Automotive
Energy & Power Analysis
Field Service
Environmental
Research & Development

DEWETRON

Training Manual

Version 1.2
January 2011

1 Math

1.1 Functions

Info: Most of the functions below are described with discrete numbers. Instead of discrete numbers also channels could be used. This was chosen to get a better and easier understanding of the functions.

1.1.1 SQR -Square

Sqr: sqr(4)= 16 The square of a number is the number multiplied with itself. No matter if the number is pos. or neg. the result will be always positive.

Sqr(-3)= 9.

1.1.2 SQRT - Square root

Sqrt: sqrt(16) = 4 The square root of number B is the number A^2. So it is the inverse function of SQR.
Sqrt(9)= 3.

Info: SQRT of a negative number sqrt(-9) will deliver always 0 instead of a complex number in DEWESoft.

1.1.3 ABS – Absolute value

ABS: ABS(45.34) = 45.34 Calculates the absolute value of number or a channel.
ABS(-33.12)= 33.12
ABS(0) = 0

1.1.4 SGN – Sign

Sgn: sgn(-8.124) = -1 Extracts the sign of a channel or a number.
sgn(19.345) = 1
sgn(0) = 0 Sign function delivers 0 if input channel or number is 0.

1.1.5 TRUNC – Truncate function

TRUNC: converts a number into integer,
Trunc(1452.457) = 1452
Trunc(-1452.457) = -1452

So every number or channel which is converted with the Trunc function will lose the part after the comma. It will become an integer value.

It is not rounded up or down, so both:

Trunc(86.248) = 86 and also
Trunc(86.848) will give 86.

For an example please refer to points 2.2 in the examples section.

1.1.6 ROUND – Round function

Round: rounds a number or channel depending on the digits after the comma to an integer value.
3.xxx if xxx is bigger or equal 0.5 it will be rounded up to the next integer value.

round(14.43) = 14
round(14.501) = 15
round(-14.492) = -14
Round(-14.51) = -15

Hint: if you want to round 14.43 to 10, or 136.3724 to 140 simply divide the value or channel by 10 round it and multiply the result by 10.

Round(13.63724)*10 = 140

If you want to round 136.3724 to 136.4, multiply the value or channel by 10, round it, then divide it by 10.

Round(1363.724)/10 = 136.4

1.1.7 RND – Random

RND: creates random numbers with the selected sampling rate between 0 and 1.
So if a sample rate of 1000Hz is selected, 1000 values per second are created.

1.1.8 LOG2 – Logarithm base 2

Log2: calculates the logarithm (base 2) of a number or an input channel.

Log2(8) = 3

Log2(a) = b the logarithm extracts b from an equation $2^b = a$.

1.1.9 LOG10 – Logarithm base 10

Log10: calculates the logarithm (base 10) of a number or an input channel.

Log10(100) = 2

Log2(a) = b the logarithm extracts b from an equation $10^b = a$.

1.1.10 LN – Natural logarithm base e

LN: calculates the natural logarithm (base e=2.71828…) of a number or an input channel.

LN(100) = 2

LN(a) = b the logarithm extracts b from an equation $2.71828..^b = a$.

1.1.11 EXP – Expotential function of e

EXP: calculates the exponential function of e from a number or an input channel.

EXP(1) = 2.71828…

EXP(b) = a the logarithm extracts b from an equation $2.71828..^b = a$.

1.1.12 IF / NAN – If, not a number function

IF(condition,result1,result2)

Outputs either the result1 or result2 depending on condition.

Condition: Expects a true false input. Example: (Channel >= 12)

So if channel or number is equal or greater than 12, the condition becomes true and result1 will become true. Otherwise condition is false and result2 will become true.

Example: IF('ID'>=1,'velocity','displacement')

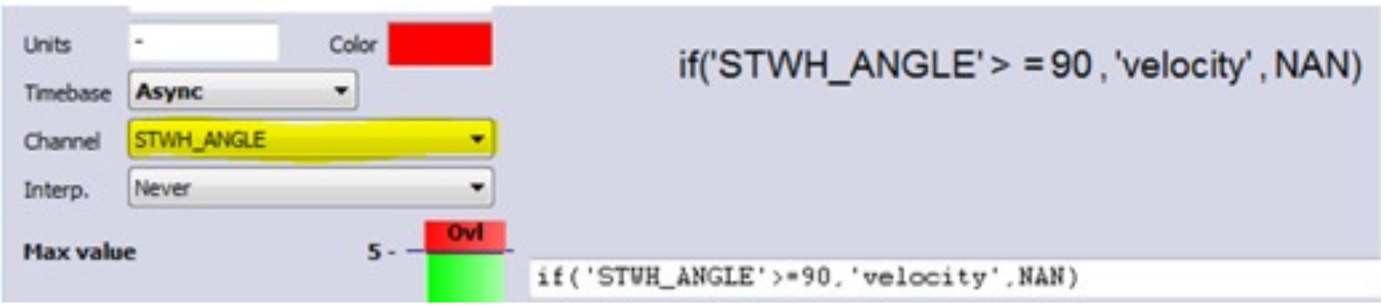
'ID', 'velocity', and 'displacement' are analog input channels.
If the ID channel is equal to or greater than 1, the statement is true and the velocity channel will be chosen as output, otherwise displacement. So either velocity or displacement are used for output depending on the condition, both at the same time are not possible.

NAN(not a number)

if('STWH_ANGLE'>=90,'velocity',NAN)

'STWH_ANGLE' = CAN Channel (asynchronous)
'Velocity' = analog input channel (synchronous)

The channel will output NAN only if STWH_Angle is smaller than 90 deg.
And the timebase of the math channel will be forced to asynchronous output.
Otherwise because a synchronous channel is used in the formula (velocity) the formula will output a synchronous channel and NAN will become zero.



The picture above is showing the time base setting of the math channel.
The asynchronous channel has to be used in the formula otherwise it is not possible to select it in the channel selector. The interpolation has to be set to never.

1.1.13 MAX – Maximum function of more channels

MAX: **MAX(Channel1,Channel2)** checks both channels and outputs the maximum value of one of the channels.

Example: **max('pressure1','pressure2')** pressure1 and pressure2 are two analog input channels.

pressure1	3	4	6	8
pressure2	2	5	4	7
output	3	5	6	8

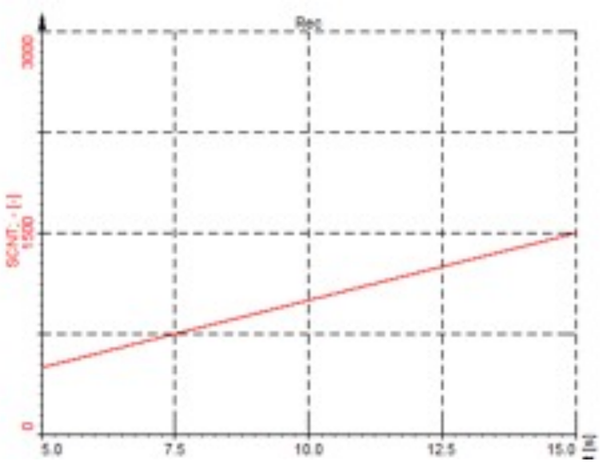
The higher value if both channels will be output.

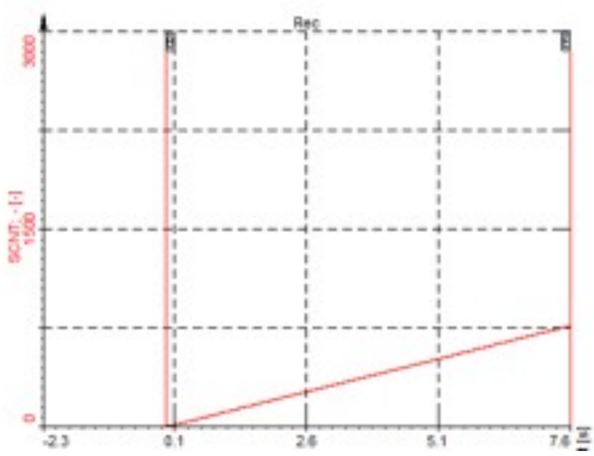
INFO: Also multiple max could be used in one formula.

Max(max('pressure1','pressure2'),'pressure3')

1.1.14 SCNT – Sample counter

SCNT: delivers the samples acquired from start of the measurement.
The counter will be reset at the start of storing.





The left picture is showing the result, while at the right picture the reset after start storing could be seen.
For and example refer to point 2.1

1.1.15 MOD – Modulus function

MOD: delivers the remainder of a mathematical division operation

<u>420 MOD 720</u> = 420 — 420 could not be divided by 720, and we will get the the remainder of the division which is 420.

<u>740 MOD 720</u> = 20 — 740 could be divided by 720 one time, with a remainder of 20.

For an example please refer to points 2.2 and 2.3 in the examples section.

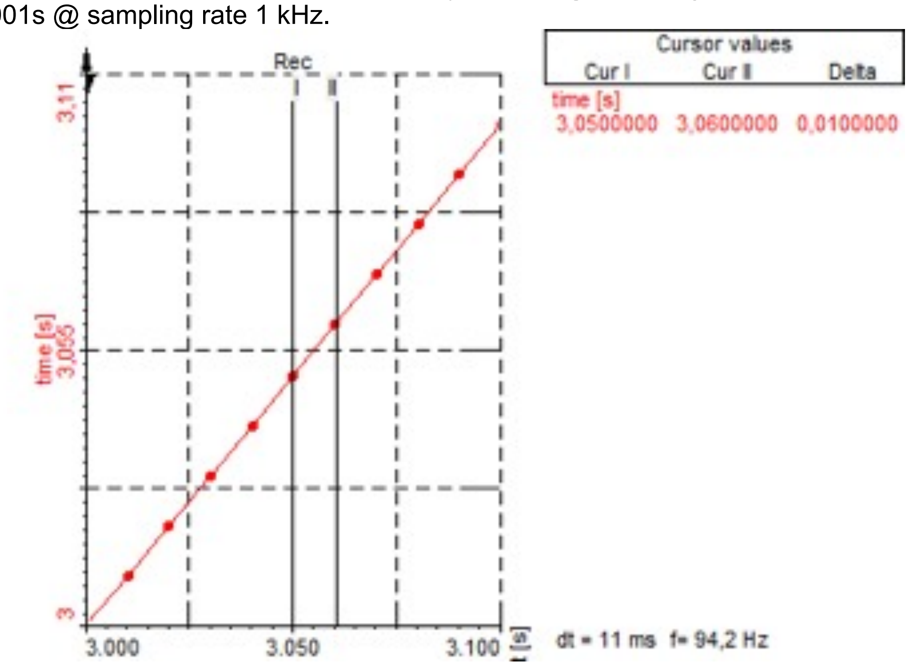
1.1.16 Time – Time function

TIME: is providing the elapsed time of the measurement in seconds.
This is a function similar to SCNT. The only difference is, instead of samples
It outputs the time in seconds, regardless which sampling rate is used.

Note: the TIME function is reset at the start of storing.

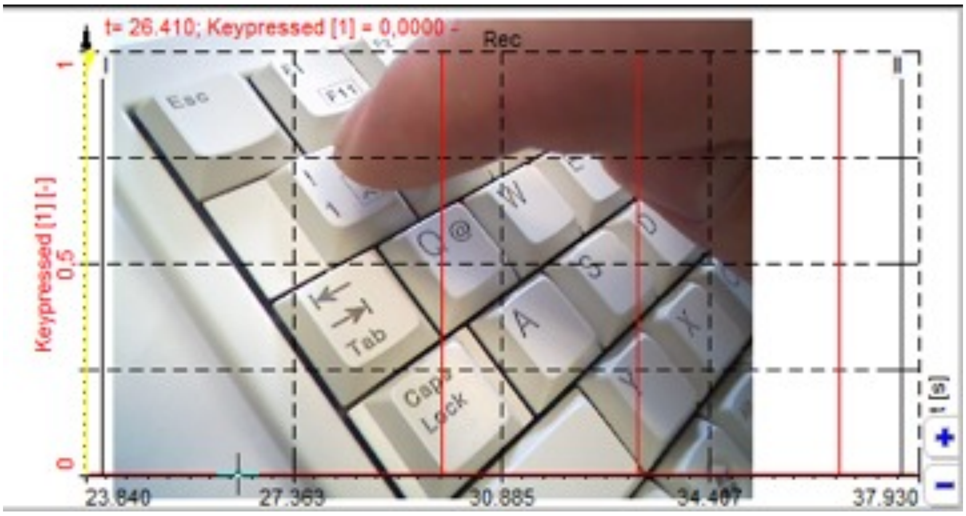
The resolution is linked with the sampling rate.

The screen shot below shows the time function in a recorder,
We can see that the resolution is 0.01s and also dt=10ms which results in
a sampling rate of 100 Hz.
So the time will count up 0.01s after every sample @ sampling rate of 100 Hz or
0.001s @ sampling rate 1 kHz.



For an example please refer to points 2.3 in the examples section.

1.1.17 Keypressed – Keypressed function



Almost any key can be used in the keypressed function.
The value in the brackets is reflecting the virtual key code in decimal.
Below you will find a list of the most popular keys. You have to convert them from Hex to decimal. So [Key 1] → 31hex → 49 dec. . The decimal value has to be entered Into the keypressed function. keypressed(49).

For an example please refer to points 2.4 in the examples section.

1.1.17.1 Major Key codes

0x41 ('A')	A
0x42 ('B')	B
0x43 ('C')	C
0x44 ('D')	D
0x45 ('E')	E
0x46 ('F')	F
0x47 ('G')	G
0x48 ('H')	H
0x49 ('I')	I
0x4A ('J')	J
0x4B ('K')	K
0x4C ('L')	L
0x4D ('M')	M
0x4E ('N')	N
0x4F ('O')	O
0x50 ('P')	P
0x51 ('Q')	Q
0x52 ('R')	R
0x53 ('S')	S
0x54 ('T')	T
0x55 ('U')	U
0x56 ('V')	V
0x57 ('W')	W
0x58 ('X')	X
0x59 ('Y')	Y
0x5A ('Z')	Z

0x60	Numpad 0
0x61	Numpad 1
0x62	Numpad 2
0x63	Numpad 3
0x64	Numpad 4
0x65	Numpad 5
0x66	Numpad 6
0x67	Numpad 7
0x68	Numpad 8
0x69	Numpad 9

0x2E	Delete
0x28	Arrow Down
0x23	End

0x70	F1
0x79	F10
0x7A	F11

0x72	F3
0x73	F4
0x74	F5
0x75	F6
0x76	F7
0x77	F8
0x78	F9

INFO: Do not use F2 because it is used as shortcut to enter Setup.

If you need any other key which is not present in the upper list,

search for **"key codes"** or "**virtual key codes**" in the internet.

Display screen selection

<F2> Switch to Setup screen
<F3> or <SHIFT> + <F1> Switch to Overview screen
<SHIFT> + <F2> Switch to Scope screen
<SHIFT> + <F3> Switch to Recorder screen
<SHIFT> + <F4> Switch to Vertical recorder screen
<SHIFT> + <F5> Switch to FFT screen
<SHIFT> + <F6> Switch to Power screen
<SHIFT> + <F10> Switch to Video screen
<SHIFT> + <F11> Switch to GPS screen

1.1.18 HOLD – Hold function

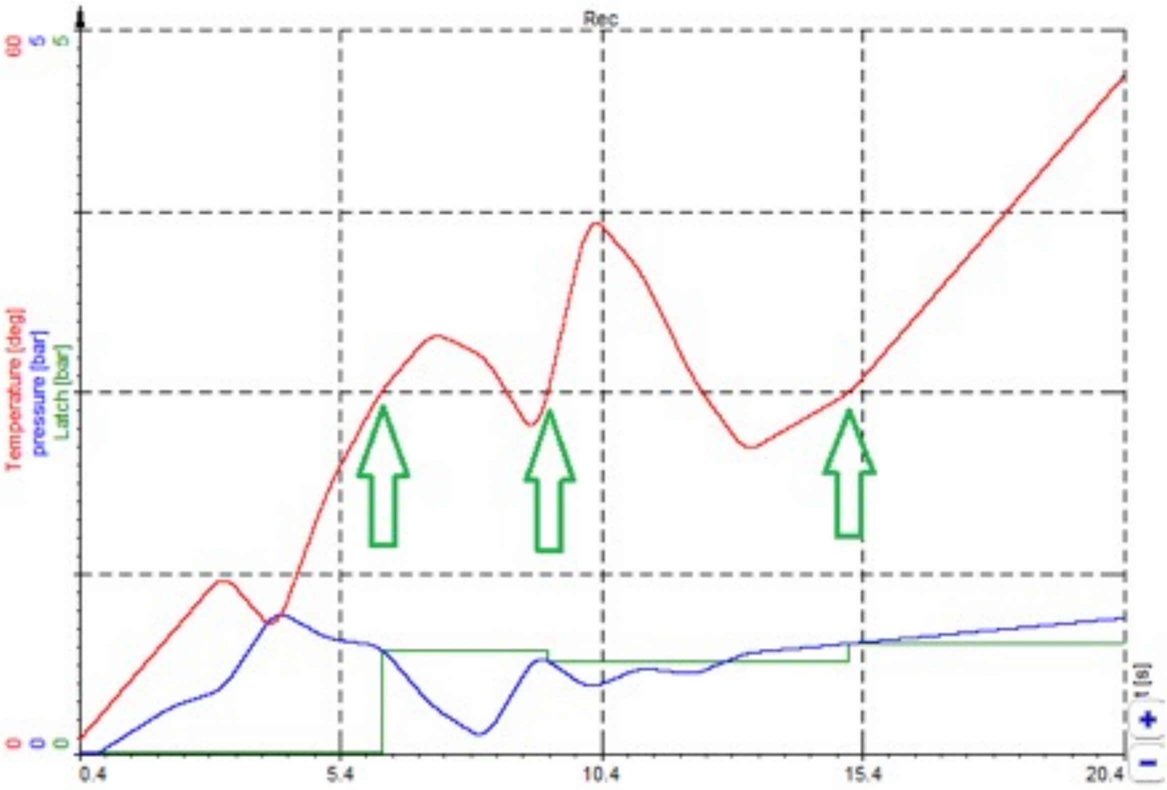
Hold(value,latchcondition,[rearmcondition])

The hold function is used to latch or hold a single value if a condition is met.

Example1: Hold('pressure', 'Temperature'> 30)

Hold(value, latchcondition)

In the example above the function will hold the actual pressure if the temperature is higher or equal to 45 degrees.



The picture above is showing the function in a recorder. All the time when the temperature exceeds 30 deg (green arrows), the actual pressure channel is latched.

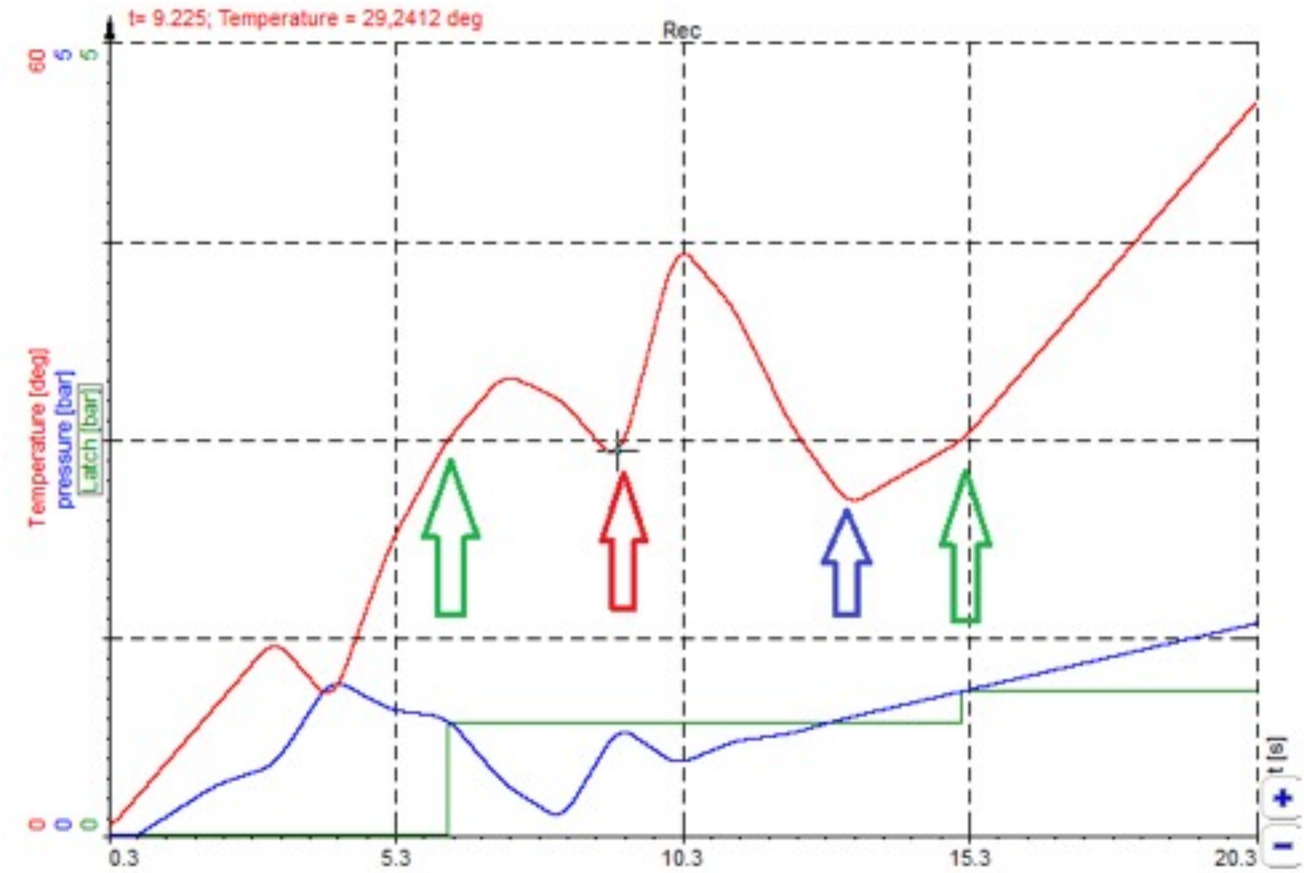
Example2: Hold('pressure', 'Temperature'> 30, 'Temperature'< 28)

Hold(value, latchcondition,[rearmcondition]) optional.

-

The function can be extended with a rearm condition. After a latch condition occurs, the ream condition has to be met first before a new latch condition and therefore a new latch can occur.

This is used to filer the latch condition a little bit. Imagine the latch condition channel has noise on it , or is fluctuating around the level (30 deg +-0.2 deg) which would cause a unintentional LATCH.



The picture above illustrates this function:

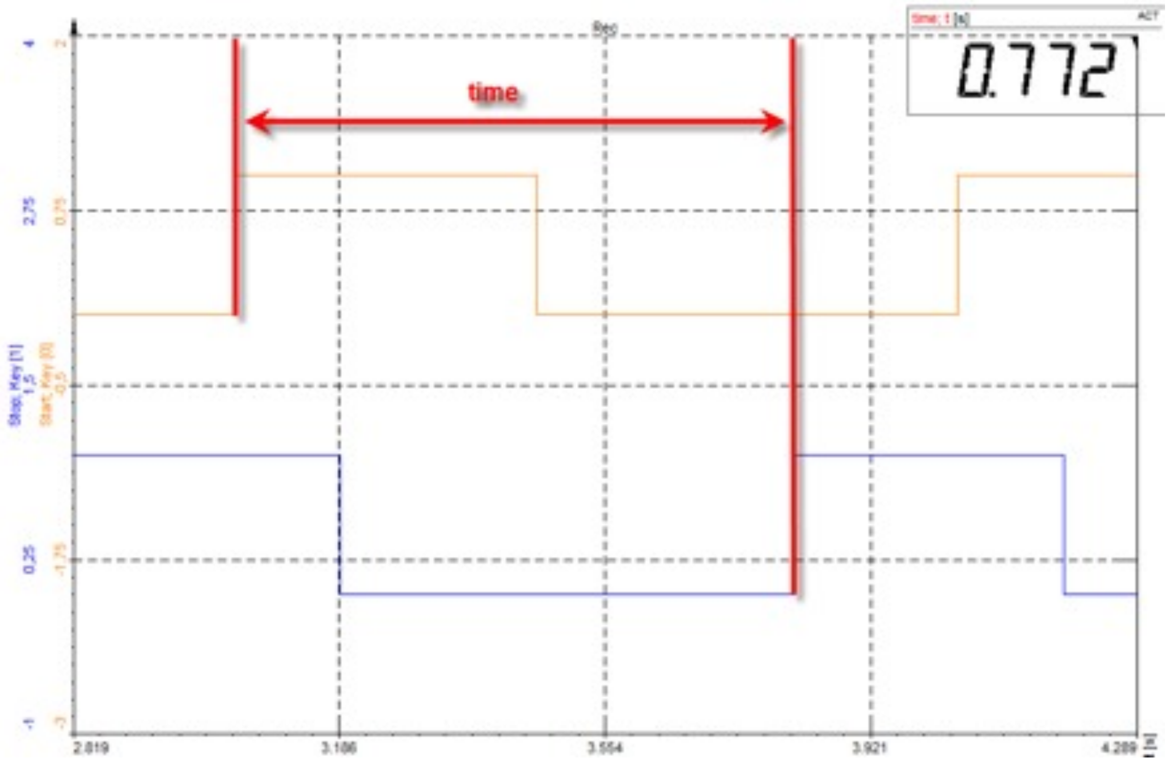
1.) Temperature rises above 30 deg → LATCH 1 green arrow
2.) Temperature goes below 30 deg and back above 30 deg → no LATCH because temperature did no go under 28 deg therefore the LATCH did not occur. Red arrow.
3.) Temperature goes below 28 deg → rearm condition is complied
4.) Temperature exceeds again 30 deg → LATCH is performed again

For and example refer to point 2.5

1.1.19 STOPWATCH – Stopwatch function

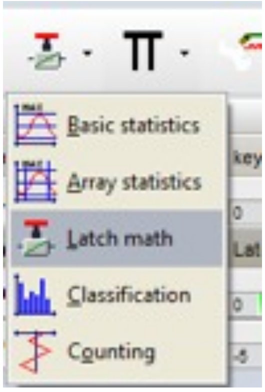
STOPWATCH: **STOPWATCH(Condition1,Condition2)** measures the time in [s] between condition1 and condition2.
Condition is a logic value that jumps from 0 to 1 (edge sensitive)

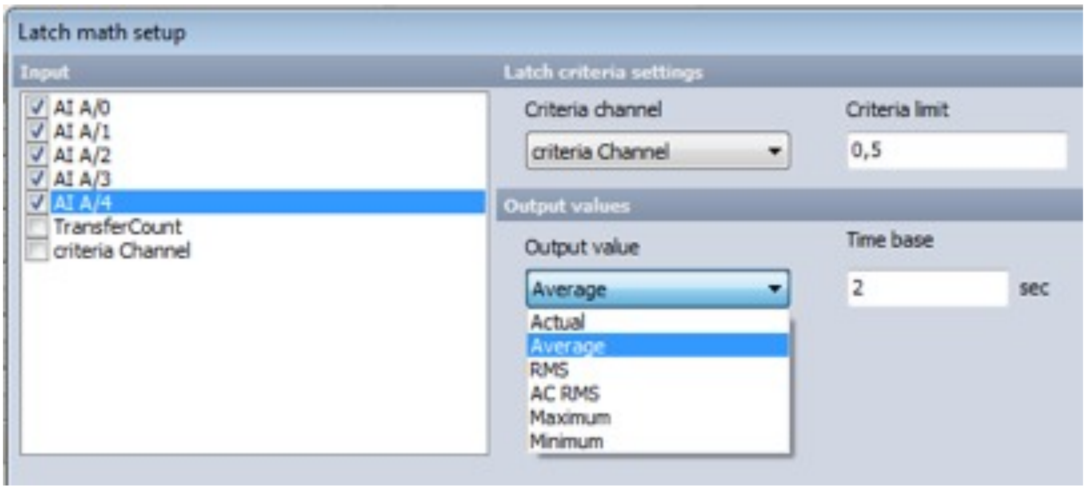
Example: **stopwatch('Start'>0.5,'Stop'>0.5)** Start and Stop could be analog or digital signal.
The logic comparison '>' turns the condition into a logic value anyway.



1.2 Statistic Functions

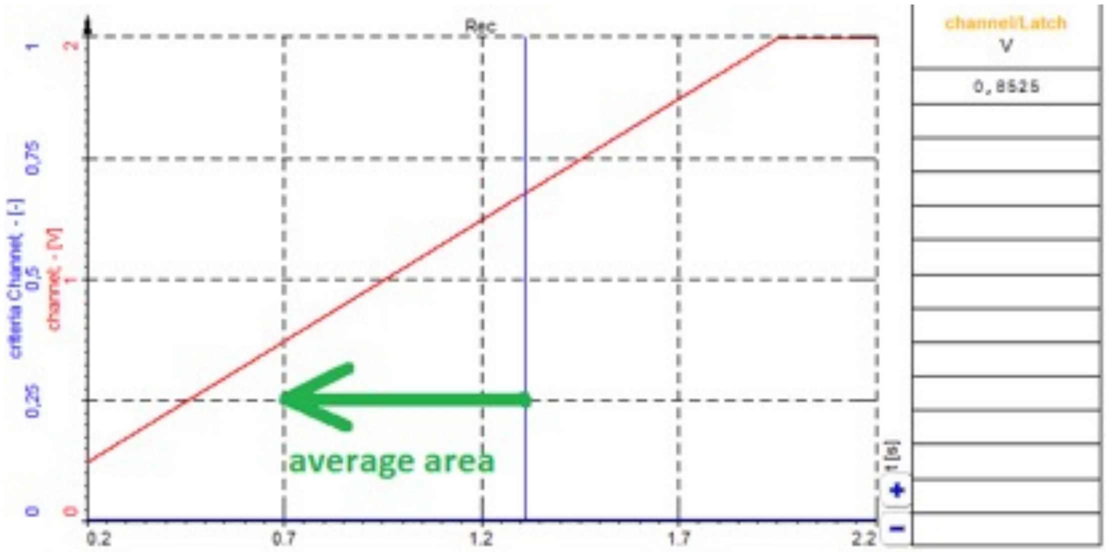
1.2.1 Latch – Latch Function





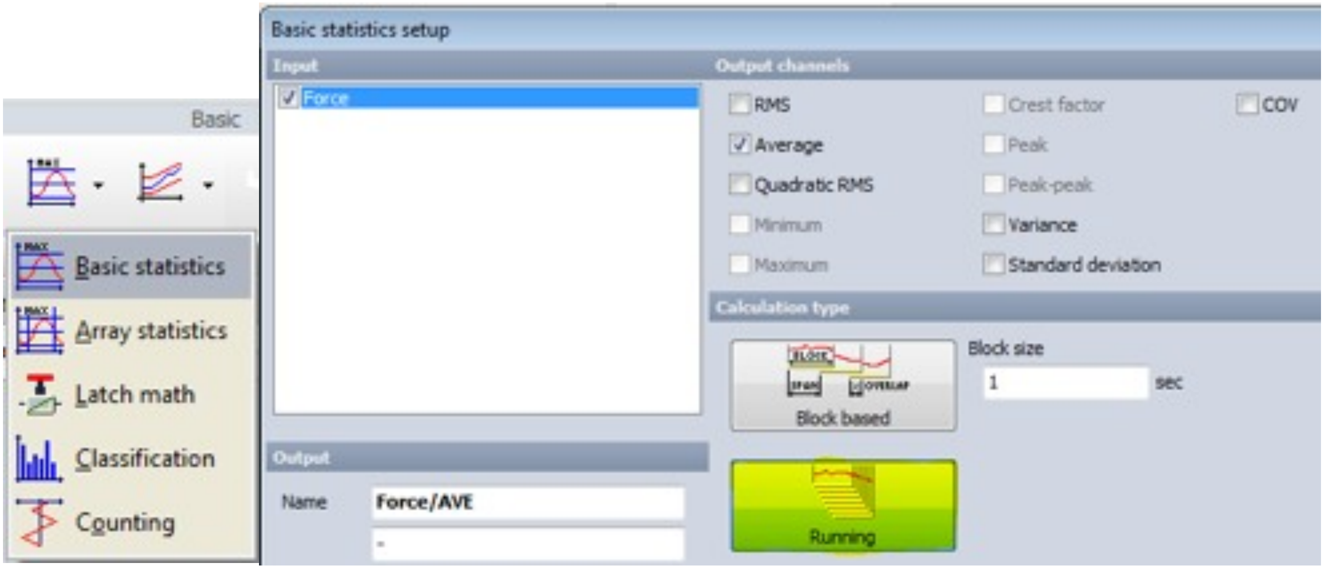
The latch function can hold a value if a trigger condition is met.
So if the criteria channel exceeds the set criteria limit the input channels will be latched.
Here the latch can be done in various modes.

Actual:	If criteria is met the ACTUAL value at that time is taken.
Average:	If criteria is met the AVERAGE value is taken. The calc. time can be set (Time Base) in sec.
RMS:	If criteria is met the RMS value is taken. The calc time can be set (Time Base) in sec.
AC RMS:	If criteria is met the AC-RMS value is taken. The calc. time can be set (Time Base) in sec. AC-RMS will calculate the RMS only on the AC part of the signal. DC components are not taken in account.
Maximum:	If criteria is met the Maximum value is taken. The average time can be set (Time Base) in sec.
Minimum:	If criteria is met the Minimum value is taken. The average time can be set (Time Base) in sec.



Info: The average is taken always the average time span before the event.

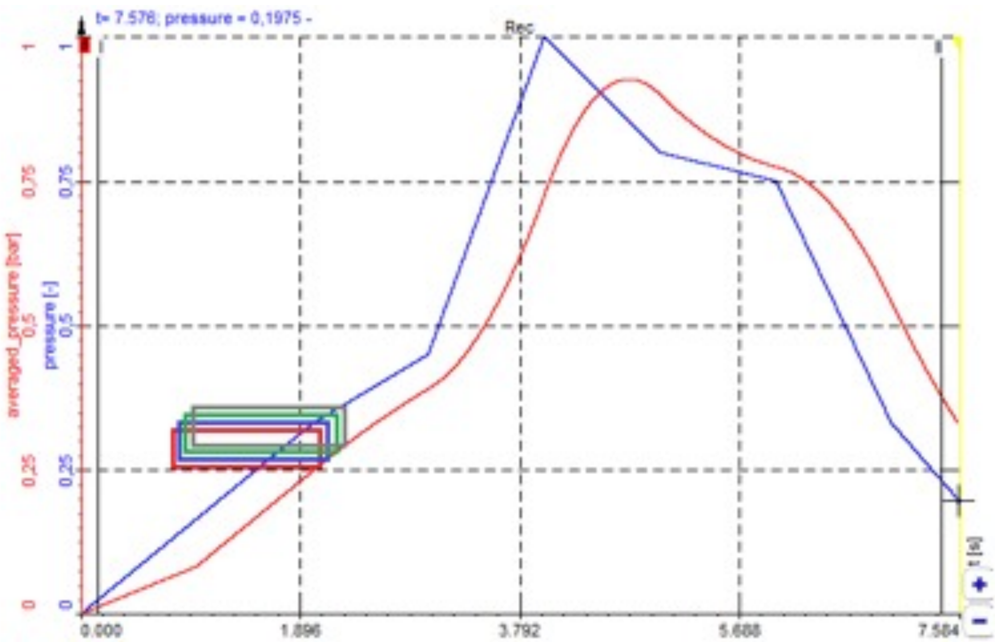
1.2.2 Average – Running average



The average function(s) is found in the Basic statistics function in the MATH section.
There are several modes how the average can be calculated. Here in this example the RUNNING mode over a block size of 1s is chosen.

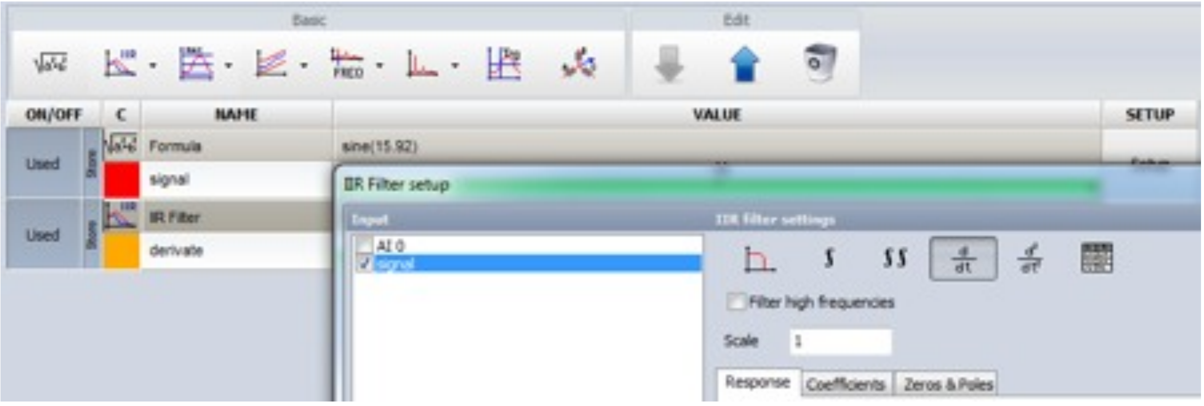
You can visualize the calculation like this:

A window with a length of 1s is moved over the channel to average. All the data in this window will be averaged and a single average value will be the result (indicated by the red, blue, green, and gray boxes in the picture below). After that, the window will be moved for one sample to the right and the average will be calculated again.
So the output will be a synchronous channel with the full sample rate.
The picture below will illustrate the function in more detail.



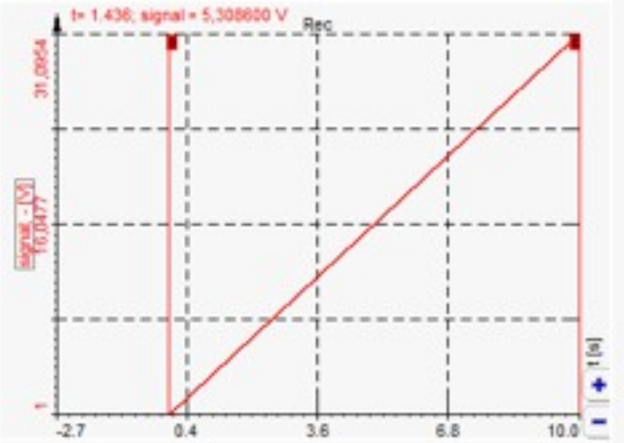
1.3 Filter Functions

1.3.1 Derivation and dY – derivation of a signal and dY



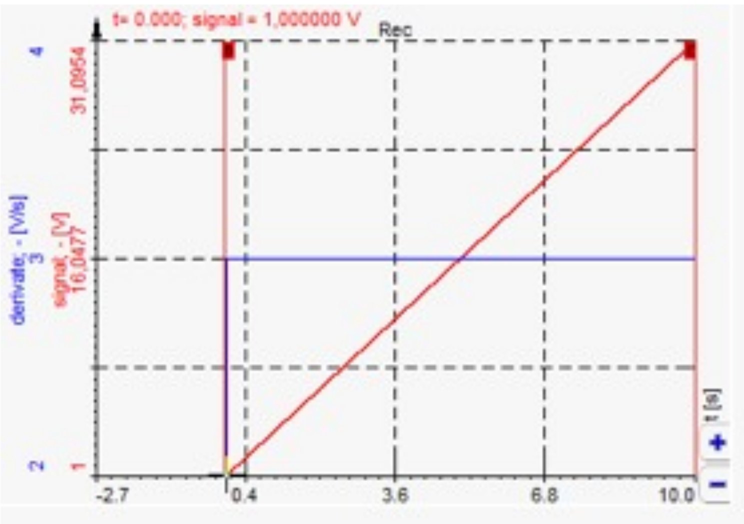
Derivation describes the <u>maximum voltage change</u> of an signal.

If we have a function:
s(t)= 3*t + 1 this function will create a signal shown in picture below.

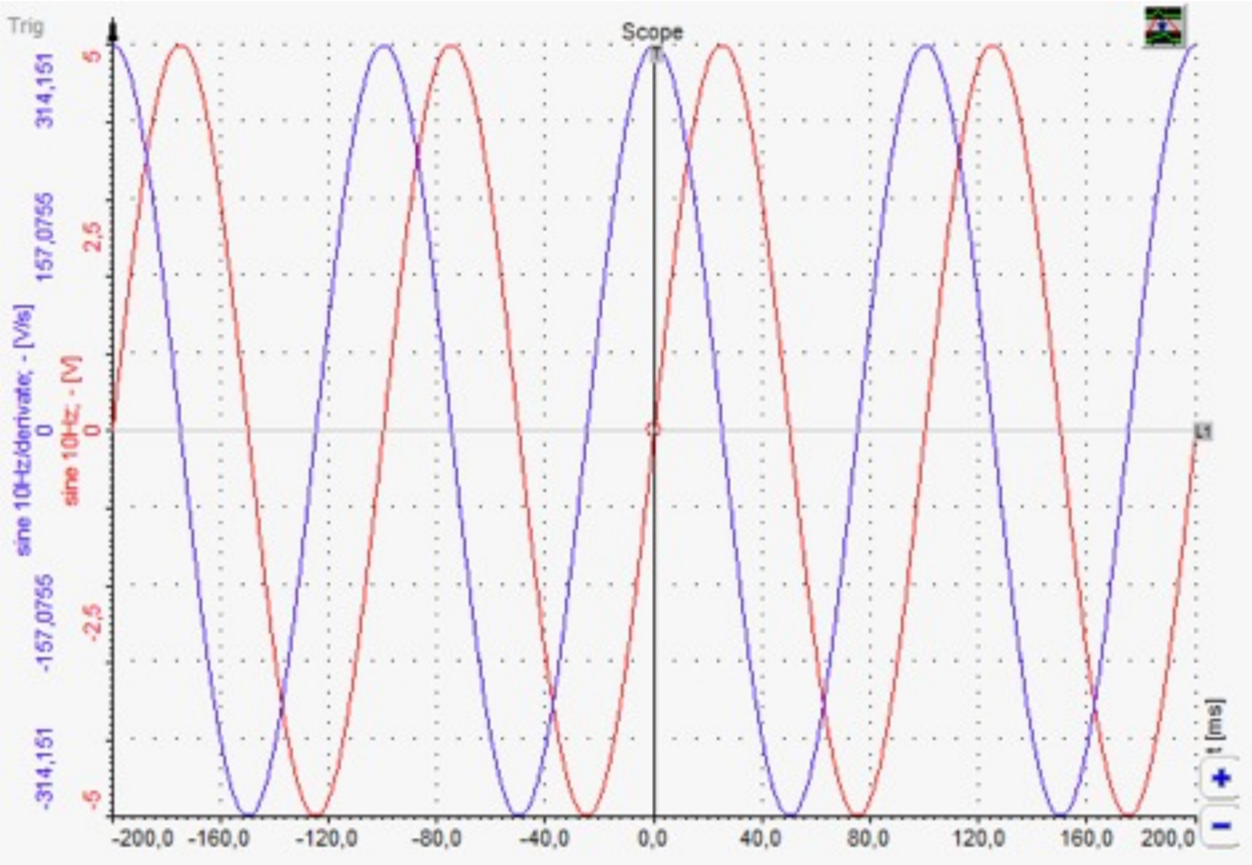


So the signal will start @ 1 and will count up 3 units after 1 second.

The derivation of this signal will be 3V/s .

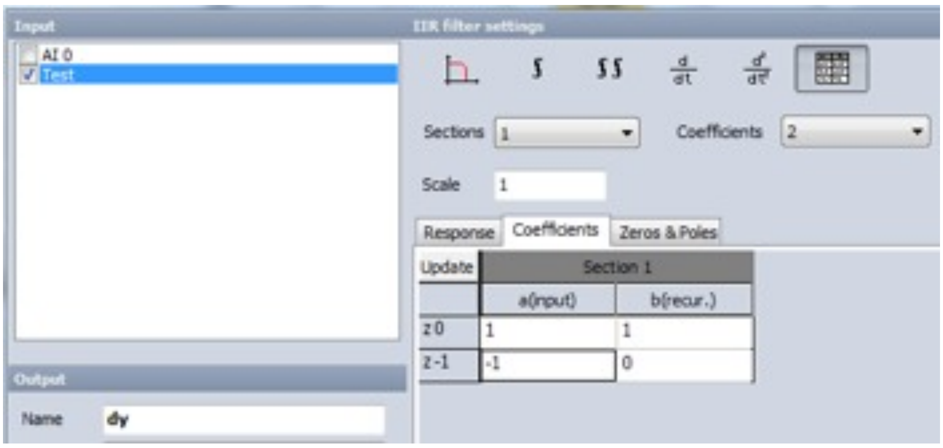


The next example shows the derivation of a sine signal.
A sine signal which is described as 5 * sine(10) will create a sine wave with 5 V amplitude and 10 Hz.
Derivation will yield: -5 * 2 * pi * 10 * cos(10) -- therefore a cosine with an amplitude of 314.15 V/s.



The filters can be also used to calculate the dY of a signal.

Example: a signal with voltage 3,5,7,8,12,15,…. will result in (5-3)=2,(7-5)=2,(8-7)=1,4,3 dY……

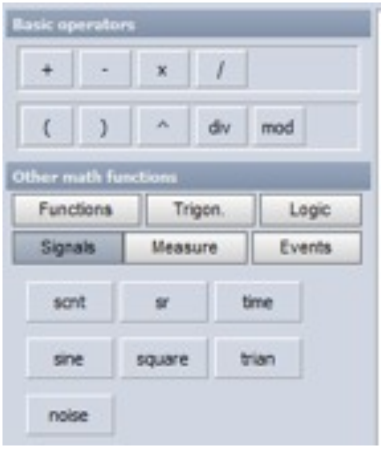


For this we have to manually enter coefficients into a filter, like shown in the upper picture. In comparison to Derivation here only dY is calculated where at derivation dY/dt is calculated. So in simply word, dY is divided with

Dt = sample rate.

2 Examples

2.1 Function: SCNT Sample Counter - Example: Create Angle Signal



SCNT: delivers the samples acquired from start of the measurement.
The counter will be reset at the start of storing.

MOD: delivers the remainder of a division
420 MOD 720 = 420
740 MOD 720 = 20

Example: If external clocking is used and the signal will be shown in an angle-based XY diagram, with the MOD function we can create an angle signal.

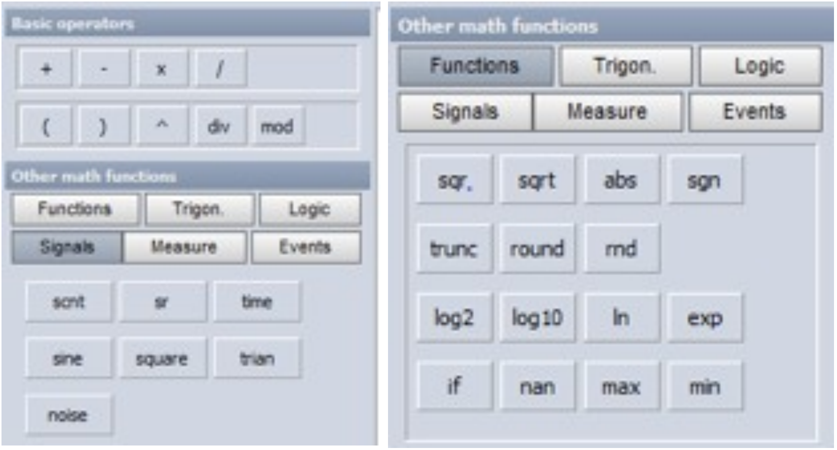
Let’s assume we are using an encoder with 720 pulses/rev.

MATH: SCNT MOD 720 will deliver a sawtooth which runs from 0 to 720.
To get the angle we have to multiply it by 0.5 deg, so at the end we get this formula:
SCNT MOD 720 *0.5 this channel can be used in a XY diagram to show the result angle based.

The only disadvantage will be, if we get wrong pulses from the encoder (i.e., noise or spikes), the angle signal will shift.

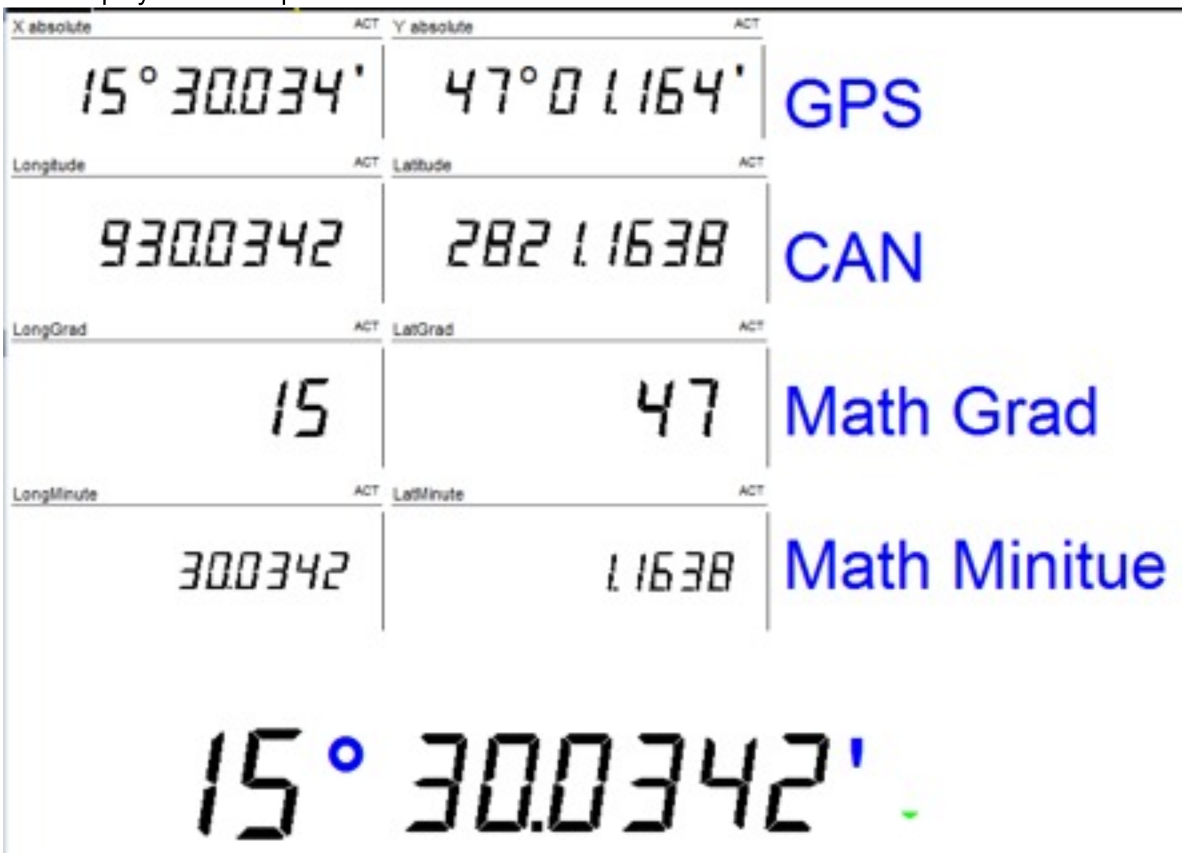
If a TRG pulse is also available, CLK and TRG, this signals could be also routed to an Orion Counter, where this is eliminated. Because the TRG PULSE will reset the counter every revolution. So a hardware counter on an ORION A/D card will typically yield better results than software functions.

2.2 Function: MOD / TRUNC – Example: Separate CAN GPS Signal to DEG:MIN,xxx



<u>TRUNC:</u> converts a number into integer,
Trunc(1452.457) = 1452

Example: The VGPS Longitude and Latitude signals received over the CAN Bus should be separated into DEG:MIN,xxx.
The CAN data is received in MIN like it is shown below. With math functions the regular GPS display should be produced. Below the result is shown.



MATH: trunc('Latitude'/60) dividing by 60 will convert it from minutes to deg, where the trunc function will eliminate the digits after the comma. → result [°]

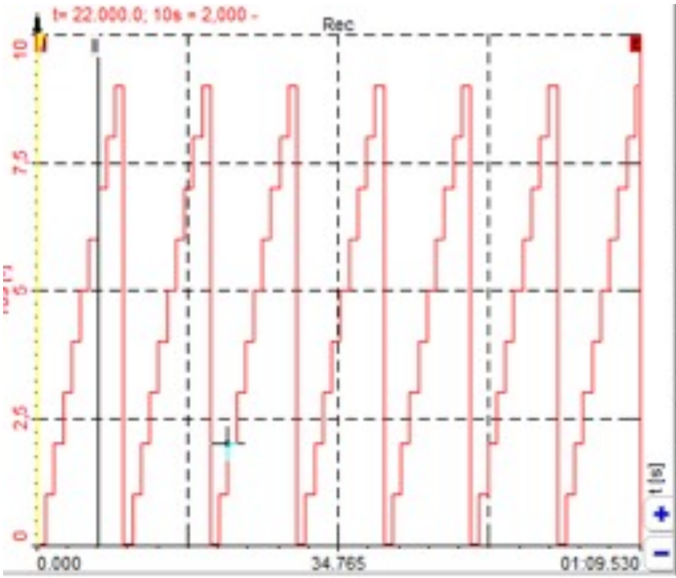
(('Latitude'*100000) mod 6000000) / 100000 will provide the minutes and the remainder.

Because the MOD function only delivers the remainder as an integer, we have to multiply the latitude with 100000 and use MOD with 60 *100 000 = 6000 000, then divide the result again with 100,000 to get the result.

2.3 Function: MOD / TIME – Example: Show actual value averaged every 10s in a list and export it to Excel.

TIME: is providing the elapsed time of the measurement.

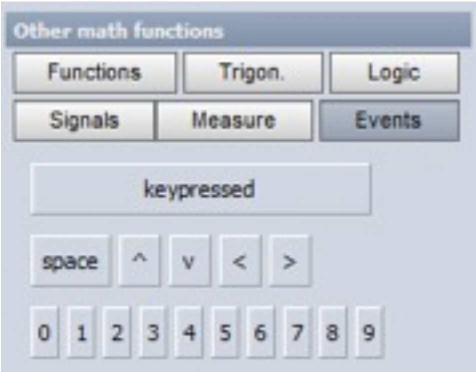
MATH: (Time MOD 60)
This will crate a saw tooth with a period time of 60s. Look to the picture below.
This is channel will be use as event channel in the LATCH math to average the actual channel and show it in a list. The averaged values could be exported to Excel or TXT asynchronously.
To export it asynchronously @ Export only the asynchrony channels have to be selected.

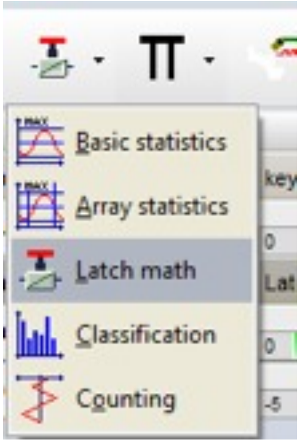


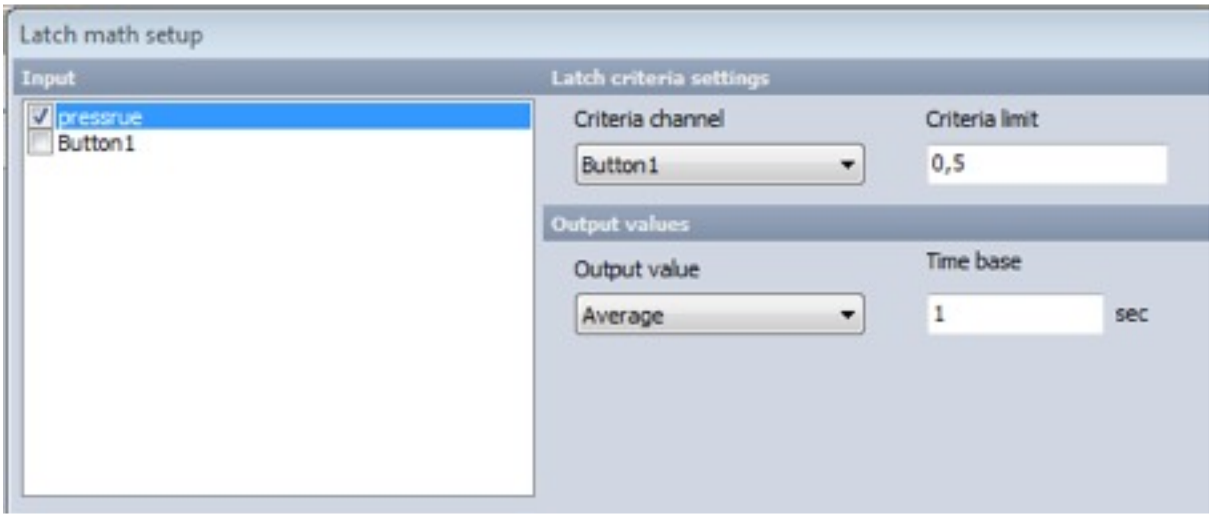
Time	pressrue/Latch bar
06.09.2010 18:00:22.537	0,9997
06.09.2010 18:00:12.537	0,9997
06.09.2010 18:00:02.537	0,9998
06.09.2010 17:59:52.537	0,9998
06.09.2010 17:59:42.537	0,9998
06.09.2010 17:59:32.537	0,9998
06.09.2010 17:59:22.537	0,9998

Exported	Index	Type	Acq. rate	Dimension	Name
No	0	AI 0	10000	Scalar	pressrue
No	1	Math 0 (Formula)	single	Scalar	Frm0/Formula 0
No	2	Math 1 (Formula)	10000	Scalar	Frm1/10s
Yes	3	Math 2 (Latch math)	0,1	Scalar	Latch0/Latch index
Yes	4	Math 2 (Latch math)	0,1	Scalar	Latch0/pressrue/Latch

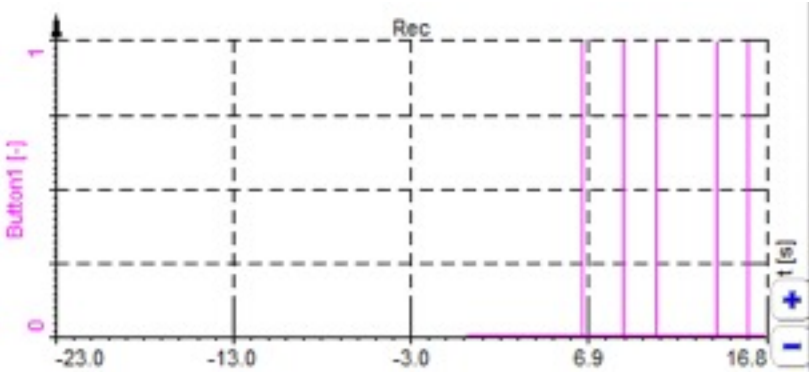
2.4 Function: KEYPRESSED / LATCH – Example: Latch Value into List







MATH: keypressed(49) produces signal from 0 to 1 with a duration of 1 sample if [1] Button is pressed. This could be used in the latch Math to latch the actual or average value of an other channel(s) in a
l list.



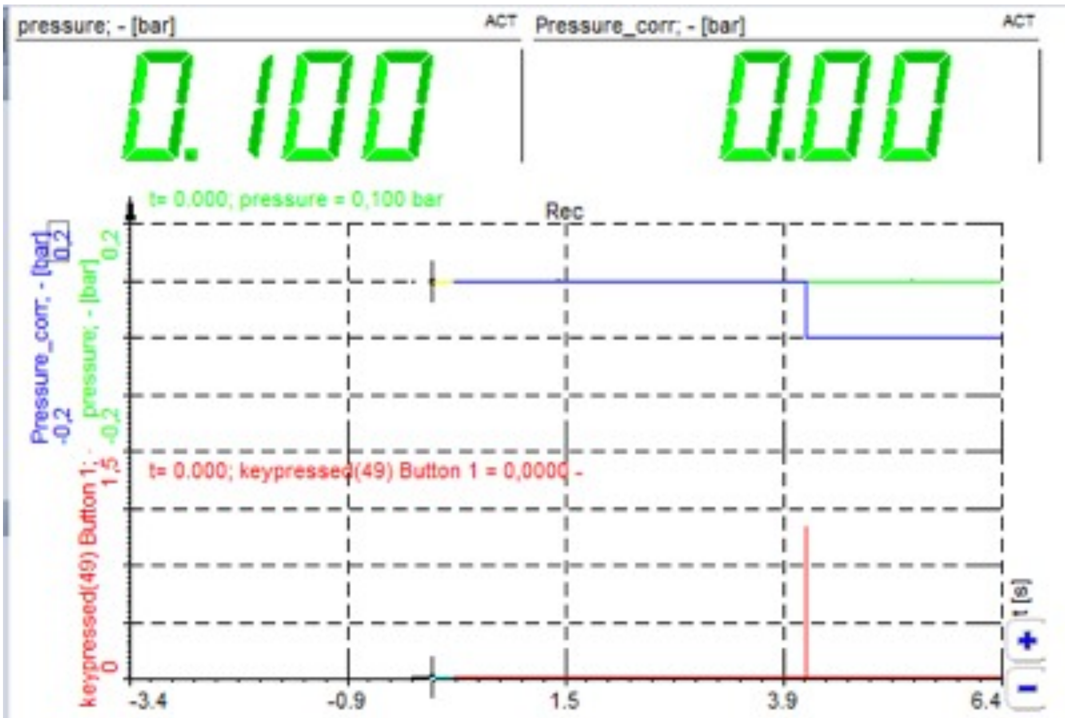
Time	pressrue/Latch bar
06.09.2010 18:42:44.999	0,9998
06.09.2010 18:42:43.234	0,9998
06.09.2010 18:42:39.904	0,9998
06.09.2010 18:42:38.040	0,9998
06.09.2010 18:42:35.642	0,9998

2.5 Function: HOLD – Example: Remove offset from a Channel

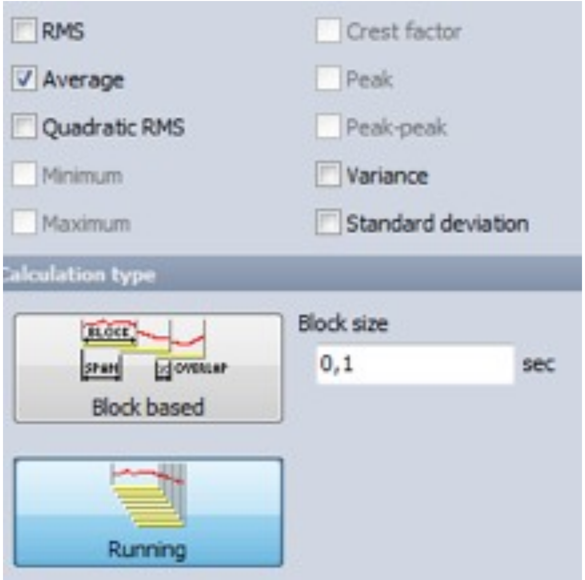
HOLD: HOLD(channel, condition) Will latch the actual value of the channel if condition become true.

MATH: 'pressure'-hold('pressure',keypressed(49)>0.5)
The actual pressure channel is subtracted with the value latched in the hold function.
The hold function will latch the actual pressure if the second statement (keypressed(49)>0.5) become true.

So even during the measurement a offset compensation could be done by pressing a specific key. The picture below is showing an example. The keypressed(49) channel is indicating the pressed key in an additional math channel to make it more abvious.



2.6 Function: Average – Example: Remove offset from channel POST Processing

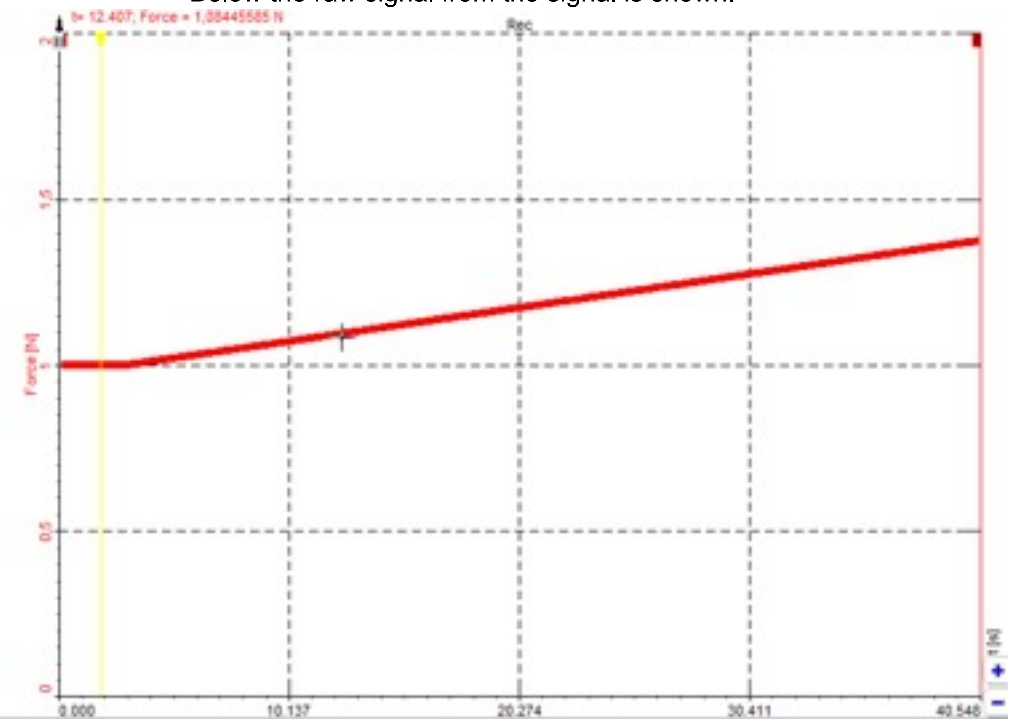


Above, “running” is selected.

Average: Calculates the running average over a Block size of 0,1s.

Example: A analog channel with an offset should be recalculated in Math. The first 0,2 seconds of the signal should be averaged and used for the offset correction.

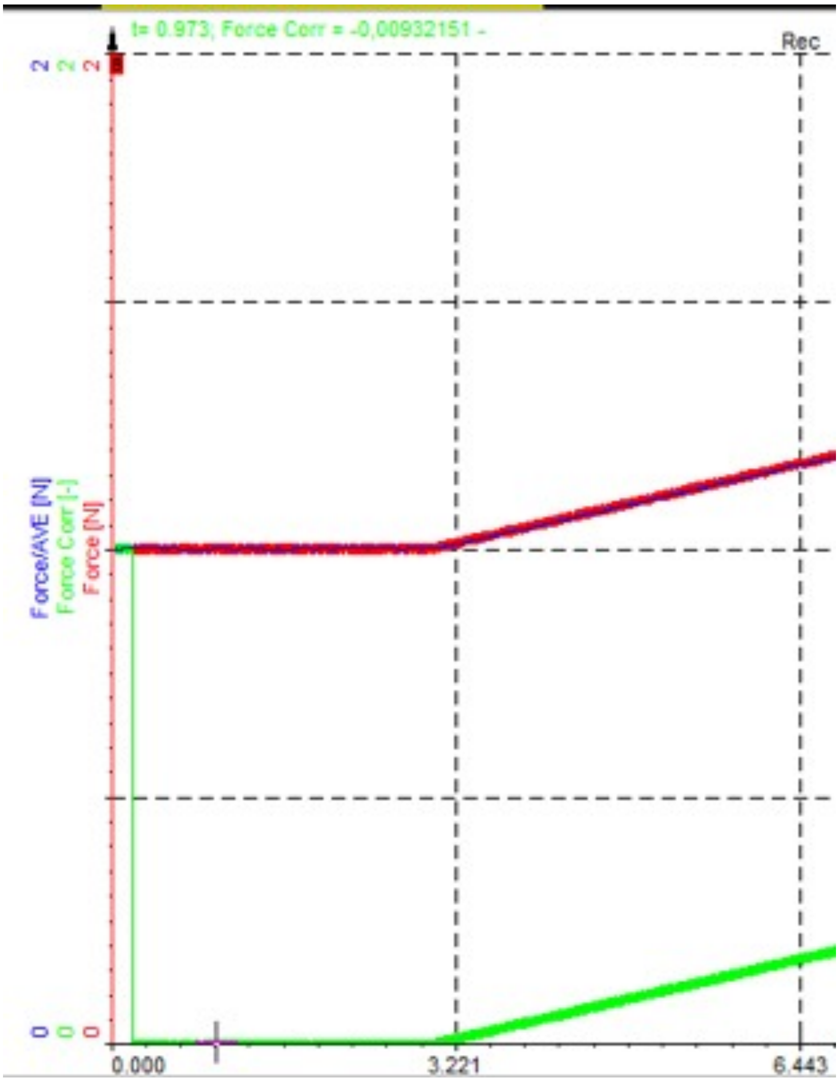
Below the raw signal from the signal is shown.



MATH: 'Force'-hold('Force/AVE',time>0.2)

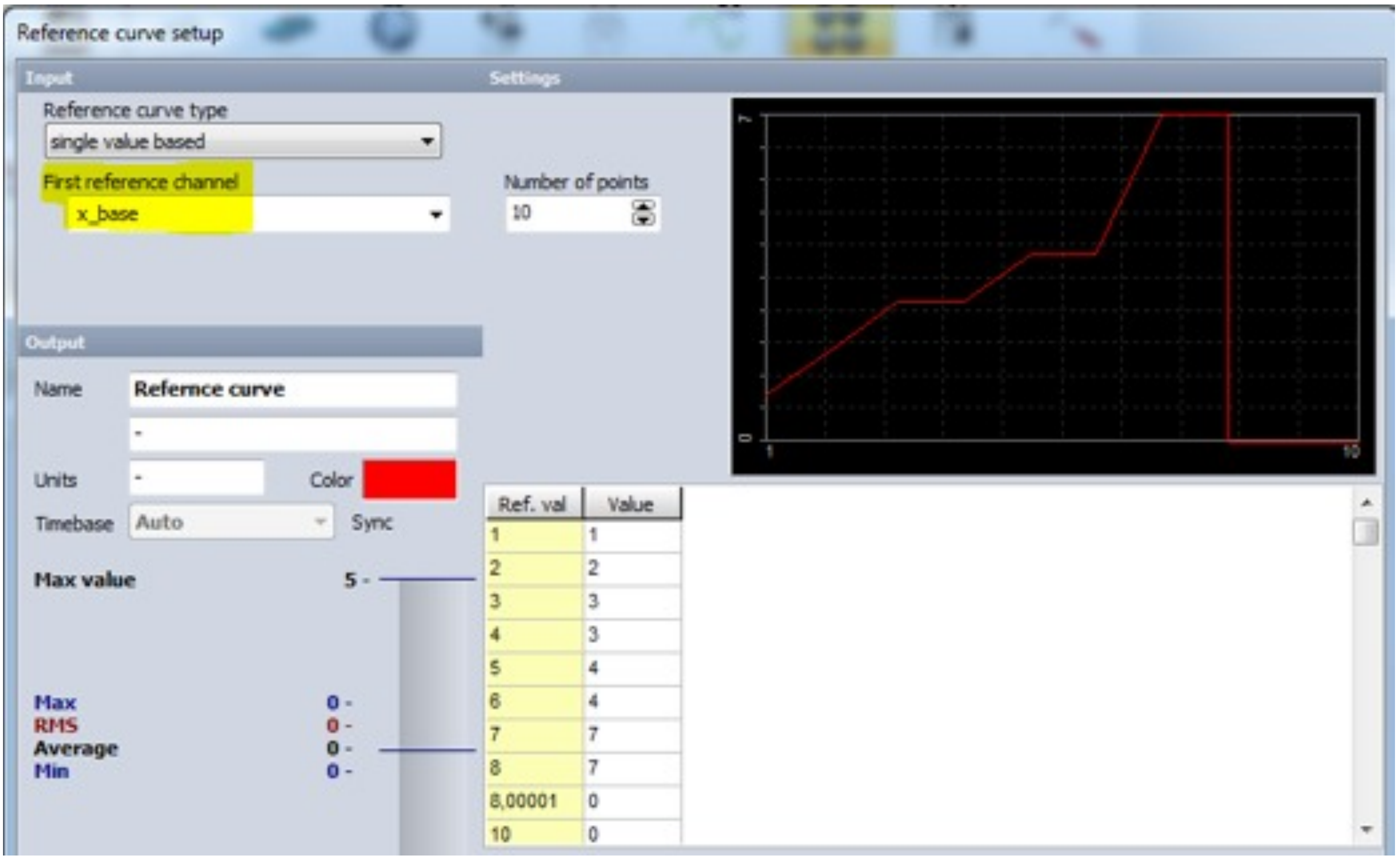
Force/AVE is the running average calculated out of the statistics. This will be latched if the time (reflecting the actual time in the dataset) is higher than 0.2 seconds, the hold function will latch the average and subtract it from the force.

The result is shown in the picture below.



2.7 Function: Create a reference curve, with restart possibility.

Example: The target is to create a free definable reference curve, with the possibility to restart.
Here the reference curve function with reference channel is used.
Below you will see a reference curve with is related to a second channel called x_base.
So if the x_base channel delivers 2 the output will be 2, if time delivers 6 the output will be 4, and so on...



No lets have a closer look to the x_base channel which is created in a math formula.

Formula: time+100-hold(time+100,keypressed(49))

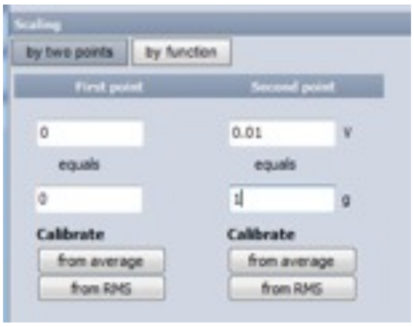
One of the target was to start and restart the reference curve on an event... which can be a keyboard event or any other event, like pressure < 12bar.

So let's have a closer look to the reference curve, after 8s it will become 0.
So at the beginning the x_base channel should start with a value higher than 8, so that the output of the reference curve is 0. That is solved by adding 100 to the time function of the formula (red part).
So at the beginning the formula will output 100 and count up ,… the reference curve will deliver 0.

The next thing would be to start the refence curve by an event. This is realized in the second part of the formula (green). If an event, (key pressed) will occur, (time+100) will be subtracted from time+100.
So the x_base channel will start from 0 and the reference curve will be output.

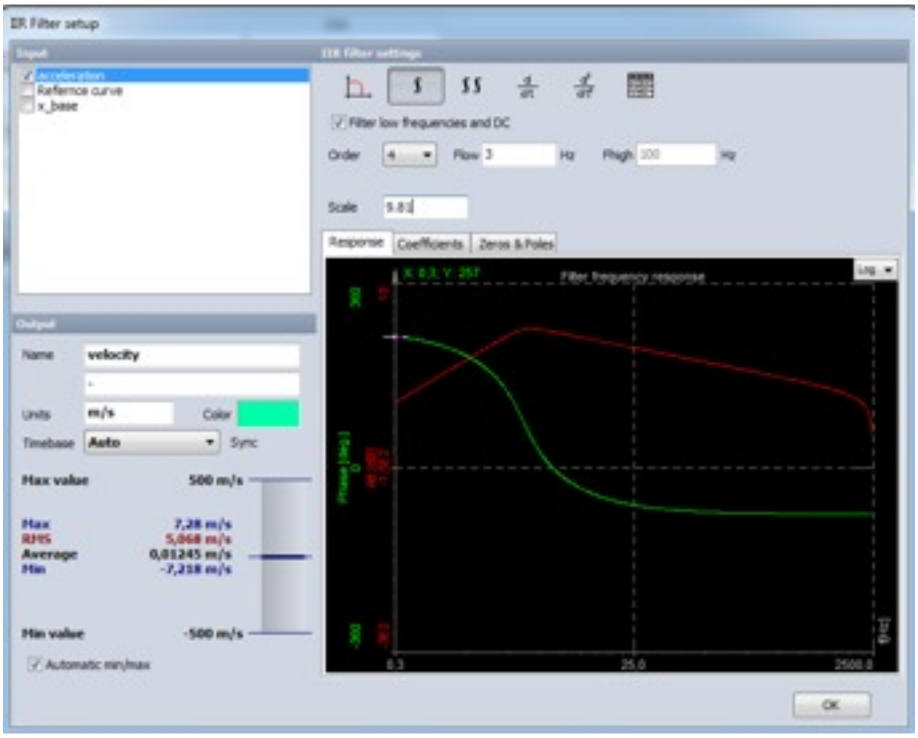
2.8 Function: Acceleration → velocity → displacement

Very often, acceleration must be converted to velocity or displacement. This is done with integration or double integration. But to do it right, we need to start at the beginning: the scaling.



1.) So the scaling is done either in g or m/s² in the analog setup of DEWESoft, where the measurement chain starts.

Velocity



2.) We have to apply the acceleration channel to a single integration to get the velocity. This will be found in the IIR filters MATH function.

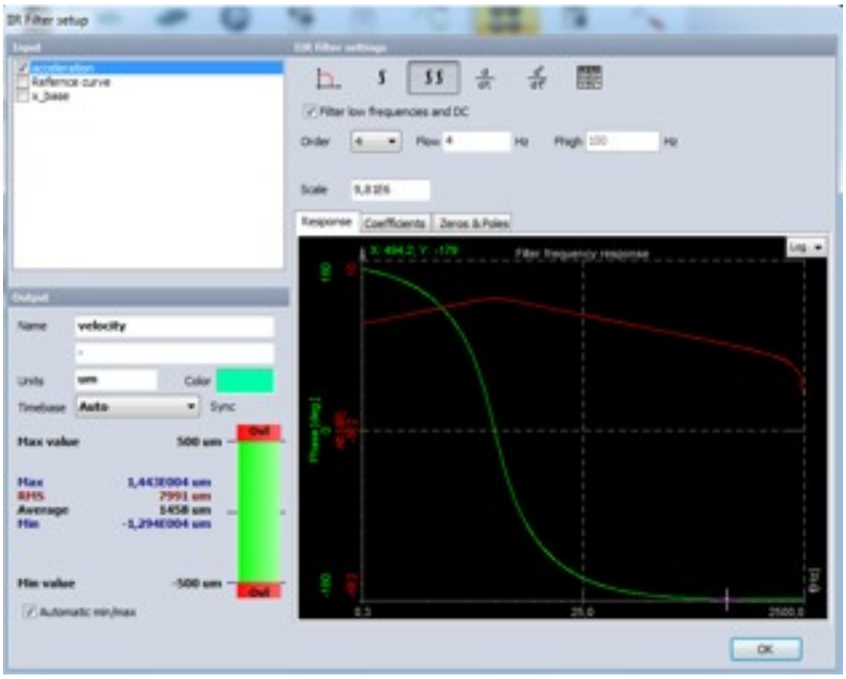
If an integration is done on low frequencies, or signals with a static offset, this will produce incorrect results. Therefore the integration function has an the option to filter low frequencies and zero out DC components (offsets). In this case, a 3 Hz Fourth order high pass filter is used which gives reliable results on most of the applications.

The unit should be m/s, and because the acceleration is scaled in g we have to add an additional scaling factor 9.81.

If the acceleration would have been scaled in m/s² in the analog setup the scaling factor is not needed.

In case mm/s is needed, either 9810 or 1000 has to be entered as scaling factor.

Displacement



3.) To get displacement we have to us the double integration. Also there the high pass filter will be needed, to avoid wrong results caused on low frequencies and dc offset.

4Hz 4 Order will give reliable results on most case. If not the filter has to be set higher (5Hz).

Most of the time the displacement is needed in um. So the scale factor has to be set depending on the analog acceleration scaling.

In this example the input was scaled in g, take a look to the picture above, so to get

um we have to enter a scale value of 9810000 or 1000 000 if the input would have scaling in m/s².

Note: As described above we have to use a high-pass filter to avoid incorrect results.

Also be aware that below the high-pass filter frequency, the output will be attenuated by the filter. So exact readings will be only available @ frequencies higher than the high-pass filter.
This is the case for both displacement and velocity.

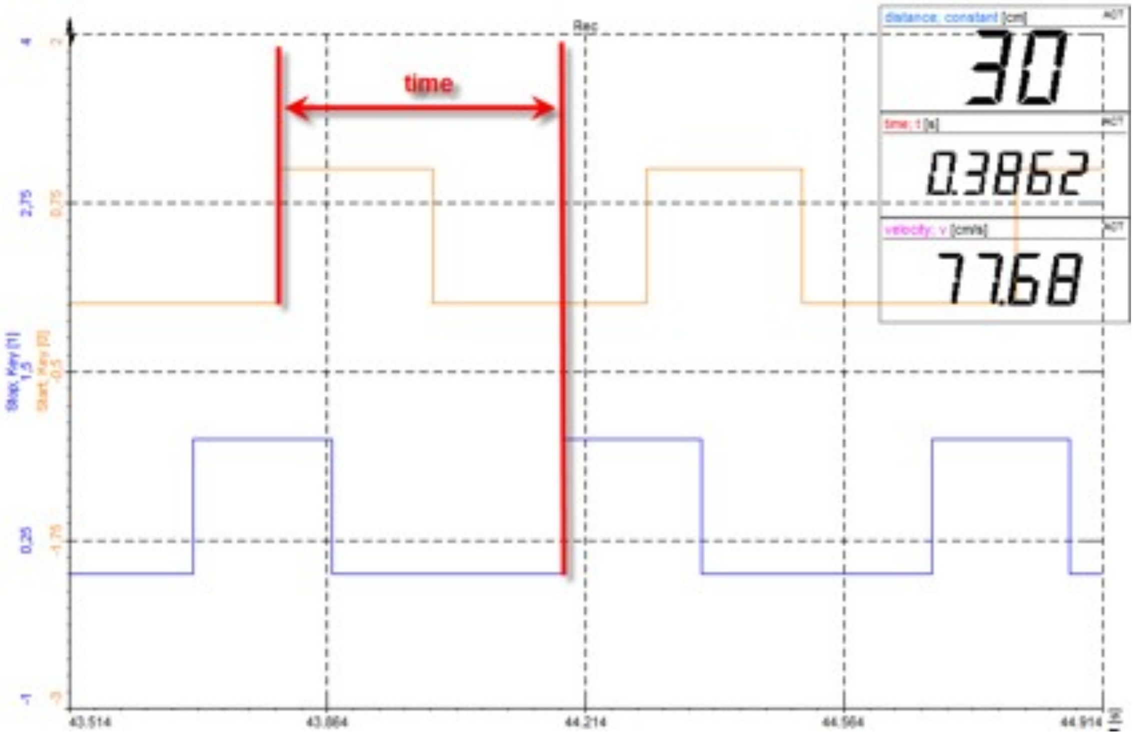
If displacement is needed below this frequency, it is preferable to use a displacement sensor in the first place and not use software functions.

2.9 Function: STOPWATCH – Example: Velocity with two light barriers

STOPWATCH: **STOPWATCH(Condition1,Condition2)** measures the time in [s] between condition1 and condition2. Condition is a logic value that jumps from 0 to 1 (edge sensitive)

Example: If the distance between the two light barriers is known as 'Distance' the averaged speed of an object passing the light barriers could be calculated.

MATH: **'Distance' / stopwatch('Start'>0.5,'Stop'>0.5)**
Start and Stop could be analog or digital signal. The logic comparison '>' turns the condition into a logic value anyway. Instead of 0.5 there could be used any logical triggerlevel. Also for digital signals the comparison is recommended.

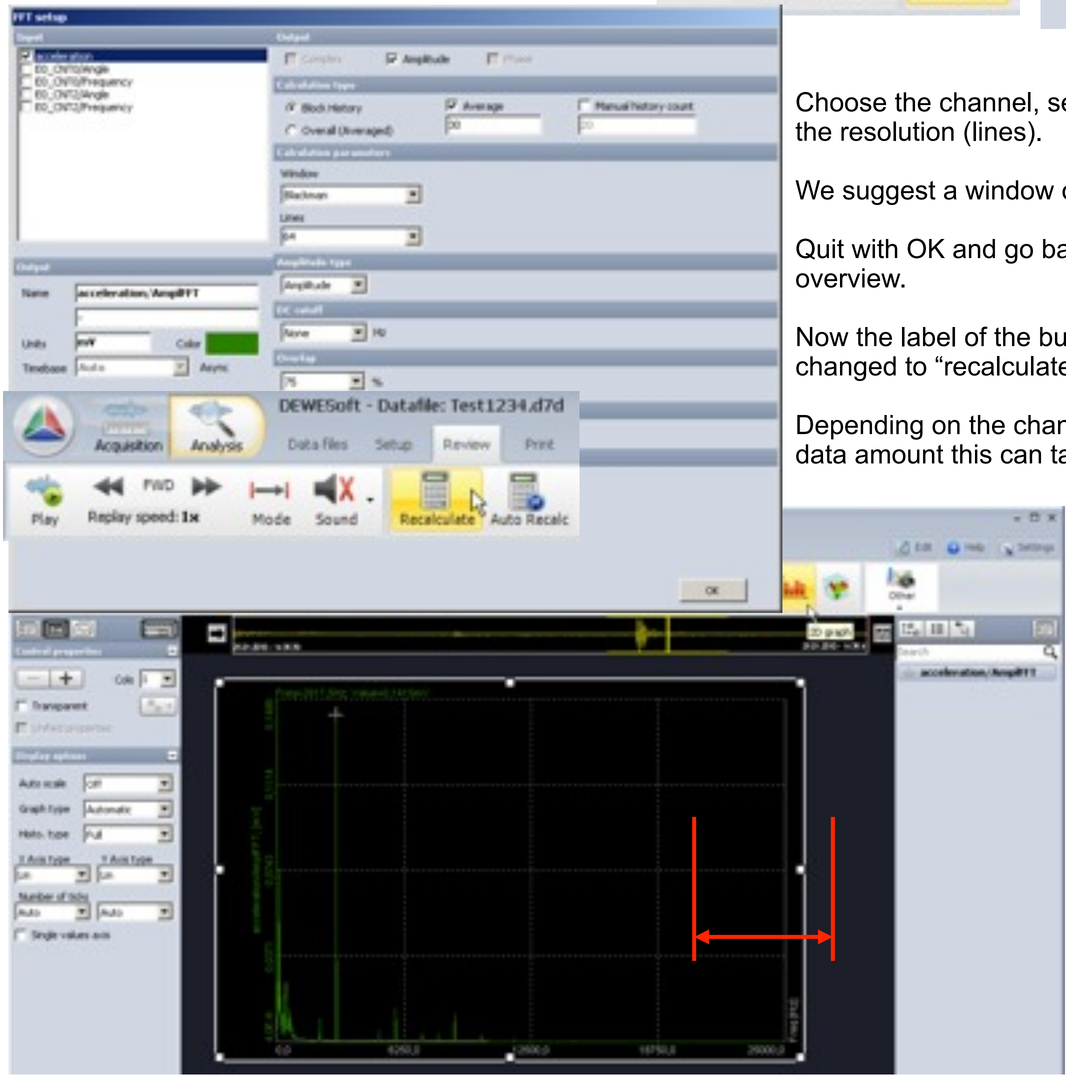


2.10 Hint: How to add an average FFT to a datafile

If you did not add an average FFT during measurement, it is a little tricky to achieve this later (because you cannot simply change the FFT instrument to Average afterwards).

But it can be done with MATH – in post-processing!

Open the datafile (*.d7d) and add in the ofline math a FFT function. This is listed under the 6th menu point.



Choose the channel, set the average cycles and the resolution (lines).

We suggest a window overlap of 75%.

Quit with OK and go back to „Review“ to the overview.

Now the label of the button “offline math” has changed to “recalculate”. Start the recalculation.

Depending on the channel count, resolution and data amount this can take a little while.

For the calculation results in a matrix, the instrument to be used is the „2D graph“.

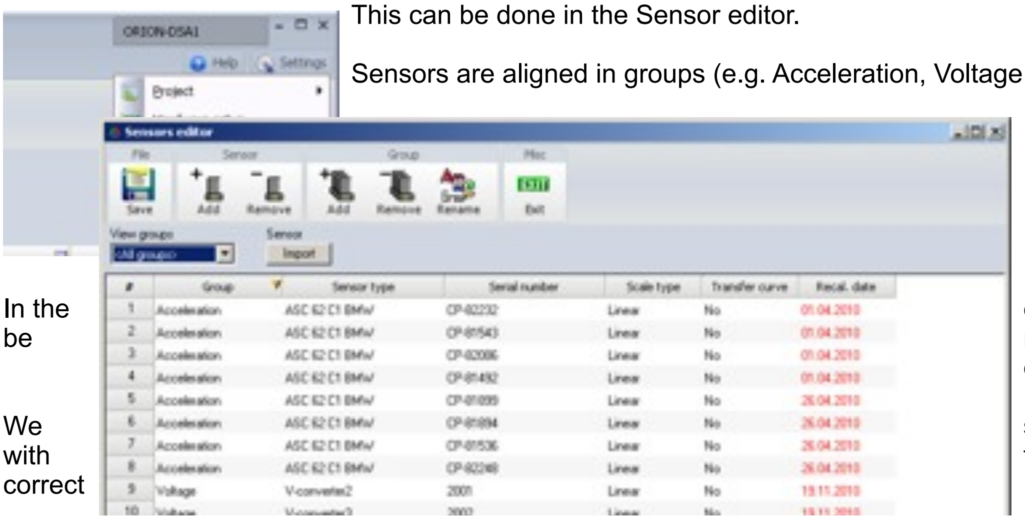
Hint when scrolling through: in the overview-instrument on top the yellow T-bar shows the FFT window-size.

2.11 Hint: Non-linear scaling of sensors

A linear sensor scaling can easily be done in the channel setup. How about correcting non-linear sensors?

This can be done in the Sensor editor.

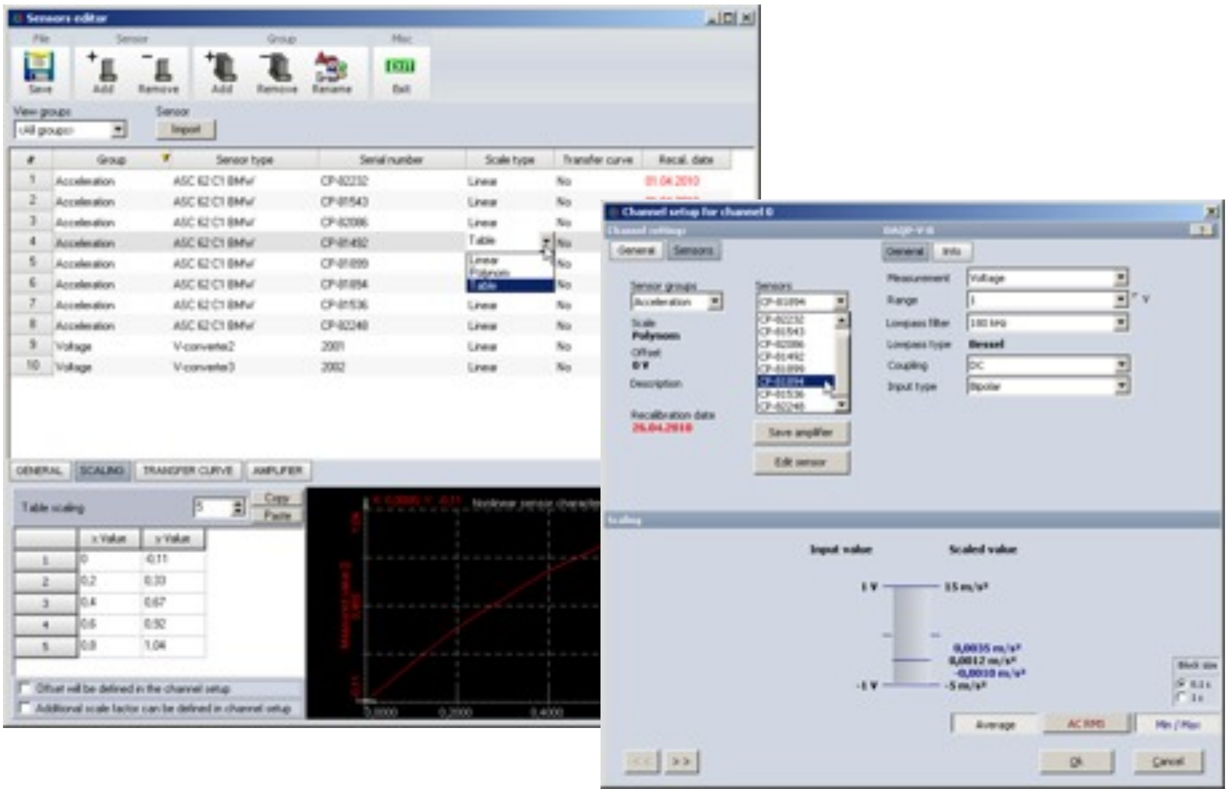
Sensors are aligned in groups (e.g. Acceleration, Voltage, Current…).



In the column „Recal. Date“ you will be remembered for the annual calibration.

We suggest to store the sensors with the serial number for the correct identifying.

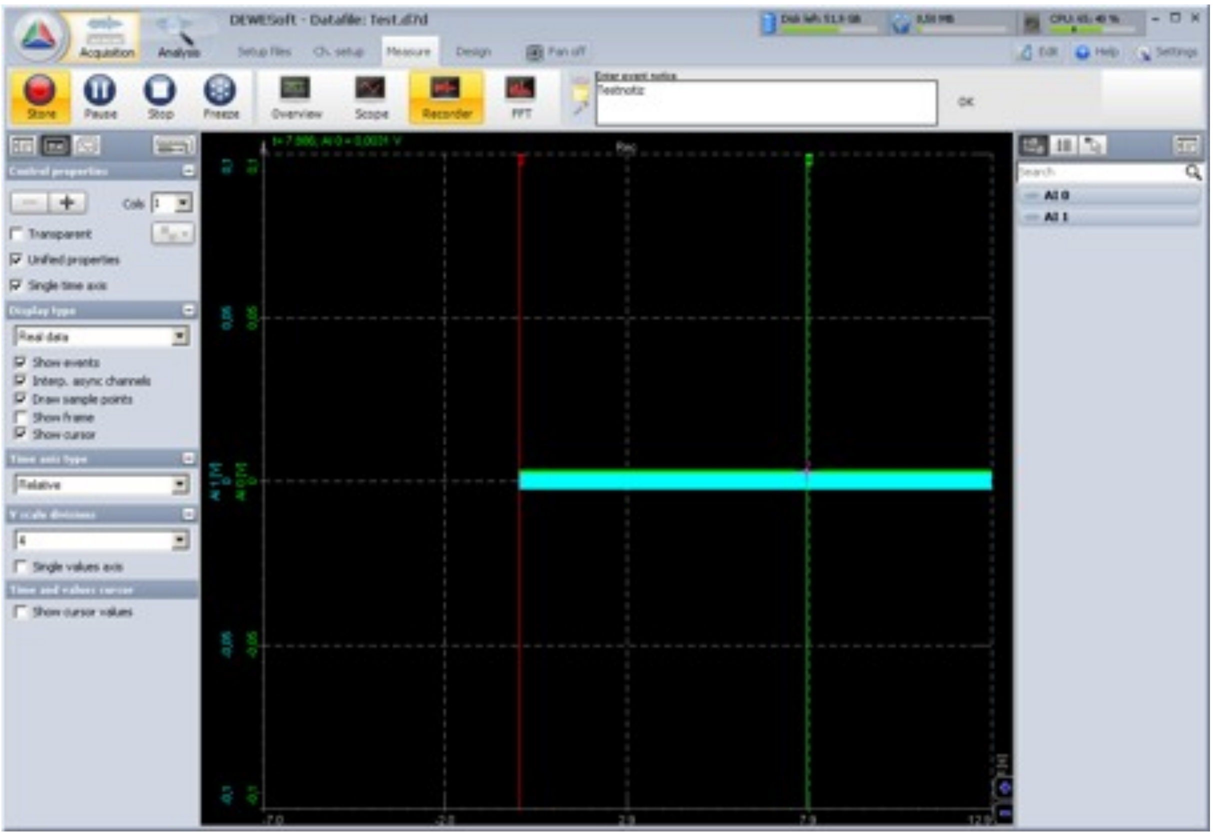
Change the scale type to Linear / Polynom / Table and enter your values below.
The sensor is then easily selected in the channel setup.



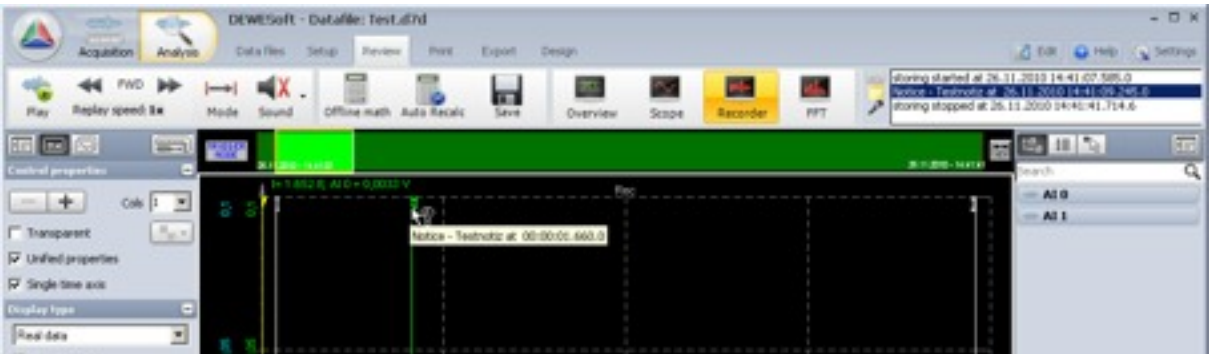
2.12 Hint: Add notes during measurement

If you press the button ‘n’ during storing a green marker occurs.
At the same time the Event window opens and you can enter your note.

You don’t have to hurry because for the note the time you pressed the ‘n’ key is stored.



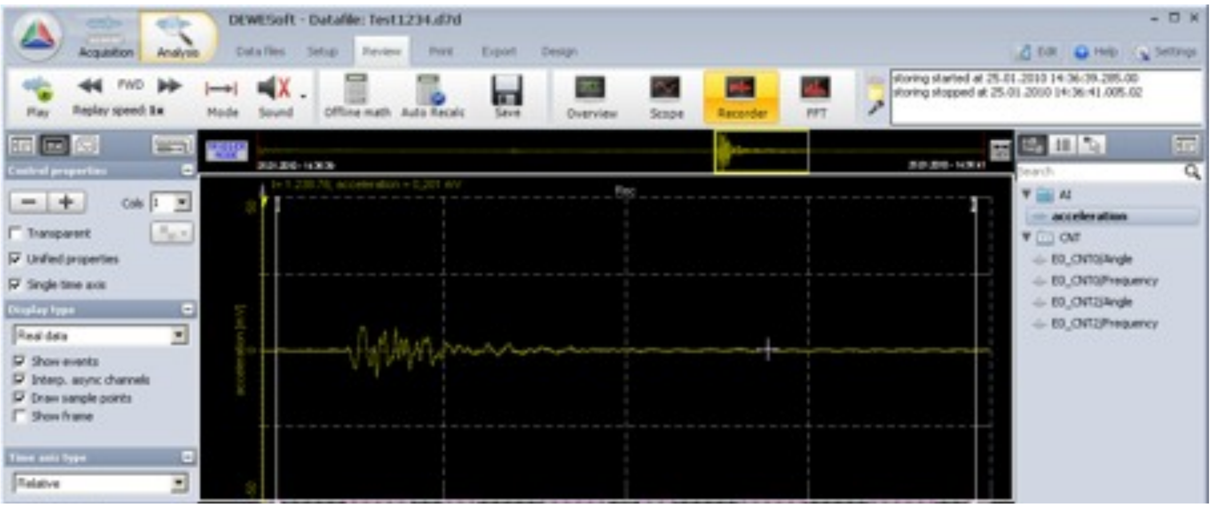
This is then saved to the datafile. Place the mouse pointer over the event and you see the note.



2.13 Hint: Downsize DEWESoft datafiles

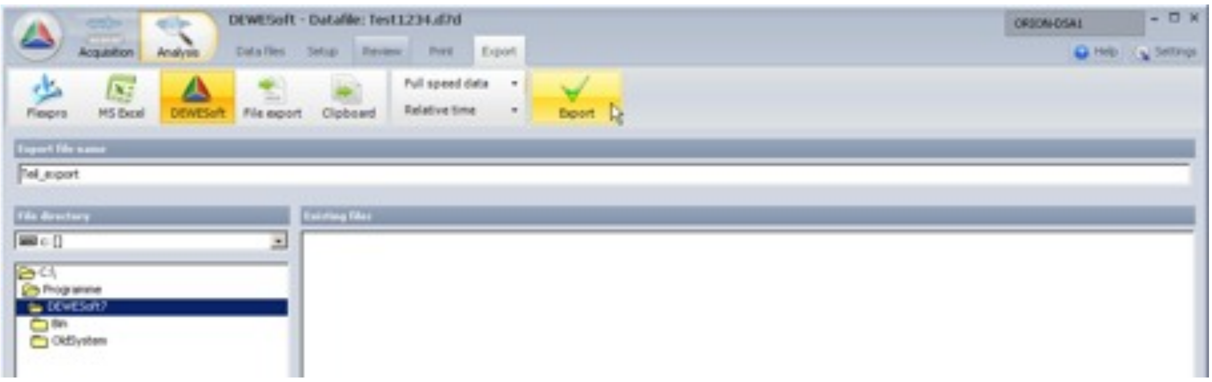
DEWESoft export -> DEWESoft

To reduce the size of a stored DEWESoft datafile you can for example export only the interesting part to a new DEWESoft datafile.
Therefore take a recorder instrument and select the area you want to export with the cursors. On the overview instrument on top you see the selection in relation to the whole file.



Then go to Export and select the option DEWESoft.

All channels are exported.

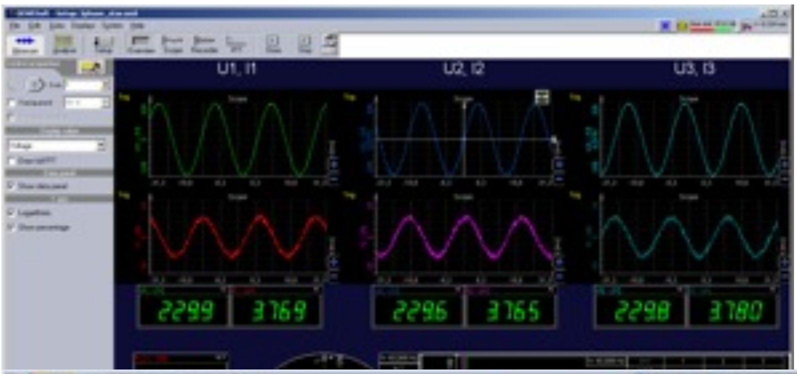


The new file is much smaller and has all the data you are interested in.

2.14 Hint: Porting DEWESoft6 setupfiles (*.dss) to DEWESoft7 (*.d7s)

One of the big advantages of DEWESoft7 is the open file structure. Let’s have a look at a practical example.

Open the old setup in DEWESoft 6.



Next save the setup as an XML file.

Close DEWESoft6 and open DEWESoft7.
Load the XML setup file in DEWESoft7, chose XML as filetype.

Hint:
You can also **copy a whole screen** of instruments.
Just right click on the background and chose “Copy Screen to Clipboard”.

Now you can Paste all the instruments to another Screen such as Recorder, Scope, … feel free to make your own layout.