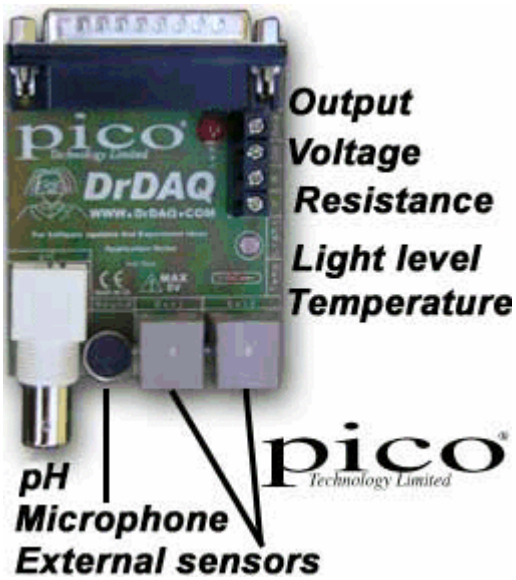
DrDAQ
Data Logger

User guide

Table of Contents

1 Introduction

1.1 Overview



DrDAQ is a low cost data logger from Pico Technology. It is supplied ready to use with all cables, software and examples. Features include:

- Built in sensors for light, sound (level and waveforms) and temperature
- Measure pH - just plug in our standard pH electrode
- Sockets for external sensors
- Use DrDAQ to capture fast signals (for example, sound waveforms)
- Digital output for control experiments
- Simply plug in to your PC and measure - supplied with both PicoScope (oscilloscope) and PicoLog (data logging) software

1.2 Safety warning

DrDAQ ground is connected directly to the ground of your computer. As with most oscilloscopes and data loggers, this is done in order to minimise interference. You should take care not to connect the ground (screw terminal, outer shell of BNC or exposed metalwork) of DrDAQ to anything which may be at some voltage other than ground: doing so may cause damage to the unit. If in doubt, use a meter to check that there is no significant AC or DC voltage.

For computers that do not have an earth connection (for example, laptops), it must be assumed that DrDAQ is not protected by an earth (in the same way a battery multimeter is not protected by an earth).

The maximum input voltage range of any input is 0 to 5V. Any voltage in excess of ±30V may cause permanent damage to the unit.

The unit contains no user serviceable parts: repair or calibration of the unit requires specialised test equipment and must be performed by Pico Technology Limited, or their authorised distributors.

1.3 Legal information

The material contained in this release is licensed, not sold. Pico Technology Limited grants a licence to the person who installs this software, subject to the conditions listed below.

Access
The licensee agrees to allow access to this software only to persons who have been informed of these conditions and agree to abide by them.

Usage
The software in this release is for use only with Pico products or with data collected using Pico products.

Copyright
Pico Technology Limited claims the copyright of, and retains the rights to, all material (software, documents etc.) contained in this release. You may copy and distribute the entire release in its original state, but must not copy individual items within the release other than for backup purposes.

Liability
Pico Technology and its agents shall not be liable for any loss, damage or injury, howsoever caused, related to the use of Pico Technology equipment or software, unless excluded by statute.

Fitness for purpose
No two applications are the same: Pico Technology cannot guarantee that its equipment or software is suitable for a given application. It is your responsibility, therefore, to ensure that the product is suitable for your application.

Mission-critical applications
This software is intended for use on a computer that may be running other software products. For this reason, one of the conditions of the licence is that it excludes usage in mission-critical applications, for example life support systems.

Viruses
This software was continuously monitored for viruses during production, but you are responsible for virus-checking the software once it is installed.

Support
If you are dissatisfied with the performance of this software, please contact our technical support staff, who will try to fix the problem within a reasonable time scale. If you are still dissatisfied, please return the product and software to your supplier within 28 days of purchase for a full refund.

Upgrades
We provide upgrades, free of charge, from our web site at www.picotech.com. We reserve the right to charge for updates or replacements sent out on physical media.

Trademarks
Pico Technology Limited, PicoScope, PicoLog, DrDAQ and EnviroMon are trademarks of Pico Technology Limited, registered in the United Kingdom and other countries

Pico Technology acknowledges the following product names as trademarks of their respective owners: Windows, Excel, Visual Basic, LabVIEW, Agilent VEE, HP VEE, Delphi.

2 Product information

2.1 Specifications

General specifications

Number of input channels	7 internal, 2 external
Typical sampling rate	15,000 samples per second (measured on 100 MHz Pentium)
Input overvoltage protection	30 V
Digital output voltage	typically 3-5 V, depending on PC and load
Digital output impedance	approx. 1-3 kΩ depending on PC
Output connector	25 way male D-type (connects to PC parallel port)

Internal sensor specifications

Channel	Range	Resolution	Accuracy
Sound waveform	±100	0.2	Not calibrated
Sound level	55 to 100 dBA	1 dBA	5 dBA
Voltage	0 to 5 V	5 mV	3% of FSD
Resistance	0 to 1 MΩ	0.1 kΩ (at 10 kΩ)	2% (at 100k)
pH	0 to 14	0.02 pH	Calibration dependent
Temperature	-10°C to 105°C	0.1°C (at 25°C)	0.3°C (at 25°C)
Light	0 to 100	0.1	Not calibrated

2.2 Connecting to the PC

DrDAQ is connected to the parallel port of your computer. Parallel ports can be easily identified as they are a female 25 way connector. It is common for parallel ports to be labelled as printer ports or simply LPT. It is possible to plug the DrDAQ directly to your parallel port, however it is recommended that the parallel port cable supplied is used for ease of access to the unit.

If another device such as a printer or scanner is already connected to the parallel port, and your computer only has one parallel port then the other device will have to be unplugged before DrDAQ can be connected.

The DrDAQ must be connected directly to the parallel port via the cable supplied, the use of other cables, printer sharers, port splitters, software dongles may result in DrDAQ not working.

If you wish to add additional parallel ports to your PC, Pico sells a dual parallel port card that can be installed inside your PC. USB to parallel port converters will not work with DrDAQ, but can be used with your printer to free up the main parallel port for DrDAQ.

2.3 Connections

Channel	Input Connection
pH	BNC connector
Voltage input	Screw terminal
Resistance input	Screw terminal
Digital output	Screw terminal
External inputs	FCC68 4-pin connector

Digital output
The output pin can be used either as a digital output or as a voltage source. Note: DrDAQ does not provide any additional protection for this output.

When using the pin as a voltage output, the output impedance will vary between computers from about 1 to 3 kilohms, but will be consistent for a particular computer. It should be sufficient to power a 10k thermistor, if you use a 10k bias resistor for the thermistor. This output can also directly power the LM35 type of IC-based temperature sensor.

External Inputs
Each external input socket has four connections:

Pin	Function
1	Input channel with 100k pull-up to 2V5
2	Ground
3	Sensor type detect with 100k pull-up to 2V5
4	Power approx 2mA at 3-5V

Note: On most computers, the printer port ground is connected to mains earth. If you connect the ADC ground to a non-zero voltage, you may damage the ADC and your computer. If in doubt, check whether there is a voltage difference between ground on you equipment and ground on the ADC before connecting the ADC to your equipment.

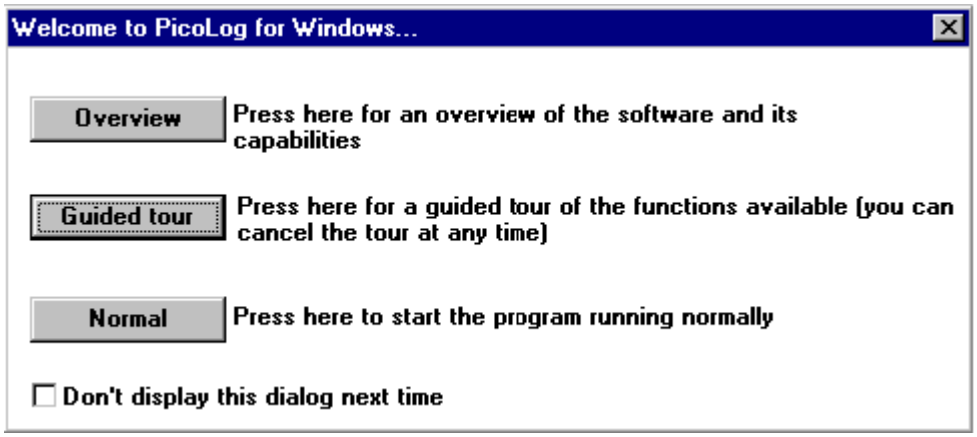
2.4 Channel scaling

The following table gives details of the scaling for each channel.

Channel number	Input	Min Value	Max Value	Decimal Places	Units
1	Sound waveform	-100	100	1	-
2	Sound level	50	100	2	dBA
3	Voltage	0	5000	0	mV
4	Resistance	0	1000	1	kΩ
5	pH	0	14	2	-
6	Temperature	0	100	1	°C
7	Light	0	100	1	-
8	External 1	Depends on sensor	-	-	-
9	External 2	-	-	-	-

2.5 Getting started with PLW

PicoLog is a powerful but flexible program for collecting, analysing and displaying data. When PicoLog is first loaded the following screen will appear:



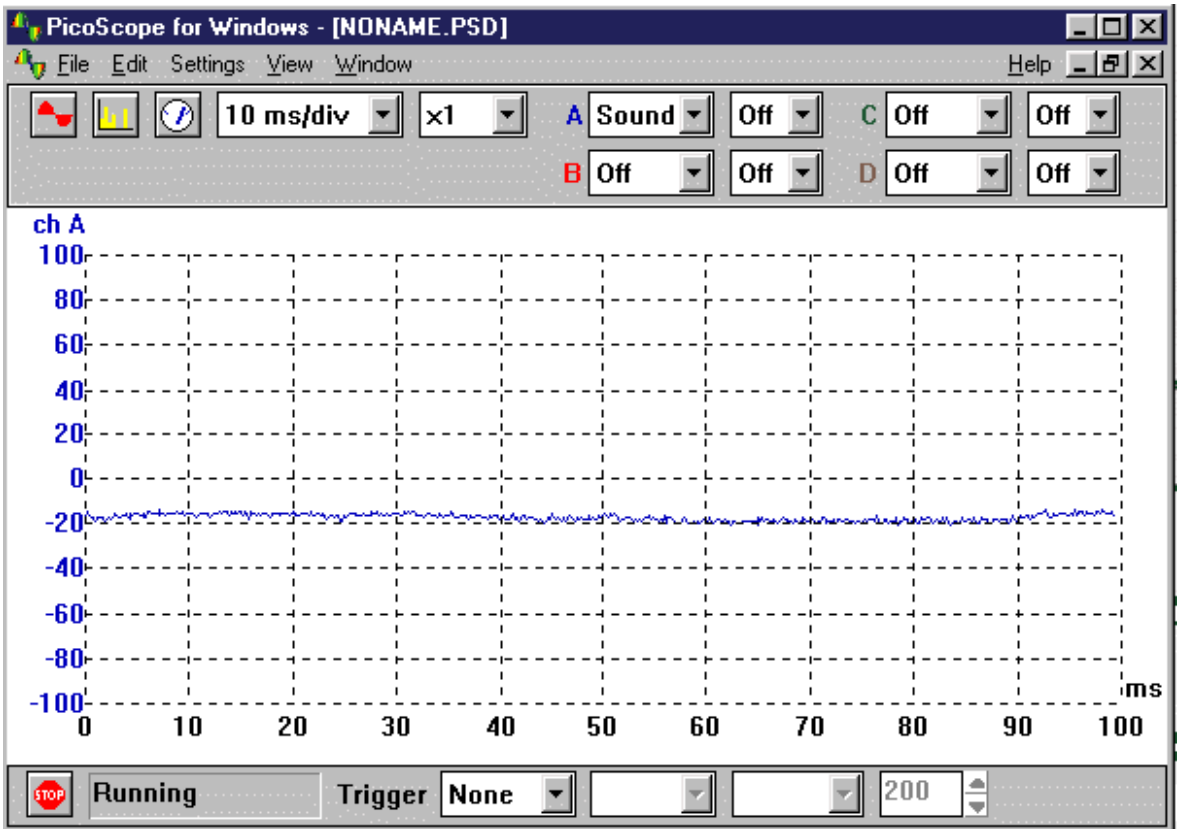
Select the guide tour which will take you through the main functions in PicoLog.

2.6 Getting started with PicoScope

2.6.1 Using PicoScope

PicoScope software turns your PC into an oscilloscope, spectrum analyser and meter. PicoLog software is an advanced data logging package. The decision on whether to use PicoScope or PicoLog depends on the signals you wish to measure. If you wish to collect fast signals (such as sound waveforms) then use PicoScope, if you want to collect data over a long period (such as plotting battery discharge) then use PicoLog.

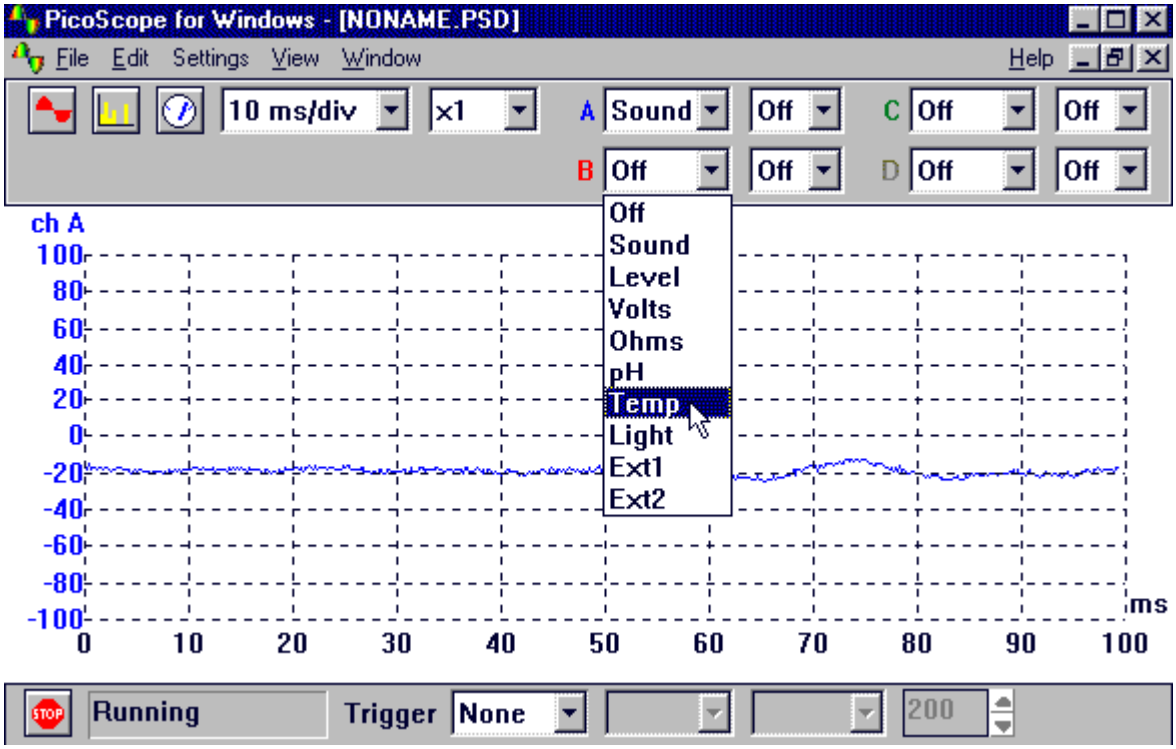
When PicoScope is first loaded, a title screen will be displayed, then the view below should appear. If this view does not appear, or an error message is displayed, then double check the connections and consult the technical support section of this help file.



Try clicking your fingers over the microphone: the trace on the screen should react. DrDAQ is now successfully installed and working.

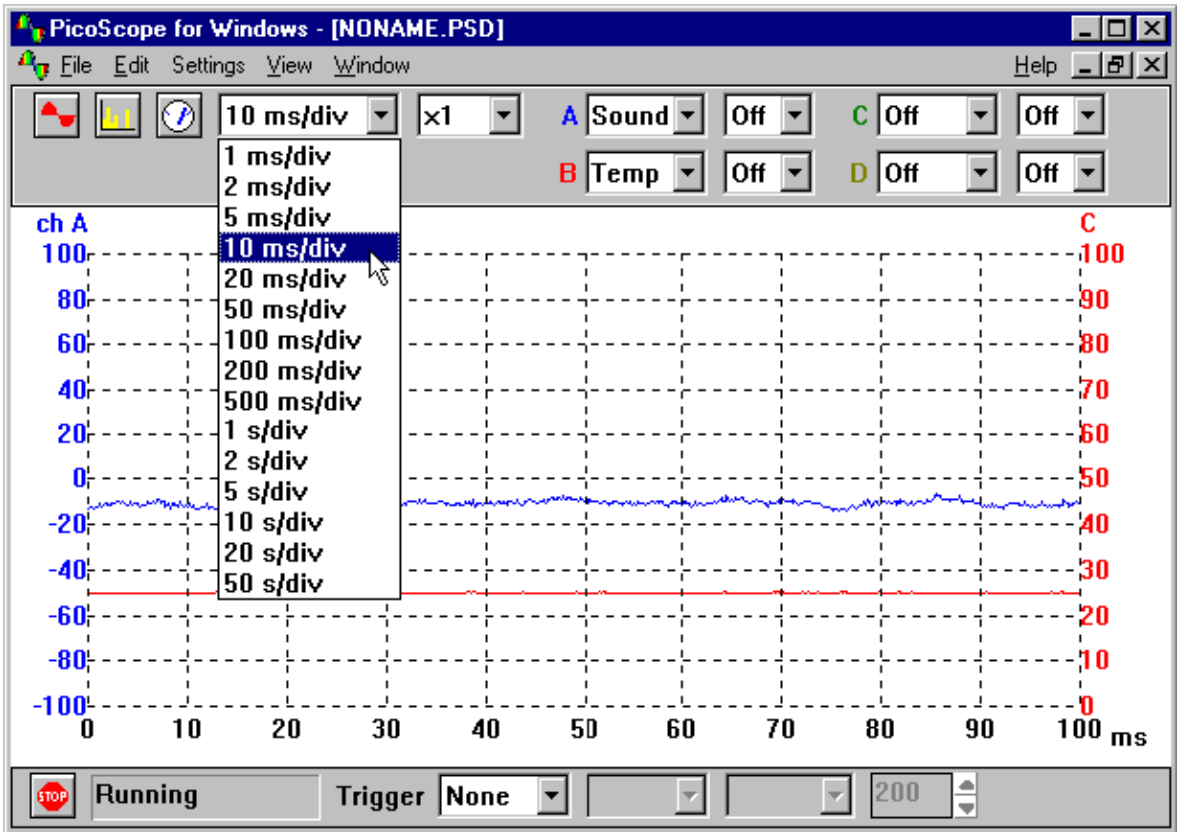
2.6.2 *Adding an oscilloscope channel*

It is possible to display up to 4 different channels on any one particular scope window. Add another channel by going to the drop-down menu B and assigning it to Temp. Put your finger on the temperature sensor and notice the rise in temperature.



2.6.3 *Changing the oscilloscope timebase*

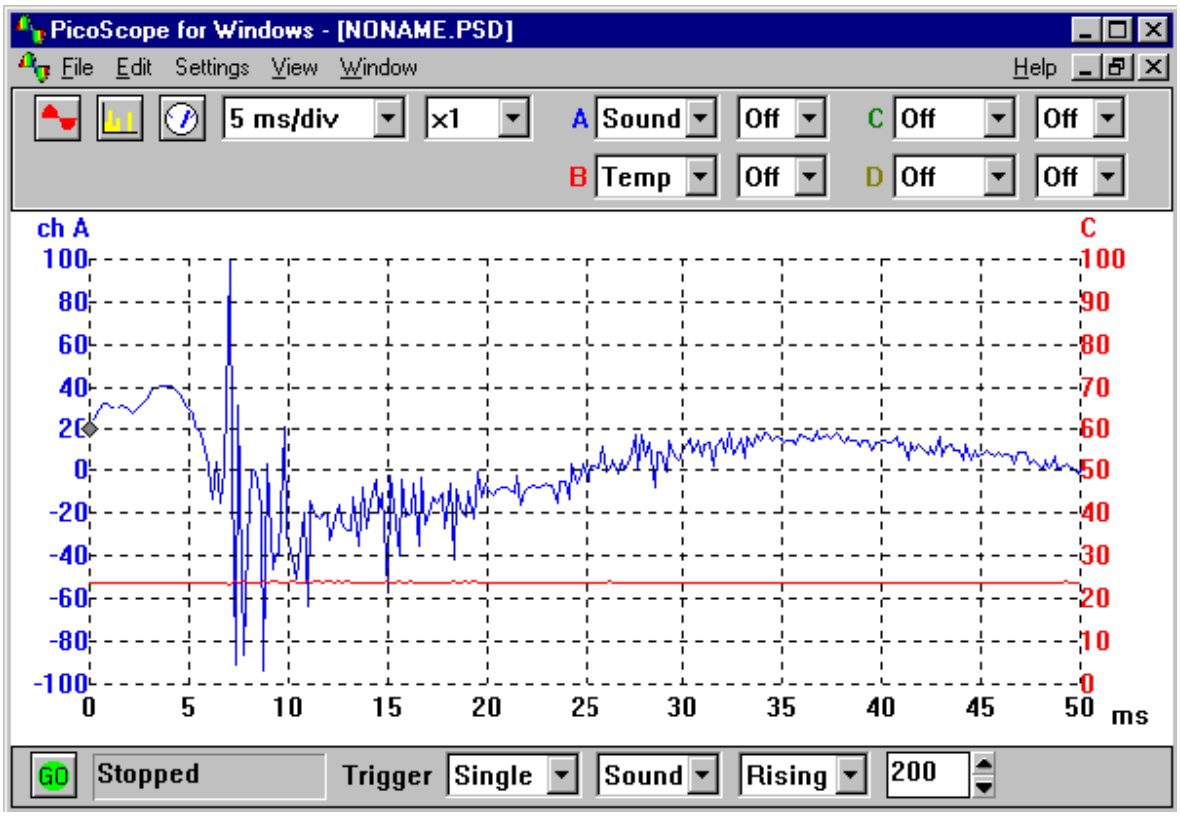
The timebase can be adjusted by selecting the drop-down menu shown below. Change the timebase to 5 ms/div.



2.6.4 *Using the trigger*

The PicoScope trigger can be used to capture infrequent or one-off events. As an example, click your fingers over the microphone: a waveform is displayed but vanishes as soon as the display updates. The correct way to capture such a signal is to use a trigger.

Stop PicoScope (click on the 'Go' icon at the bottom left hand corner of the window, or press the space bar) and select a single trigger using the trigger options at the bottom of the screen. Press the space bar to start PicoScope and click your fingers over the microphone. The single trigger option means that PicoScope stops running when it has captured a signal.



2.6.5 *Adding spectrum and meter views*

So far we have looked at the oscilloscope functions. PicoScope allows DrDAQ to be used as a spectrum analyser and meter as well. The three buttons on the top left-hand side control the oscilloscope, spectrum and meter views.

 - Oscilloscope view (view signals against time)

 - Spectrum analyser view (view amplitude against frequency)

 - Meter view (measure DC volts, AC volts, dB and frequency)

2.7 External sensors

2.7.1 Overview

As well as the built in sensors, DrDAQ has sockets for optional external sensors. When a sensor is plugged in to the external sensor sockets, the software detects it and automatically scales readings. For example, if a temperature sensor is plugged in, readings are displayed in °C, or if a humidity sensor is plugged in, readings are displayed in % RH.

External sensors are optional extras so can be purchased at any time. For an up-to-date list of the available external sensors consult the DrDAQ web site at http://www.drdaq.com/.

- Click here for the DD100 Temperature Sensor
- Click here for the DD011 pH Electrode
- Click here for the EL029 Reed Switch
- Click here for the DD101 Humidity Sensor
- Click here for the DD103 Oxygen Sensor

2.7.2 DD100 Temperature Sensor

DD100 Temperature Sensor

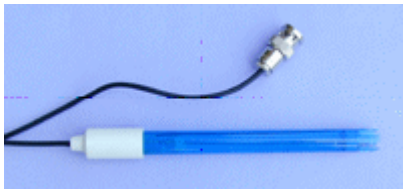
High-accuracy general purpose temperature sensor with a 2 metre lead. Suitable for air, surface or liquid measurements.

Range	-10°C to +105°C
Resolution (at 25°C)	0.1°C
Accuracy (at 25°C)	0.3°C

2.7.3 DD011 pH Electrode

DD011 pH Electrode

pH is measured using a standard electrode with a BNC connector. Pico supplies a robust epoxy bodied pH electrode ideal for educational use. It covers the full 0 to 14pH range.



Pico supplies two different makes of pH electrode. Both operate in exactly the same way and are interchangeable in use. Before using your electrode please remove either the protective cap (top photograph) or storage bottle (bottom photograph).

Size	12 x 120 mm
Operating temperature	0 to 60 °C
Resolution	0.02 pH

Using DrDAQ with pH Electrodes
The pH input on DrDAQ is a very high impedance input that is suitable for use with any standard pH electrode. For most applications no calibration is required: just plug in an electrode and measure. If, however, you require very accurate pH measurements, then you should calibrate the probe before use (see below).

If you are using a pH probe and not getting the results you expect then the most probable cause is a defective pH electrode. If cared for properly, pH electrodes will last for a number of years (see the instructions supplied with the electrode). If they are not stored properly, then they will be destroyed in a few weeks. Trying to calibrate out errors from a defective probe will not work and may increase errors further.

pH measurement and temperature calibration
The output from a pH electrode is proportional to absolute temperature (kelvin). A temperature difference of 10°C will cause the probe output to change by approximately 4%. This is a major source of error with most pH meters and data loggers. To minimise these errors, DrDAQ uses the internal temperature sensor to compensate for temperature changes.

Testing pH Electrodes
The best way to test a pH electrode is to use pH buffers (pH4, pH7 and pH10 buffers are widely available). If you do not have any pH buffers, then distilled de-ionised water will have a pH near 7 (some still mineral water bottles have the typical pH printed on the label). Most fizzy cola drinks have a pH in the 2.5 to 3 range.

With the pH electrode connected to DrDAQ, display the pH channel using PicoLog. Measure the pH of each buffer in turn. Allow 30 seconds for the reading to stabilise and be sure to wash the electrode in clean water before swapping solutions. If the pH measured is within 1pH of the expected value then you can be fairly sure the electrode is working correctly. If the readings are wrong then the electrode is probably defective - replacement electrodes are available from Pico Technology.

If you suspect that the pH input on DrDAQ may be defective, then short out the BNC connector using a terminator plug (or paper clip). The reading on the screen should be pH7 (0.5), if not, then the most likely reason is that someone has calibrated the probe incorrectly. Select **File/New Settings** from PicoLog to delete the user calibration and return to the default calibration. If the reading is still wrong then contact technical support.

Calibrating for accurate pH measurements
As described above, calibration is only required when accurate (better than 0.5pH) measurements are required. Calibration should be performed just before the measurements are made. User calibration information is stored separately in PicoScope

and PicoLog, so if you wish to measure pH accurately with both programs, 2 calibrations will be required.

To calibrate a pH electrode you will require at least 2 pH buffers (pH4, pH7 and pH10 buffers are widely available). Calibration should be performed with the ambient temperature close to 25°C. A container of clean water is also required to wash the electrode before moving it from one buffer to another.

1. Connect the pH electrode to DrDAQ and display pH using either PicoScope or PicoLog (depending on which program you are calibrating for).
2. Place the electrode in the first solution and wait for at least 30 seconds for the readings to stabilise (gently stirring helps).
3. Make a note of the reading and repeat the procedure for each of the buffer solutions (do not forget to wash the probe clean before swapping from one buffer to another.
4. You should now have a table of readings similar to the one below:

pH Buffer value at 25°C	**Measured value from DrDAQ**
4.01	4.06
7.00	7.07
10.01	9.92

pH calibration with PicoLog

As already mentioned you will need to note down the measured pH value and compare it with the actual value to create a lookup table. Then follow the steps below:

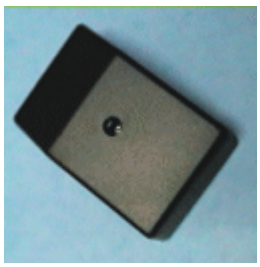
1. Got to 'Settings' and click on 'Input Channels' on the pull down menu.
2. The converter input type will then be displayed, check settings and click 'OK'. You should now see the DrDAQ measurements box.
3. Select pH and click 'Edit'.
4. Click 'Options' from the Edit measurement box.
5. Now click on 'Scaling' and select 'Look up table' from the menu box.
6. Enter the real measured pH value below Raw and the actual value in 'Scaled' (similar to the table in the previous section)
7. Click 'OK' for all open boxes and verify the calibrated pH readings.

Storing pH electrodes

If KCl solution is not available then any pH4 buffer solution will be suitable for storing electrodes. (KCl is recommended because this is inside the bulb of the electrode). Tap water would be acceptable but is not ideal. Do not under any circumstances use de-ionised or distilled water.

2.7.4 EL029 Reed Switch

EL029 Reed Switch



A reed switch sensor can be used to detect the presence of a magnetic field such as that from a bar magnet or an electromagnet. Alternatively, a simple, single-pole switch can be connected to terminals inside the EL029.

Size	72 x 45 x 28 mm
Operating range	0 to 99%
Maximum response time	2 ms

The EL029 Reed Switch may be connected to either Ext 1 or Ext 2 on DrDAQ.

To determine the optimum position for the magnet:
Hold the EL029 with its connecting socket towards you and the screw that fixes the lid facing up.
The best position for the magnet is about half way along (towards the bottom) of the right hand side of the EL029 case. The label on the bottom of the EL029 indicates this position.

When a magnet is in place next to the EL029 the Reed Switch inside the EL029 closes. This is shown in PicoScope and PicoLog by a change from 99% (switch open) to 0% (switch closed). Note that this percentage is an indication of the proportion of the sampling period that the switch is in the open state.

You may use the EL029 to connect a simple, single-pole switch (such as a micro-switch) to DrDAQ. You will need to obtain a switch and some insulated connecting wire. Remember to keep magnets or magnetic fields away from the EL029 when you use it with an external switch. Magnetic fields will still make the internal Reed Switch operate overriding the open setting of the external switch.

To connect the switch follow these steps:

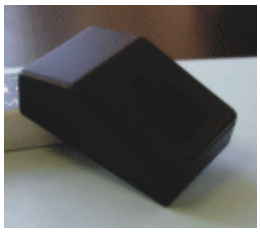
1. Make sure that the leads from the external switch have about 5 mm of bare wire to form the connections.
2. Remove the lid.
3. Put the EL029 on a firm flat surface, such as a table top, with the fixing screw facing up and the connection socket towards you.
4. Unscrew the fixing screw.
5. Lift off the lid of the EL029 and put it and the screw to one side. The lid should lift off easily.
6. Connect the wire.
7. Find the small terminal block, with two screws in its top, towards the back of the EL029 circuit board.
8. Loosen the screws in the terminal block. Do not take them out altogether.
9. Thread the two wires from the external switch through one of the holes in the bottom of the EL029.
10. Note: If you do not wish to use either of the two holes in the bottom of the EL029

case, make a hole in the side of the case just big enough for the leads to pass through. Take care not to damage the electrical components.

11. Leaving enough free wire to make the connection, secure the leads by looping them around the central boss of the EL029 case. Alternatively, put a 'tie wrap' around the leads inside the case and near the hole. Make sure that the leads are securely held in the EL029.
12. Push the bare ends of the leads into the holes in the back of the terminal block. There should be one lead in each hole.
13. Tighten the screws in the terminal block and check that the leads are held firmly in place. Do not overtighten the screws.
14. Put the lid back on to the EL029. Make sure it is on the right way round.
15. Tighten the fixing screw being careful not to overtighten it.
16. Check the operation of the external switch. Connect up the EL029 and make sure that PicoScope or PicoLog shows near to 100% when your external switch is open and 0% when it is closed.

2.7.5 DD101 Humidity Sensor

DD101 Humidity Sensor



DD101 Humidity Sensor is used to measure humidity by a 'non-condensing' technique. It has a short response time and plugs into the external sensor connections of DrDAQ.

Size	72 x 45 x 28 mm
Operating range	20% - 90% Relative Humidity
Overall accuracy	Reading + or - 10%
Operating temperature	0 - 60 °C
Resolution	0.2% Relative Humidity
Minimum response time	60 seconds with vigorous air movement
Maximum response time	60 minutes in still air

Caution: Do not allow the DD101 sensor to become wet. The DD101 is a non-condensing sensor and liquid entering the case (including condensation) may damage it.

The DD101 may be connected to either Ext1 or Ext2 on DrDAQ.

Tips for use of DD101 Humidity Sensors
The sensor responds to humidity changes more slowly in still air. If you need to increase the response time, then increase the air flow around the sensor. For example, swing the sensor gently on its lead or create a constant draft using a fan. Do not blow into the sensor, as your breath is very humid and will produce incorrect results. Never allow the sensor to get wet; for example, do not take a cold sensor into a warm humid environment where condensation may form inside the sensor's case.

Making accurate measurements
The DD101 will give good readings of humidity and show trends well. However if you need to make more accurate measurements you will have to calibrate the sensor. The calibrated sensor could have an accuracy as good as + or - 5% though this depends on how well the calibration procedure is carried out.
The output of the sensor may vary over time. For this reason you should calibrate the sensor regularly (at least once a year).

Basis of the calibration method
Calibration relies on the general physical properties of saturated solutions. The humidity above such solutions in closed containers is known quite accurately and is used to calibrate the sensors. Chemical salts used for this purpose include potassium chloride, magnesium chloride and magnesium nitrate.
Note: The humidity that exists above all such solutions varies with temperature.

Warning: The standard solutions may be harmful to your skin, eyes or when swallowed. Take all necessary precautions to avoid contact when preparing and using the standard solutions.

Note: For the highest accuracy you should compare the readings from you DD101 with the readings from a calibrated 'laboratory standard' reference humidity gauge.

Calibration of the sensor involves:

- Preparation of the calibration equipment.
- Measurement of the standard saturated solutions.
- Creation of calibration data for the DrDAQ software.
- A check that the calibrated sensor is accurate.

Note: You must measure at least two different standard solutions to provide two or more fixed points for calibration.

Equipment you will need:

- DrDAQ with DD101 sensor
- Pico DD100 Temperature Sensor
- At least two Standard Solutions
- A label for the DD101
- A test container
- An insulated box
- A stand for the DD101

[Optional: high accuracy, calibrated reference 'laboratory standard' humidity gauge]

Stand
It is very important that the standard solution does not enter the case of the DD101. The solution may damage the electrical components of the sensor. Therefore the DD101 needs to be suspended in the test container clear of the standard solution. A small stand is the best way to hold the sensor above the solution. The stand may be any object (open framework) that is impervious to the standard solution. Make sure that the material that your stand is made from is compatible with the test solutions you intend to use.

Test container
Your test container should be only just big enough to hold the DD101 and DD100 sensors, the stand, and a small quantity of standard solution (the solution should fill approximately 5% of the volume of your container). It will also have to be water-tight with a seal that allows you to pass the leads for the sensors into the container.
Note: The smaller your container, the more quickly the humidity will stabilise. You can reduce the time taken for the system to stabilise by forcing the air in the container to circulate. This will require a fan that does not introduce heat into the container (it must be driven by a shaft or magnetic coupling so that the motor is outside the container).
[If you are going to compare the values measured with the DD101 and a calibrated reference, your container will have to be large enough for the reference sensor too.]
Make sure that the material that your container is made of is compatible with the chemicals you are going to use. If the chemicals react in any way with the container, the humidity reference will not be correct.

Standard Solutions
Warning: The standard solutions may be harmful to your skin, eyes or when swallowed. Take all necessary precautions to avoid contact when preparing and using the standard solutions.

There are several chemicals that have been measured under laboratory conditions to find out what humidity they provide. Whichever chemicals you choose, make sure that the humidity they give is within the operating range of the DD101 sensor (that is, greater than 20% and less than 90% relative humidity). You should use at least two standard solutions to give two fixed points for your calibration. Standard reference books such as Kaye & Laby "Tables of Physical and Chemical Constants" (Longman) give tables with the humidity of standard solutions. For example:

Chemical	**Humidity at 20 C**
Potassium Chloride	85%
Magnesium Nitrate	54%
Magnesium Chloride	33%

Note: The humidity produced by the standard solution depends on the temperature. If the temperature changes from 20 to 21 Celsius this could give a change in relative humidity of as much as 3% for a nominal 50% value. This change would be worse for higher values of relative humidity.

Insulated box
Because the humidity produced by the standard solution depends on temperature, you should use your test container inside an insulated box. A domestic cool box (without the cold blocks) is suitable.

Preparation
Warning: The standard solutions may be harmful to your skin, eyes or when swallowed. Take all necessary precautions to avoid contact when preparing and using the standard solutions.

Prepare your standard solutions before you start the calibration procedure. Follow the instructions given by the manufacturers of the solutions you are using. A slurry of un-dissolved chemical in your solution should improve the stability. Make sure that all the containers you use for preparation are thoroughly clean before use as contamination of the solution will alter the humidity. Do not use tap water to make up the solution as this is insufficiently pure: use distilled and de-ionized water. Allow time for the solution to reach room temperature before use. Always use fresh solutions to ensure that the chemicals have not become contaminated or degraded.

Allow the standard solutions and other equipment to reach the same temperature (ideally 20°C) before you start.

Measure humidity
1. Setup the equipment
2. Clean the equipment. Make especially sure that the test container is clean; rinse it out with distilled water.
3. Put the test container into your insulated box.

Warning: The standard solutions may be harmful to your skin, eyes or when swallowed. Take all necessary precautions to avoid contact when preparing and using the standard solutions.

Caution: It is very important that the standard solution does not enter the case of the DD101. The solution may damage the electrical components of the sensor.

1. Put some of the standard solution into the test container. The solution should occupy about 5% of the volume of your container.
2. Put the stand into the test container. The stand should give you a clear platform above the level of the liquid. Do not allow any standard solution to spill on to the top of your stand.
3. Put the DD101 and DD100 sensors onto the stand. [If you are using a calibrated reference put this in too.]
4. Connect the DD101 and DD100 sensors to DrDAQ.
5. Take your measurements
6. Start the PicoLog Recorder software for DrDAQ. Make sure DrDAQ is receiving readings from the sensors on Ext 1 and Ext 2.
7. Seal the test container and close the insulated box (if you are using a fan, start the fan).
8. Record the temperature and humidity inside the test container for at least one hour. You must wait for the temperature to stabilise and the DD101 to provide correct readings. This may take up to eight hours if you have used a large container.
9. Check that the DrDAQ plots for temperature and humidity have been constant over the last few minutes of your measurements. [If you are using a calibrated reference take a reading of this now. Take care to minimise the changes in the setup so that the reading does not change significantly.]
10. Save your results.
11. Dispose of the solution as recommended by the supplier of the chemical. Remember to take adequate precautions to protect your skin and eyes when disposing of the chemicals.
12. Do the measurements again for the other standard solutions. You should end up with a set of recorded measurements for each standard solution. Remember that you must provide at least two fixed points for calibration of the sensor.

Create calibration data
When the measurements are completed you need to make a written calibration table.

1. From your recorded results find the place near the end of the measurement time where the readings are most stable.
2. Write down these temperature and humidity readings for the standard solution in a table.
3. Look up the humidity that the standard solution should give for the temperature that you have recorded. Write this value in your table next to the value measured by the DD101 (the manufacturer of the standard solution should have provided you with a table for humidities for different temperatures).
4. Fill in entries for all the standard solutions you have measured.

You should end up with a table something like this:

Chemical	Measured humidity	Standard humidity	Temperature
Potassium chloride	81%	85%	20°C
Magnesium nitrate	50%	54%	21°C
Magnesium chloride	30%	33%	20°C

Keep this table safe for further reference. It will be useful if you have to reset the software with the PicoLog Recorder 'New Settings' command.

At this stage compare the measured and standard values. If you find that the measured values differ from the standard values by more than 10%, then there is something wrong. You may have had a non-saturated solution, some contamination in the solution, inadequate sealing of your test container, or possibly a damaged sensor. Check your calibration routine. If you still get large errors contact Pico for assistance.

[If you are using a calibrated reference this should give values very close to those quoted for your standard solutions. If these values disagree by more than a few percent suspect your calibration procedure. When the two are in agreement use the values given by the calibrated reference as the correct figures for your data table.]

Once you have your calibration data you must enter it into the Pico software to calibrate the sensor:

1. Open the PicoLog Parameter Scaling Dialog
2. Start PicoLog Recorder
3. Click on Settings
4. Click on Input Channels - the DrDAQ Measurements dialog appears
5. Select the input channel for the humidity sensor - for example 'External 1 Humidity'
6. Click on the Edit button - the Edit DrDAQ Measurements dialog appears
7. Click on the Options button - the Parameter Options dialog appears
8. Click on the Scaling button - the Parameter Scaling dialog appears
9. Enter your calibration data
10. Pull down the Scaling Method list and select 'Table lookup'
11. To start your calibration table, click in the white rectangle - the text cursor appears
12. Type in the value that you measured with the DD101 under 'Raw'. Type a space then enter the standard value under 'scaled'. Press return (the enter key).
13. Enter all the pairs of measured and standard values that you have to build your table. You should end up with two columns of values. Note: Your table must have at least two pairs of values.
14. Click OK to close the dialogs
15. Click on the OK button to close the Parameter Scaling dialog - the Parameter Scaling dialog disappears
16. Click on the OK button to close the Parameter Options dialog - the Parameter

Options dialog disappears
17. Click on the OK button to close the Edit DrDAQ Measurements dialog - the Edit DrDAQ Measurements dialog disappears
18. Click on the OK button to close the DrDAQ Measurements dialog - the DrDAQ Measurements dialog disappears

This completes the entry of the calibration data.

Note: You can remove the calibration data by opening the Parameter Scaling dialog and setting the drop-down list to 'none'. If you want to completely delete the calibration scaling you can either edit and delete the entries in the Parameter Scaling dialog or use the 'New Settings' command from the main menu. If you use the 'New Settings' command all scaling data will be lost.

Check the calibration

To be sure that the calibration has been successful, you must repeat the measurement stage of the procedure. When the check measurements have been completed there should be very close agreement between the measured and standard values (that is, within 5% of the value). If this is not the case, check that you have entered the calibration data correctly and repeat the process.

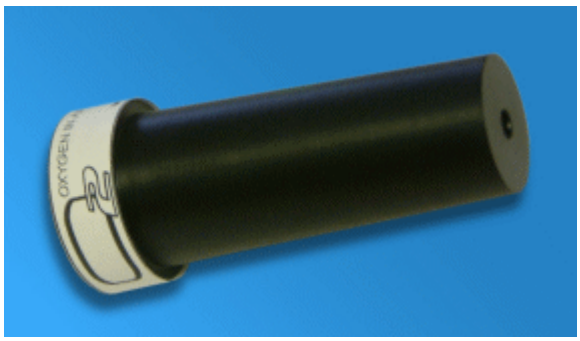
Once you have successfully calibrated your DD101, write the date on the label and stick the label on the sensor (do not block the holes in the sensor with the label).

Testing Sensors

You can check whether a sensor is working properly or not by running through the calibration procedure. If you find that the sensor values differ from the standard values by more than 10% you may have a damaged sensor. In this case contact Pico for assistance.

2.7.6 DD103 Oxygen Sensor

DD103 Oxygen Sensor



The DD103 Oxygen Sensor is used to measure the percentage of oxygen in a gas. The sensor plugs into the external sensor sockets of DrDAQ (Ext1 and Ext2) using the supplied cable.

Specification

Sensor Type	Galvanic Cell (lead-oxygen with weak acid electrolyte)
Input Range	0 - 100% oxygen
Accuracy (Calibrated)	±3.0% over operating conditions range
Response Times	< 15 seconds for 90% response < 25 seconds for 97% response
Max Humidity	0 - 95% Non-condensing
Operating Temperature	5^{O}C to 40^{O}C
Storage Temperature	-15^{O}C to 50^{O}C

Setting up the oxygen sensor
Since the Oxygen Sensor has an output of 0mV at 0% oxygen, a single point **calibration** calibration can be carried out. The procedure below guides you through setting up the oxygen sensor to work with PicoLog and calibrating it using the oxygen in the atmosphere as a reference point (calibration is necessary to use the sensor with ±3% accuracy).

If you have not yet set up DrDAQ with PicoLog, do so using the PicoLog help file before continuing below:

1. Connect the Oxygen Sensor to the socket labelled **Ext1** on the DrDAQ unit
2. From the **File** menu, select **New settings**
3. Click **OK**
4. Check that **DrDAQ** is selected as the **Converter type** and that the correct **port** is selected (Usually LPT1)
5. Click **OK**
6. Click **Add**
7. Under **Channel**, select **External 1** (The Scaling type will automatically change to **oxygen**)
8. Go back to the main monitor view by clicking **OK** on the two dialog boxes.

DrDAQ should be receiving information from the oxygen sensor. The sensor should be calibrated using the procedure below.

Calibrating the oxygen sensor
Both PicoLog and PicoScope provide the facility to enter scaling look-up tables. These instructions will concentrate on doing so in PicoLog (for instructions on Custom Ranges in PicoScope, see the PicoScope help file).

This is the information we have so far:

Raw Values after built in Scaling File (/%)	Calibrated Values known to be true (/%)
0	0
	20.9 (Known oxygen in air at sea level)

Only one more value is needed to complete the look-up table, if you are measuring the normal atmosphere then this reading is currently displayed on the PicoLog monitor view. Complete the table above with this value and follow the instructions below:

1. Select the **Settings | Input channels** menu item and click **OK**
2. Select the **Oxygen sensor** from the list then click **Edit**
3. Click on **Options**, then **Scaling**
4. From the drop down **Scaling method** menu, select **Table lookup**
5. From the look-up table you completed above, enter these values in the following format (replace 19.9 with whichever value you read from the monitor view):
   0 0
   19.9 20.9
   (Separate the values above with spaces)
6. Close all of the dialogs by clicking **OK** in each one

PicoLog's monitor view should now display 20.9% (or very close) for the oxygen content in air. If so, you've successfully calibrated your DrDAQ oxygen sensor.

Looking After the Oxygen Sensor
Ensure that the storage temperature and humidity ranges in the specification are not exceeded. The sensor is not designed for use in liquids and using it in this way will damage the sensor.

The sensor has a lifetime of 900,000 O_2% hrs; in air at sea level the sensor will operate effectively for over 4 years (this will decrease if the sensor is exposed to high oxygen concentrations for prolonged periods).

The lifetime of the sensor can be affected by exceeding the storage / operating ranges and exposure to some gaseous chemicals in high concentrations; please avoid the following: sodium hydroxide, acetone, MEK.

2.8 Making your own sensors

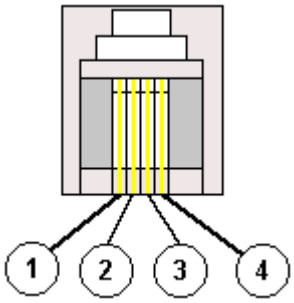
2.8.1 Overview

Making your own sensors for DrDAQ is quite straightforward, provided that you follow these guidelines.

Designing a DrDAQ sensor overview

Each external sensor socket has two channels ,one is an auto detect to inform the software which type of sensor it is and one analogue input that represents the sensor reading.

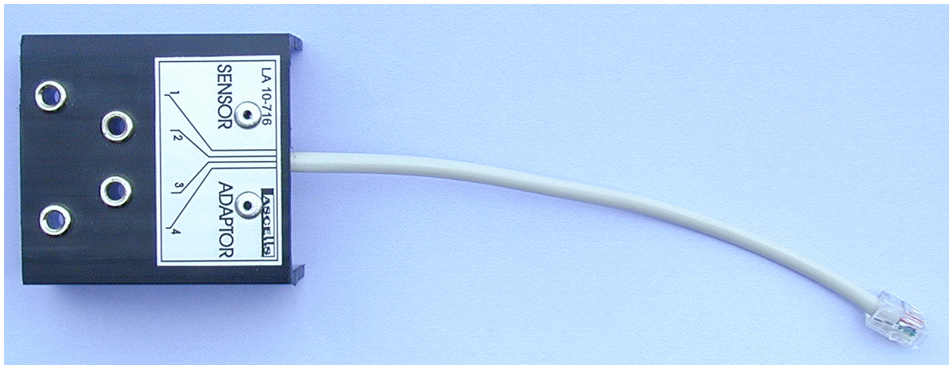
There are four pins on the External inputs. This view is looking into the Ext socket on DrDAQ in the direction of entry of the plug.



Pin 1 : Signal Input
Pin 2 : Ground
Pin 3 : Auto Detect
Pin 4 : VCC

Suitable connectors to these external inputs are FCC 68 4/4 plugs.

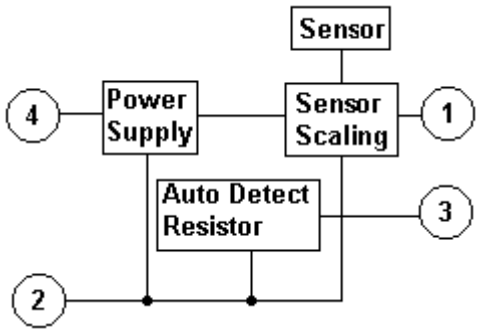
Alternatively, the DrDAQ sensor adapter (break out box) from Lascells, can be used. The pins are clearly labelled on the box.



The range of Auto Detect and Signal Input voltage must be between 0 and 2.5 volts. This is a hardware limitation and any signals outside of this range will not be read by the unit. Any voltages greater than +/- 30 volts fed directly into the device are likely to damage the unit, and can cause errors with all other DrDAQ readings.

The Signal Input channel has 100 kΩ pull-up resistors to 2V5, so that the input can be resistive as well as voltage.

A general DrDAQ sensor can be broken down into blocks:



1: = Signal input
2: = Ground
3: = Auto Detect
4: = Power

2.8.2 Powering the sensor

The definition of a sensor is:

"Device giving signal for detection or measurement of a physical property to which it responds"

The sensor properties must be known before work can begin in designing some way of interfacing it to the DrDAQ (Sensor scaling).

There are two types of sensor :-

Active Sensors: These sensors require power (Excitation) from an external source to generate an output signal, examples of active sensors include:

Property	**Sensor**	**Output**
Temperature	Silicon	Voltage / Current
	RTD	Resistance
	Thermistor	Resistance
Force / Pressure	Strain Gauge	Resistance
Acceleration	Accelerometer	Capacitance
Humidity	Capacitor	Resistance
Light	LDR	Resistance
Position	LVDT	AC Voltage

Passive Sensors: These do not require any power to generate an output, typical examples are:

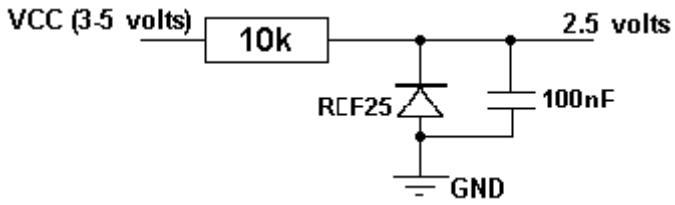
Property	Sensor	Output
Temperature	Thermocouple	Voltage
Force/pressure	Piezoelectric	Voltage
Position	Variable Resistor	Resistance
Light Intensity	Photodiode	Current

When selecting a sensor for an application you should consider the following:

1. Is the sensor Active or Passive?, if Active then can it be powered by the DrDAQ?
2. What is the output of the sensor?, can the sensor be plugged directly into the DrDAQ?
3. Is the sensor already available on the DrDAQ, or are there more suitable sensors out there? ... Why re-invent the wheel?

The power available from the parallel port will vary from PC to PC, however will typically be 2 mA at 3 - 5 V. Because the voltage varies, this needs to be taken into account when designing a sensor (it should be essential that the sensor can be used with DrDAQ on all machines). In order to guarantee a consistant power supply to your sensor it is necessary to assume the worst, i.e. the minimum voltage according to the DrDAQ specification could be 3 V. See the circuit below for an example of a simple power supply circuit.

Power supply example:



- VCC is drawn from Pin 4 on the DrDAQ socket
- GND is connected to Pin 2 on the DrDAQ socket
- REF25 is a 2.5 V reference

2.8.3 *Scaling*

Sensor Scaling - Software
It is necessary to create a scaling file so that the software can extract details about the sensor. Details on creating scaling files can be found on the DrDAQ Scaling Files (.DDS) topic. Scaling in software also provides an opportunity to perform a calibration on the sensor, this can compensate for any manufacturing irregularities.

Sensor Scaling - Hardware
With both these types of sensors it is essential that the DrDAQ displays an accurate representation of the property to be measured, there are many factors to take into account when designing scaling circuitry:

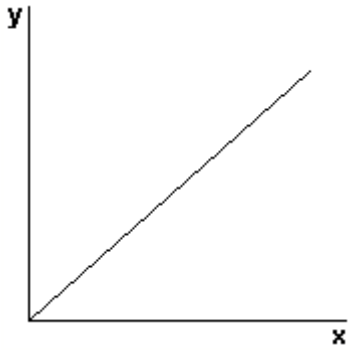
Sensitivity
The DrDAQ has 10-bit resolution over the 0 to 2.5 V input range, this means the sensitivity is:

2500 mV / 2^10 = 2.44 mV

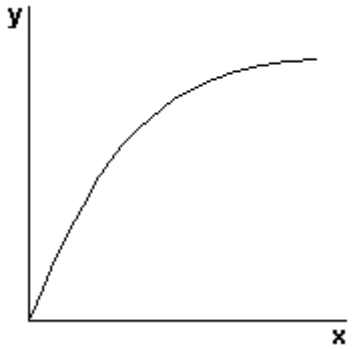
This means that the DrDAQ can detect changes in voltage as low as 2.44 mV. To make the most of the resolution the signal output from the sensor should use as much of the input range of the DrDAQ as possible.

Linearity
As illustrated below, not all sensors have a linear response.

Linear Response



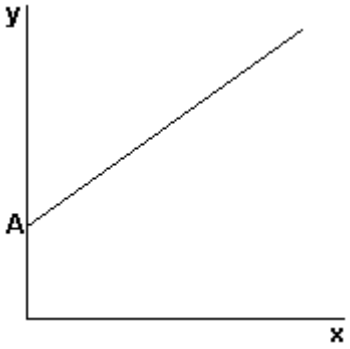
An Example of Non-Linear Response



A linear response is one whose output-versus-input graph is a straight line. A typical example of this type of response is a photodiode.

Note: a non-linear response curve cannot be calculated using a simple $y = ax + b$ formula. A typical example of this type of response is an LDR.

Offsets



The above linear/non-linear responses could have a voltage/current/resistive, etc offset that may need taking into account. It is quite usual to see offsets in output signals from sensors.

Sensor output
The DrDAQ requires an input signal in DC volts or resistance, the sensor in question could give an output in other units such as resistance, current or AC volts.

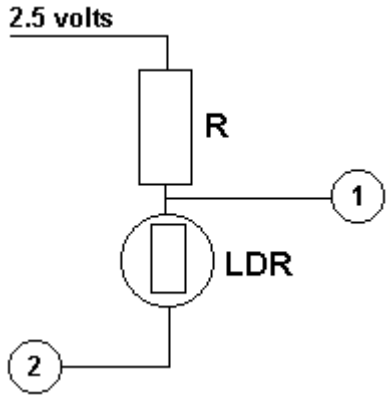
Drift
It is possible that the sensor output drifts over time, this may involve re-calibration of the sensor at set time periods, or some self calibration.

Hysteresis
The sensor may give a different output value when rising to a voltage, than if it falls to a voltage. This is something that is difficult to eliminate and should be taken into account.

Once the sensor has been selected and all of the above has been considered the design of some scaling circuitry can begin. There are many ICs on the market that can be purchased cheaply that deal with the above potential problems, look into these before re-inventing the wheel and spending expensive development time designing a circuit that can be purchased for a few pounds.

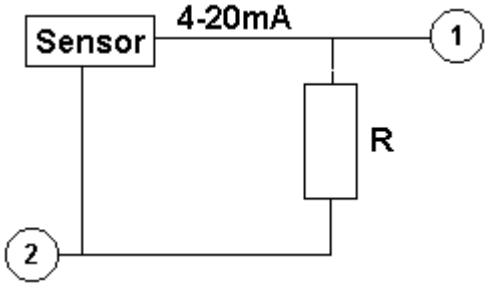
Some simple scaling examples:

Example 1: Interfacing a Resistive sensor



Notes:
Light-dependent resistor (LDR) can be used in conjunction with a fixed resistor to measure light level. A suitable LDR sensor can be obtained from Maplin Electronics (part number N53AY). A resistor (R) of around 500k is suitable.

Example 2: Measuring 4-20 mA Current



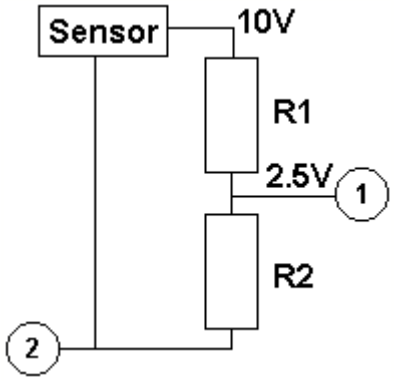
Notes:
-20 mA output is the Industry standard for interfacing between a control center (data logger) and a remote process such as a sensor, for this reason 4-20 mA sensors are very common.

For relatively small currents a simple shunt resistor can be used to convert the current into voltage which the DrDAQ can then measure.

A suitable resistor (R) for the DrDAQ is 120 ohms. This would give (using Ohm's Law):

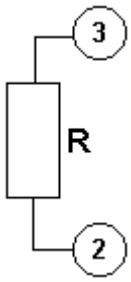
0.4 volts at 4 mA
2.4 volts at 20 mA

Example 3: Measuring 10 volts



Notes:
This is a simple circuit that uses a potential divider to reduce the voltage going into the DrDAQ by four. R1 = 3k & R2 = 1k The resistors are calculated using Ohm's Law, V=IR.

Auto-Detect



Auto detect resistor is placed between Ground (Pin 2) and Signal Detect (Pin 3). The Value for the resistor should be selected from one of the following common values: 1k0, 2k2, 3k3, 5k6, 7k5, 10k. It is necessary to have an auto-detect resistor so that the software can automatically scale the input signal into the property that it represents.

2.8.4 *DrDAQ scaling files (.DDS)*

The DrDAQ driver has built-in scaling for each of the built-in and Pico-supplied sensors. You can incorporate scaling for your own sensors by adding a file called drdaq.dds, which contains details of your sensor.

The values returned by the driver are integers that represent fixed-point decimal number. For example, the driver treats pH as a value with two decimal places, so a pH of 7.65 is returned as 765.

You can call the routine drdaq_get_channel_info to find out how many decimal places a channel is using, and also to get a divider that converts the integer value to the corresponding real number. For pH, the returned divider is 100, so 765 divided by 100 gives 7.65.

For some sensors, there is more than one possible scaling available. You can call drdaq_get_scalings to get a list of valid scaling codes, then call drdaq_set_scaling to select one of them. Once selected, drdaq_get_channel_text and drdaq_get_channel_info will return full information about the selected scaling. If you do not use drdaq_set_scaling, the driver will automatically select the first available scaling for each channel.

DrDAQ scaling files can be used to supplement the scalings built into the driver. Several DDS files may be used and these must be placed in the current working directory, where the DrDAQ software is installed (normally C:). The filename must conform to the DOS format of eight characters followed by the .dds extension. The total number of sets of scaling data in all the files used must not exceed 99.

Each scaling file may contain more than one set of scaling data. Each scaling must have a unique scaling number, contained in the `[Scale...]` section heading: Numbers above

100 are reserved for Pico Technology, customers should use numbers below 100 for custom scaling.

A typical entry from a .DDS file is shown below:

```
[Scale100]
Resistor=330
LongName=Temperature
ShortName=TempC
Units=C
MinValue=-40
MaxValue=120
OutOfRange=0
Places=1
Method=0
IsFast=Yes
NoOfPoints=32
Raw1=2.385
Scaled1=-30
Raw32=1.32
Scaled32=100

[Scale101]
Resistor=330
LongName=Temperature
ShortName=TempF
Units=F
MinValue=32
MaxValue=160
.....

[Scale103]
Resistor=270
LongName=Light
ShortName=Light
Units=LUX
MinValue=0
MaxValue=20000
.....
```

Explanation of each term in the .DDS file is as follows:

```
[Scale100]
```

A unique number to identify this entry. Pico created numbers are from 100 upwards. Customers should use numbers below 100.

```
Resistor=330
```

The ID resistor value in kohms. Values should be entered as decimals, i.e. 3.3 not 3k3.

Note: If your DDS file contains two scales that make use of the same resistor value PicoScope and PicoLog behave differently. PicoScope will not let you choose which of the scales to apply. If you have only one DDS file PicoScope will use the first occurrence of the scale in the file and ignore those that come after. If you have more than one DDS file the scale that PicoScope uses will be arbitrary. PicoLog will allow you to select which of the scales you wish to apply.

For external sensors, this resistor should be fitted in the sensor. Customers should use one of the following resistors: 1k0, 2k2, 3k3, 5k6, 7k5 and 10k. The resistor used should be 1% tolerance or better.

The following resistor values are reserved and must not be used: 330k, 220k, 180k, 150k, 120k, 110k, 100k, 91k, 82k, 75k, 68K.

For internal sensors, the following 'virtual' resistor values should be used:

1	Sound Waveform	1200
2	Sound Level	1300
3	Voltage	1500
4	Resistance	1600
5	pH	1400
6	Temperature	1100
7	Light	1000

```
LongName=Temperature
```

Used in PicoLog

```
ShortName=TempC
```

This field is not used by DrDAQ running PicoScope or PicoLog.

```
Units=C
```

Displayed on graphs

```
MinValue=-40
MaxValue=120
```

Note: For PicoScope these values will determine the maximum and minimum values displayed in Scope View. For PicoLog these values determine what Maximum range is displayed in the Graph View (set in the Graph Options dialog).

```
Places=1
```

Number of decimal places. The options are 0,1,2 and 3. With places=1 the value 15.743 would be returned as 157, meaning 15.7. With places=2, the same number would be returned as 1574.

```
Method=0
```

This specifies the scaling method. At the moment, 0 (table lookup) is the only method.

```
OutOfRange=0
```

This specifies what to do if the raw value is outside the range of the table lookup. The options are:
0 - treat as a sensor fail
1 - clip the value to the minimum or maximum table value
2 - extrapolate the value using the nearest two table entries.

```
Fast=Yes
```

This is Yes if the sensor is capable of generating high-speed information.

```
NoOfPoints=32
```

This is the number of table lookup points.

```
Raw1=2.385
```

Raw value for the first point in the look up table. The value is in V (volts) and should not be greater than 2.500 V.

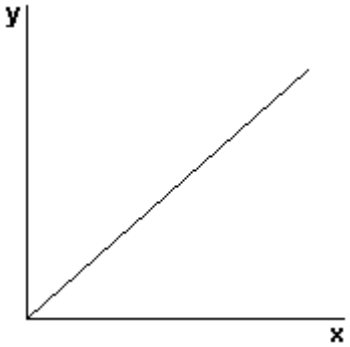
```
Scaled1=-30
```

Scaled value for the first point in the look up table. The units are specified by the units parameter.

2.8.5 Calibration

Calibration with DrDAQ and its sensors, is achieved with Scaling Files. Calibration involves measuring a known value (such as the temperature of boiling water, or the resitance of a known resistor, or the values shown by a calibrated sensor) then converting these values into the units you need.

Linear



Single-Point Calibration
When calibrating a sensor with a linear response curve it may only be necessary to perform a **Single-Point Calibration**. This only occurs in one circumstance:

One pair of values is known - for example on the DrDAQ oxygen sensor. The sensor is known to output 0mV at 0% oxygen content. Therefore we only need to find one other pair of values. All we are doing here is changing the gradient of the response curve. See the Oxygen Sensor description for a full example.

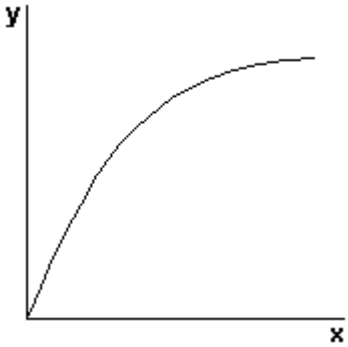
Two-Point Calibration
A two-point calibration should always be carried out if possible, since it is the most accurate calibration method for a sensor with a linear response curve.

It is important that the readings are taken for both extremes of the sensor's **operating input range** to ensure accuracy. For example:

A temperature sensor has an **input range** of -200°C to +200°C but we only want to use it between -5°C and 110°C therefore this will be its **operating input range**. The first calibration point should be close to -5°C, so the sensor will be placed in melting ice, which is known to be 0°C. The second calibration point should be close to 110°C, so the sensor will be placed in boiling water, which is known to be 100°C.
It is rare that such a convenient range is available, but the principle of two point calibration stays the same.

It is important to distinguish between the operating range and the full input range of the sensor, since it would be far more difficult to calibrate the sensor between -200°C and +200°C and would yield no accuracy benefit for the user between -5°C and 110°C.

Non-Linear



Multi-point Calibration

You will need more than two points to describe a non-linear response curve, therefore you will need to create a look-up table experimentally. Since you will be measuring a series of readings at unconfirmed intervals, the only way to know the calibrated value, is to perform the same measurement with both the sensor you are calibrating and a sensor which you know has already been calibrated. Create a table of raw and scaled values similar to below.

x (raw) new sensor	y (scaled) calibrated sensor
0	0
5	8
10	12
15	14
20	15

Both PicoLog and PicoScope have the facility to manually enter look-up tables (see the respective help files). You can also make a more permanent reusable scaling table with DrDAQ Scaling Files (.DDS).

3 Software information

3.1 Software updates

Our software is regularly updated with new features. To check what version of the software you are running, start PicoScope or PicoLog and select the **Help/About** menu. The latest version of software can be downloaded free of charge from the DrDAQ web site at http://www.drdaq.com. Alternatively, the latest software can be purchased on disk or CD from Pico Technology.

To be kept up-to-date with news of new software releases join our e-mail mailing list. This can be done from the main Pico Technology web site at http://www.picotech.com/.

3.2 Writing your own software

DrDAQ is supplied with driver routines that you can build into your own programs. The driver routines are supplied as Dynamic Link Libraries for Windows 95, 98, NT and 2000. These can be used with C, Delphi and Visual Basic programs, and programs like Microsoft Excel, where the macro language is a form of Visual Basic.

Once you have installed the software, the DRIVERS sub-directory contains the drivers and a selection of examples of how to write your own programs. It also contains a copy of this help file in text format.

The following table describes each of the routines supplied with the drivers:

Routine	Function
drdaq_get_driver_version	Check that this is the correct driver
drdaq_open_unit	Open the driver to use a specified printer port
drdaq_set_unit	Select which DrDAQ unit to use
drdaq_close_unit	Close the specified printer port
drdaq_apply_fix	Modify behaviour of DLL
drdaq_set_do	Set the digital output
drdaq_set_led	Set the LED
drdaq_get_value	Get a single reading from one channel
drdaq_set_trigger	Set a trigger event from a specified channel
drdaq_set_interval	Set the channels and time interval for the next call to drdaq_get_values, or drdaq_get_times_and_values
drdaq_get_values	Get a block of readings at fixed intervals
drdaq_get_time_and_value	Get a reading and the time the reading was taken
drdaq_get_times_and_values	Get a block of readings and their times, at fixed intervals
drdaq_get_unit_info	Get information about an DrDAQ unit
drdaq_get_channel_text	Get text information about a DrDAQ channel (eg name, units)
drdaq_get_channel_info	Get numeric information about a DrDAQ channel (eg min/max value)
drdaq_get_scalings	Get the scaling options for a channel
drdaq_set_scaling	Specify the scaling for a channel

The driver offers the following facilities:

- Specify the printer port that is connected to DrDAQ
- Get information about the scalings available for a channel
- Select a scaling for a channel
- Take a single reading from a specified channel
- Specify a trigger event from a specified channel (only available in block mode)
- Collect a block of samples at fixed time intervals from one or more channels

You can specify a sampling interval from 50 microseconds to 1 second. If you specify an interval that is shorter than your computer can manage, the driver will tell you how long it will actually take to collect the specified number of samples.

Under Windows, the sampling may be affected by Windows activities. At the least, there will be gaps in the data every 55 milliseconds due to the Windows timer function. There will be additional gaps if you move the mouse, or have other programs running. We therefore recommend using the drdaq_get_times_and_values routine, so that you can determine the exact time that each reading was taken.

The normal calling sequence to collect a block of data is as follows:

```
1.  Check that the driver version is correct
2.  Open the driver
3.  Select channel scalings (if required)
4.  Set trigger mode (if required)
5.  Set sampling mode (channels and time per sample)
6.  While you want to take measurements,
7.        Get a block of data
8.  End while
9.  Close the driver
```

3.2.1 Driver formats

3.2.1.1 Windows 95/98

When running under Windows 95, an application is not in complete control- Windows can interrupt at any time. Interruptions occur every 55 milliseconds, and are also caused by mouse and keyboard input. As a consequence, the driver cannot always take readings at fixed time intervals. To deal with this, the driver returns the time at which each reading was taken.

The 32-bit Windows 95 driver, PICO.VXD, is installed in windows, It is loaded using a reference in system.ini:

```
[386enh]
......
.....
device=pico.VXD
```

The 32-bit Windows 95/98 driver is accessed using the file DRDAQ32.DLL: it is installed in drivers. The DLL uses STDCALL linkage conventions, and undecorated names.

The 32-bit DLLs for Windows 95 and Windows NT use the same calling conventions, so a 32-bit application will run without modifications on either system. Note, however, that the two operating systems require different versions of the DLL file.

3.2.1.2 Windows NT/2000

The Windows NT/2000 driver, PICO.SYS, is installed in windows. The operating system must be told that the driver is available: this is normally done automatically by the setup program, but can also be done manually using the the regdrive.exe program which is copied into the PICO directory. Type in :

```
regdrive pico
```

The Windows NT 32-bit driver is accessed using the file DRDAQ32.DLL: it is installed in drivers. The DLL uses STDCALL linkage conventions, and undecorated names.

The 32-bit DLLs for Windows 95 and Windows NT use the same calling conventions, so a 32-bit application will run without modifications on either system. Note, however, that the two operating systems require different versions of the DLL file.

3.2.2 Routines

3.2.2.1 drdaq_get_driver_version

```
PREF1 short PREF2 drdaq_get_driver_version (void);
```

This routine returns the version number of the DrDAQ driver. You can use it to check that your application is used only with the driver version that it was designed for use with.

Generally speaking, new driver versions will be fully backward compatible with earlier versions, though the converse is not always true, so it should be safe to check that the driver version is greater than or equal to the version that it was designed for use with.

Arguments:		
	`none`	

Returns:		
		- driver version. Upper byte is the major version; lower byte is the minor version. For example, 0x0301

3.2.2.2 drdaq_open_unit

```
PREF1 short PREF2 drdaq_open_unit (short port);
```

This routine opens the DrDAQ unit connected to the specified port. It then calibrates the timing functions for the computer.

If you wish to use more than one DrDAQ unit at the same time, you should call drdaq_open_unit for each of the units, then use drdaq_set_unit to select which unit to use next.

Arguments:		
	`port`	- the number of the parallel port that the DrDAQ is connected to (1 for LPT1, 2 for LPT2, and so on)

Returns:		
		- TRUE if successful. If it is not successful, you can call drdaq_get_unit_info to find out why it failed

3.2.2.3 drdaq_apply_fix

```
PREF1 void PREF2 drdaq_apply_fix (short fix_no, short value);
```

This procedure is used to modify the behaviour of the driver.

Arguments:		
	fix_no	- the attribute to modify: 0: Temperature unit 1: pH compensation
	value	- the new value for the attribute: If fix_no = 0 (set temperature unit): 0: Celsius 1: Fahrenheit 2: kelvin If fix_no = 1 (select pH compensation): 0: Internal 1: External 1 2: External 2 3: None 4: Auto select between External 1 and Internal

Returns:		
		- none

3.2.2.4 drdaq_close_unit

```
PREF 1 void PREF2 drdaq_close_unit (short port);
```

This routine closes the DRDAQ unit on the specified port.

Arguments:		
	port	- the number of the parallel port

Returns:		
		- none

3.2.2.5 drdaq_set_unit

```
PREF1 short PREF2 drdaq_set_unit (short port);
```

This routine is used to select the unit to use for subsequent operations. It is only necessary to use this function if you wish to have more than one unit open at the same time.

Arguments:		
	port	- the parallel port number, as outlined in drdaq_open_unit

Returns:		
		- TRUE if successful; FALSE if a unit is not open on that port

3.2.2.6 *drdaq_set_do*

```
PREF 1 void PREF2 drdaq_set_do (short do_value);
```

This routine sets the state of the digital output pin for the currently selected DrDAQ unit. Any non-zero value will turn the digital output on, zero will turn it off.

Arguments:		
	do_value	- 1 - turns the digital output on - 0 - turns the digital output off

Returns:		
		- none

3.2.2.7 *drdaq_set_led*

```
PREF 1 void PREF2 drdaq_set_led (short do_value);
```

This routine sets the state of the LED for the currently selected DrDAQ unit. Any non-zero value will turn the LED on, zero will turn it off.

Arguments:		
	do_value	- 1 - turns the LED on - 0 - turns the LED off

Returns:		
		- none

3.2.2.8 *drdaq_get_value*

```
PREF 1 short PREF2 drdaq_get_value (short channel);
```

This routine reads the current value of one channel from the currently selected DrDAQ. Depending on your computer, it will take approx 200µs to take one reading.

Value scaled using the current scaling for this channel.

See also drdaq_get_value_and_time, which reports the exact time at which the reading was taken.

Arguments:		
	channel	- the channel number is between 1 and 9: See channel scaling for more information.

Returns:		
		- none

3.2.2.9 *drdaq_get_value_and_time*

```
PREF 1 void PREF2 drdaq_get_value_and_time (  short channel,
                                              unsigned long *
sample_time,
                                              short * value);
```

This routine reads the current value of one channel from the currently selected DrDAQ, and the time in microticks at which the reading was taken. Depending on your computer, it will take approx 200 µs to take one reading.

Value scaled using the current scaling for this channel.

Arguments:		
	channel	- the channel number is 1 through to 9: see channel scaling for more information
	sample_time	- is the time in microticks for the reading. There are 2^{32} microticks per hour or 1,193,046 per second. The sample time will therefore wrap around once an hour
	value	- a pointer to

Returns:		
		- none

3.2.2.10 *drdaq_set_trigger*

```
PREF1 void PREF2 drdaq_set_trigger (    short enabled,
                                        short auto_trigger,
                                        short auto_ms,
                                        short channel,
                                        short dir,
                                        short threshold,
                                        short delay);
```

This routine defines a trigger event for the next block operation, and specifies the delay between the trigger event and the start of collecting the data block. Note that the delay can be negative for pre-trigger.

Arguments:		
	enabled	- this is TRUE if the DrDAQ is to wait for a trigger event, and FALSE if the DrDAQ is to start collecting data immediately
	auto_trigger	- this is TRUE if the DRDAQ is to trigger after a specified time (even if no trigger event occurs). This prevents the computer from locking up, if no trigger event occurs
	auto_ms	- specifies the time in ms after which auto_trigger will occur
	channel	- specifies which channel is to be used as the trigger input. The channel number is between 1 and 9: see channel scaling for more information
	dir	- the direction can be rising or falling
	threshold	- this is the threshold at which a trigger event on the specified channel takes place. It is scaled using the currently selected scaling for the trigger channel. For more information, see Scaling
	delay	- this specifies the delay, as a percentage of the block size, between the trigger event and the start of the block. Thus, 0% means the first data value in the block, and -50% means that the trigger event is in the middle of the block

Returns:		
		- none

3.2.2.11 *drdaq_set_interval*

```
PREF1 unsigned long PREF2 drdaq_set_interval ( unsigned long
us_for_block,
                                               unsigned long
ideal_no_of_samples,
                                               short * channels,
                                               short no_of_channels);
```

This routine specifies the time interval per sample and the channels to be used for calls to drdaq_get_values or drdaq_get_times_and_values.

Arguments:		
	us_for_block	- target total time in which to collect ideal_no_of samples, in us
	ideal_no_of_samples	- specifies the number of samples that you intend to collect. This number is only used for timing calculations: you can actually collect a different number of samples when you call drdaq_get_values
	channels	- a pointer to an array, listing the channels to be used. The channel numbers are between 1 and 9: see channel scaling for more information
	no_of_channels	- specifies the number of channels used

Returns:		
		- the time taken, in ms, to collect the block (might be greater than time requested)

An example of a call to this routine using channels 2, 3 and 5 is:

```
int channels [3];
channels [0] = 2;
channels [1] = 3;
channels [2] = 5;


drdaq_set_interval (10000, 100, channels, 3);
```

The routine returns the actual time to collect this number of samples. This actual time may be greater than the target time if you specified a sampling interval that is faster than your computer can manage. If the specified sampling rate was too fast, you have the following choices:

- if the total time is important, collect fewer than the ideal number of samples so that the total block time is correct
- if the number of samples is important, collect the same number of samples then allow for the fact that they took longer to collect.

3.2.2.12 *drdaq_get_values*

```
PREF 1 unsigned long PREF2 drdaq_get_values ( short HUGE * values,
                                              unsigned long
no_of_values);
```

This routine reads in a block of readings at intervals and from channels specified in the most recent drdaq_set_interval call.

Arguments:		
	values	- a buffer for the readings. The readings are scaled using the currently selected scaling for each channel. If multiple channels are selected, the readings for the channels are interleaved. For example, with channels 1, 3 and 8, the readings in the buffer would be 1,3,8,1,3,8,1,3,8....
	no_of_values	- the number of readings to collect

Returns:		
		- the actual time for the block (or zero if a key is pressed)

If a key is pressed while collecting, the routine will return immediately. The return value will be zero if a key was pressed, and the total time in microseconds if a block was successfully collected.

3.2.2.13 *drdaq_get_times_and_values*

```
PREF1 unsigned long PREF2 drdaq_get_times_and_values (    long HUGE *
times,
                                                          short HUGE *
values,
                                                          unsigned long
no_of_values);
```

This routine takes a block of readings at nominal intervals specified in the most recent drdaq_set_interval call, and returns the actual times for each reading.

Arguments:		
	time	- the time, in microseconds, at which each reading was taken. The trigger event is at time 0
	values	- a buffer for the readings. The readings are scaled using the currently selected scaling for each channel. If multiple channels are selected, the readings for the channels are interleaved. For example, with channels 1, 3 and 8, the readings in the buffer would be 1,3,8,1,3,8,1,3,8....
	no_of_values	- the number of readings to collect

Returns:		
		- the actual time for the block (or zero if a key is pressed)

If a key is pressed while collecting, the routine will return immediately. The return value will be zero if a key was pressed, and the total in microseconds if a block was successfully collected.

3.2.2.14 *drdaq_get_unit_info*

```
PREF1 void PREF2 drdaq_get_unit_info (  char * str,
                                        short str_lth,
                                        short line,
                                        short port);
```

If the unit on the specified port failed to open, this routine fills str with a string that explains why the unit was not opened.

If the specified unit is open, The routine returns version information about the DrDAQ DLL, the Windows driver and the sampling rate.

Arguments:		
	str	- text buffer for returned string
	str_lth	- length of str (ie maximum string that can be returned)
	line	- line number (0..4)
	port	- lpt port number (1..3)

Returns:		
		- none

The return value is the length of the string placed in `str`: it will be zero if the line or port are invalid.

3.2.2.15 *drdaq_get_channel_text*

```
PREF1 short PREF2 drdaq_get_channel_text (    char * str,
                                              short channel,
                                              short field);
```

This routine returns one of five text strings, depending on the value of field.

Arguments:		
	`str`	- text buffer for returned string
	`channel`	- line number (1..9 - see channel scaling)
	`field`	- field number 0 - channel name 1 - short channel name (5 chars max) 2 - scaling name for currently selected scaling 3 - short scaling name (5 chars max) 4 - units for currently selected scaling

Returns:		
		- 1 if successful; 0 if unsuccessful

The return value is the length of the string placed in `str`: it will be zero if the line or port are invalid.

3.2.2.16 *drdaq_get_channel_info*

```
PREF1 short PREF2 drdaq_get_channel_info (    short * min_value,
                                              short * max_value,
                                              short * places,
                                              short * divider,
                                              short * is_fast,
                                              short channel);
```

This procedure returns as set of information about the currently selected scaling for the specified channel. If a parameter is not required, you can pass a null pointer to the routine.

Arguments:		
	min_value	- The minimum value that the channel can take
	max_value	- The maximum value that the channel can take
	places	- The number of decimal places
	divider	- The number that values should be divided by, to give real numbers
	is_fast	- TRUE (1) if the channel is capable of providing rapidly changing signals
	channel	- This specifies the channel to return details for

Returns:		
		- 1 if successful; 0 if unsuccessful

3.2.3 Programming support

3.2.3.1 C++

C++ programs can access all versions of the driver. If drdaqw.h is included in a C++ program, the PREF1 macro expands to extern "C": this disables "name-decoration", as Microsoft calls it, and enables C++ routines to make calls to the driver routines using C headers.

See the C examples for sections of C code to interface with the driver.

3.2.3.2 C

The C example program is a generic windows application- ie it does not use Borland AppExpert or Microsoft AppWizard. To compile the program, create a new project for an Application containing the following files:

```
            drdaqtes.c
            drdaqtes.rc
either      drdaq32.lib (Borland applications)
or          drdaqms.lib (Microsoft Visual C applications)
```

The following files must be in the same directory:

```
            drdaqtes.rch
            drdaqw.h
            drdaq32.dll  (all applications)
```

3.2.3.3 Delphi

drdaq.dpr is a complete program that demonstrates the use of DrDAQ .

The file drdaqfm.inc contains a set of procedure prototypes that you can include into your own programs.

3.2.3.4 Visual Basic

The DRIVERS sub-directory contains the following files:

DRDAQ32.VBP
DRDAQ32.BAS
DRDAQ32.FRM

3.2.3.5 Excel

The easiest way to get data into Excel is to use the Picolog program.

However, you can also write an Excel macro which calls drdaqxx.dll to read in a set of data values. The Excel Macro language is similar to Visual Basic.

The example drdaqxx.XLS reads in 20 values from channels 1 and 2, one per second, and assigns them to cells A1..B20.

Note: it is usually necessary to copy the .DLL file to your directory.

Excel 97 example
The file DrDaq97.xls contains a more advanced example for Excel 97.

The default setting for the example takes 100 readings from the light, sound and temperature channels and displays the results in separate cells

The following can be configured via the settings button:

- Port number
- Channels to take readings from
- No of readings to be taken
- Continuous readings on/off (Displayed in a single cell only)
- LED on/off
- Set up a trigger
- Target time to collect number of samples

To stop the application running press the Stop button.
To run the application press the Run button.
To clear the screen of data press the Clear button.

3.2.3.6 LabVIEW

The routines described here were tested using LabVIEW for Windows 95 version 4.0.

While it is possible to access all of the driver routines described earlier, it is easier to use the special LabVIEW access routines if only single readings are required. The drdaq.llb library in the DRIVERS sub-directory shows how to access these routines.

To use these routines, copy drdaq.llb and drdaq32.dll to your LabVIEW user.lib directory. You will then find a sub-vi to access a DrDAQ channel, and an example sub-vi that demonstrates how to use it.

You can use one of these sub-vis for each of the channels that you wish to measure. The sub-vi accepts the port (1 for LPT1) and channel and returns a value.

3.2.3.7 HP-Vee

The example routine `drslow.vee` is in the drivers sub-directory. It was tested using HP-Vee version 5 under Windows 95 and Agilent Vee 6 (beta) under Windows 98.

The example shows how to collect a block of data from DrDAQ.

Index

A

C

D

E

G

H

L

M

O

P

R

S

T

V

W

Pico Technology Ltd

The Mill House
Cambridge Street
St Neots PE19 1QB
United Kingdom
Tel: +44 (0) 1480 396 395
Fax: +44 (0) 1480 396 296
Web: www.picotech.com