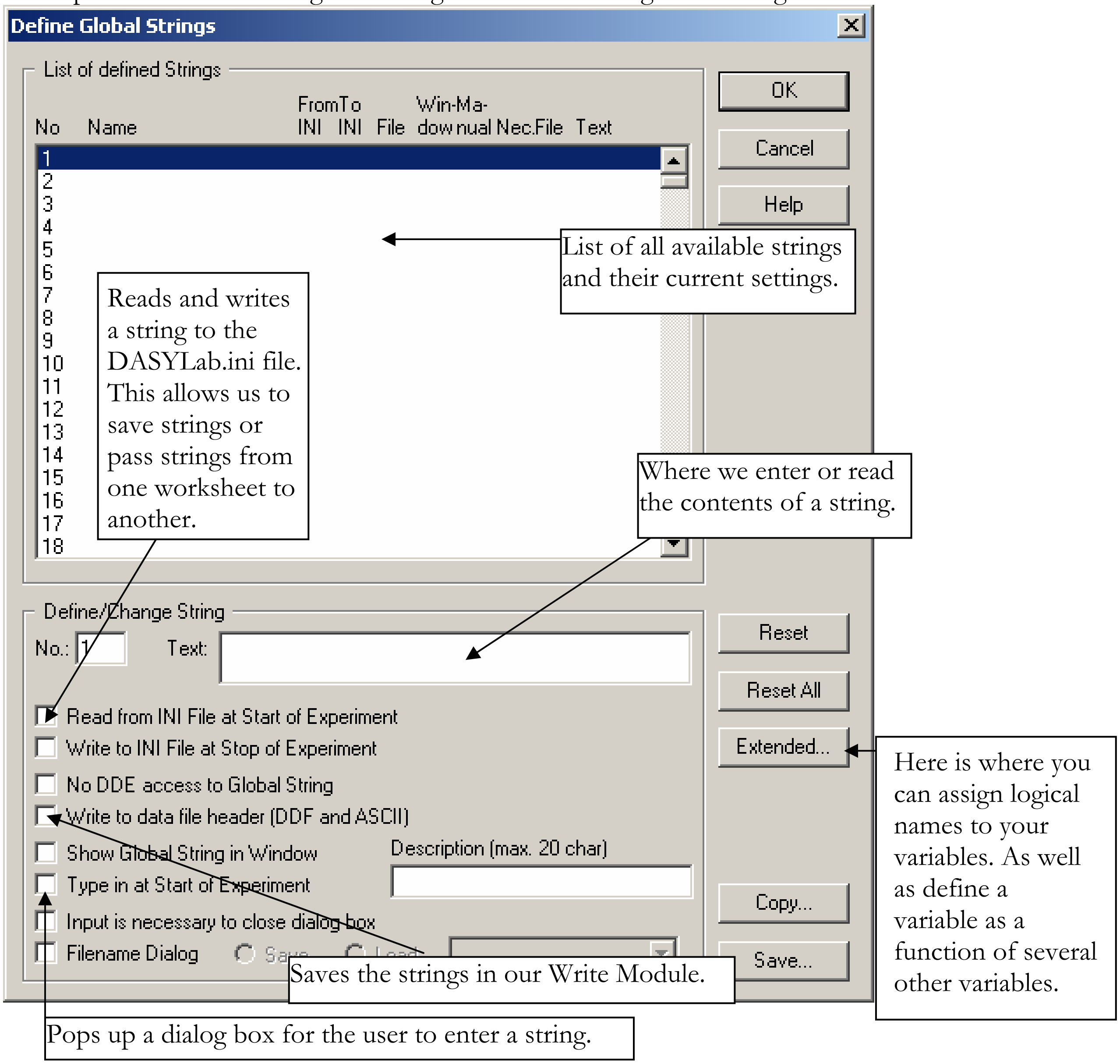


Here you can place a logical name. For example you can name this variable "Set_point_1" or anything you like. To use this variable in DASYLab type: ${Set_Point_1}

You can create a formula based on other global variables and/or global strings. The result of this formula will continuously re-calculated and placed into this global variable.

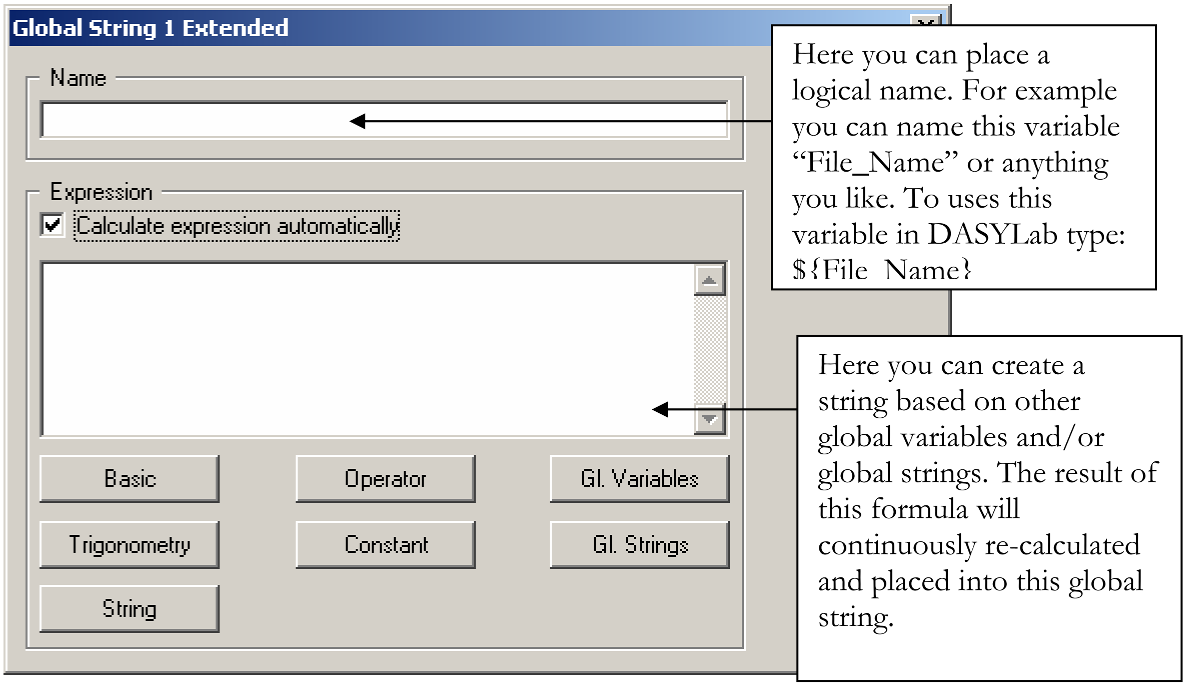
GLOBAL STRINGS

Global Strings work much the same as Global Variables; however, they are much better suited for documentation and file naming purposes. To access the global strings menu click on "Options" then "Define global strings…". You should get the dialog box as follows:

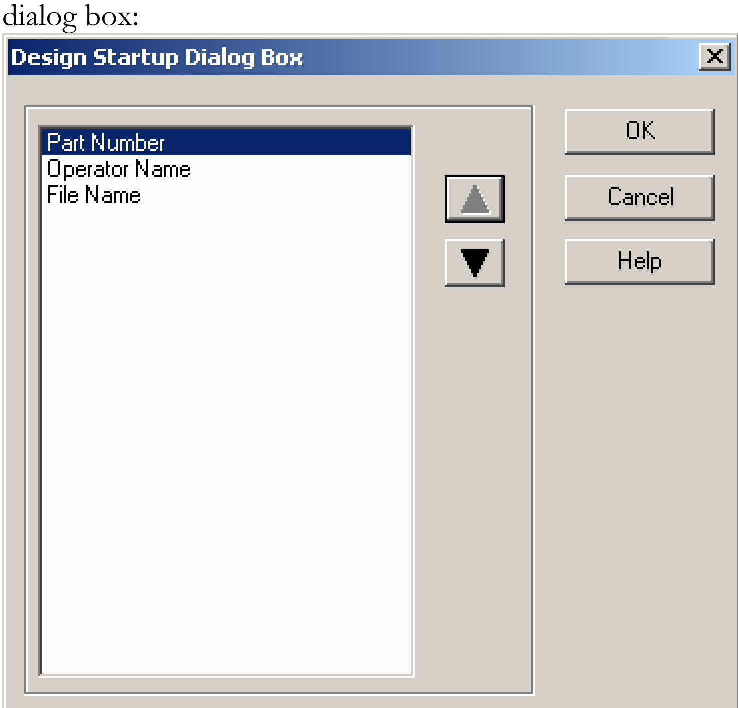


Once a Global String is set we can interpreted it by using ${str_#} where # is the number of the Global String we wish to interpret or ${X} where x is the logical name for the string. For example to read Global String 1 we would use ${str_1}.

To set a logical name or to define the string as a function of other strings click on the Extend… button.



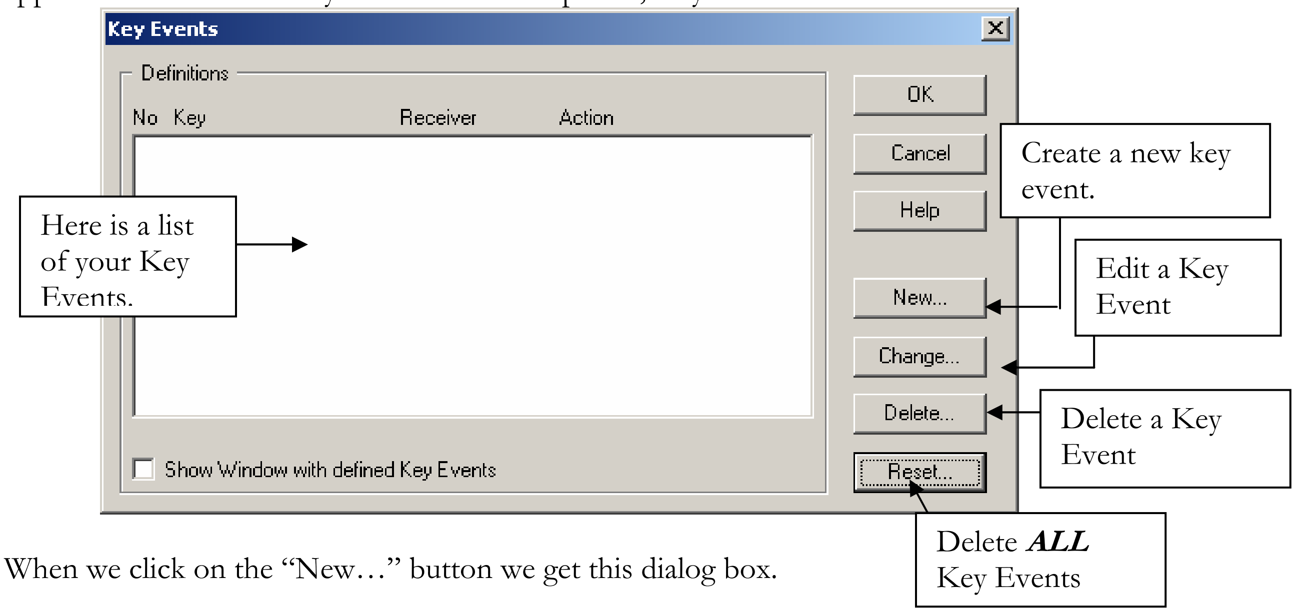
If you wish to have you user enter several Global Variables or Global Strings at experiment start, or any other time, you can arrange the order they are displayed. To set this order click Options, Design Setup Order. Once you have selected that you should get the following dialog box:



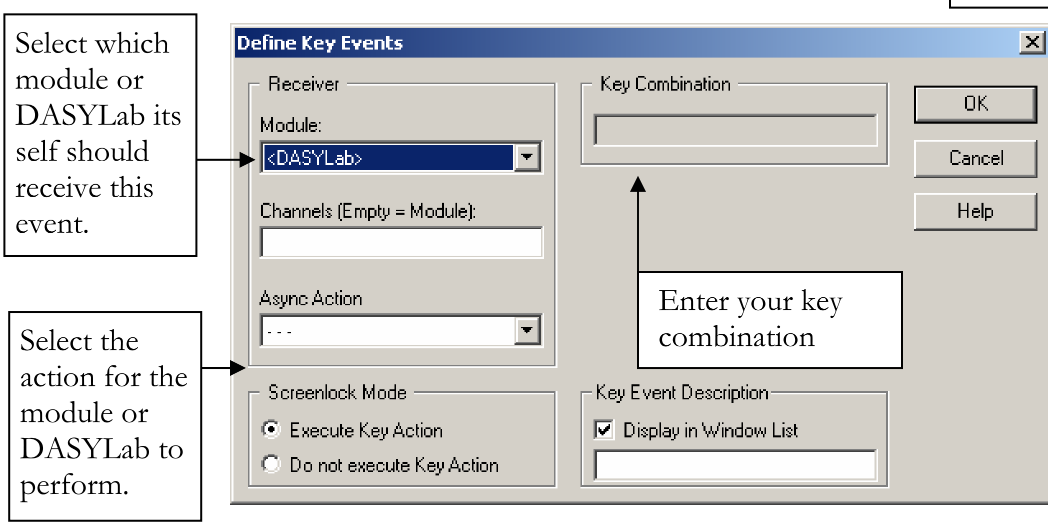
The Up and Down arrows move items either up or down in the "Enter Variable" window.

KEY EVENTS

In **DASY*Lab*** we use keyboard keys or key combinations to perform actions and control applications. To define key events click on Options; Key Events…



When we click on the "New…" button we get this dialog box.



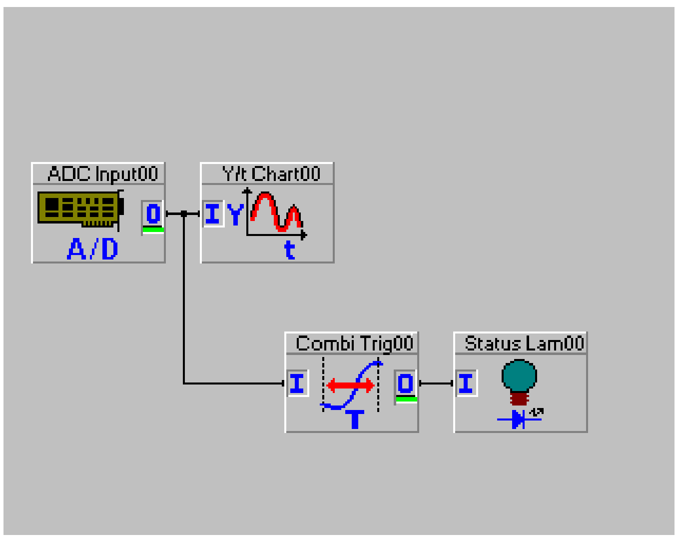
There are a few keys you can't assign to key combinations, for example F1, F5 and ESC.

CHAPTER 8: ADD-ONS AND ADVANCED MODULES

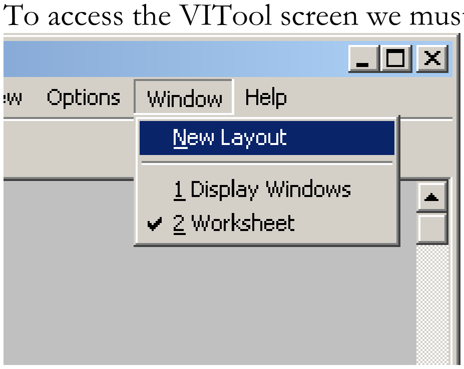
VITOOL

Once a worksheet is created we can now generate a graphical front-end display. This display is also known as a *Virtual Instrument*; in **DASY*Lab*** these virtual instruments are stored in layouts. The **VITool** is used to create these layouts. These virtual layouts are useful for onscreen displays as well as generating reports to be printed.

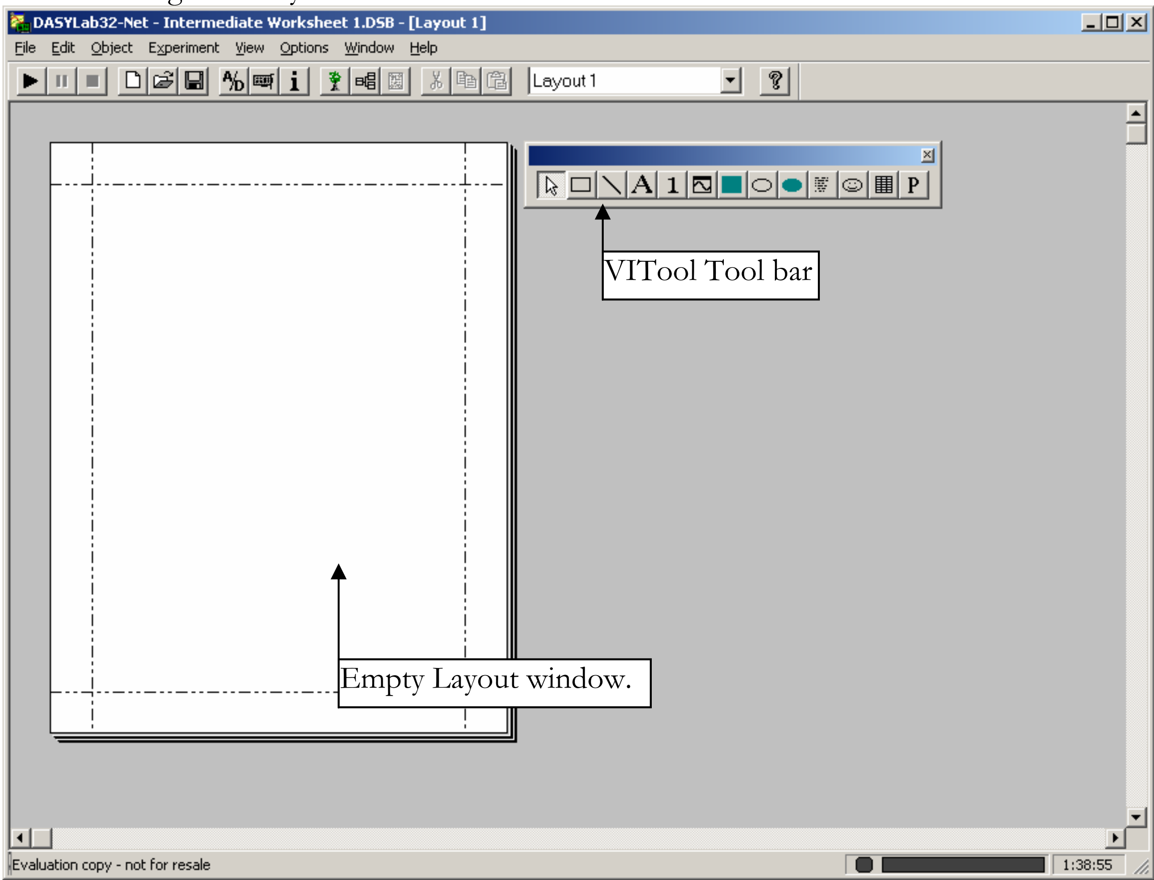
We will create our first Virtual Instrument using out advanced data logger with alarms worksheet we created a few chapters ago. That completed worksheet looked like this:



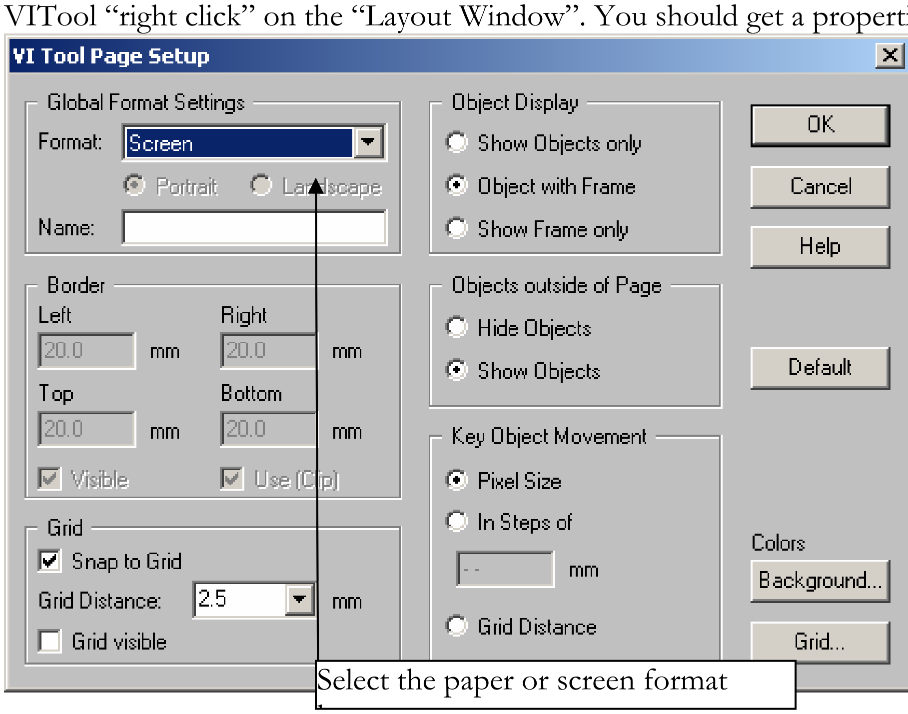
To access the VITool screen we must click on Window, New Layout.



After selecting New Layout we should see a screen as follows:

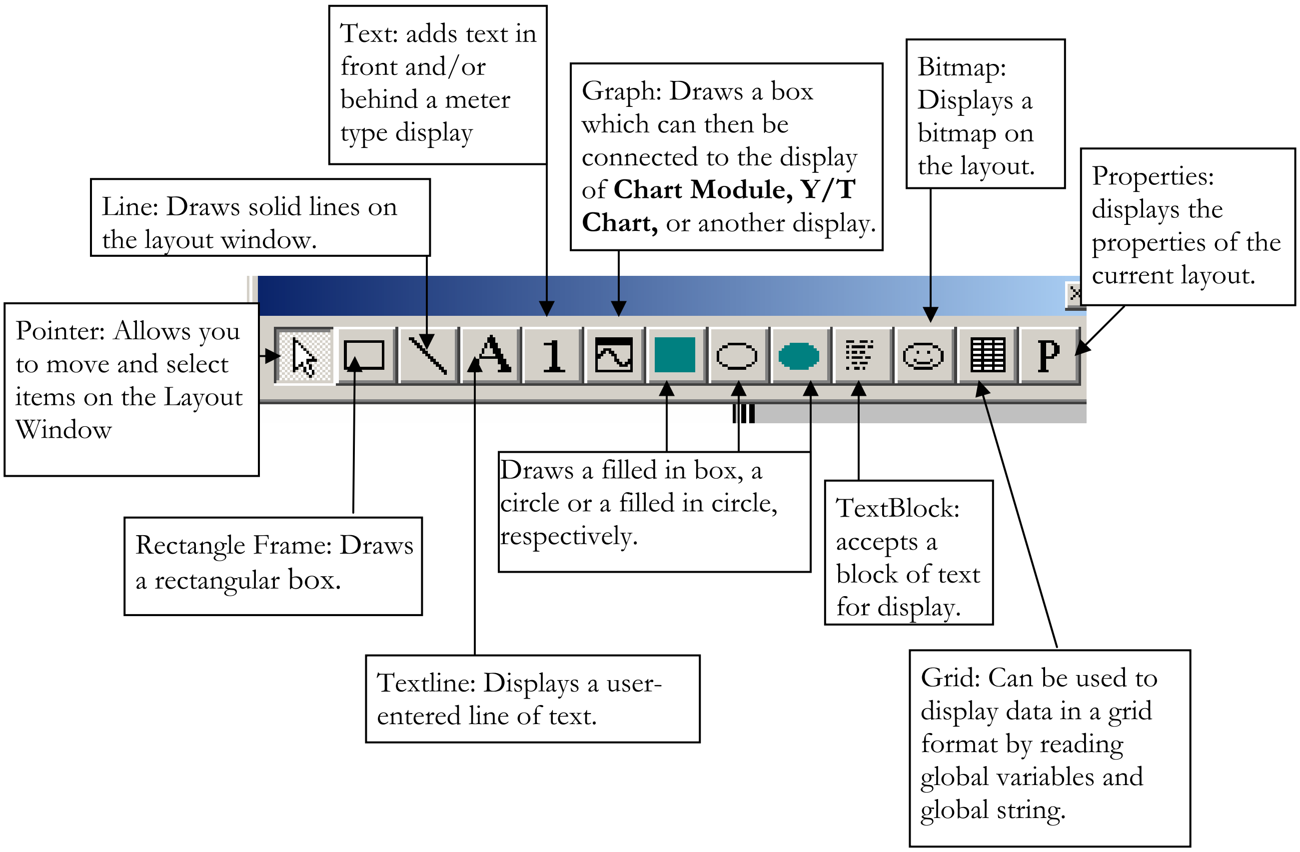


There are several different forms that the layout screen may be. The view above is in "Legal" format designed to be printed on legal size paper. For this experiment we want to be in the "Screen" form so that it appropriately displays on the monitor. To view the properties of the VITool "right click" on the "Layout Window". You should get a properties menu as follows:

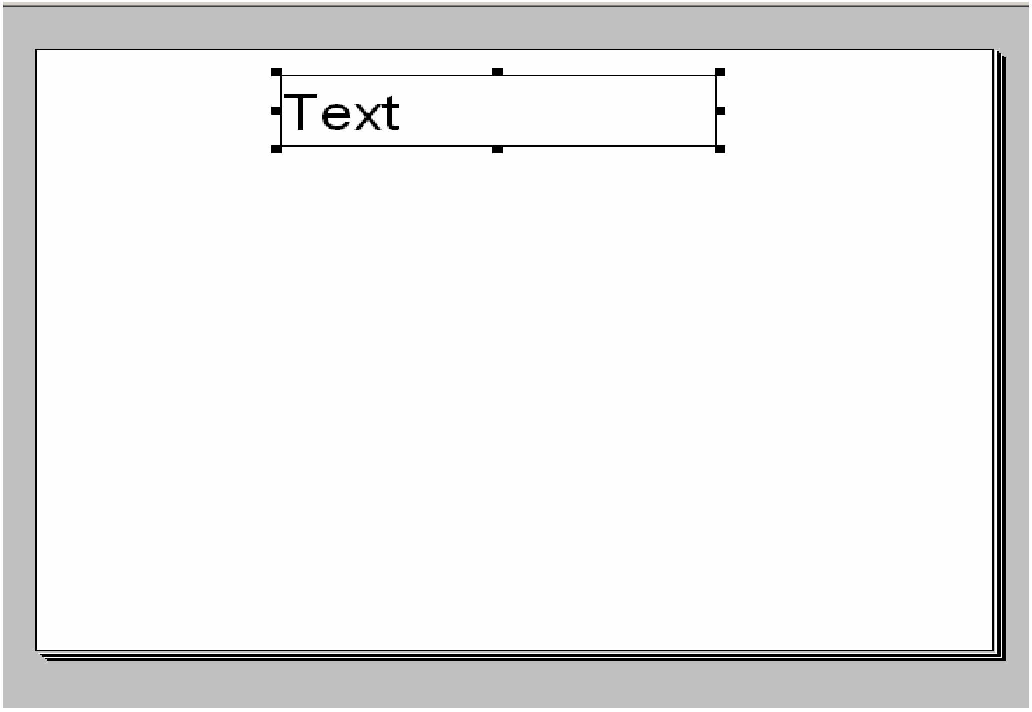


Once Screen is selected, the display should change to reflect the dimensions of a computer monitor.

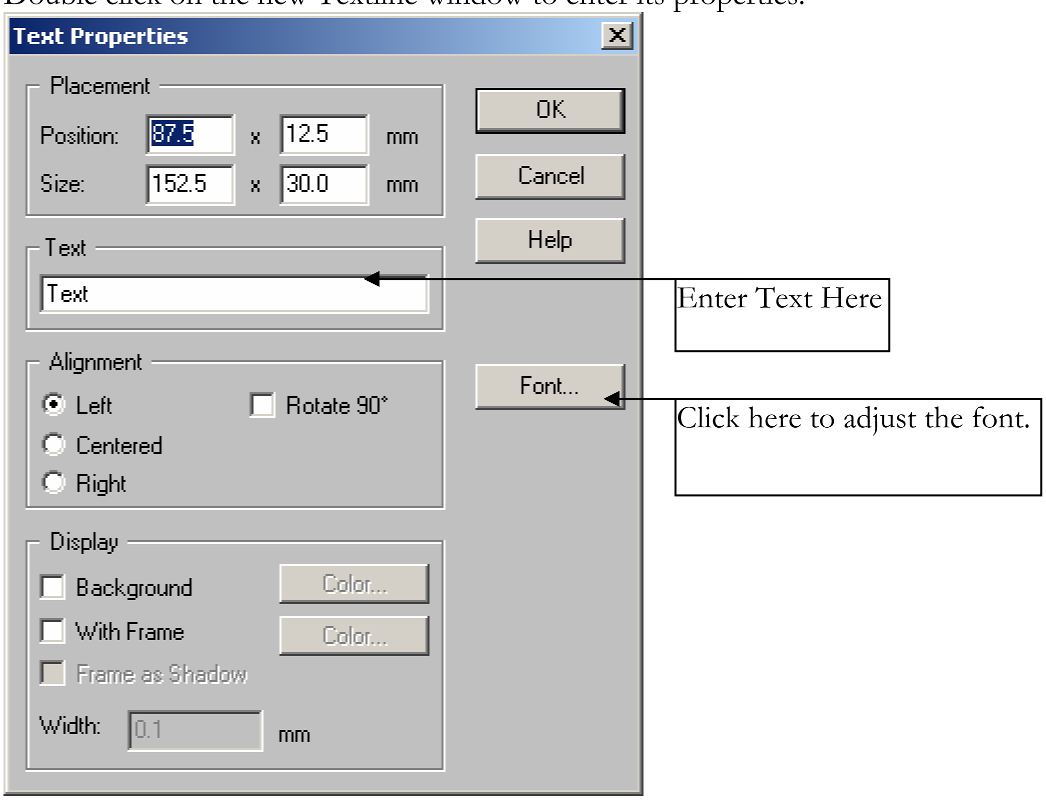
In the VITool we have a new menu bar, the new bar look like this:



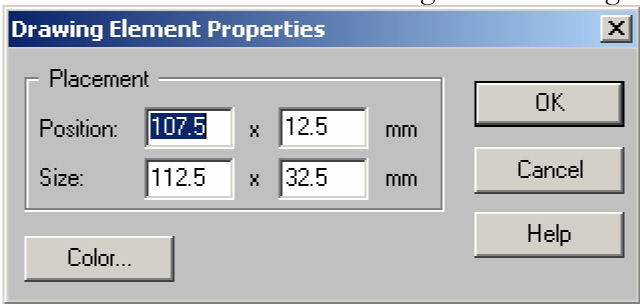
This layout will include a title bar, chart recorder and a status lamp. We will first create the title, click on the *Textline* button and draw a box. The box will look something like this:



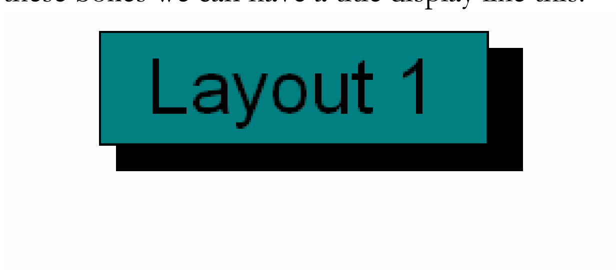
Double click on the new Textline window to enter its properties.



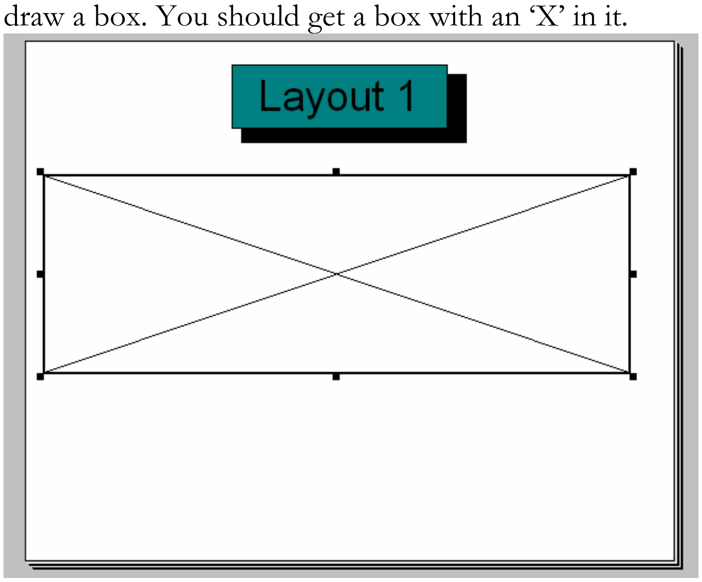
I am simply going to title this Layout 1, however you can be more creative if you so choose. I will also click the "Centered" radio button to center the text in the Textline window. I would also like a background around my title so I will select the Filled in box and drag a box over the title. After double clicking on the box I get a properties menu:



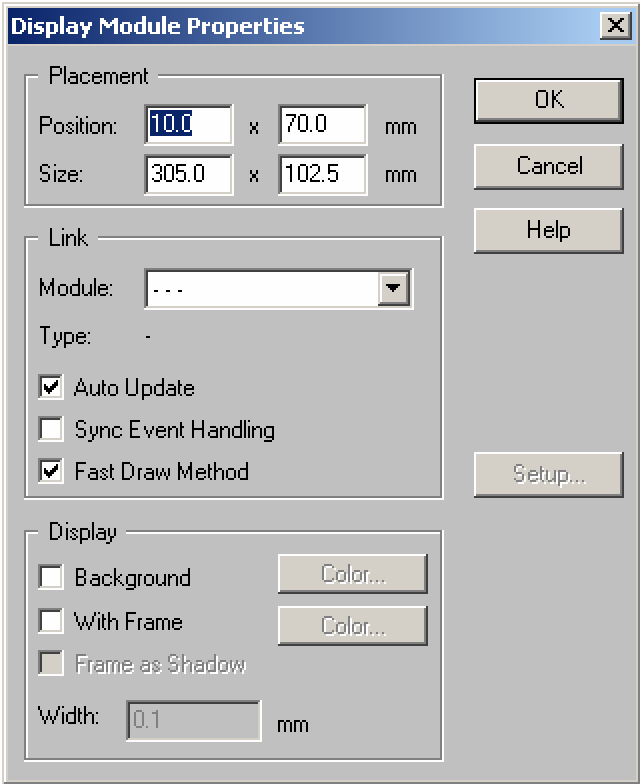
We can click on the color button and select the color of our box. Then click OK. To send the box behind out text "right click" on the box and select "Down" to send it one level lower or "To Background" to send it to the bottom. With the careful application of two of these boxes we can have a title display like this:



We can now place our data displays on the Layout as well. Click on the "Graph" button and draw a box. You should get a box with an 'X' in it.

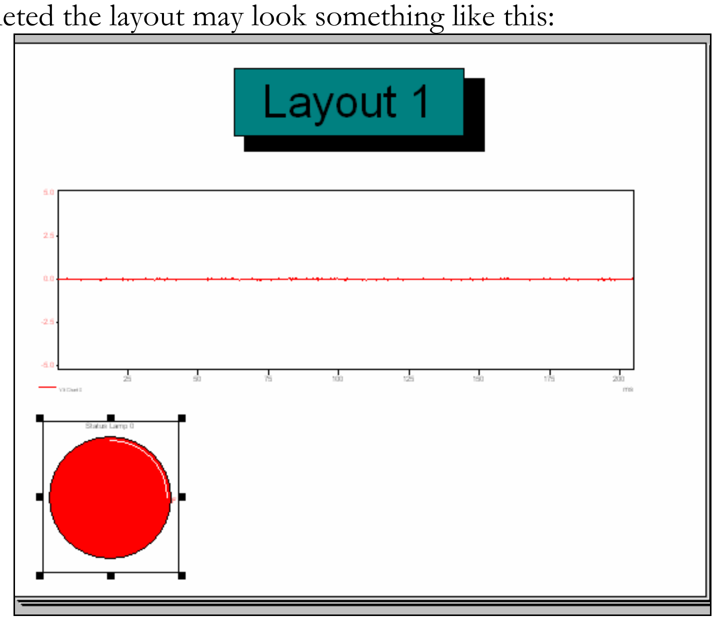


Double clicking on the Graph box will give us the properties of the box:

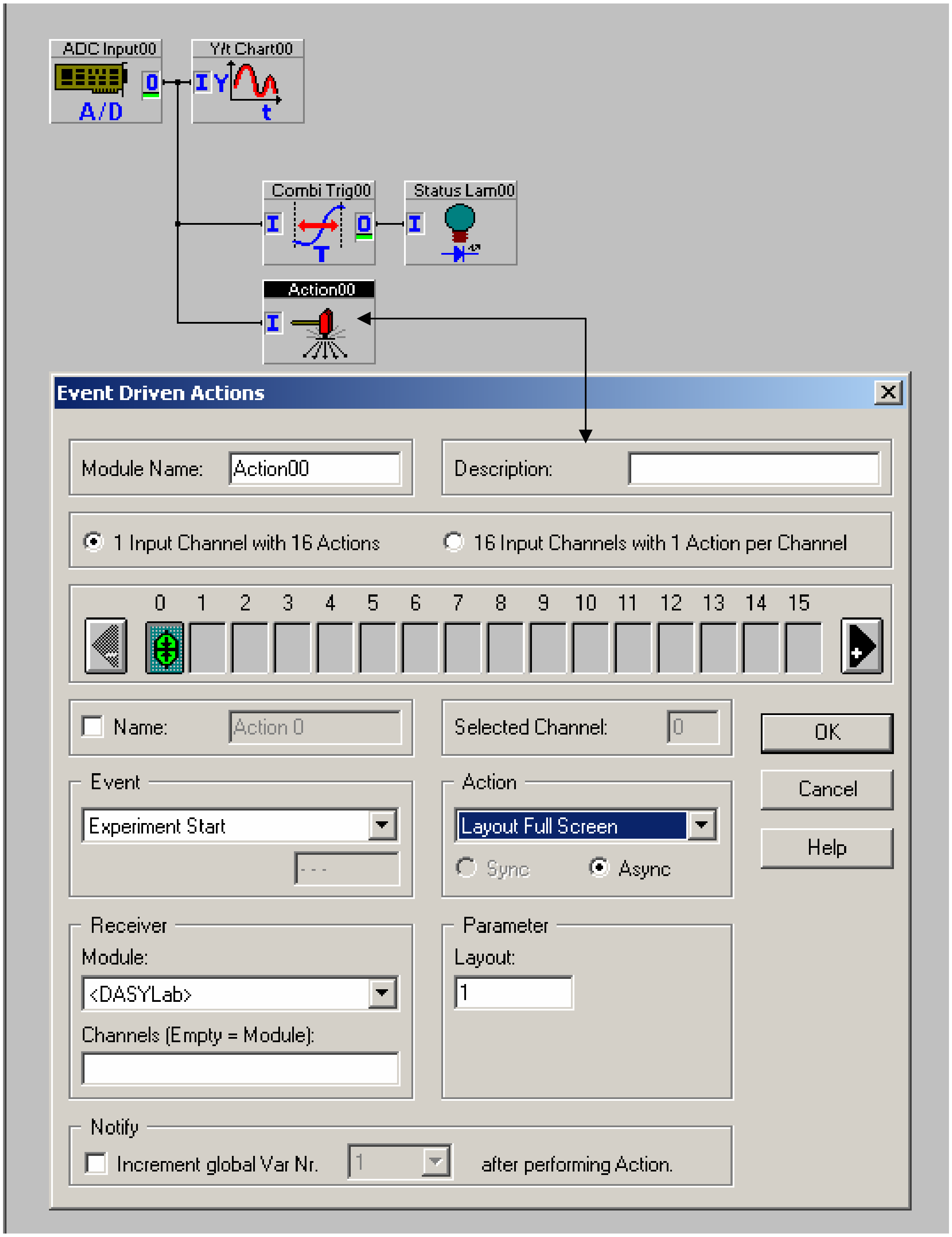


By clicking on the "Module:" dropdown we get a list of all the modules that are available with displays. Select "Y/t chart00" then click OK. The layout should now have out title and a Y/t chart display where we had out Graph Box. This box is now resizable and configurable just like the Y/t Chart it's self, to change its settings, double click on the Y/t chart and click on "Setup…"

Repeat this process again, however this time select the Status lamp for the display. Completed the layout may look something like this:



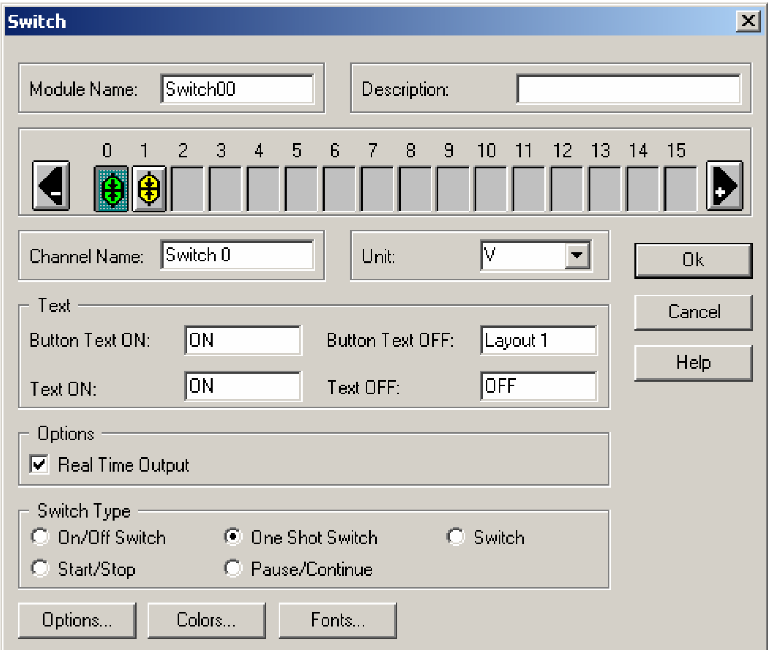
If we run the worksheet we should see that data displayed on the screen in out Layout. If the CRTL and F keys are pressed then the display should go "Full Screen", pressing the ESC key will return the screen to normal. We can automate the full screen process by using an action module in the worksheet as follows:



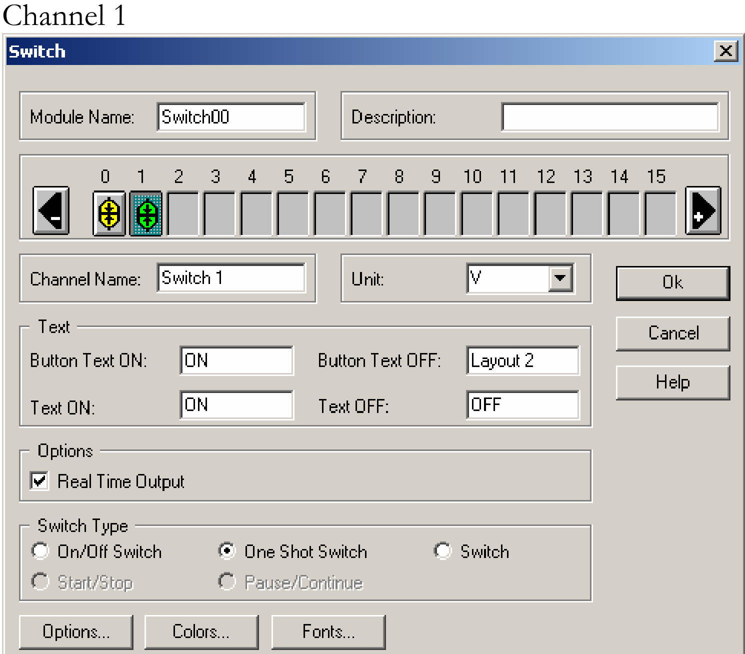
This module can be connected anywhere inside the worksheet. Note the "Experiment Start" condition and the "Layout Full Screen" action with the Parameter of which layout to display.

A layout screen is limited to 1000 objects, and in many instances the display gets very crowded looking long before that point. We can create a menu screen, which will allow us to select from many different layouts and navigate through them all. First, however, we will need another Layout. Click on windows and select "New Layout" to create a blank layout.

Back at the worksheet we will need a **Switch Module** and an **Action Module**. To control two layouts the **Switch Module** settings should look as follows:
Channel 0:

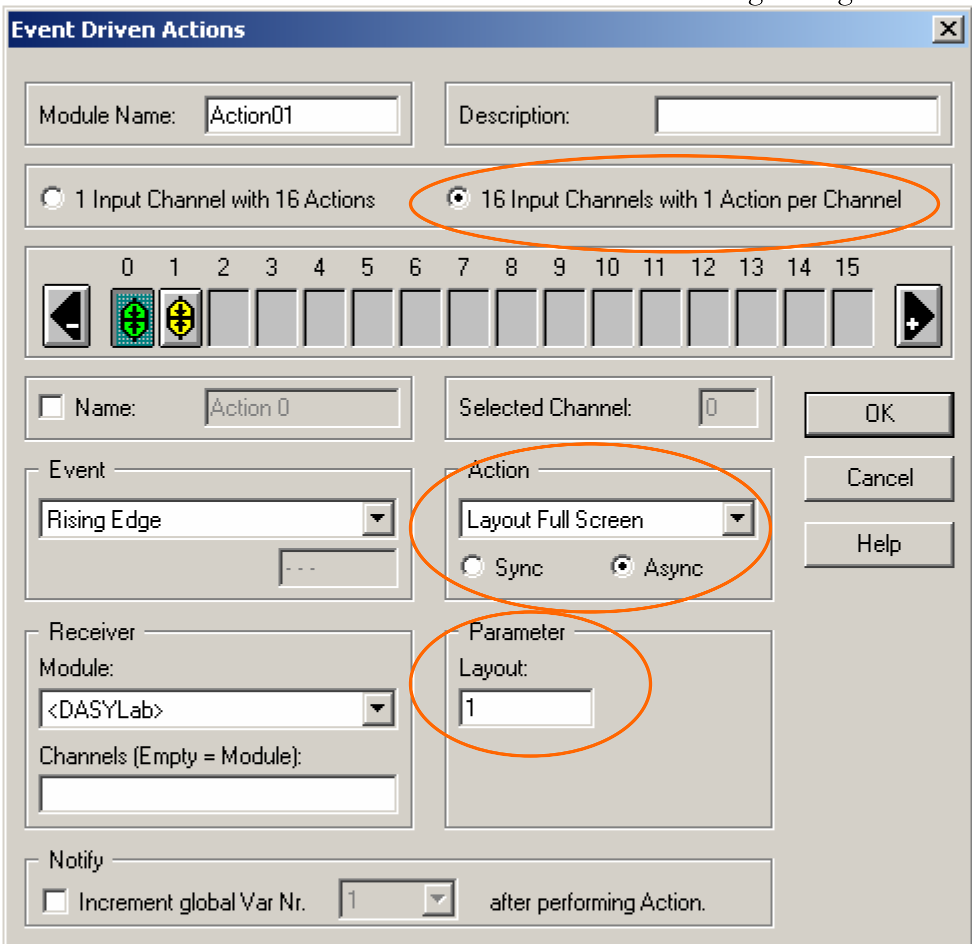


Channel 1



Note that both switches are "One Shot" and the Button Text OFF is the titles of the layout we wish to see.

Connect this switch to an **Action Module** with the following settings:

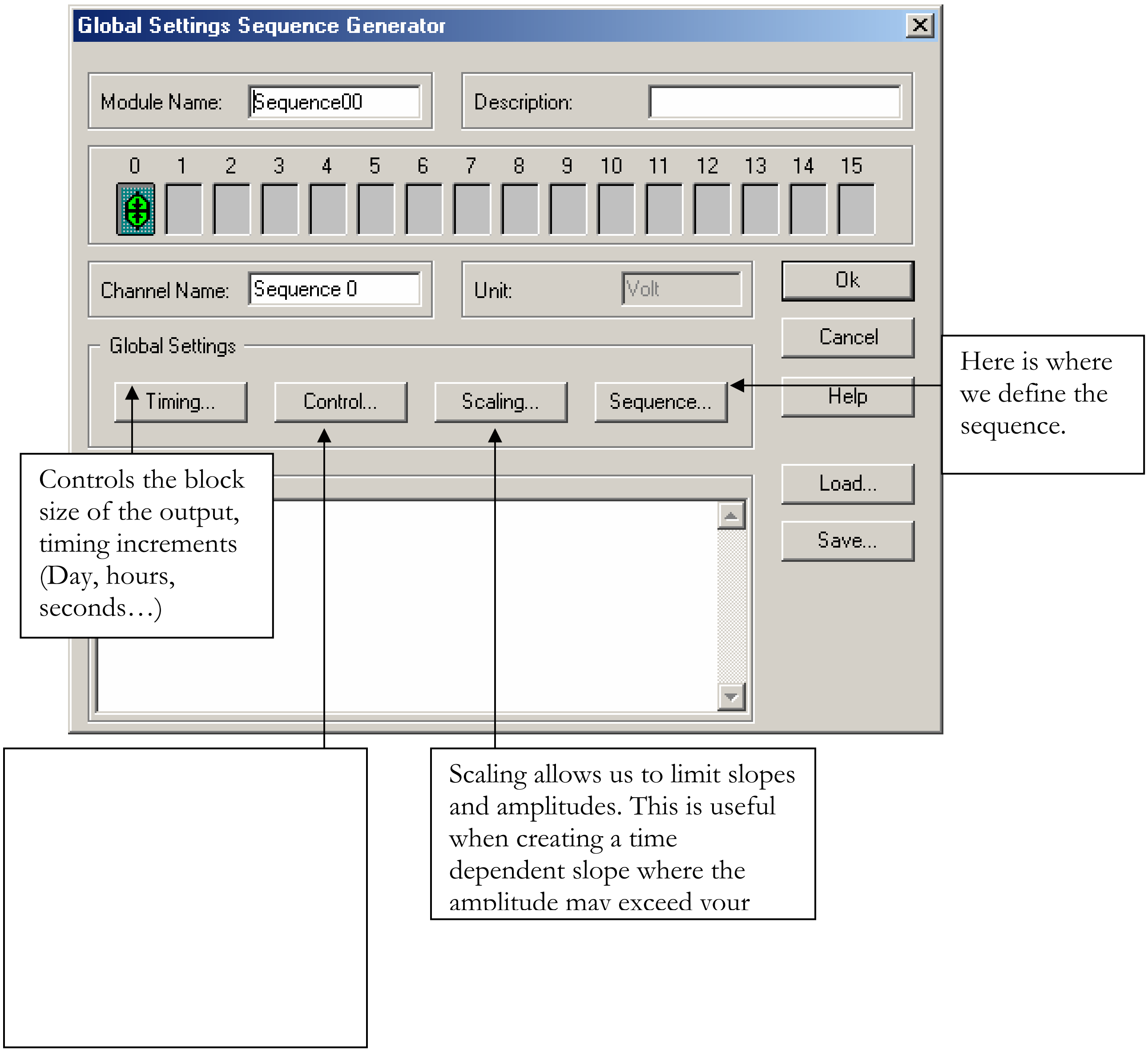


The only change made for channel 1 is that the "Parameter" is Layout 2.

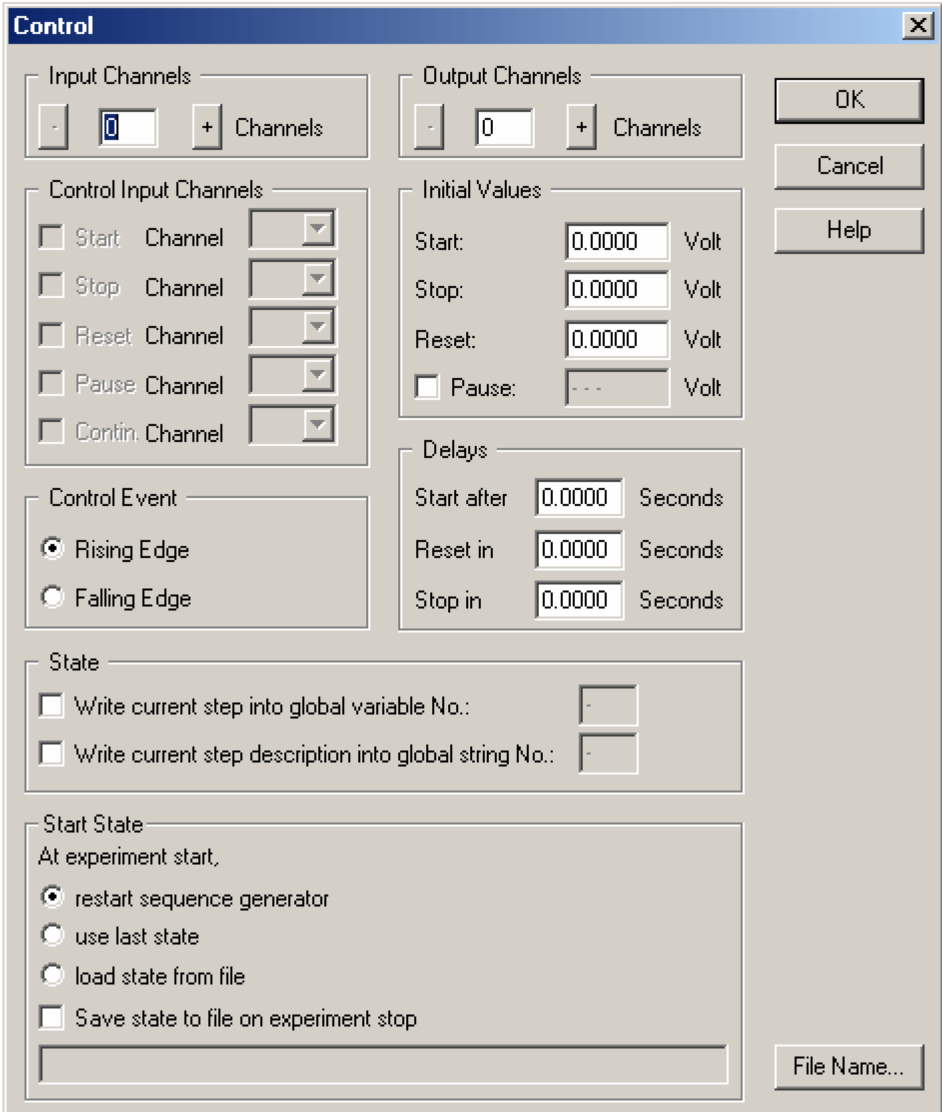
If we place the module Switch00 in both of our Layout windows then it is possible to toggle between the two windows. By adding more channels to both the **Switch Module** and the **Action Module** we can change up to 16 different displays before we add another **Switch Module** and **Action Module** pair. **DASY*Lab*** can handle up to 200 Virtual Instrument Layout Windows.

SEQUENCE GENERATOR

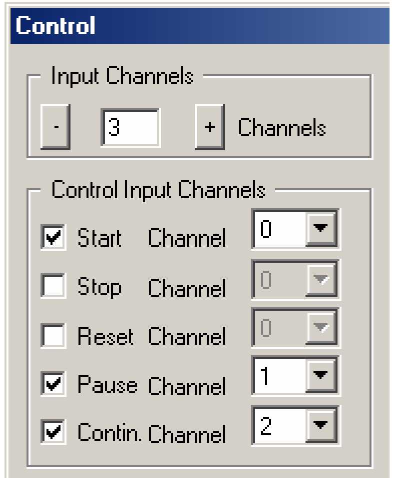
The **Sequence Generator Module** allows us to create a user-defined waveform. Common applications for this module are timed "Ramp and Soak" applications as well as waveform generation. The **Sequence Generator Module** can create two Analog waveforms and 14 digital waveforms. This module can also accept inputs for timing. The **Sequence Generator** is found under the Controls Menu. The properties of the module look like this:



The first step is creating the sequence is to define the controls, click on the "Control…" button:



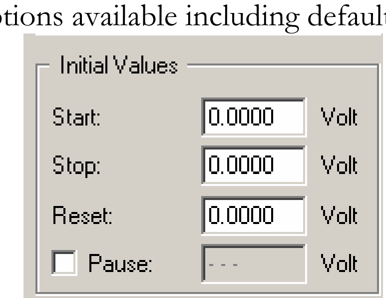
If you wish to have control inputs then we need to increase the number of inputs. In this example we will have a start, pause and continue inputs. When configured that section should look like this:



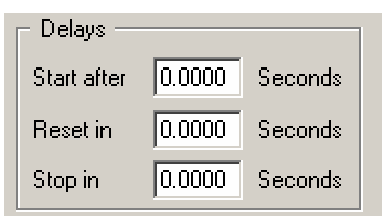
Note that I have increased the inputs to three and have assigned each of the functions I wish to use to a channel. These numbers correspond to the channel number on the LEFT side of the module. The module will change shape and add these channels when you click ok to close the properties menu.

Next in this example I'm going to control 4 digital outputs also. Therefore we need to increase the number of outputs to 5 (1 analog and 4 digital). The number displayed indicates the number of *additional* channels, one channel is assumed.

There are several advanced options available including default values:

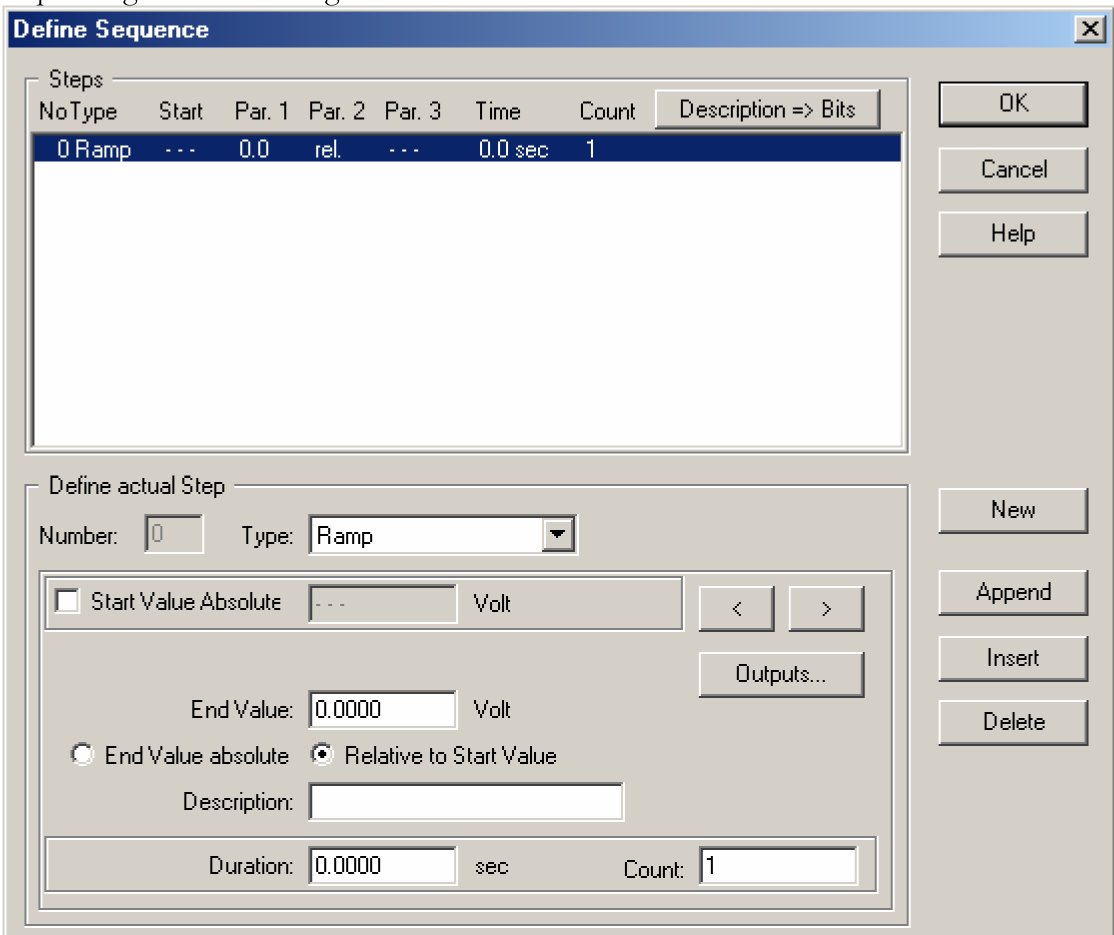


and delays:



We won't be using any of these today, however they may be useful in other applications. We can click OK to close this box.

We can now define out Sequence. When we click the "Sequence…" button we get the sequence generation configuration box:

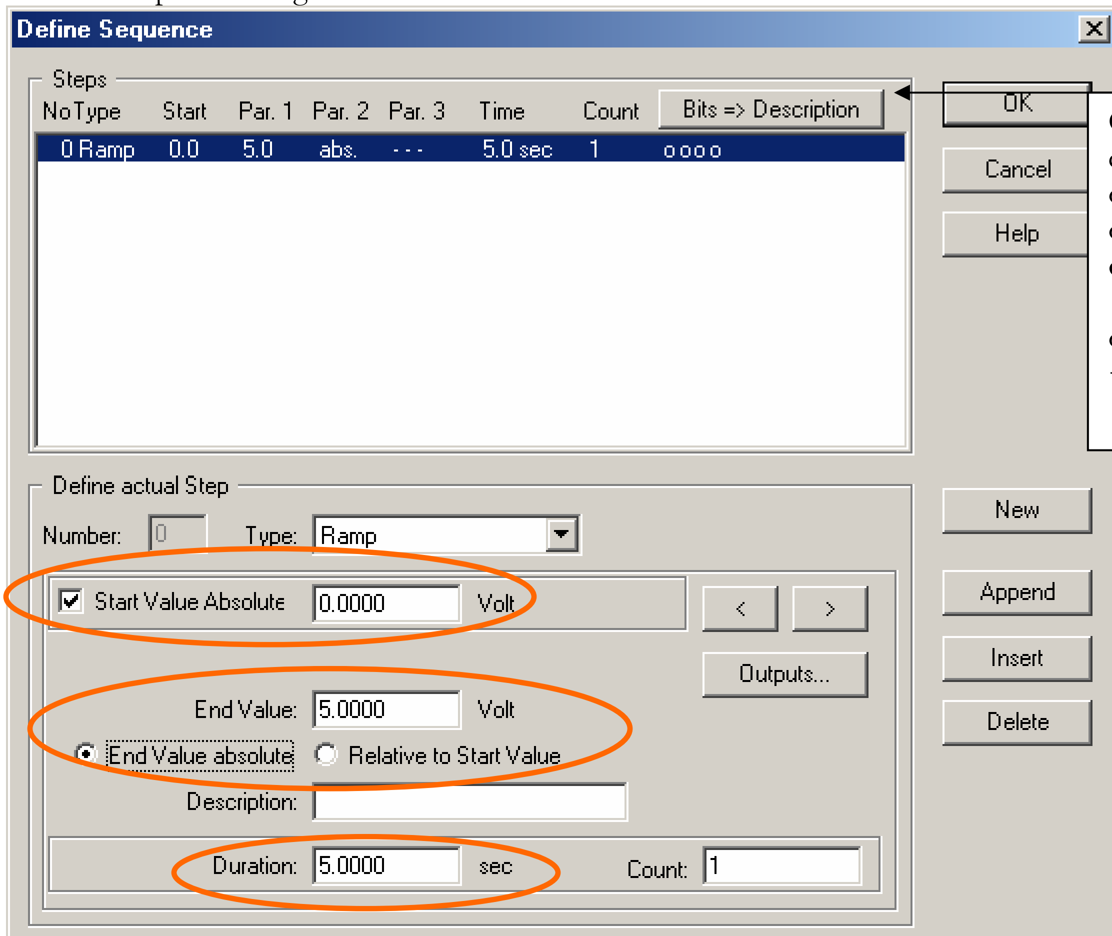


Here are a few important things to note about setting up a sequence.

- If you want to stop or start at an exact value then you are interested in an *Absolute* value.
- Durations and values can be either constants or global variables

The waveform we will be creating will ramp form 0 to 5 and hold there for 3 seconds, then produce 1 SINE wave and ramp back to 0 in 10 seconds. Also when we start we will turn on 2 digital output, toggle them at the hold and turn them all off when we reach 0 again.

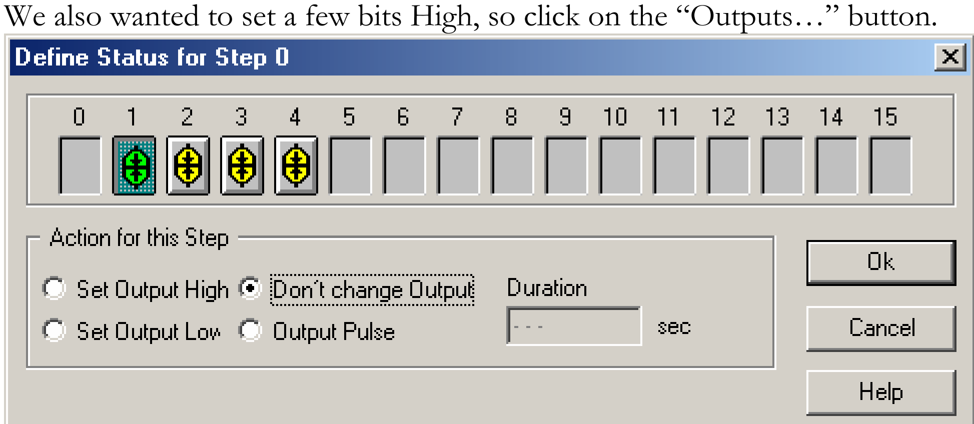
The first step is defining the RAMP from 0 to 5:



Click here to display the state of the digital outputs we created.

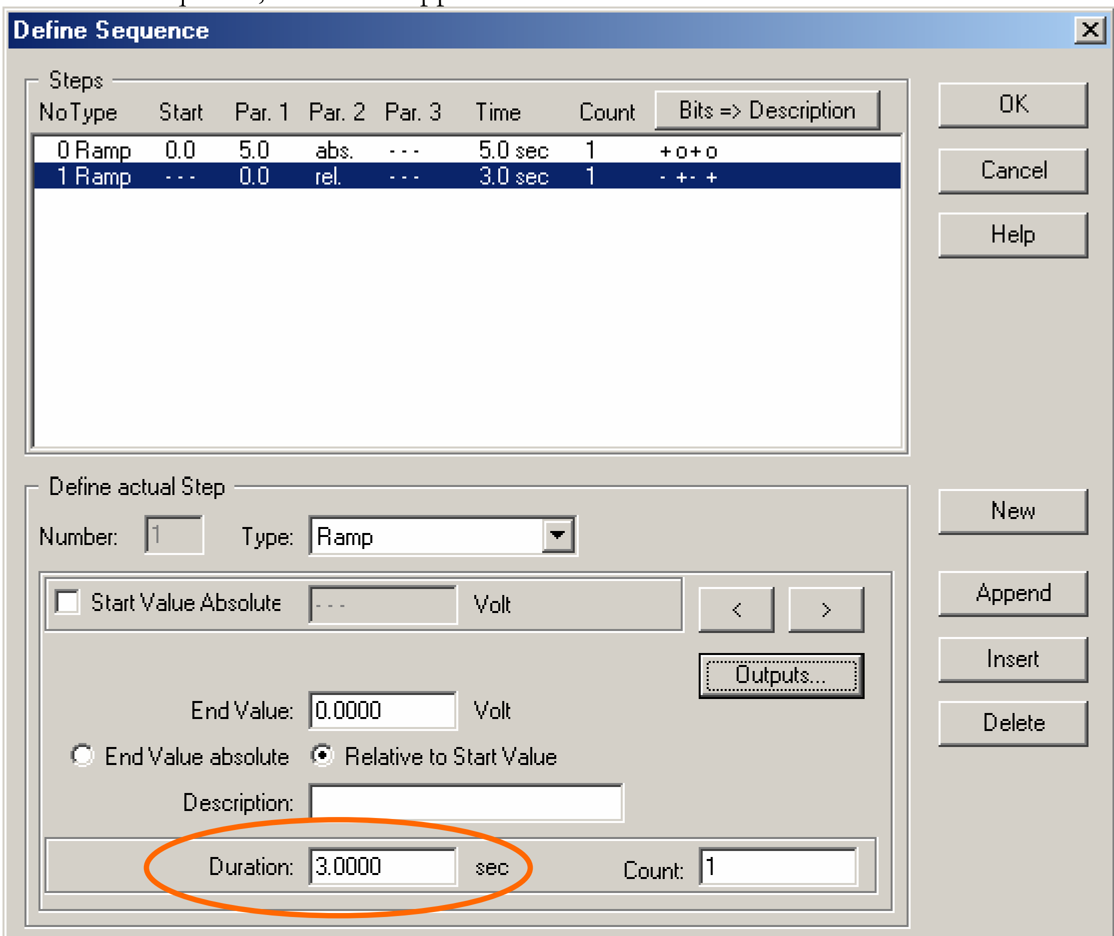
o = no change
\+ = High
\- = Low

We also wanted to set a few bits High, so click on the "Outputs…" button.



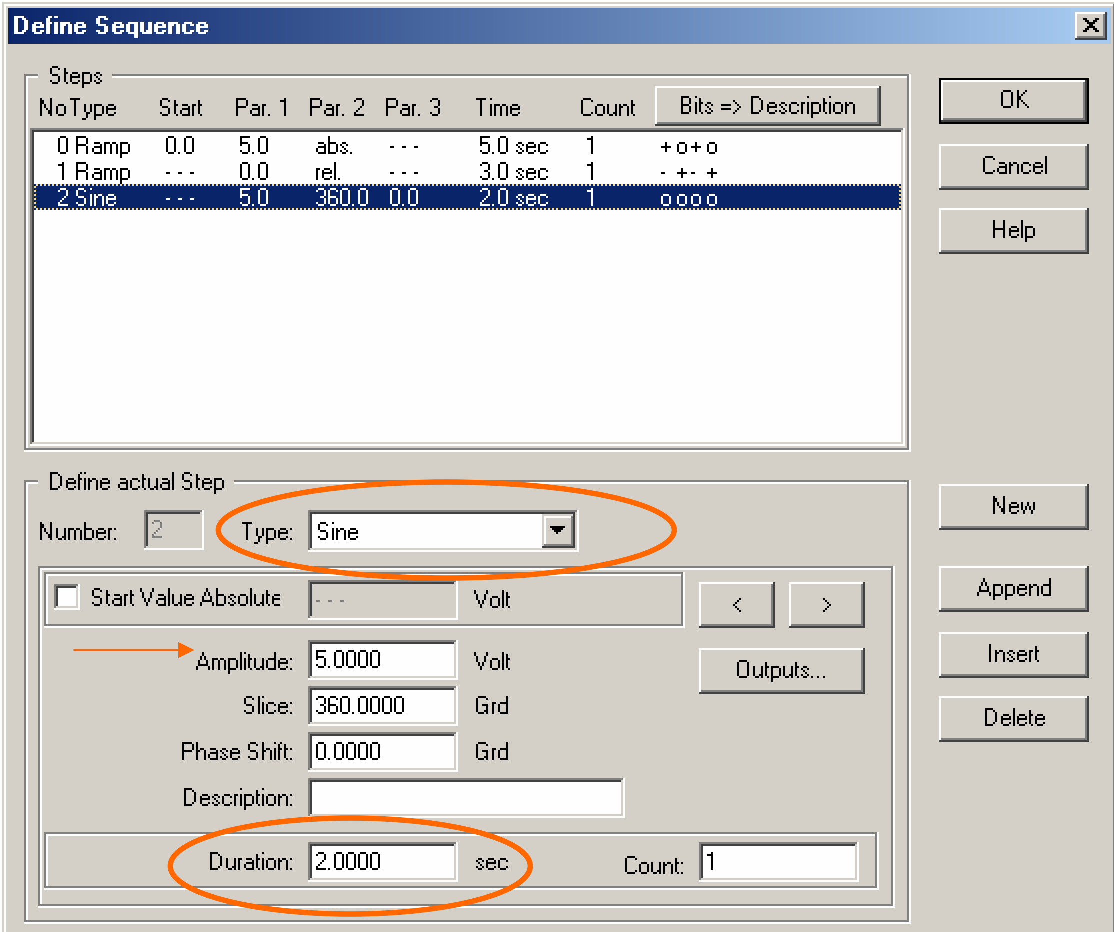
Select a channel and select a state for this step in out generator. I'm going to set bits 1 and 3 high.

Next we will generate a 3 second hold or "soak" and toggle the bits. To add a step to the end of our Sequence, click the "Append" button.

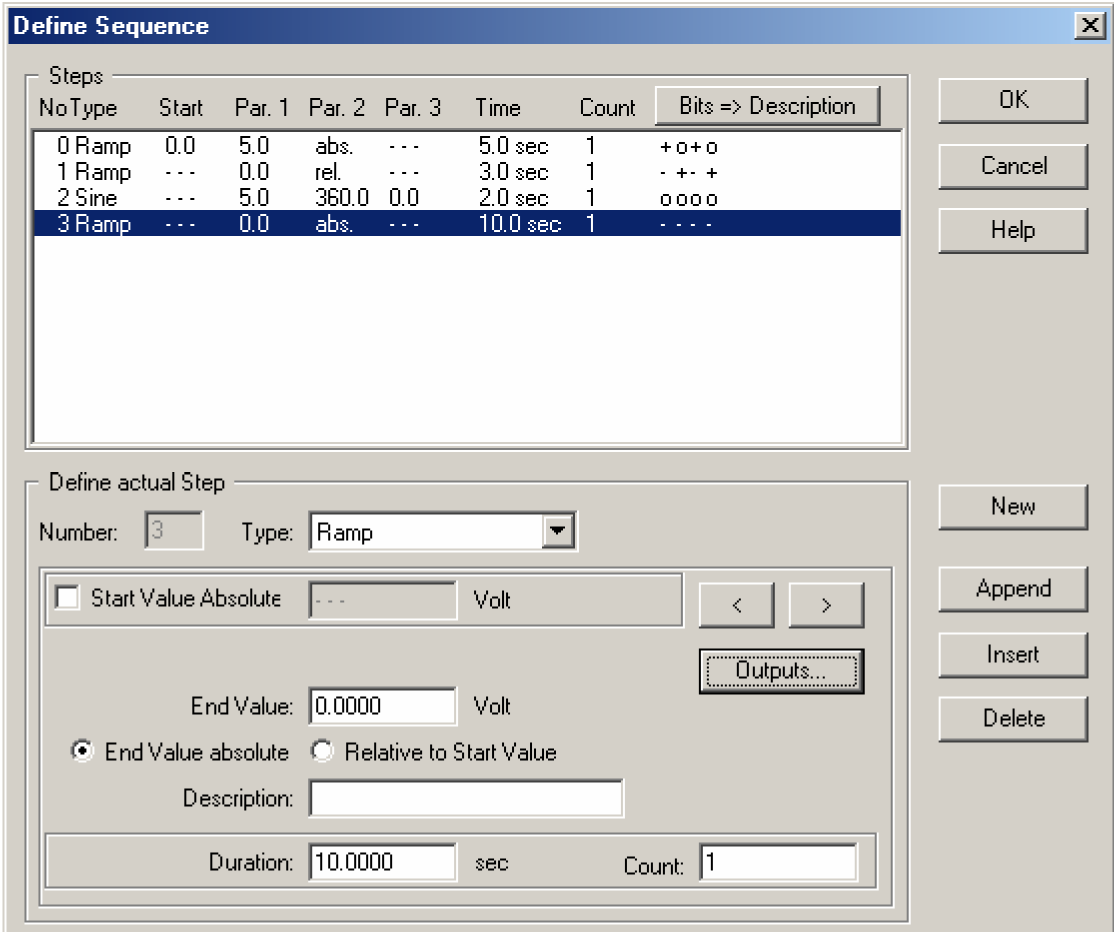


You may notice that the "End Value" is 0, however it is *relative* to the start value of 5 produces no change. Also click on the "Outputs…" button and change the state of the digital outputs. The next step we will add a SINE wave to the output.

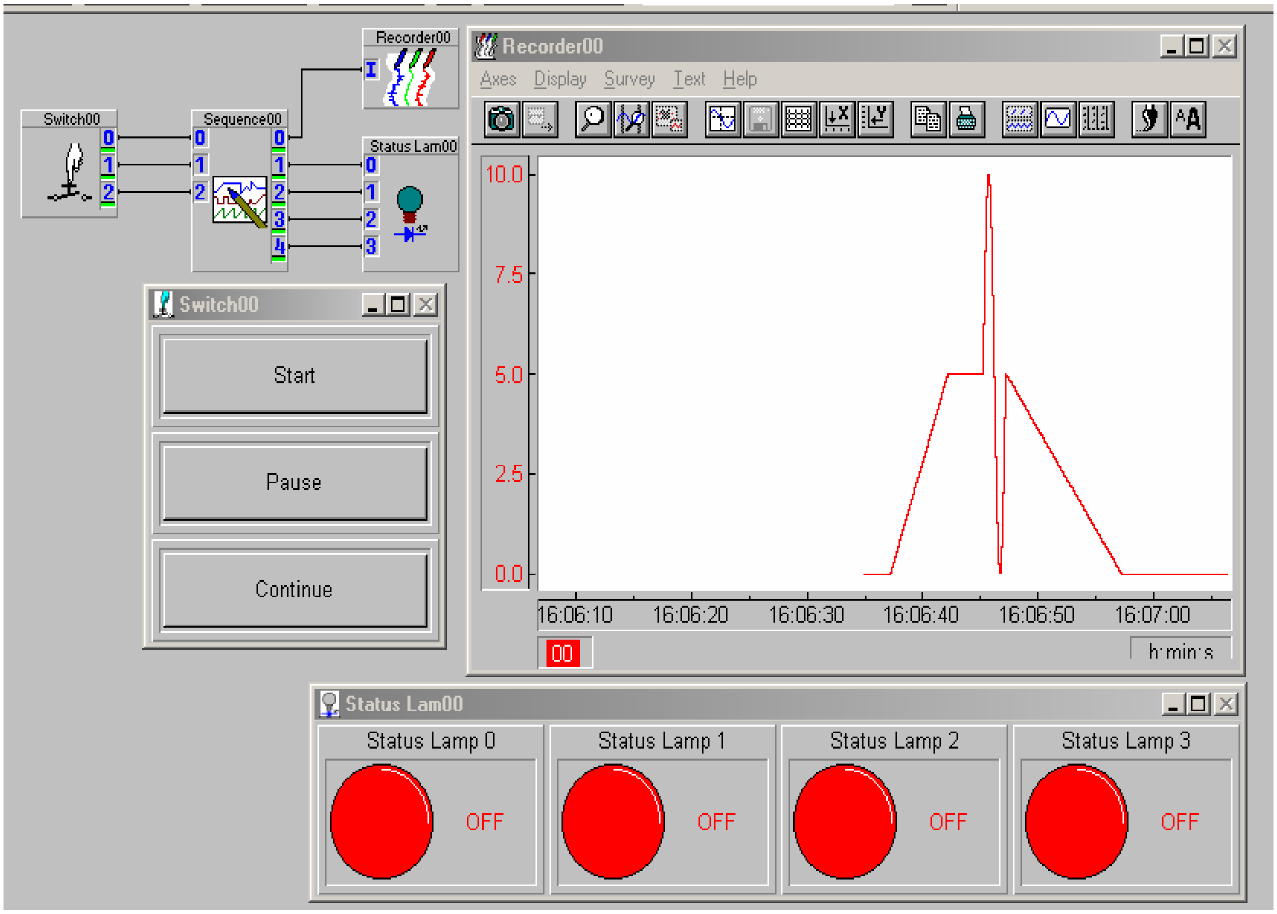
Creating a SINE wave works the same as creating the RAMP, however we must change the type from RAMP to SINE then define the period of the wave under duration. The completed setup:



If we wished to have more then 1 period of a SINE wave then all we must do is increase the “Count” option. Lastly we want to ramp back down to 0 and turn all the digital outputs to off. By appending another step to this waveform and configure it appropriately.



Notice that the "End Value absolute" option is selected and that I have set all the digital outputs to 0. We are now ready to use this **Sequence Generator** in our worksheet.



Notice that I have connected a **Switch Module** to the input of the **Sequence Generator Module**; these switches correspond to the three controls we included in our module. I have connected a **Chart Recorder Module** to output 0 (displaying 0 to 10 on the X axis) and a set of **Status Lamps** to the rest of the outputs. As you can see the wave for is as we defined it and when running the status lamps blink as expected.

DASYLAB RUNTIME

Not all applications created in **DASY*Lab*** are meant for internal use or to be modified. It is also expensive to distribute full copies of **DASY*Lab*** to your customers, especially if they don't need to change the worksheet. **DASY*Lab*** has a version called Runtime, this version has all the power and performance of **DASY*Lab*** PLUS with VITool however it lacks the development ability. What you get with **DASY*Lab*** Runtime is a copy of **DASY*Lab*** that can load and run worksheet, however no changes can be made.

DASY*Lab* Runtime requires its own serial number, which can be purchased from your local **DASY*Lab*** distributor.

SETTING UP AN AUTO-START SYSTEM

Many times we want the computer to boot right into **DASY*Lab*** and start running the worksheet. This allows the system to be fault tolerant, recovering from power failures, or even an automatic restart. The first step is to open a completed worksheet in a development version of **DASY*Lab***. Select the "Auto Start" option found under the "Experiment" menu, and save the worksheet. This worksheet is now ready for distribution.

On the target system, install **DASY*Lab*** (Development or Runtime). Place the **DASY*Lab*** worksheet in the "START UP" folder found under the Windows Program Menu. The items in this folder are run when the computer starts.

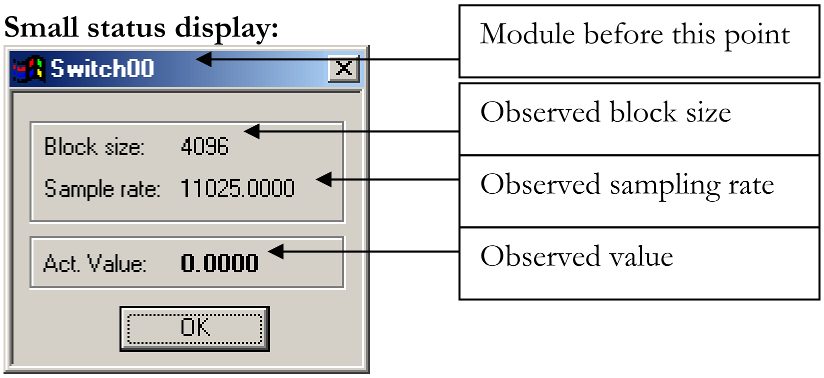
Note: be sure that your computer associates the **DASY*Lab*** file extensions (.DSB and .DSA) with the **DASY*Lab*** application. The **DASY*Lab*** installation will do this, but you must choose CUSTOM install and select the option.

Be sure to test the system before distributing. Sometimes subtle changes exist between two computers, which might make the hardware not communicate appropriately. This is best discovered in the home office and not on the customer site.

DEBUGGING IN DASYLAB

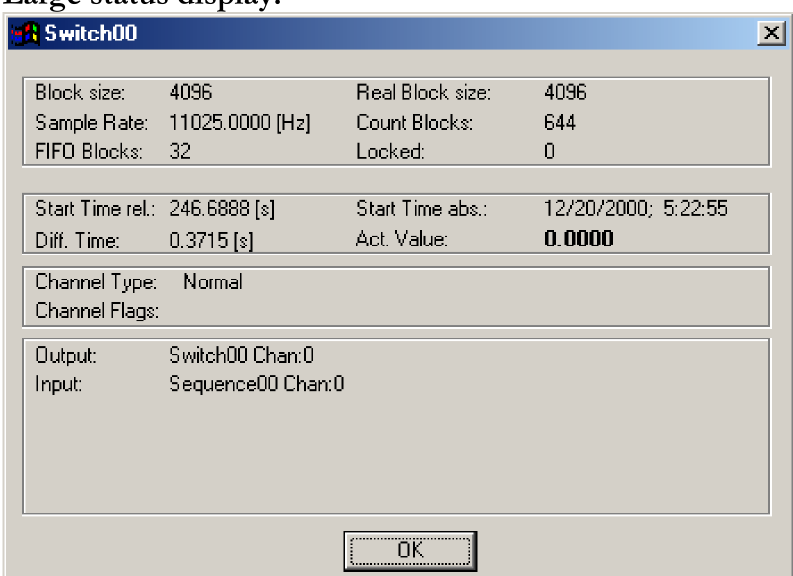
DASY*Lab* comes equipped with different debugging tools to assist you in finding and resolving any data flow or logic errors. These tools are only available while **DASY*Lab*** is in the Acquisition mode, after the play button is clicked.

Small status display:



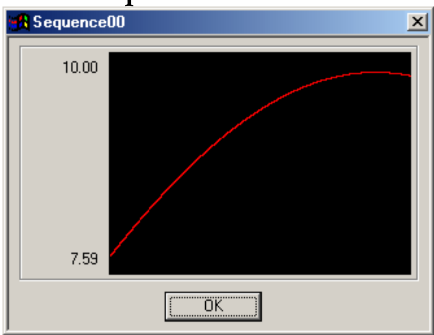
To access this display simply click on a wire in **DASY*Lab***.

Large status display:



This tool shows the same information as the previous tool, however it goes into more depth. This display shows not only the observed values but the global values as well. To access this display, hold down the CONTROL key while clicking on a wire. (CTRL-Left Click)

Mini Graph:



This graph shows 1 block of data at a time. The graph is also self-scaling to always show the peak-to-peak value. To access this tool, hold down the SHIFT key and click on a wire.

Additional tools:

System Global Variables: there are several System Variables available to assist you in determining how your worksheet is running. Information such as Total Delay and Total Load may be displayed. Additionally, the Time Base Module (Plus version) allows you to display the current "Distance to Real time" – an indicator of how long it takes the data from the Data Acquisition device to reach the Time Base Module.

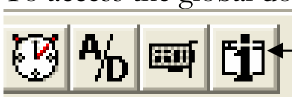
CHAPTER 10: DOCUMENTATION

There are several methods available in **DASY*Lab*** to document your worksheets. There are methods to document the whole worksheet, individual modules, and the graphical worksheet. This chapter will explain the common methods of creating and displaying this documentation.

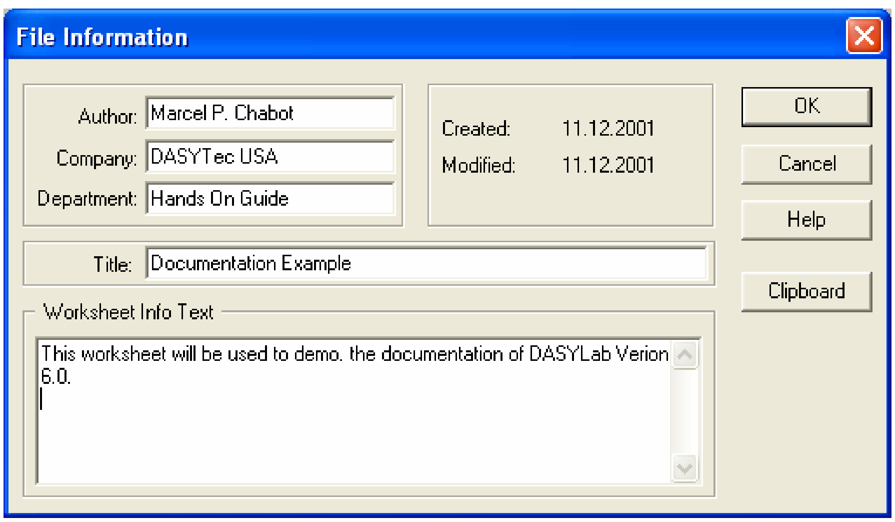
GLOBAL DOCUMENTATION

Global documentation is for the entire worksheet. Here you should explain who made the worksheet and which company and department was involved. In the global documentation you can also give a brief explanation of the worksheet.

To access the global documentation click on the "File Info" button:



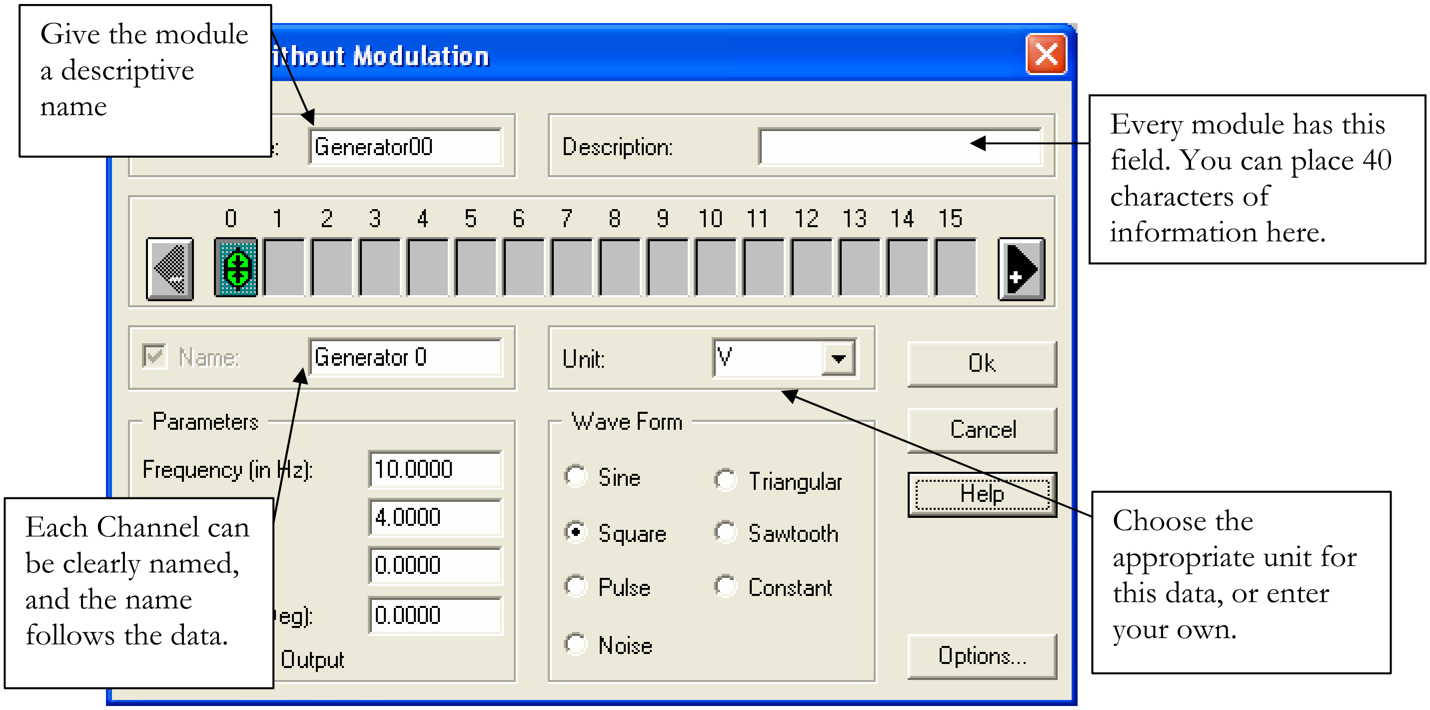
Click here to access the "File Info" dialog box.



This information will appear at the top of each saved file, in the header of several data files as well as in all documentation printouts.

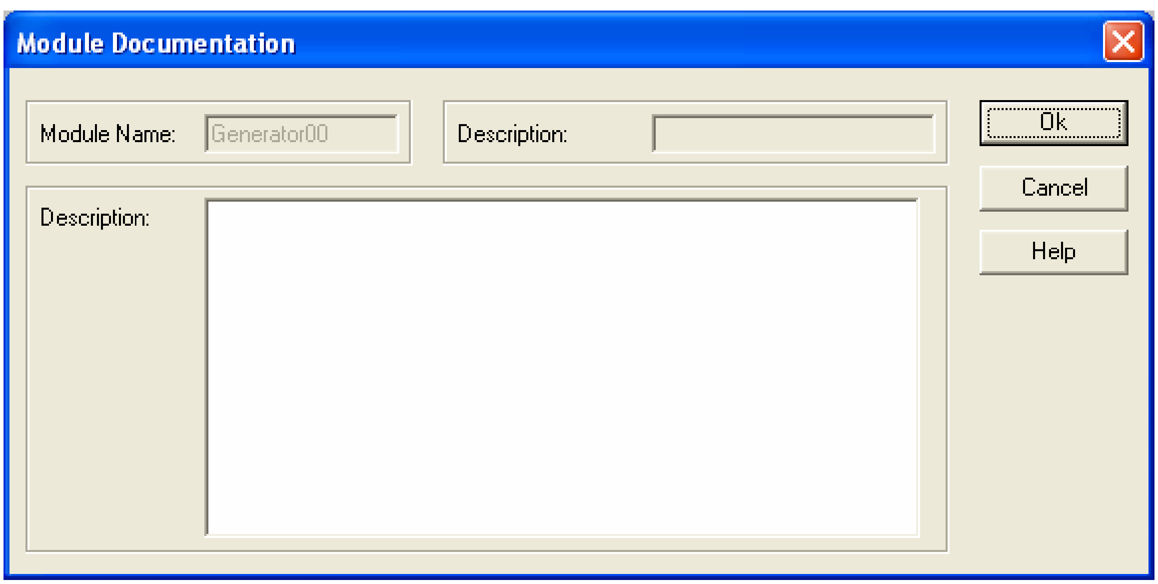
MODULE DOCUMENTATION

Each module in **DASY*Lab*** has the ability to store documentation about its function. This documentation is will be printed out with the worksheet documentation. There are two places to store this information; the first is in the module description



In addition, use the Module Name, Channel Name, and Unit fields to document your data.

If you wish to add more documentation to your **DASY*Lab*** module you can *right click* on the module and select "Module Documentation" to access this dialog box:

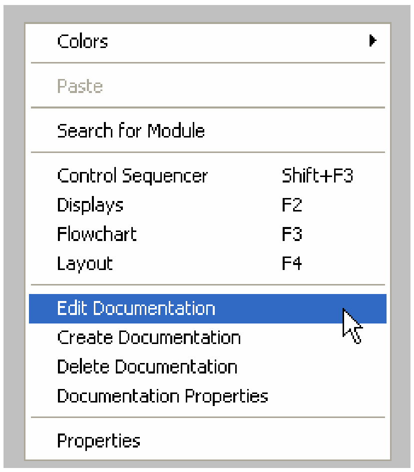


This dialog box can hold 256 characters.

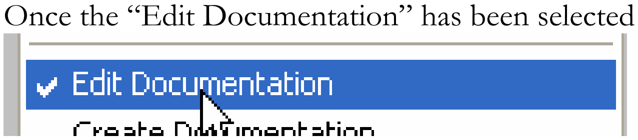
WORKSHEET DOCUMENTATION

There is another layer of documentation available in **DASY*Lab*** version 6.0. This layer allows you to draw squares, assign them a color and add text into each square. This graphical documentation is placed under the modules. This allows you to graphically segment your worksheet and document blocks of logic. This documentation is not saved in to the documentation Text file

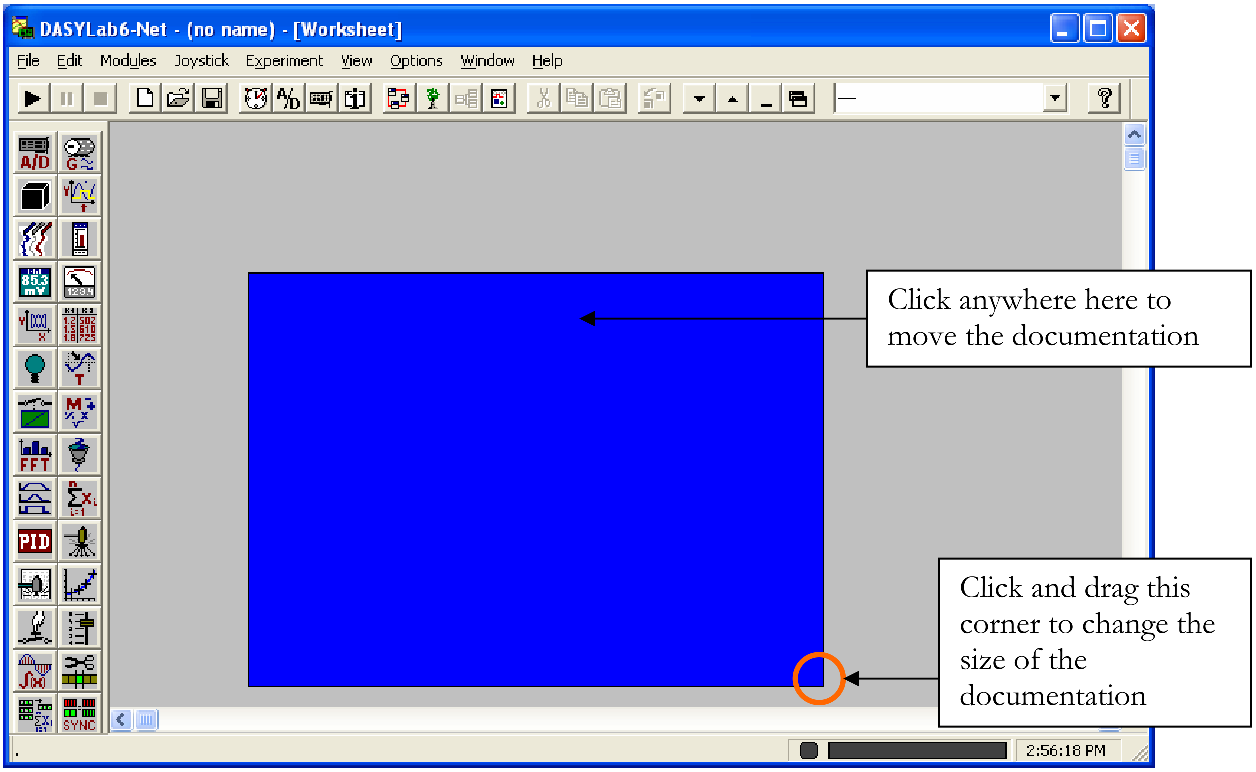
To access the documentation layer; right click on an open area of the screen to get this drop down menu.



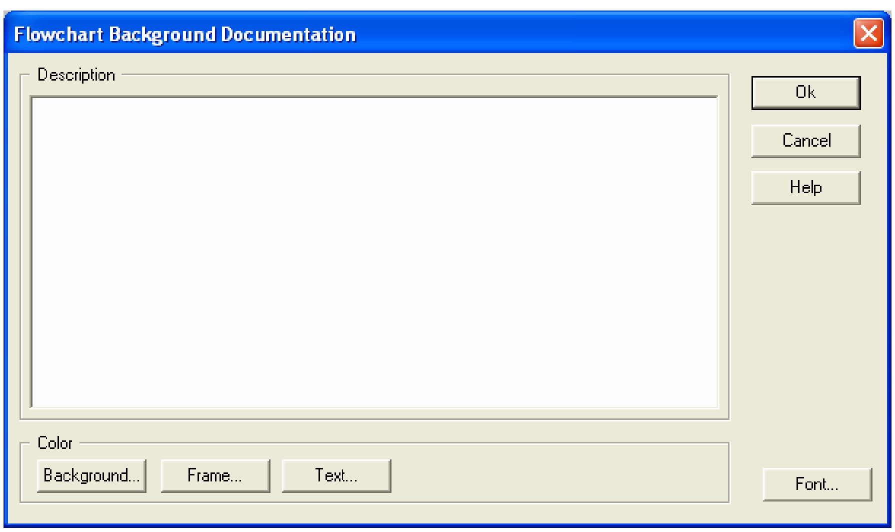
Once the "Edit Documentation" has been selected a check will appear next to this option.



Once this has been done then we can start creating the graphical documentation. To draw a new documentation box first Right Click and select "Create Documentation", then Left Click and drag a box shape. When you release the mouse button you should have a blue square on your screen like this:



To set the boxes color, border, text and font we need to right click and select "Documentation Properties". You should see this dialog box:



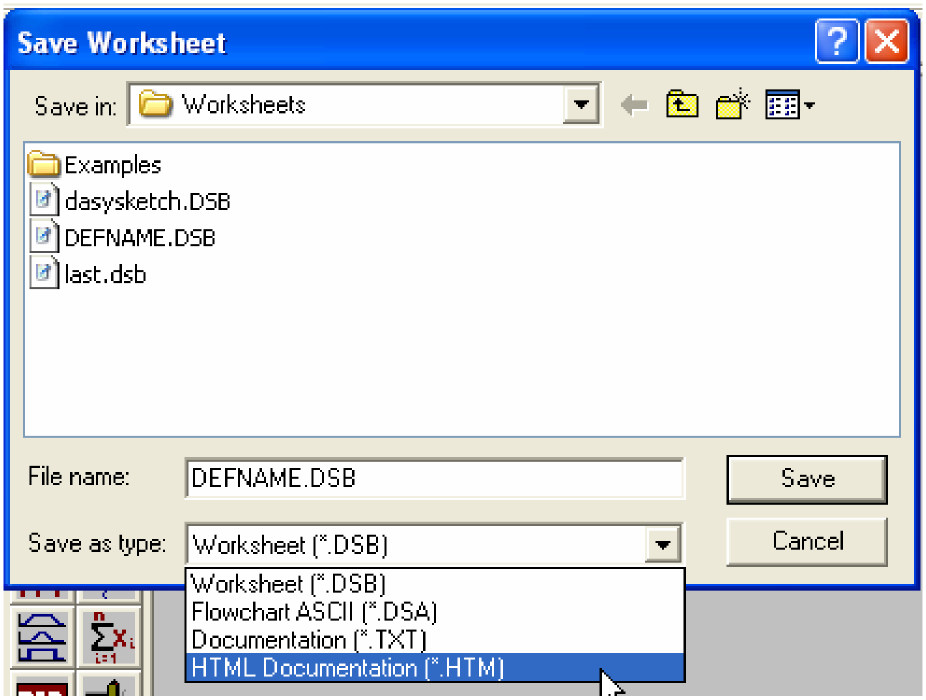
In this dialog box you can enter the text you wish to display in the "Description" field. You can also set the "Background…" color, "Frame…" color, and "Text…" color. You can also set the font by clicking on the "Font…" button.

Deselect the "Edit Documentation" option once you are done creating or editing your graphical documentation.

PRINTABLE DOCUMENTATION

There are two methods of creating documentation that is not connected to a **DASY*Lab*** worksheet. This documentation can be saved/archived, printed or published to document and protect your worksheet. ***DASYLab*** can save its documentation as either a text file or a hyper linked html file (web page). This documentation includes all the setting of each module, where they get their data, where the data gets sent to as well as the documentation you enter into each module.

To save you worksheet off into one of these formats select the File Menu and click "Save As…"



By selecting Documentation (*.TXT) or HTML Documentation (*.HTM) you will save you worksheets documentation to the hard drive.

These formats are not available to be loaded back, so be sure you save a DSB or DSA format of your worksheet.

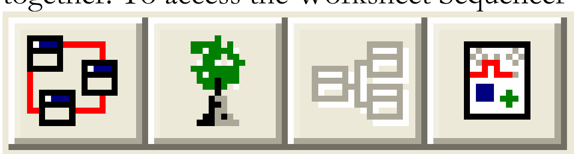
CHAPTER 11: WORKSHEET CONTROL SEQUENCER

LINKING SEVERAL WORKSHEETS TOGETHER TO FORM A SINGLE TEST SEQUENCE

In many test situations it may be best to have several different worksheets each with a different function. For example it may be easiest to read or generate calibration settings in one worksheet, pass those settings on to the next worksheet. This worksheet acquires the data and saves it into a file. The last worksheet analyzes and displays the data.

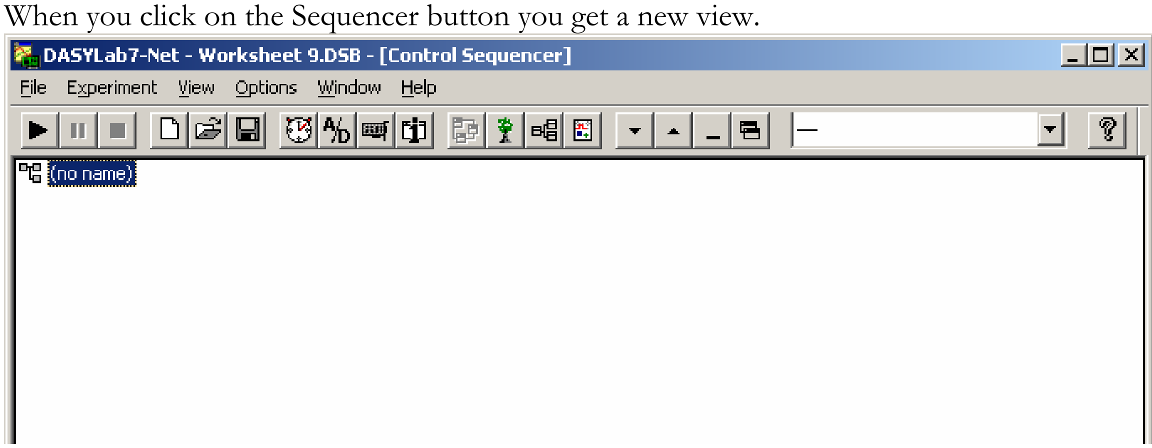
DASY*Lab* makes this easy by providing a layer to create and over view the sequencing of several worksheets. Global variables or global strings will control the changing of worksheets. Starting or stopping of a worksheet can also trigger other actions.

After you have created and saved your **DASY*Lab*** worksheets you can start sequencing them together. To access the Worksheet Sequencer click on the sequencer button.



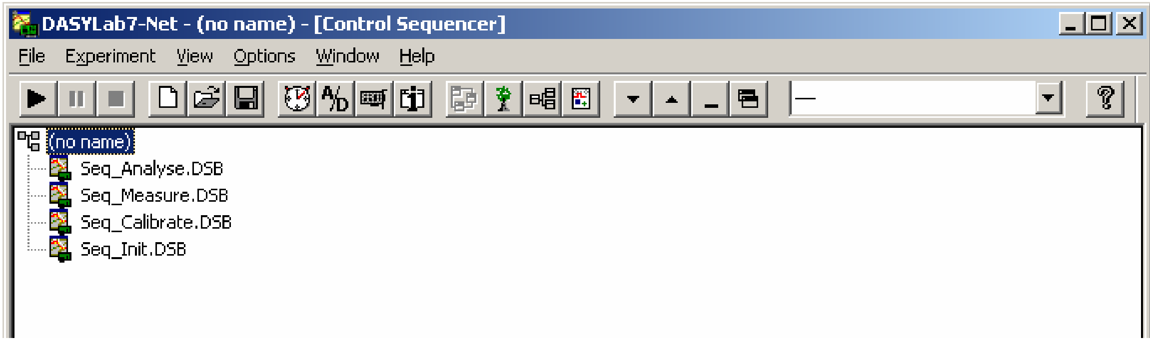
This is the Sequencer button.

When you click on the Sequencer button you get a new view.

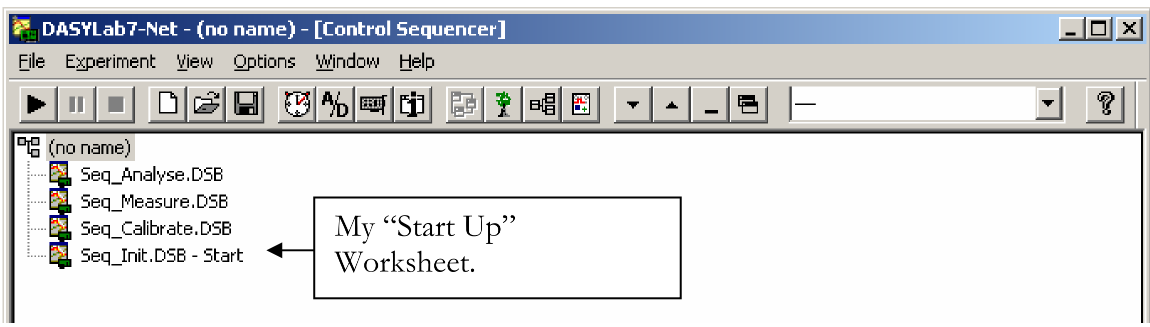


In this view we will create a tree diagram of worksheets and actions.

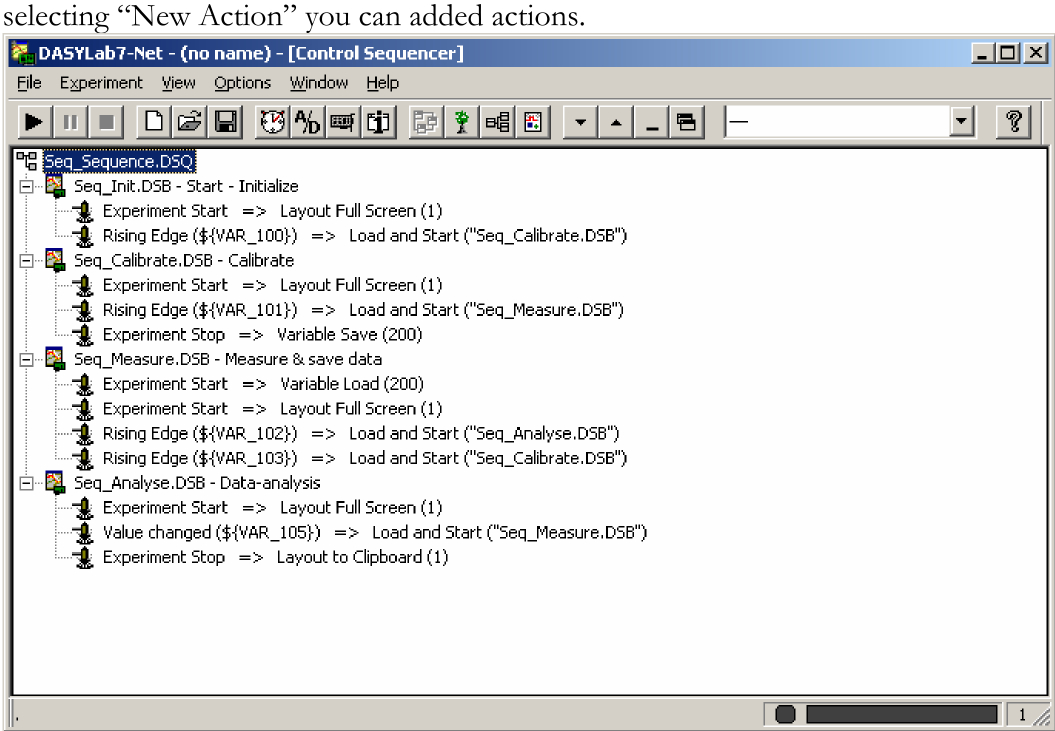
First we must create the **DASY*Lab*** worksheets we wish to link together. Once that is done Right Click on "(no name)" and select "New Flowchart" and load your worksheet. Repeat this for all the worksheet you wish to have linked together.



Now a worksheet must be selected as the first or "Start Up" worksheet. This will be the first worksheet to run every time the sequencer starts. To set a worksheet as the start up worksheet, Right Click in the worksheet name and select "Set Start Flowchart".



The next step is to add your actions to the sequence. By Right Clicking on a worksheet and selecting "New Action" you can added actions.



The Actions can be triggered by many different events. For example we can monitor for the start or stop of the experiment, or any number of values or states of a global variable or string.

APPENDIX A: HARDWARE INSTALLATION

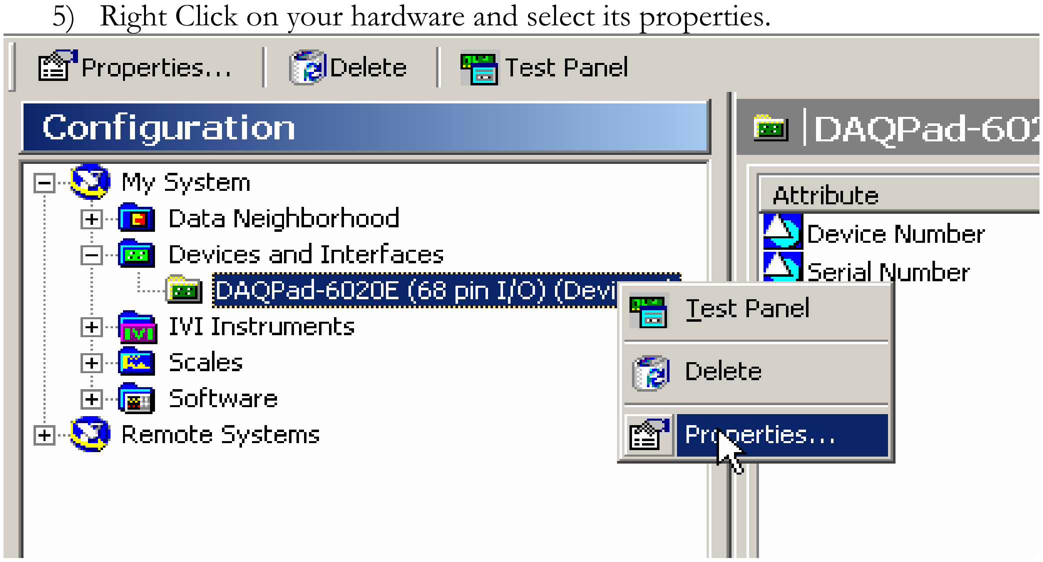
If you have hardware that interfaces with **DASY*Lab*** through a third-party driver, like National Instruments, IOtech or GW Instruments there may be additional steps. In this section we will walk through the additional steps in installing this hardware.

NATIONAL INSTRUMENTS

To install National Instruments hardware you should first read all the instructions provided with your hardware. If you are ever in doubt please contact your local sales representative for assistance. There are several steps involved in installing the hardware.

These steps are for non-Field Point or NICAN hardware.

1) Install the drivers provided with your hardware. The CD should have been included with your device. This CD should install both the driver and "MAX 2.1" and NIDAQ 6.9+. If these packages were not installed you can download them at www.ni.com.
2) Shut down your computer and install your hardware as defined in your owners manual.
3) Restart the PC and Open MAX.
4) Click on the "+" next to "Devices and Interfaces" to view the list of available hardware.
5) Right Click on your hardware and select its properties.

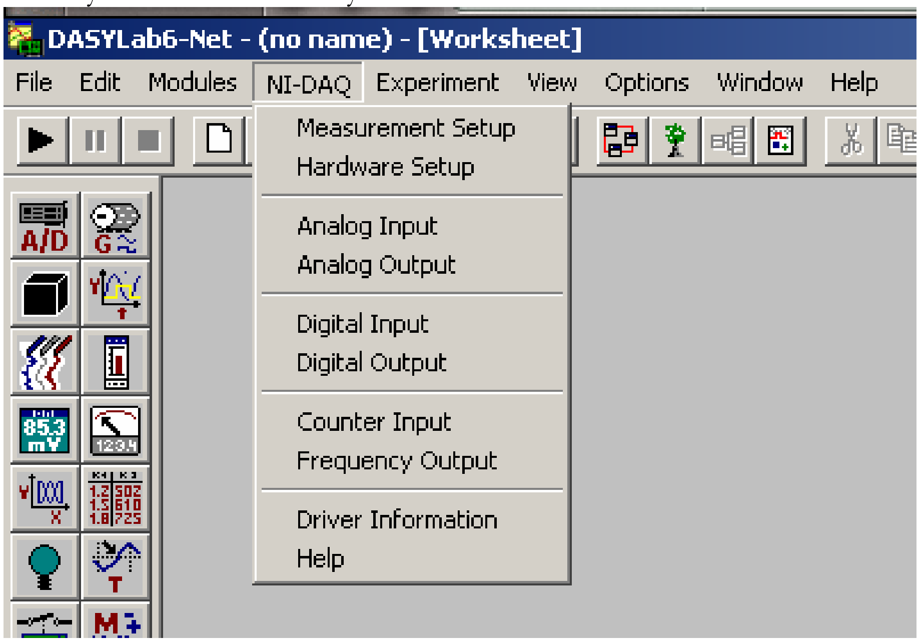


6) Click on "Test Resources".

If the device passes the test then we are ready to start working in **DASY*Lab***.

Close MAX before starting **DASY*Lab***. **DASY*Lab*** and MAX cannot run at the same time!

When you start **DASY*Lab*** you will see a new menu item:



From this menu you select your inputs and outputs as well as setup the acquisition rate and block size. This menu will allow you to set the acquisition rate and block size for each board separately, allowing you to sample slowly on one board and fast on another board.

INSTRUNET

The InstruNet devices require that you install the "InstruNet World" software package provided with your hardware device. InstruNet also requires a special **DASY*Lab*** serial number; if you ordered **DASY*Lab*** with your InstruNet then this should have been taken care of by your distributor. If your not sure that it has been included please feel free to contact your distributor.

Once InstruNet World has been installed and you hardware has been tested then a new drop down should appear in **DASY*Lab*** titled "InstruNet". From here you can access you InstruNet device, as well as InstruNet World to make settings and changes.

Do not attempt to run InstruNet World at the same time as **DASY*Lab***.

IOTECH

Contact IOtech for advanced instructions on the proper installation and configuration of their hardware.

APPENDIX B: RS-232 INTERFACE

DASY*Lab* has a universal RS-232 interface, which allows **DASY*Lab*** to communicate with virtually any RS-232 device. The RS-232 Modules in **DASY*Lab*** are suitable for communicating with any COMM port defined in WINDOWS, which includes RS-232, RS-422 and RS-485. Because RS-232 has no set standard for how the device communicates it can be rather difficult to configure **DASY*Lab*** to properly communicate with a given device. In this section I will walk through the steps of setting up a common RS-232 Device.

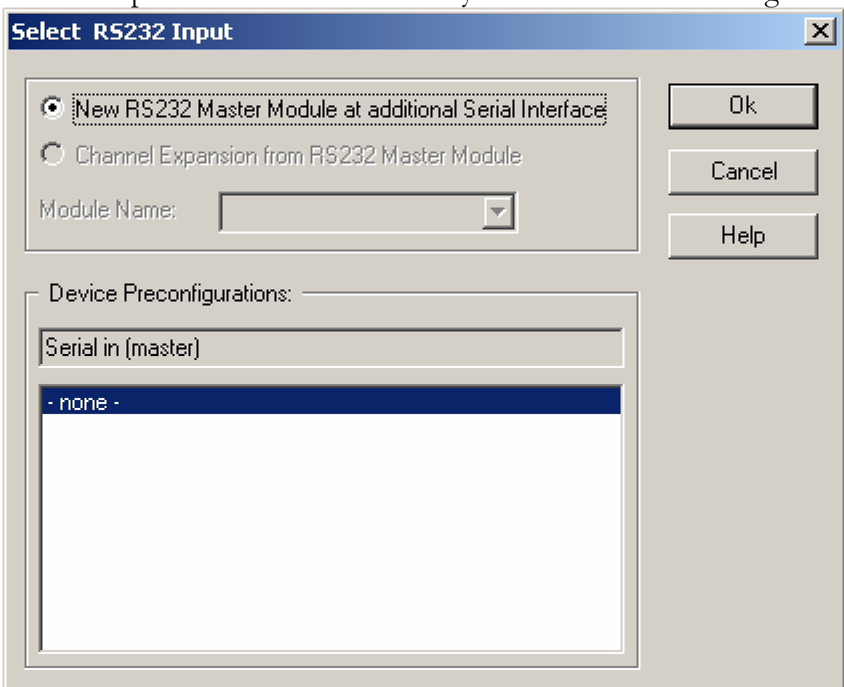
First of all you'll need the manual provided by the device manufacturer. Search the manual and fill out this table:

Baud Rate	
Parity	
Stop Bits	
Data Bits	
Handshake	

You will also need to find out if your device requires a "Data Request" or "Start Command". If it has either make note of it here:

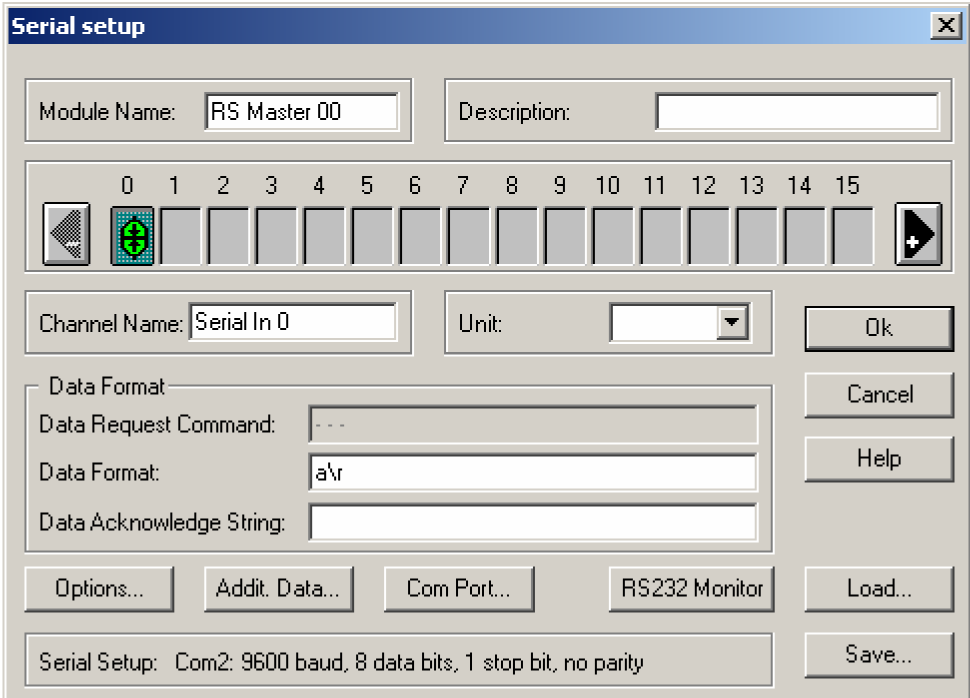
SETTING UP THE RS-232 INPUT MODULE

We are now ready to start working with **DASY*Lab***. Place an RS-232 Input Module onto the worksheet. These can be found under Modules; Input/Output; RS232 Input. When the module is placed on the works sheet you should see this dialog box:

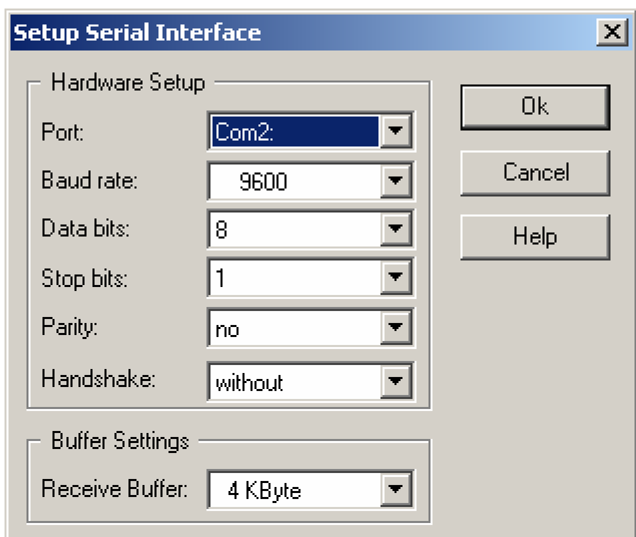


If you were provided with a Pre-configuration you could select it here, otherwise click OK.

The RS-232 Input Module dialog box looks like this:



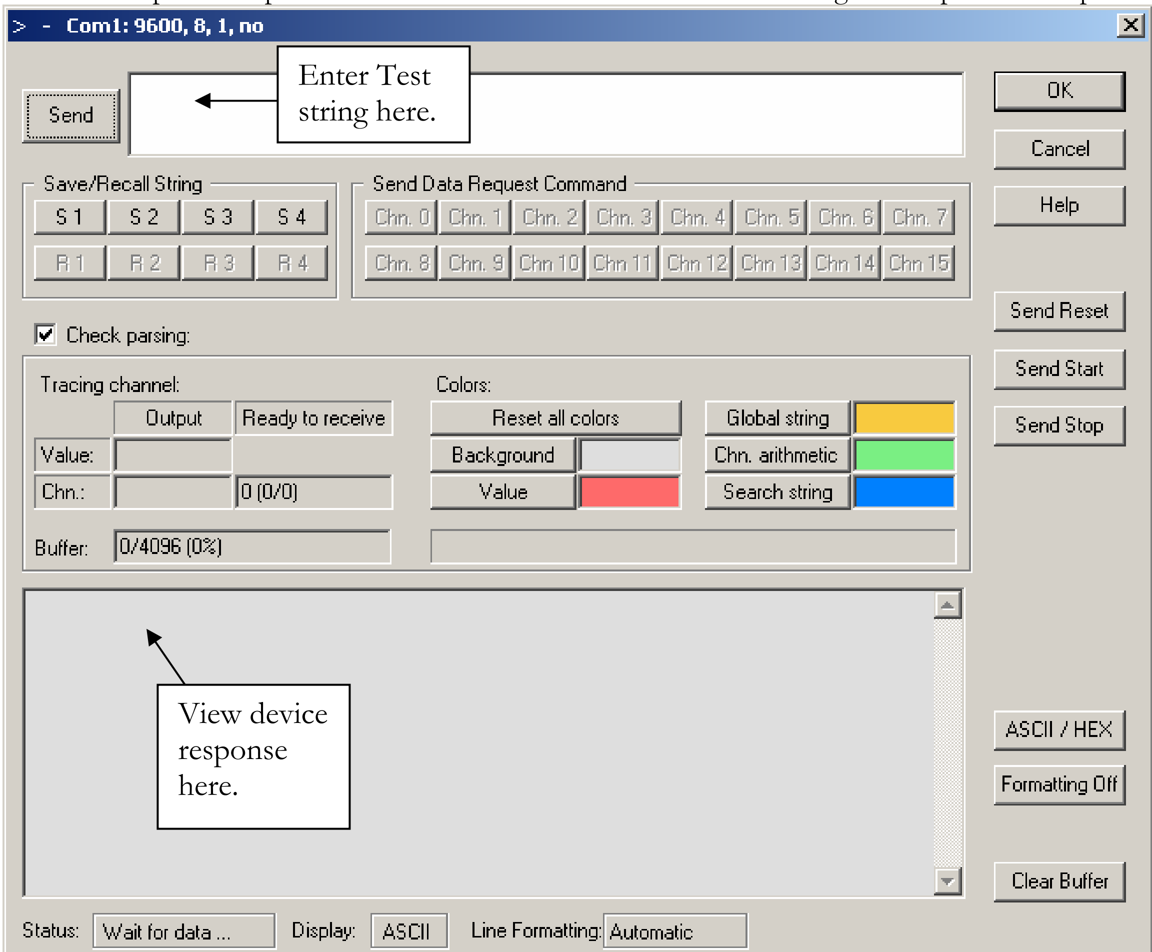
Step 1 is to configure the comm. Port for your device. Click on the "Com Port… " button to access this dialog box:



Set the properties to match the setting of your hardware device. If you don't know the handshake settings its okay, we can find them by trial and error later, however the rest of the information is rather critical and must be correct. The "Buffer Settings" can be ignored for most applications, and left at "4 KByte". Once you have these settings correct, click OK.

If you get the "Error configuring the serial interface" error message then there may be another device using that comm. Port, or you may have selected a comm. Port that is not available on your PC. Check your hardware settings to be sure that you can use that comm. Port.

The next step is to make "first contact" with the device, we will do this in the RS-232 Monitor. To access the monitor click on the "RS-232 Monitor" button in the RS-232 Input Module. In this dialog box we will "poke and prod" at the device to test its communication and get a sample of its response.

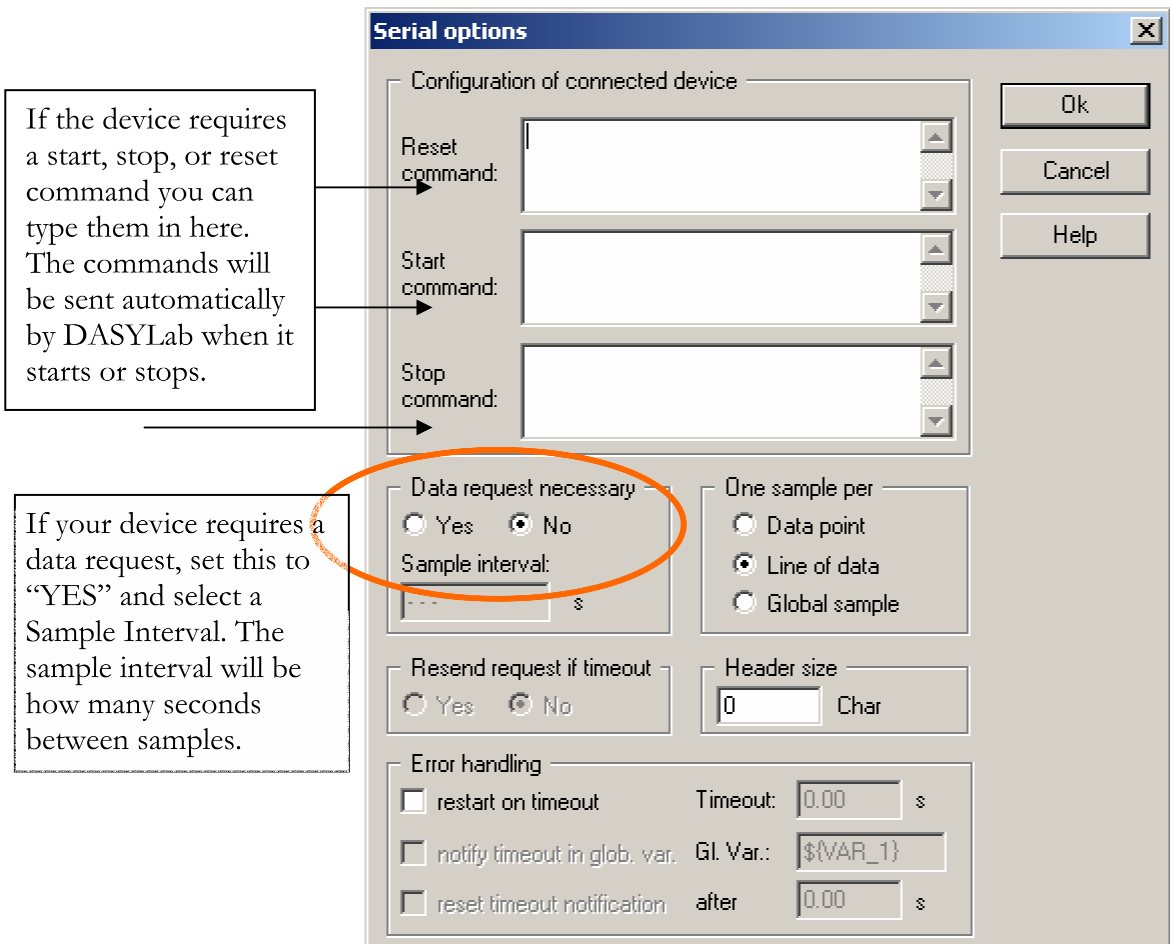


In the large white box next to the send button enter your "Data Request" or "Start" string and click the "Send" button. If the device doesn't require either then it should be sending data already. There are a few special characters you may have to us in your "Data Request" or "Start" command. Here is a table some of them.

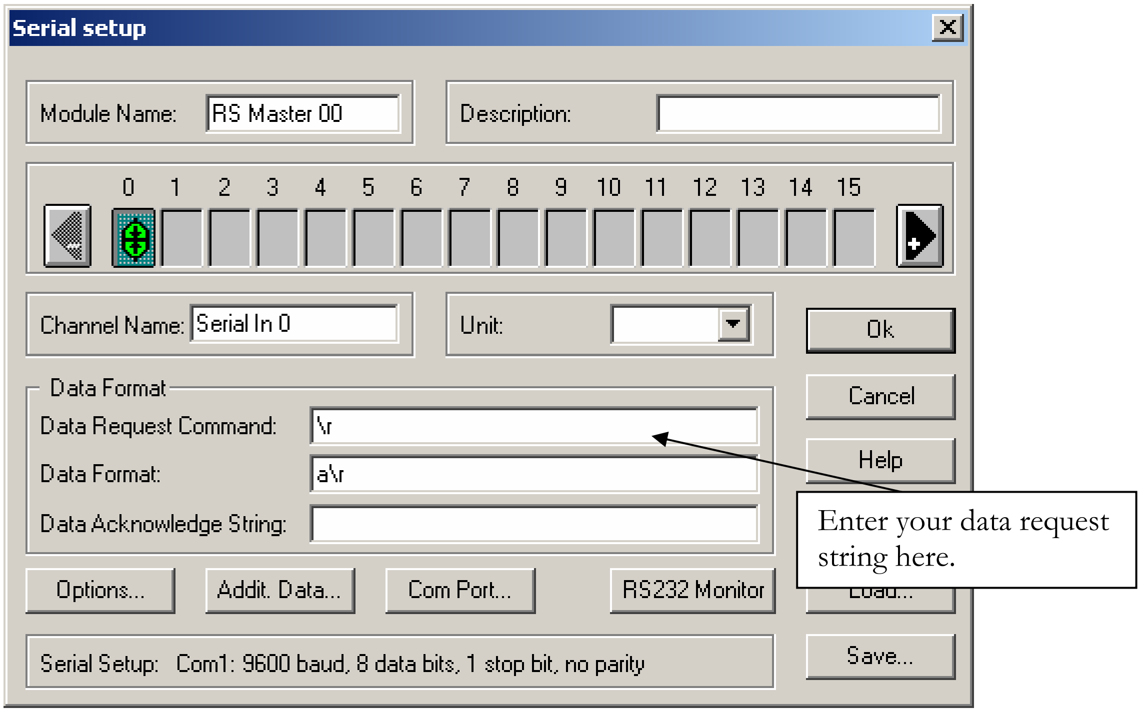
Command	Character	Example
Carriage return	\r	1DSQ\r
Line Feed	\n	1BLA\n
TAB	\t	3DOW\t
Form Feed	\f	2WOW\f
Vertical Tab	\v	Q4\v
Backspace	\b	Q1\b
Pause x, where x is time in ms.	\px	1LA\p100TEDA

If everything is setup correctly we should see a response from the device in the response window. If we do then we are ready to start parsing our data, other wise we have to de-bug our communication. If all the com port parameters are correct (Baud rate, parity, stop bit, and data bits) then we may have a problem with our handshake. Click OK to close the RS-232 Monitor and back up a step and select the first handshake from the "Com Port…" dialog. Try each handshake in turn until the device starts to communicate.

We have a few other options to set before **DASY*Lab*** is ready to talk with your device. If you are still looking at the RS-232 monitor, make a note of what you sent to it and what it sent back then click OK to close it. Now click on the "Options…" button.



If your device requires a Start, Stop or Reset command, enter them here. If you selected "Data Request Necessary" we will enter that in a different screen. The rest of the setting can stay as default for the moment.



If your device requires a data request string you should enter it here. The data request string you enter here should be the same one you tested in the RS-232 Monitor. Once this is all set up then we can define our data format. The data format is an explanation to **DASY*Lab*** on how to parse the values of the return string.

PARSING RS-232 DATA

Once we have established communication with our device we can start the get the data from it. There are two different thoughts when parsing data, and it depends on how the device communicates, “Fixed Format” or “Flexible Format”.

If the length of the string and the position of the data in the string don’t change, then we will use the “Fixed Format” method of data parsing. An example of a fixed format string might look like:
#2b+1.23<cr>
#2b+2.34<cr>
#2b+12.00<cr>

In this example when a “Data Request” is sent the data always occurs 3 characters in (skipping the #2b” and ends with a <cr>. The data is also ASCII compatible. If I wanted to parse out the important part of the data I would use “\x3 a \r” (minus the “”). This will tell **DASY*Lab*** to skip 3 characters, read the ASCII characters until the <cr>(carriage return).

String	Parsing String	Parsed String
#2b+12.00<cr>	\x3 a \r	#2b+12.00<cr>

A slightly more complex string might look like:
$2b3q123.45qb<cr>

Where we need to skip some character in front of the data, and a few character after the data before a terminating carriage return. The parse string would be “\x5 a \x2 \r” (minus the “”).

String	Parsing String	Parsed String
$2b3q123.45qb<cr>	\x5 a \x2 \r	$2b3q123.45qb<cr>

What if there is no terminating character? We can tell **DASY*Lab*** how characters make up the value we are looking for. For example if we take the last example and remove the <cr>:
$2b3q123.45qb

We could parse it our with:
“\x5 5a” (minus the “”)

String	Parsing String	Parsed String
$2b3q123.45qb	\x5 5a	$2b3q123.45qb

Here we’ve told **DASY*Lab*** how many characters to skip, then how many to read. The rest of the data is ignored until the next data request.

If the strings are not fixed length or it is easier to find a fixed pattern in the string then the "Flexible Format" is what you should use. The flexible format works by matching a given pattern and parsing out from there. To have **DASY*Lab*** search for a string in the returned data place the string in quotes then the rest of the parsing information after that. For example:

Q:W34E32N54.432DATA123.45F:JUNK

If we want the 6 characters after the word "DATA" we would use the following parsing string: "DATA" 5a

String	Parsing String	Parsed String
Q:W34E32N54.432DATA123.45F:JUNK	"DATA" 5a	Q:W34E32N54.432DATA123.45F:JUNK

The other parsing techniques from the fixed format can also be applied if the data is not directly fixed string. For Example:

Q:W34E32NDATAJUNK123.45F:JUNK

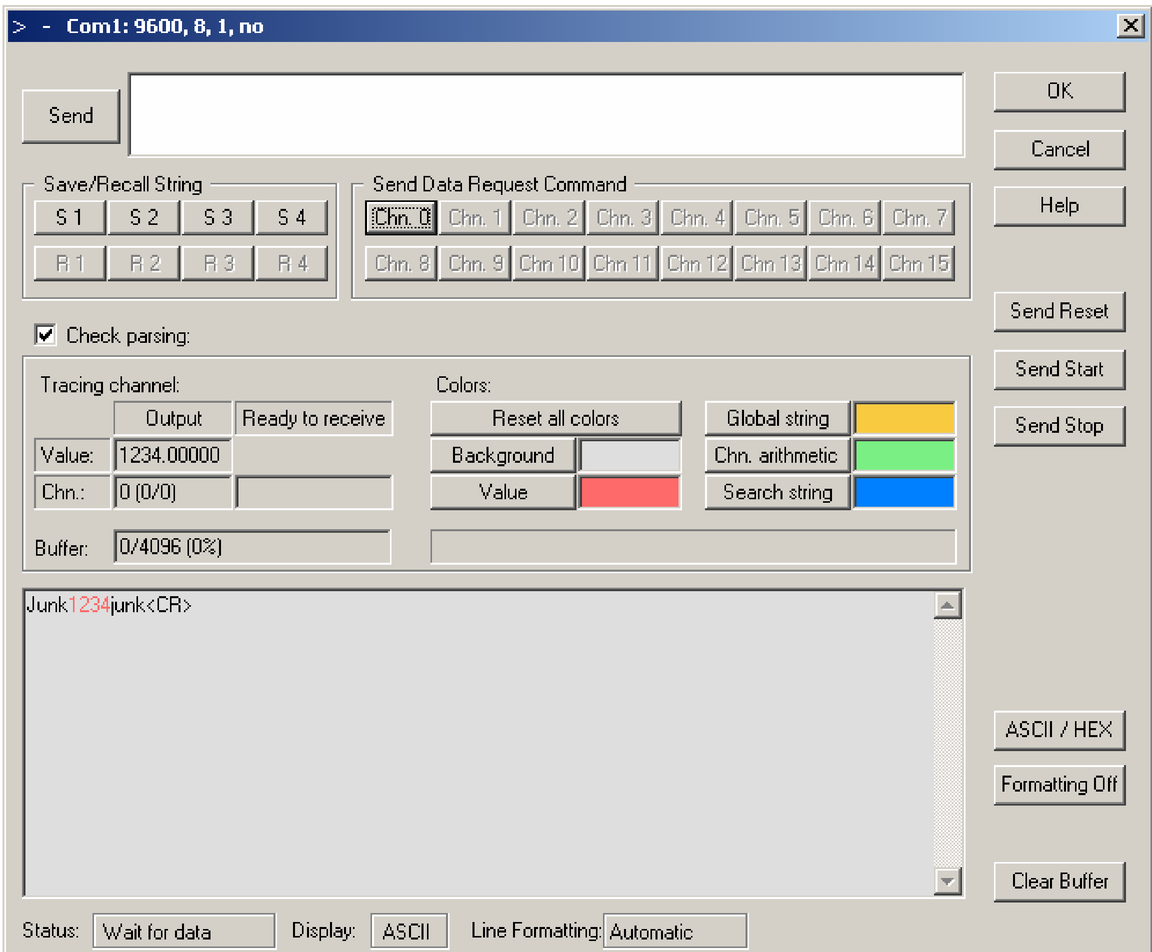
Would parse with:
"DATA" /x4 5a

String	Parsing String	Parsed String
Q:W34E32NDATAJUNK123.45F:JUNK	"DATA" /x4 5a	Q:W34E32NDATAJUNK123.45F:JUNK

For these examples all the data has been in ASCII format. However **DASY*Lab*** can handle ASCII Hex and Binary Data. Here is a table of data formats and their key character:

Data Type	Number of Bytes	Format Code
ASCII text	optional	a
ASCII hexadecimal	optional	ah
Byte without sign	1	b
Integer with sign	2	i
Integer with sign (Motorola)	2	iy
Integer without sign	2	w
Integer without sign (Motorola)	2	wy
Long Integer with sign	4	l
Long Integer with sign (Motorola)	4	ly
Long Integer without sign	4	u
Long Integer without sign (Motorola)	4	uy
IEEE Floating point	4	f
IEEE Double Floating point	8	d

Once you have entered you data format string you can test it in the RS-232 Monitor. For example if our device responds with Junk data junk\r, we would use a 4x a 4x \r as the data format. When we send that data request out in the RS-232 monitor, using the "Send Data Request Command" buttons, we get a color coded example of our parsed data. In addition to color coding our data the "Tracing channel" dialog shows the value of the parsed data.



NOTES