

1 Introduction to DALONO

DALONO data logger is excellent value logger for voltage based sensors. The instrument is based on high quality, multi-channel, USB voltmeter and custom made software.

The precision of the instrument is in microvolt range so reading sensors and thermocouples accurately is no problem. The footprint of the system is a fraction of that of a laboratory bench-top multi-meter, and so is the cost of the unit.

The software includes possibility for sensor calibration, automatic data saving in .csv / text format, and appealing live graphs.

The instrument has 48 nV resolution and 0.1% accuracy on the selected range, ranges from ± 39 mV to ± 2500 mV.

The instrument can be connected up to 8 sensors. The software can be calibrated to linear sensors as well as complex (polynomial, etc.) fitting expressions, and can log and plot all 8 channels simultaneously.

On request some or all of the channels of the instrument can be configured also log current based sensors such as 4-20 mA.

2 Installation

Do not plug in the instrument to USB port before the drivers are installed.

2.1 Software

Download the latest version from <http://www.norecs.com/index.php?page=Software> -> DALONO

The file is downloaded as .zip

Unfortunately Windows shows the contents of .zip files like if they were regular files in regular folders which they are not, and this can cause confusion sometimes.

To unpack the .zip archive: Navigate so that you see the DALONO2 yyyy-mm-dd.zip and right click the file. Select unzip, extract or similar option and follow the instructions until the files are extracted to the disk.

After done extracting, delete the DALONO2 yyyy-mm-dd.zip

Move all the extracted files and folders to your desired location such as C:/DALONO/

Please note that you cannot place the files to 'program files' or any folder underneath 'program files' folder. Windows prevents programs to write files and folders to this location, and DALONO autosave needs to write files and folders.

Go the DALONO/DLL/ folder and run the DPInst.exe in the 32bit/ or in the 64bit/ folder. If you do not know which folder to use, pick one and the computer will tell you if you are in the correct folder. If you had the wrong folder, try the other one. Then follow instructions given by the DPInst.exe and accept and allow and confirm whatever Windows asks.

2.2 Instrument

Once the drivers-installer in the DALONO/DLL folder has been successfully ran, you can connect the instrument to a free USB port on your computer.

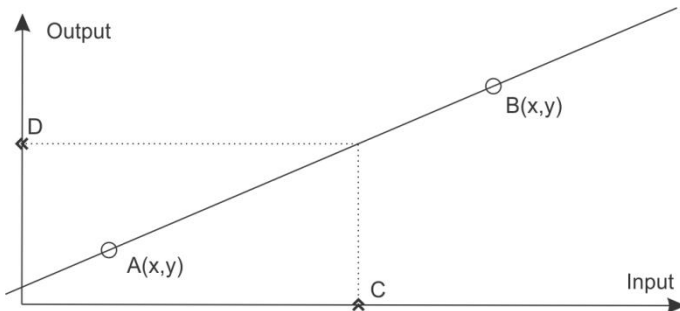
2.3 Sensors

Depending on which options were included with your DALONO setup, you may have enclosure housing sensors. Plug in the 25 pin cable from the sensor housing to the instrument.

2.4 Test

Run the DELONO2.EXE and take note of the feedback.

3 Calibration



For a sensor with linear transformation calibration is a matter of measuring the sensor voltage (Input) at two known conditions (Output) and solving the line equation for those two points in form of Output, $Y = \text{Slope} * \text{Input}, X + \text{Offset}$. This resulting equation can be used to get output D for any input C. The software

has a tool to automatically create the line equation based on the two given points A and B. The line equation is entered into a text field called transformation expression, that can be any math expression and will get solved for each measured point.

The 'raw value', the input is stored in the raw data tab, and the transformed value is also saved, in the 'transformed data' tab, as well as plotted on the graph if the channel is marked to be plotted on the graph.

3.1 Example - Linear voltage oxygen sensor

We decide to express the oxygen content of measured gas in ppm (parts per million). We flush the sensor with bottle pure N2 and measure 0.000 007 V. We assume small oxygen leakage into our gas stream through tube joints etc., and therefore we decide our low point is A(x,y) is A(0.000 007, 30). Next we flush our sensor with bottled air and measure 0.004 242 V, our bottled air has 20.9% oxygen, so our point B is B(0.004 242, 209 000). We type these into the tool 'Create linear transformation expression', and click 'Apply and Save Channel' and the channel is ready to measure.

| X | Y |
|-------------|-------------------|
| 0.123456789 | 6091482.75261629 |
| -1 | -49943880.9309424 |
| 0 | -315.404958677686 |
| 15 | 740153167.475797 |
| 3.14159 | 155016936.613932 |
| 0.001 | 49028.160866706 |

4 Transformation expression

The given expression is pre-treated by replacing all occurrences of INPUTX by the currently measured value for the channel in question, and then solved to obtain the output value. For no transformation, leave the field value to the default value of INPUTX.

4.1 Example 1 - Temperature with K-type thermocouple

Let us assume we are measuring K-type thermocouple where the cold junction is held at 25°C.

Temperature in Celsius, $T = \text{TCK}(\text{INPUTX}, 25)$ where TCK is built-in function for thermocouple C. Using the following as the expression

$\text{TCK}(\text{INPUTX}, 25)$

will evaluate in the expression parser, giving the temperature at the tip of the K-type thermocouple, when cold junction compensation was 25°C

4.2 Example 1 - Vapor pressure of water

The Tetens equation $P = 0.61121 \cdot \exp(17.27 \cdot T / (T + 237.3))$ where T is temperature in Celsius and P is water vapor pressure in kPa. If we replace the T from the above example, we get:

$0.61121 \cdot \exp(17.27 \cdot \text{TCK}(\text{INPUTX}, 25) / (\text{TCK}(\text{INPUTX}, 25) + 237.3))$

which will evaluate in the expression parser, giving the water vapor pressure of the temperature that was measured with the K-type thermocouple, when cold junction compensation was 25°C

4.3 Supported mark-up

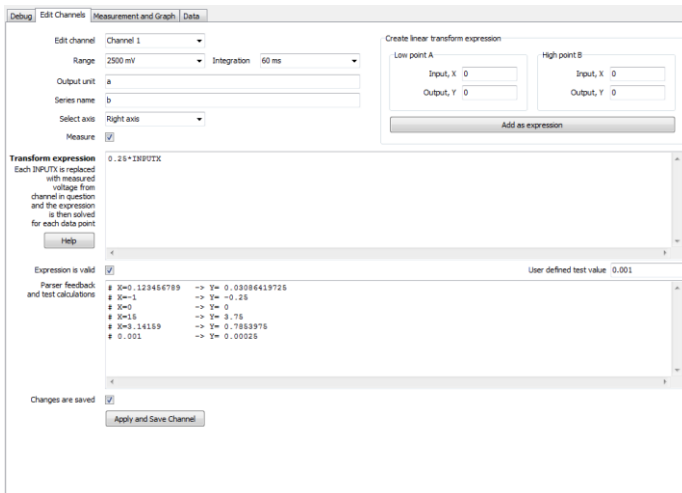
| Symbol | Full name | Notes |
|-------------|---------------------------|---|
| + | Addition | Returns the sum of left side added to right side. |
| - | Subtraction | Returns the subtraction of right side from the left side. |
| * | Multiplication | Returns the multiplication of left side and right side. |
| / | Division | Returns the left side divided by the right side. |
| ^ | Exponentiation | Returns the left side to the power of the right side. |
| % | Integer division | Returns the integer part of left side divided by right side. |
| () [] {} | Brackets | Any level of nesting brackets is supported. Expression inside brackets gets evaluated first. |
| < > = <= => | Comparison | Returns 0 if comparison is false and 1 if comparison is true. $-10 > 20 = 0$ or $-10 < 20 = 1$. |
| & | Comparison and | Returns 1 if left side $<>0$ and right side $<> 0$ |
| | Comparison or | Returns 1 if left side $<>0$ or right side $<> 0$ |
| ABS(A) | Absolute value | Return the absolute value of A. |
| ATAN(A) | Arctangent | Returns Arctangent of A. |
| CEIL(A) | Ceiling | Returns the next highest integer value. |
| COS(A) | Cosine | Returns cosine of angle A. |
| COSH(A) | Cosine hyperbolic | Returns hyperbolic cosine of angle A. |
| COTAN(A) | Cotangent | Returns Cotangent of A. |
| EXP(A) | Exponent | Returns e to the power of A. |
| FLOOR(A) | Floor | Returns the next lowest integer value. |
| IF(A,B,C) | If then else | Returns B if A is $<> 0$ (A=true), returns C if A = 0 (A=false). |
| INTPOW(A,B) | Integer power | Raises A to an integral power of B. $INTPOW(2, 3) = 8$ or $INTPOW(2, 3.4) = 8$ |
| LN(A) | Logarithm natural | Returns natural logarithm of A. |
| LOG(A) | Logarithm 10 | Returns 10 based logarithm of A. |
| LOGN(A,B) | Log base | Returns the log base A of B. Example: $LOGN(10, 100) = 2$ or $LOGN(2,8) = 3$ |
| MAX(A,B) | Maximum | Returns the higher of A and B |
| MIN(A,B) | Minimum | Returns the lower of A and B |
| MOD(A,B) | Modulus | Returns modulus of $trunc(A) / trunc(B)$. $MOD(5,3) = 2$ or $MOD(5.5,3) = 2$ |
| POW(A, B) | Power | Raises A to the power of B. Effectively same as A^B . |
| RANDOM(A) | Random float | Generates a random floating point number such that $0 \leq Result < A$. |
| RND(A) | Random integer | Generates a random integer number such that $0 \leq Result < A$. |
| SIGN(A) | Sign | Returns -1 if $A < 0$; +1 if $A > 0$, 0 if $A = 0$. |
| SIN(A) | Sine | Returns sine of angle A. |
| SINH(A) | Sine hyperbolic | Returns hyperbolic sine of angle A. |
| SQR(A) | Square | Returns A^2 |
| SQRT(A) | Square root | Returns square root of A. |
| TAN(A) | Tangent | Returns Tangent of A. |
| TRUNC(A) | Truncate | Discards the fractional part of a number. |
| TCK(V, T) | Thermocouple K type | Returns a temperature in °C. Parameter V is voltage (from K-type thermocouple) with valid range of -6 mV to 54 mV. Parameter T is the ambient temperature in °C with valid range of -100°C to 100°C. Function returns -1000°C if parameters are out of valid range. For the parameter T a known room temperature can be entered or for more accurate results a true temperature can be calculated using TTT function or by other means. |
| TCS(V, T) | Thermocouple S type | Returns a temperature in °C. Parameter V is voltage (from S-type thermocouple) with valid range of -0.235 mV to 18.693 mV. Parameter T is the ambient temperature in °C with valid range of -100°C to 100°C. Function returns -1000°C if parameters are out of valid range. For the parameter T a known room temperature can be entered or for more accurate results a true temperature can be calculated using TT2 function or by other means. |
| TPP(T) | Thermopower Platinum | Returns the absolute thermoelectric power of Platinum in Volts/degree Celsius. Parameter T is temperature, given in degrees Celsius. Function fitted from data presented in Thermoelectricity in metals and alloys - R.D.Barnard, Thermoelectric power of metals – Blatt, with maximum error (from the data in the book) of 0.08 microvolts above 250°C region. |
| TT2(R) | Thermistor to Temperature | TT2(R) Thermistor To Temperature returns a temperature in °C. Parameter R is resistance measured from thermistor type EPCOS serie:B57861S type:B57861S0103F45. The function is valid between 2986Ω-32650Ω, a temperature range of 0 to 55°C accordingly. |

5 Tabs explained

5.1 Debug tab

This tab is shown on program start-up, when the software seeks to find the measurement instrument. Feedback is shown in the text box.

5.2 Edit channels tab



Edit channel is a dropdown with list of the 8 channels to select from. Once chosen, all the details on the page refer to that specific channel, and apply once the 'Apply and Save Channel' button has been clicked.

Selecting **range** closest to the foreseeable voltage will ensure best resolution and accuracy. In case voltage exceeds the selected range the exceeding part is not displayed. The instrument is protected against voltages up to 30 VDC. Range is given as \pm the selected value.

Integration is a noise reduction filter, the longer time the channel measures the less noise the resulting value will have.

Output unit is note to the user about the transformation expression, to tell what the resulting unit is without having to read the expression.

Series name is note to the user of what the selected channel is measuring.

For any series the user can **select axis** between the left axis and the right axis.

Measure checkbox toggles this channel to measure or to not measure. This also determines the initial visibility of a channel when the program is first started.

Transform expression and **Create linear transform expression** are explained elsewhere in this manual.

Expression is valid is not interactive checkbox, it just verifies if the entered expression can be parsed and executed.

User defined test value allows user to feed the **Parser feedback and test calculations** textbox one value (last line) to pass through the expression and the resulting output.

Changes are saved is not interactive checkbox but indicates if anything (of significance) on the page was changed, but not applied and saved by clicking **Apply and Save Changes**.

5.3 Measurement and Graph tab

Measurement, Active defines if the software will try to measure the active channels.

Interval, s textbox sets the internal frequency of the software to given seconds. It tries to measure each x seconds. Each channel takes some time to measure depending on the integration setting, so to measure all channels with 660 ms integration may take longer than the interval defined here, in such case, new measurements are taken as soon as possible.

Save button will pop up save dialog to select a location and filename for the measurement. The saved file will have header first with each channel number, series name, series unit, series expression. This data is followed by ; separated list of values same as is seen in the Data tab. Default filename 'experiment yyyy-mm-dd hh-mm.csv' if no other filename was defined where the end part is year-month-day hour-minute.

The program will autosave every 5 minutes if the program is marked active. The assigned filename with _autosave suffix is used.

Visibility toggles visibility of each channel/series. The channel will still measure if it was designated to do so, even if it is not visible.

Edit graph opens extensive editor to change any property of the graph.

Clear data removes all data from all series on the graph. Already measured data remains the the data tab.

Left click and drag right zooms into the graph, dragging left same way unzooms the graph.

The page also has a vertical divider that can be used to resize the properties side and the graph itself, by dragging with mouse.

6 Miscellaneous

One computer can run more than one copy of the DALONO software. In such case each copy of the software will need to be installed to a separate location on the disk, and in the corresponding .ini file, the setting [Software] Instrument handle=1 must be changed so that each copy of the software has integer number one larger than the previous, for example 1, 2, 3.