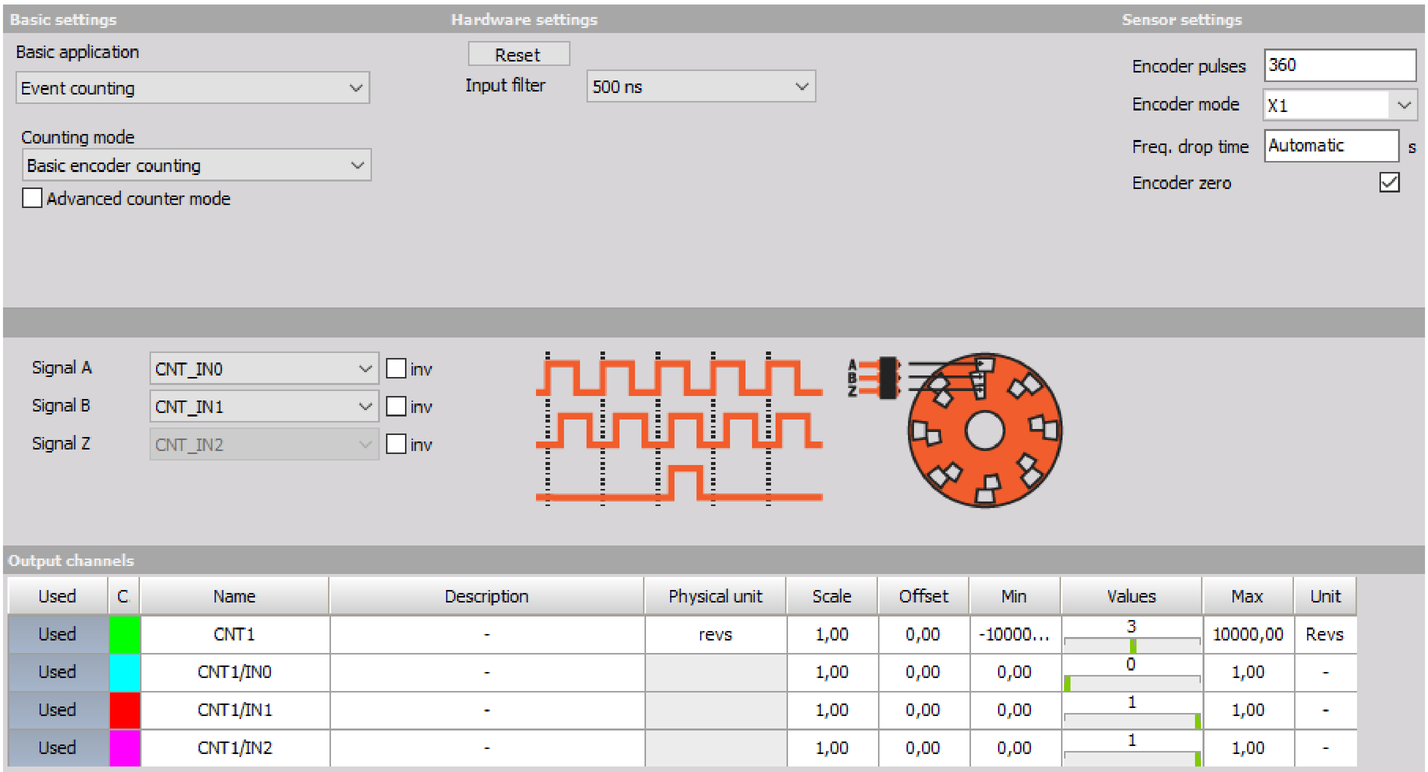
DEWESoft®

Measure Angle and Frequency With Digital Counters



What is a digital signal?

Most engineers understand that data acquisition systems are used to measure time-history signals, such as voltages, temperatures, and currents, vibrations sensed by accelerometers, strain sensed by Wheatstone bridge strain gage sensors, and more.

But often it is necessary to measure discrete events and angle of rotation signals which are synchronous with the more common time-history data. Discrete events are those events that have only two possible values, e.g. on / off switches, etc.

These are sometimes called digital signals since they are fundamentally composed of high-low (aka on/off) voltage states. In the sections that follow, we will present some examples of these additional signal types, and discuss how they are best measured and synchronized with the rest of the data.

What are Discrete Signals / Digital Signals?

Let's take the case of a proximity switch or sensor, which outputs a low voltage (0 V in this example) when the unit under test (UUT) is not nearby, but then outputs a higher voltage (5V) when the UUT comes within range. It may be necessary to record this discrete state in sync with our measuring system in order to put the analog data into context. So how can we do it?

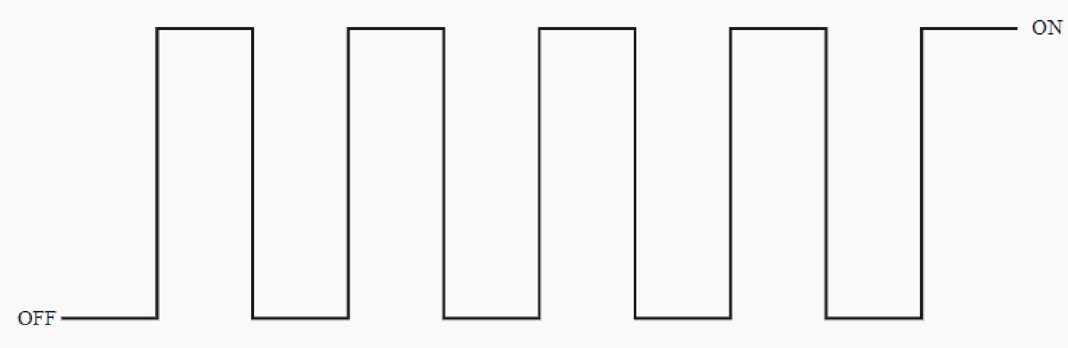


Image 1: Representation of the ideal TTL on/off system

One simple way is to take these electrical signals and input them into the analog inputs of the measuring system. This will work and only occupies one analog channel. However, what if we needed to record the states of eight proximity sensors or ten or more?

In that case, it would be an enormous waste of our wide-ranging and relatively expensive analog inputs to use them for such simple discrete signals. In addition, sometimes discrete inputs require higher bandwidth than the relatively slow analog inputs can support, so again an analog input may not be a good fit regardless.

When multiple discrete signals need to be acquired, it is more efficient and far less expensive to use a digital input that is designed for this task. In the case of Dewesoft DAQ systems such as the SIRIUS product line, each counter input can handle a variety of sensors with discrete digital outputs such as counters and encoders, as well as a number of discrete digital inputs.

And since counter/encoders typically run at a very high rate, the timebase of these inputs is quite high at 102.4 MHz, which provides a 10 MHz bandwidth - far in excess of typical analog inputs for physical measurement.

The simplest of digital inputs is the on/off type of signal that looks like a square wave if you look at it. These are sometimes referred to as discrete channels or event channels. Since they have only two states, they are often used to show the state of a door is open or closed, or a circuit being on or off, or a blade passing by, and a thousand other yes/no possibilities that we might need to measure.

What is a TTL signal?

Discrete inputs are normally output from a relay or transducer at TTL (transistor to transistor logic) levels, which are based on a 5V pull-up. In theory, the perfect TTL on/off signal would be 0V representing OFF (meaning a digital value of 0), and 5V representing ON (meaning the digital value of 1). However in practice, it is nearly impossible to achieve such precision, so the acceptable ranges have become 0 to 0.8V for OFF and 2V to 5V for ON.

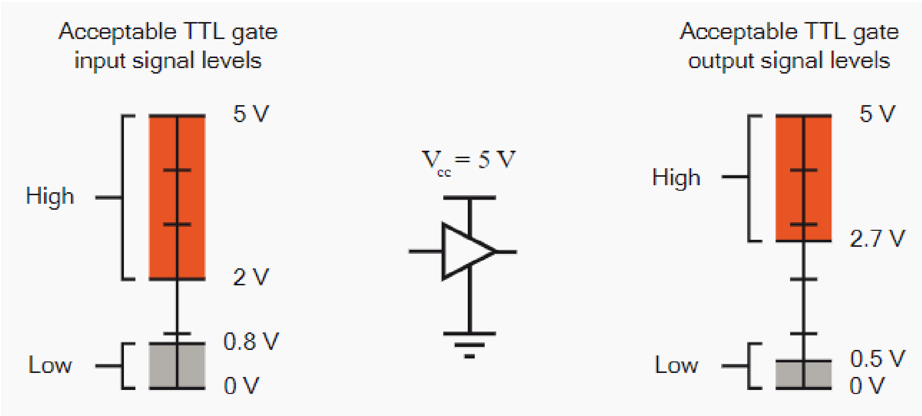


Image 2: Acceptable TTL level input and output levels

Dewesoft Digital / Discrete Inputs

Dewesoft's Counter/Encoder inputs provide three inputs which can be used for discrete/digital inputs. Certain models also provide dedicated DI (digital input) lines separate from the counter inputs.

What is Counter or Encoder and application where its used

What is Counter or Encoder?

Counters and encoders are in the business of counting pulses. But why? Sometimes the application really is just counting, but very often it is done to measure the angle or angular position.

For example, the steering wheel in your car - it is important to know exactly which way the car is being steered in real-time. So an encoder within the steering wheel divides the 360Â° of rotation into thousands of discrete steps. It is also configured so that the top-dead position (steering straight ahead) is at a known rotational position value.

All of today's safety and collision avoidance features, as well as self-driving features on passenger cars, farming vehicles, and more, rely on this encoder to know the exact position of the steering wheel at all times.

But just a short distance away from our steering wheel is a rotary dial on the dashboard where you can change your radio's volume or turn to a different channel. This dial is really an encoder with a digital output that feeds into a microcontroller that reads the encoder position you have selected and allows you to step through the possible choices, around and around in either direction.

Usually, the volume encoder is set with a start and stop position, while the channel changing encoder is allowed to spin around and around all of the stations. These are just three encoder applications that most people use every day in their automobiles.

Counter and Encoder Applications

Position and Angular position sensors can be found in a wide range of applications:

- Steering wheel position sensing
- Pedal position sensing
- Throttle position sensing
- Torque-sensing
- Process machine monitoring and control (thousands of applications)
- Maintaining absolute position references in CNC machines
- Controlling absolute patient position in CAT (Computerized Axial Tomography) and MRI machines
- Position feedback in robotics of all kinds
- Electronics systems, especially for the human interface
- Conveyor belt applications
- Parking sensors

An angular position sensor measures the angular position of a shaft. Angle sensors are available in a variety of packages and resolutions, from simple inductive sensors that simply count each time the shaft rotates, to high-resolution encoders that provide hundreds or even thousands of positions around the 360Â° rotation, and also report the direction of rotation.

There are several kinds of counter and encoder sensors that are used in data acquisition today, including:

Sensor type	Description
Proximity Sensors	Detects an object coming within a prescribed distance of the sensor and outputs a pulse. Used for counting, tachometer, and rotations speeds applications
Rotary Encoders	Rotary shaft sensor that outputs A, B, and Z signals with up to thousands of pulses of resolution around 360Â°
Linear Encoders	Same technology as a rotary encoder except that these encoders work in a linear fashion, i.e., in a straight line
Gear Tooth Sensors	Sensors with the defined number of pulses per revolution (usually 60), sometimes with missing teeth (60-2) for angle or starting point is known,

Sensor type	Description
Optical sensors	Non-contact optical angle sensor that detects either hole in a rotating disc or white/black stripes on tape affixed to a shaft.

What is a Digital counter?

An electronic counter is a digital input circuit that counts the number of pulses that are input from proximity sensors, encoders, and similar devices. The simplest type of digital counters track and count time and pulses. Much like a stopwatch, they can be reset and can count up to a certain value.

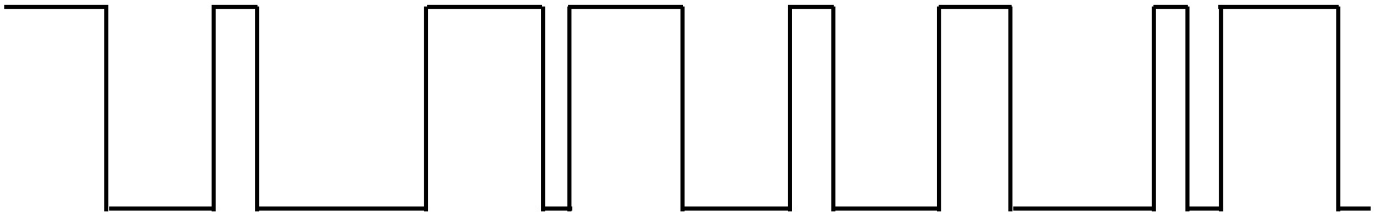


Image 3: This digital pulse train from a proximity sensor indicated a variable rotational speed

The simple counters provided by most manufacturers usually provide ample capability when used by themselves, but when they are part of a data acquisition system that is simultaneously measuring analog data and perhaps even additional inputs, like CAN bus data, PMC data, video frames, etc. they rarely provide true-time synchronization across all of these inputs. This is where SuperCounter technology from Dewesoft comes in.

Dewesoft SuperCounterÂ®

Counters are mainly used for measuring RPM and angle of rotating machines. Dewesoft super-counters work on a 102.4 MHz internal time base, always, independent of the current sample rate. In comparison to the standard counter, which only outputs whole numbers like 1,1,2,2,3,4,... one sample later, Dewesoft is able to extract the accurate values like 1.37, 1.87, 2.37,... fully time- and amplitude-synchronized! This is done by measuring the exact time of the rising edge of the signal with an additional counter. Each counter has three digital inputs that are fully synchronized with analog data.

Dewesoft SuperCounter is compatible with a broad range of encoders, gear tooth sensors, proximity sensors, etc. Dewesoft systems like SIRIUS, DEWE-43A, MINITAURs, KRYPTON, etc. can be configured with one or more SuperCounter inputs. They are usually provided on a rugged locking LEMO connector, but there are other connector possibilities with some models.

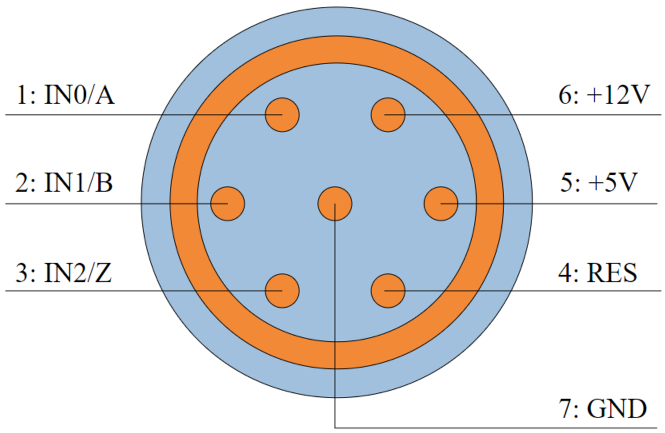


Image 4: Typical Dewesoft SuperCounter on LEMO 7-pin connector

There are typically three inputs because encoders require it. If you want to measure discrete inputs (TTL on/off signals), then you can use these three inputs as independent discrete inputs instead of for a counter. Then there are +12V and +5V sensor supply voltages available, a digital output (to be discussed in a different article), and a ground connection.

The inputs are TTL level, meaning that their low state must be below 0.8V, and their high state must be higher than 2V (up to 5V). Let's look a little closer at the electronic specifications for SIRIUS counters:

SuperCounterÂ® Inputs	
Timebase	102.4 MHz
Timebase accuracy Typical	5 ppm, Max: 20 ppm
Max. Bandwidth	10 MHz
Input filter	500 ns, 1 Î¼s, 2 Î¼s, 4 Î¼s, 5 Î¼s, and 7.5 Î¼s
Input level compatibility	TTL (Low: <0.8, High: >2V)
Input impedance	100kÎ© pull-up to +3.3V
Input protection	Â±25Volt continuous
Alarm output	Open collector, max. 100mA/30V
Sensor supply	5V/100mA, 12V/50mA

Before we get into all of the digital input operating modes and how you can use them, we should review a very important aspect of what makes SuperCounters so special - and it has to do with how they precisely align counter data with the analog and other data.

Aligning Counter Data with Analog Data

The standard counters available on most DAQ systems today provide only integer resolution outputs (e.g. 1, 1, 2, 2). As a result, their outputs are always one sample behind the analog sensor data. This can be a real problem in applications like rotational or torsional vibration when a phase shift of even one sample can change the results.

SuperCounters solve this problem completely by extracting floating-point values like 1.37, 1.87, 2.37, and then aligning them precisely in time with the rest of your data. In fact, a SuperCounter is really two counters in one. The input is fed in parallel into both counters, and the sub counter measures the exact time of the rising edge of the signal. Thus the real value of the counter with respect to the analog values is calculated and perfectly aligned.

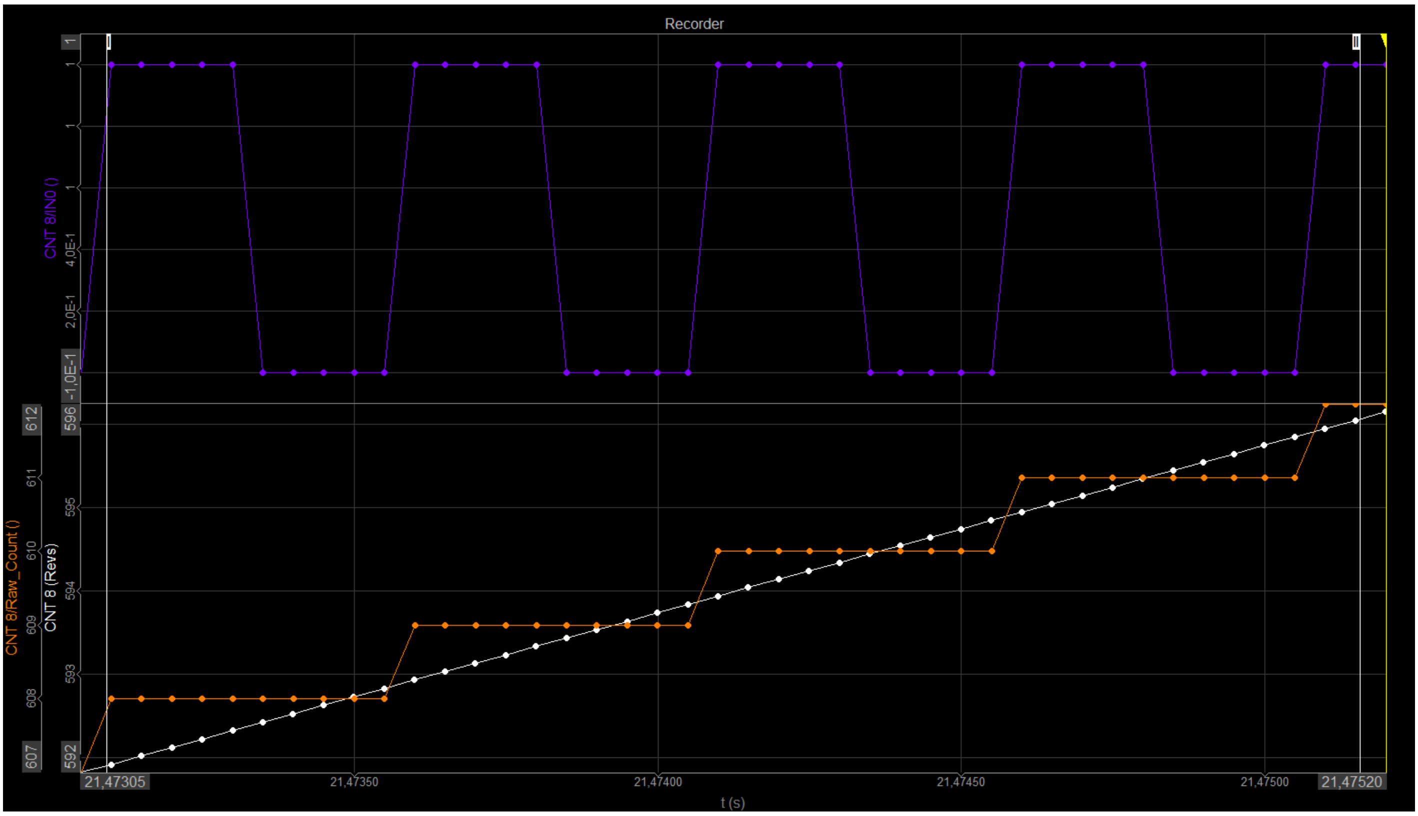


Image 5: SuperCounter values are aligned because of the interpolation between the values

The video below shows how SuperCounter technology measures counter signals fully synchronized with the analog channels. This video includes a real-world comparison between normal counting mode and SuperCounting mode.

[Video available in the online version]

Other data sources, like CAN bus, XCP, video, and others are also synchronized with the analog data in all Dewesoft data acquisition systems.

The other secret behind this technique is that Dewesoft's SuperCounters run on a 102.4 MHz time base that is independent of and much higher than the analog sampling rate.

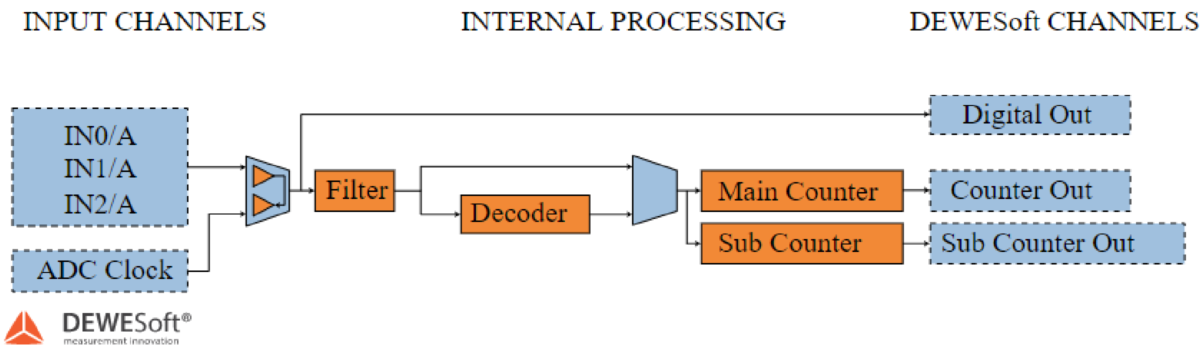


Image 6: Dewesoft SuperCounter architecture

There is also a special counter pinout on Sirius slice, called STGM-DB.

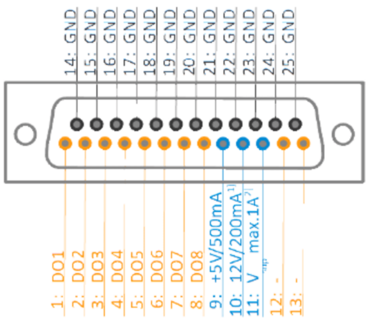


Image 7: Special counter pinout on Sirius STGM-DB

We can connect the digital inputs to several Dewesoft DAQ devices. Take a look at the image below for more information.

DI - Availability

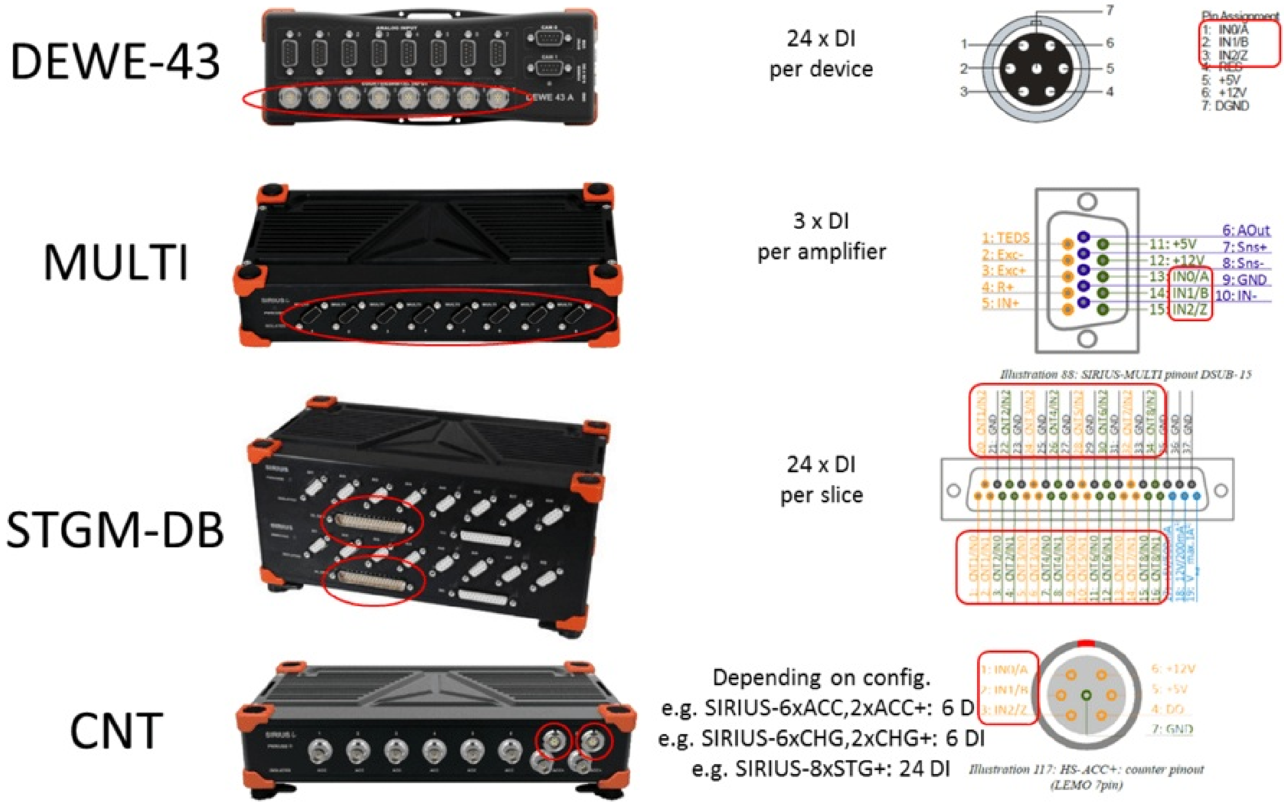


Image 8: Digital inputs on Dewesoft DAQ devices

Counters in Dewesoft

An electronic counter is a sequential digital device, which is used for counting the number of pulses that are coming at the input line in a certain time period.

There are many different types of counters, which are designed regarding the particular needs of the customers. The simplest type of digital counters track and count time and work similarly as a well-known stopwatch. This means that they can be reset and can count up to a certain value of digits.

Dewesoft Counter module is used to perform counting and frequency measurements. Typical applications are:

- event counting (basic, gated, up/down, basic encoder)
- sensor (encoder, tacho, CDM, 60-2,...)
- waveform timing (period, pulse-width, duty cycle)

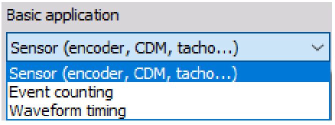


Image 9: Basic counter application

A special mode, called super-counter, allows exact frequency measurement and direct support of the most commonly used sensors. All other counters offer only a specific subset of operation.

> The maximum frequency bandwidth for the counters is 10 MHz!

Basically, the input to the counter should be a clean digital signal, where most of the cards support 5 V levels as the default (some cards offer higher input signals or even variable trigger levels). If the user has lower or higher signals, some conditioning in front of the counters should be used.

Input filter and the importance of filtering

The Importance of Filtering

In the real world, noise and glitches on the counter outputs are not uncommon. The problem is that if glitches are high enough in amplitude they can be counted as pulses, resulting in wrong values. Dewesoft SuperCounters provide advanced filtering on their inputs that you can use to mitigate this problem, just as you do in the analog domain.

Setting the filter

We need to set the input filter for the counters. The input filter is needed to prevent glitches and spikes in the digital encoder pulse signal. It can be set from 100ns to 5us, the optimal setting is derived from the following equation:

$$InputFilter[s] \leq \frac{1}{10 \cdot \frac{RPM_{max}}{60} \cdot PulsesPerRevolution}$$

where:

- RPM_{max} are maximum revolutions per minute [RPM]
- PulsePerRevolution is the number of pulses per revolution of the sensor

Factor 10 in the equation means that we take 1/10 of the pulse width at max RPM.

Example: Let's say that our machine is running at 3000 RPM and we have the encoder with 512 pulses per revolution. If we insert this number in the equation above, we get the next result:

$$InputFilter[s] \leq \frac{1}{10 \cdot \frac{RPM_{max}}{60} \cdot PulsesPerRevolution} = \frac{1}{10 \cdot \frac{3000}{60} \cdot 512} = 3,9\mu s$$

We need to set the filter to react a bit faster than what we expect the events, we need to set it a bit slower than the expected frequency of the glitches. With a manual switch and a 102.4 MHz base clock, some glitches can be expected.

The red curve shows the digital signal from the switch and the blue curve shows the counter value. The counter value is increased by each transition from low to high.

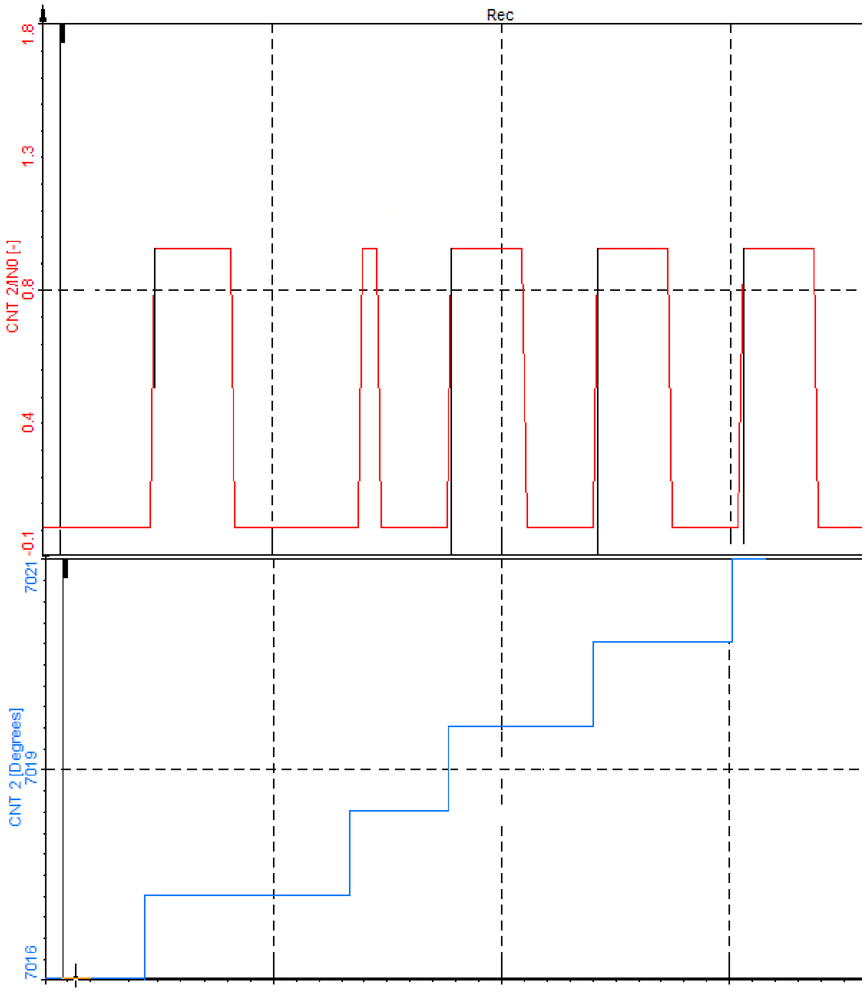


Image 10: A glitch being miscounted as a pulse

We can see at some points that the values are counted up when there is a glitch (no real pulse). This is because a counter can see every glitch (even below 20 nanoseconds) in the signal. Therefore, we need to use a filter to filter out these glitches.

For example, we use the filter of 500 nanoseconds. Filter checks if each pulse is high at least 500 nanoseconds. If it is, the counter value is increased.

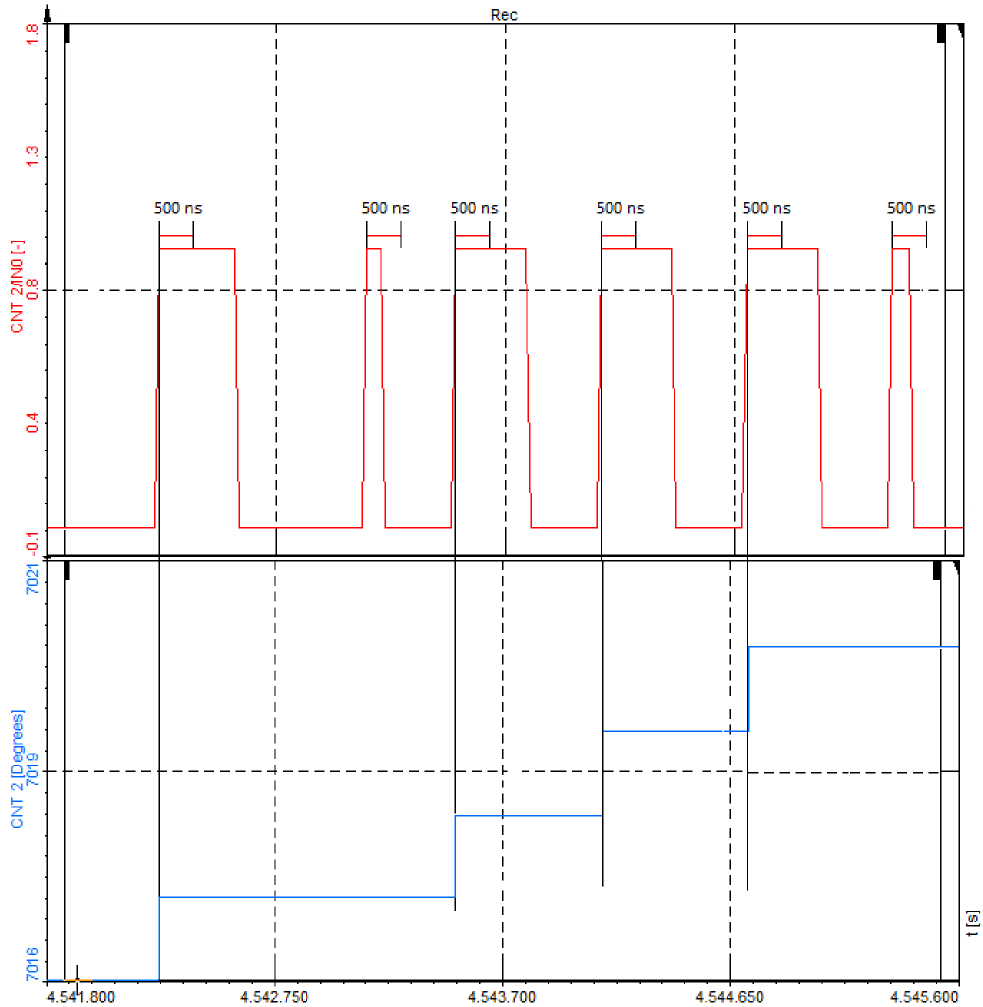


Image 11: With filter, the glitches are not counted as a pulse

Basic event counting

Basic event counting is the mode where we can count either falling or rising edges of the signal. You only need to connect the signal to one of the counter inputs and ground.

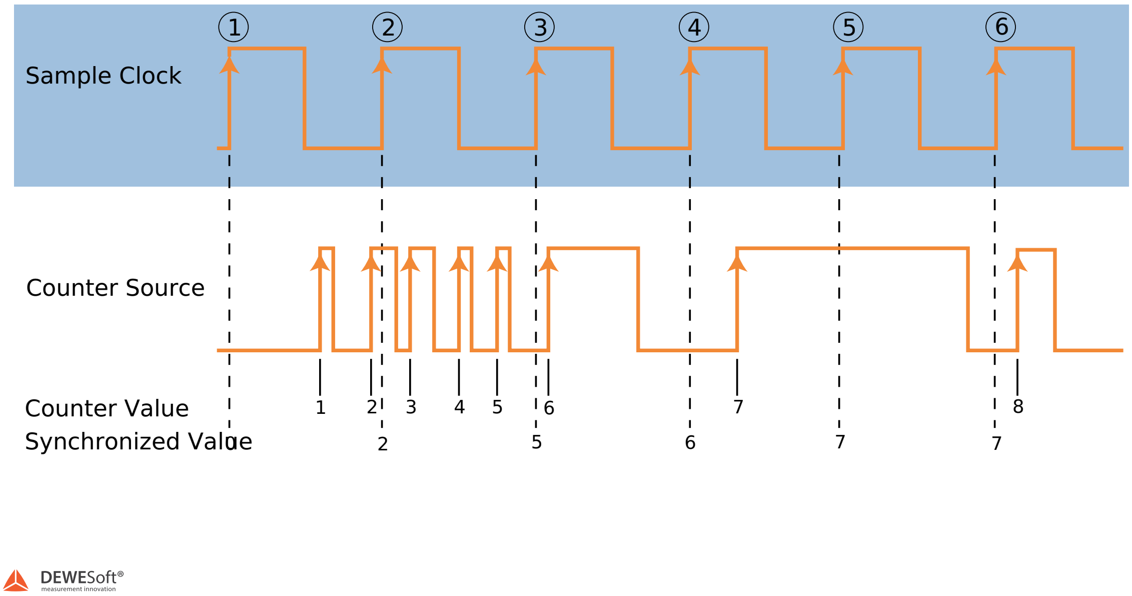


Image 12: Basic event counting mode

In this case, we will just add a switch between the counter input and the ground. Since all inputs have pull-ups, shorting the switch will give zero and opening it will give one

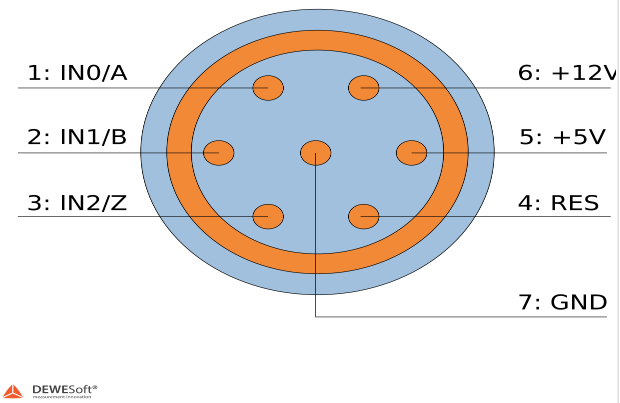


Image 13: LEMO 7-pin connector

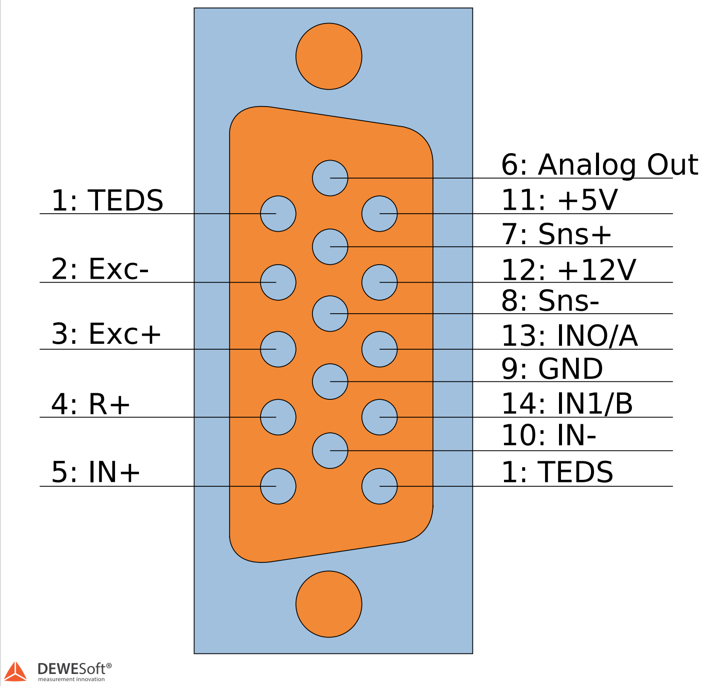


Image 14: MULTI connector

In the software, select the Event Counting application, then the Basic Event Counting mode. Now just let the software know which input pin you connected the signal to. You can elect to have the system count UP or DOWN. There's a checkbox to reset the count at the start of measurement or not.

The normal state (when the switch is not selected) is high, therefore, it is nice to invert the signal by choosing the "inv" checkbox.

This has two effects: firstly, the levels will change, so if the button is not pressed, the level will be low and consequentially the counter will count on falling edges.

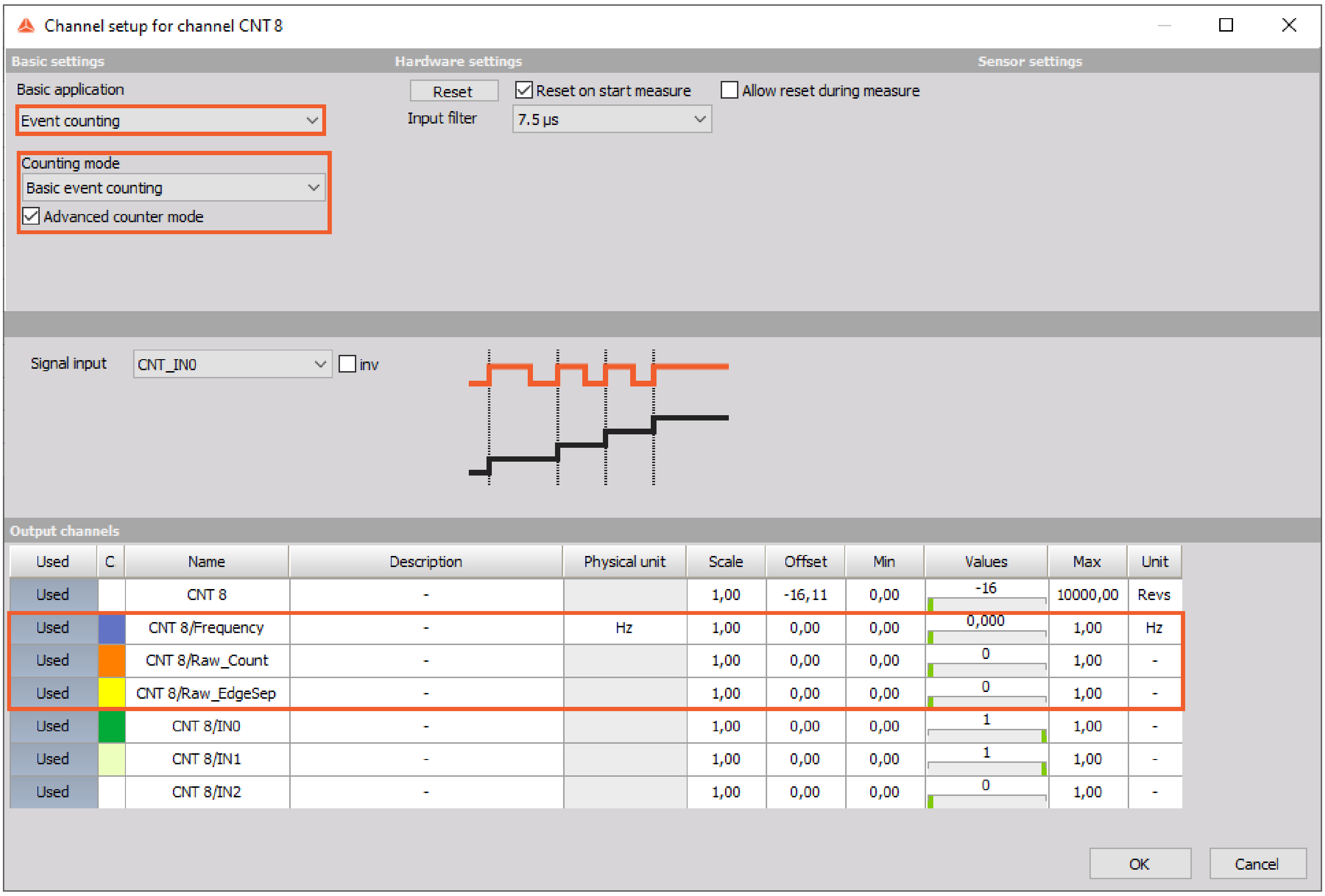


Image 15: Basic event counting setup in Dewesoft

Measurement with basic event counting.

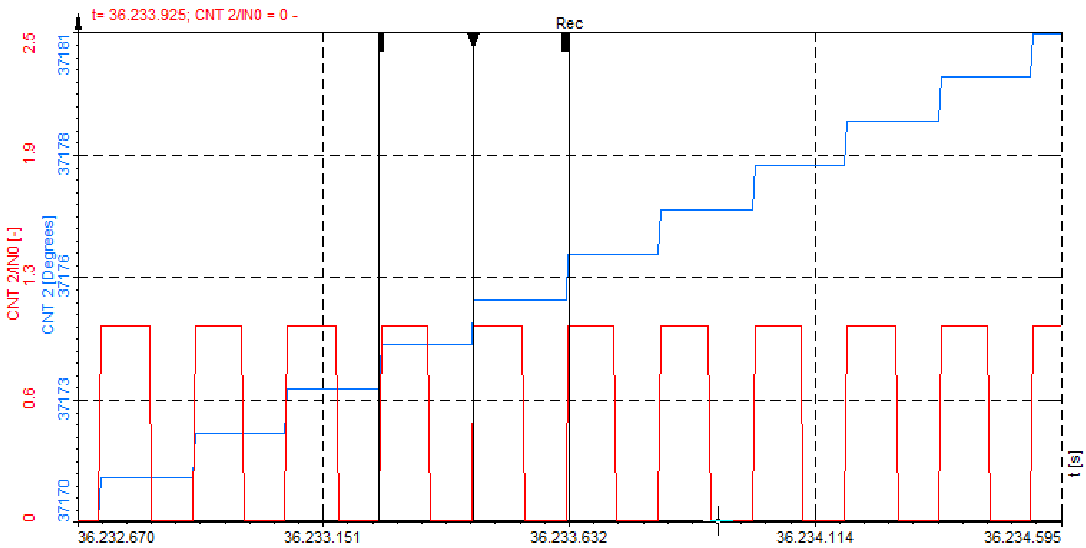


Image 16: An example of measurement with basic event counting

Gated event counting

In gated counting mode we will only count a pulse when a gated signal is high. You connect both the pulse signal itself to input IN0 (and ground) and connect a second signal to input IN1, to serve as the gate signal

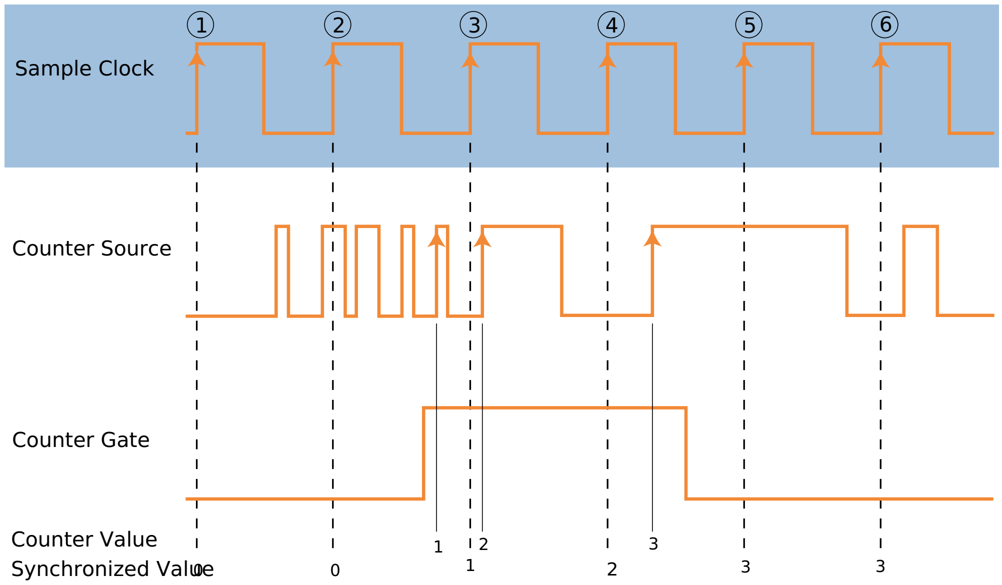


Image 17: Gated event counting mode

The counter signal is connected to the same input as with Simple event counting, but we need to connect an additional second signal - the gate. We should connect the second switch to CNT_IN1.

Then we perform the setup of the counter channel. We should choose Gated event counting mode and set the Signal input (CNT_IN0) and the Signal gate (CNT_IN1).

The counter will count the transitions from low to high only when the signal gate is high. Since the Signal gate is inverted (normally it is high), it is necessary to also invert the gate signal so it will count only when a button is pressed.

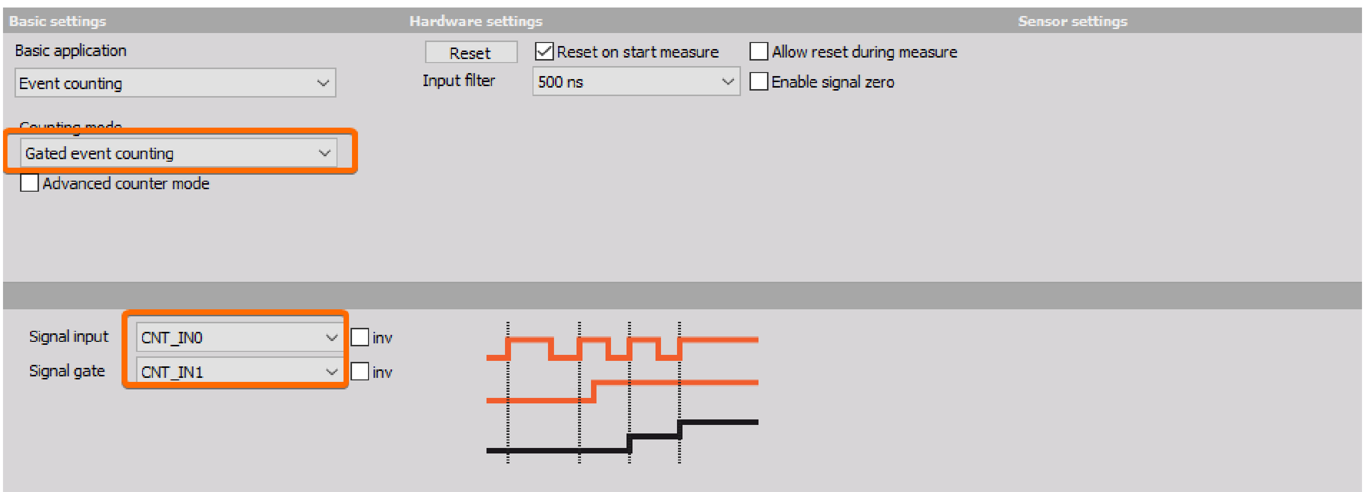


Image 18: Gated event counting setup

In the example below, we can see how this counter works.

The green signal (from the counter) counts up when the red signal (gate signal) makes a transition from low to high and when the gate signal (red) is high.

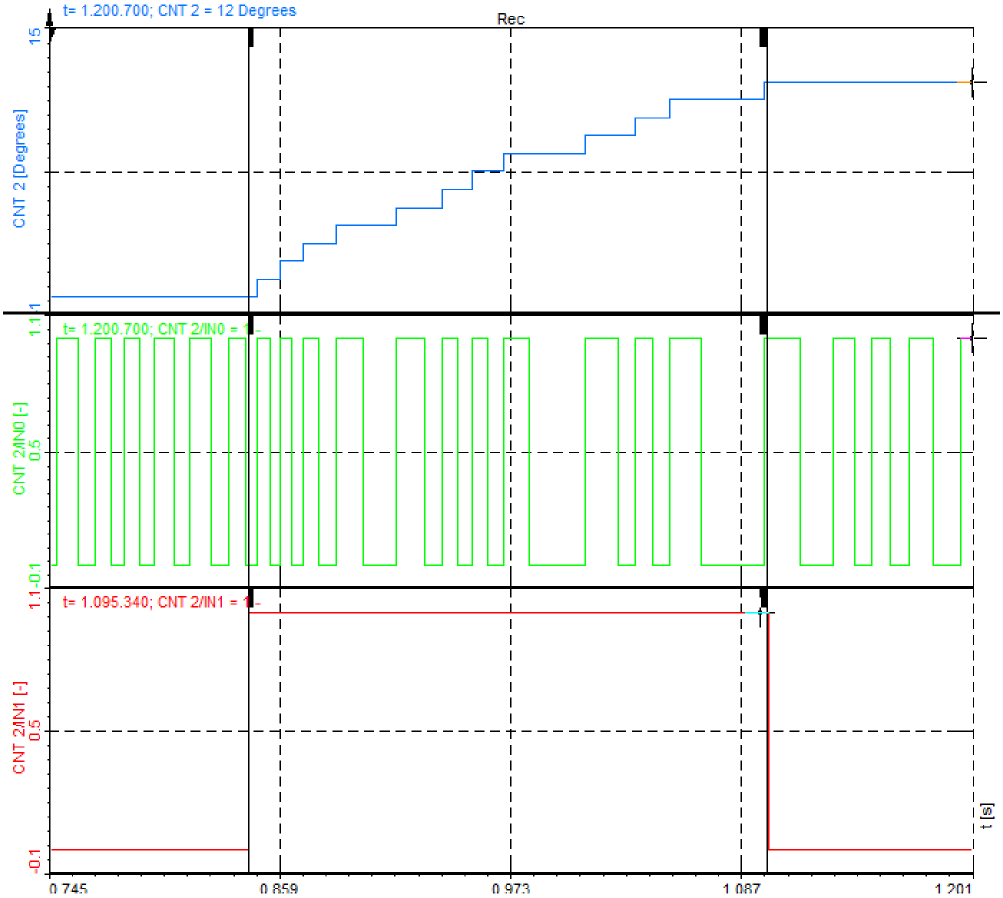


Image 19: An example of measurement with Gated event counting

Up/down counting

Up/down counting is a counter operation that counts up when the gate is high and counts down when a gate is low. The encoder operation is similar to this one, in fact, we can use this mode to make X1 mode encoder measurements, but some more electronics are needed.

If you look at Image 20, you can see that the Counter Value at the last Counter Source pulse equals -2.

How?

At the first Counter Source pulse, it counted -1, as the Counter Aux gate was low at that time, and it continued counting in minus until the 4th Counter Source pulse. After the Counter Aux gate changed to high, which means it will start counting in a positive direction. So at the end of the positive Counter Aux gate staying high, we have a Counter value of -1. Then it continued to count in a negative direction, so we ended up with -2 at the last Counter Source pulse.

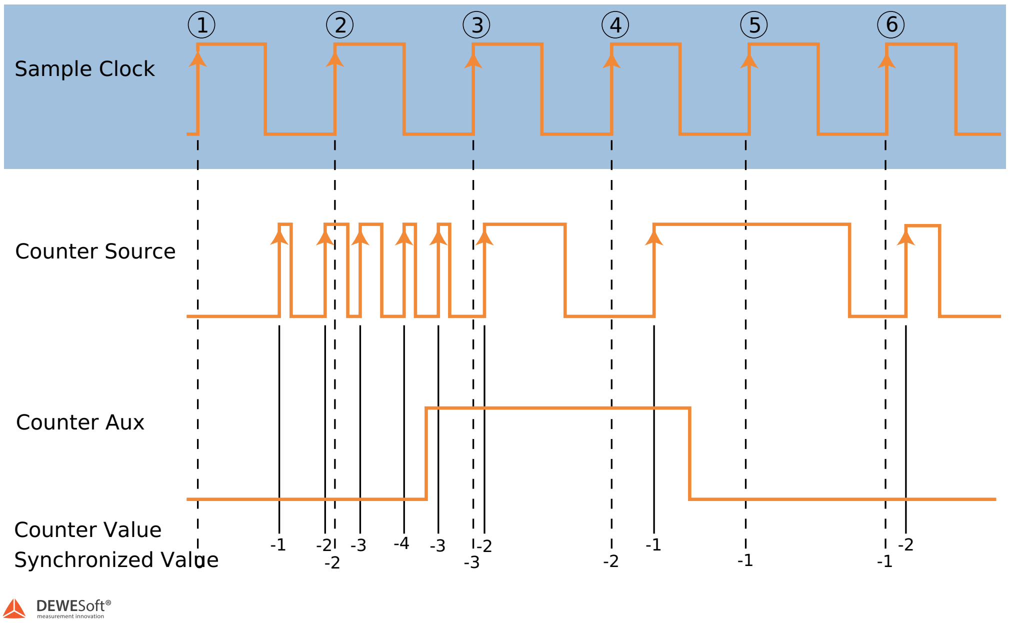


Image 20: Up/down counting mode

The connection is the same as in the previous example - gated event counting. We should choose Up/Down counting as Counting mode in the Channel setup and select the Signal input (CNT_IN0)and the Signal up/down signal (CNT_IN1).

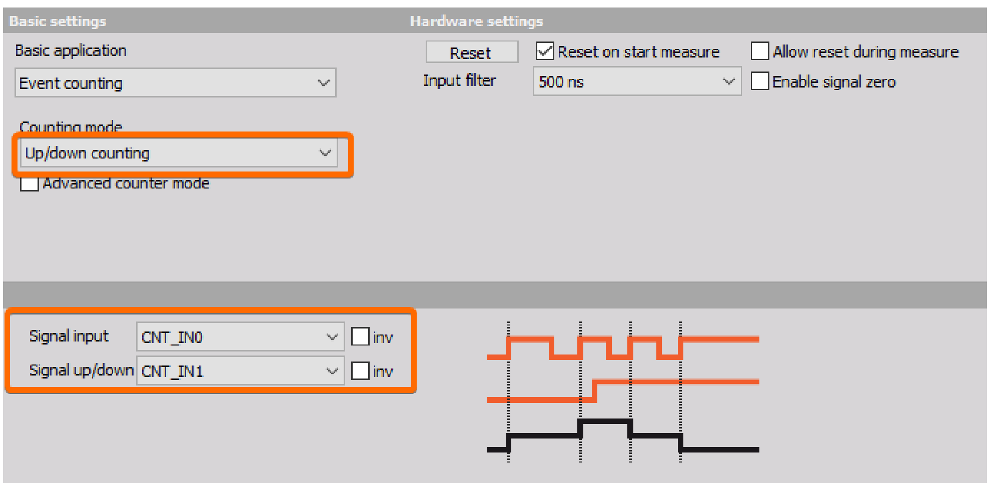


Image 21: Up/down counting setup

On the measurement, we can see that when the gate (red signal) is down, the counter counts in the negative direction (blue signal). When the value of the gate is high, the counter counts in a positive direction.

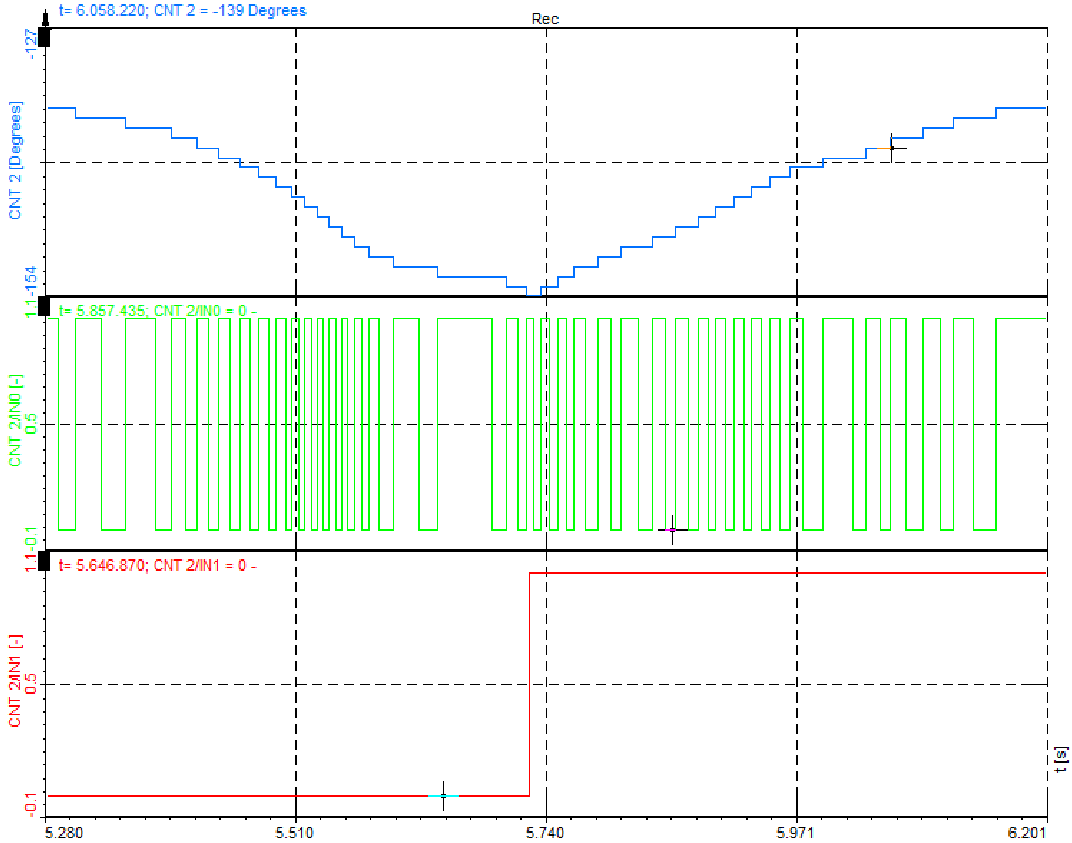


Image 22: An example of measurement with up/down counting

Basic encoder counting and X1, X2 and X4 modes

The simple encoder mode can be selected by choosing the Event counting and Basic encoder counting.

First, let's set up the encoder. Signal A is CNT_IN0 and Signal B is CNT_IN1. We set the Encoder mode to X1 mode and we set the Input filter to match our highest frequency. vScaling is easy - simply select counts, revs, or degrees from the drop-down menu or enter any other scale factor.

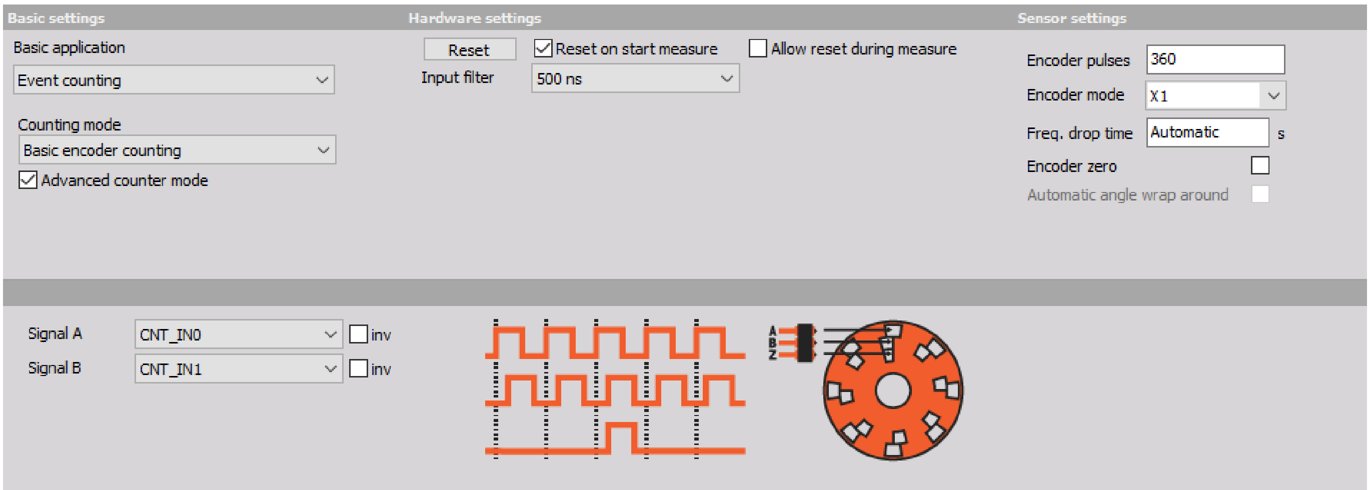


Image 23: Basic encoder mode setup

Now let's take some measurements. This counter counts up when signal A leads signal B and counts down when signal B leads signal A. The positive edges of signal A are used to calculate the counts.

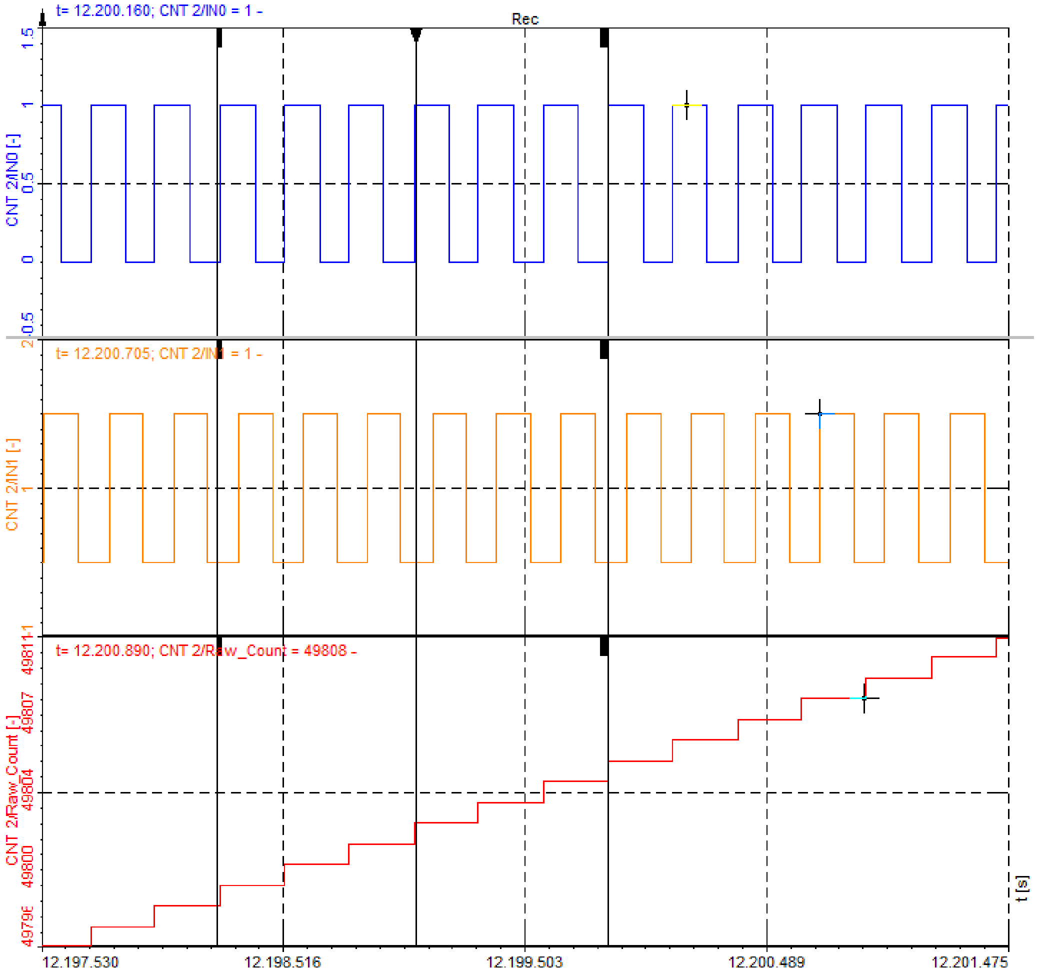


Image 24: An example with X1 mode

If we choose X2 mode in the setup, the counter will count rising and falling edges of source A and therefore the resolution will be increased by a factor of 2. Everything else stays the same.

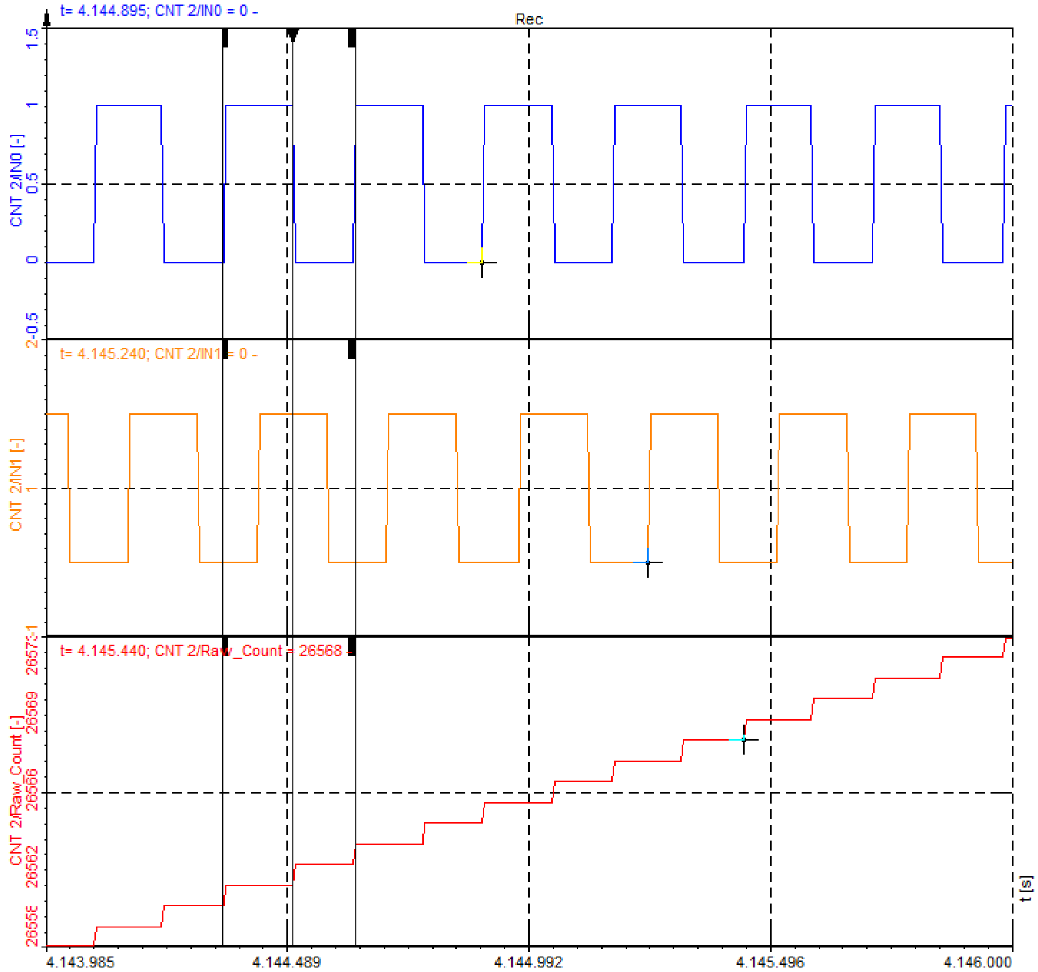


Image 25: An example with X2 mode

X4 mode counts the rising and falling edges of signal A as well as signal B. The resolution of the measurements is, therefore, increased by a factor of 4.

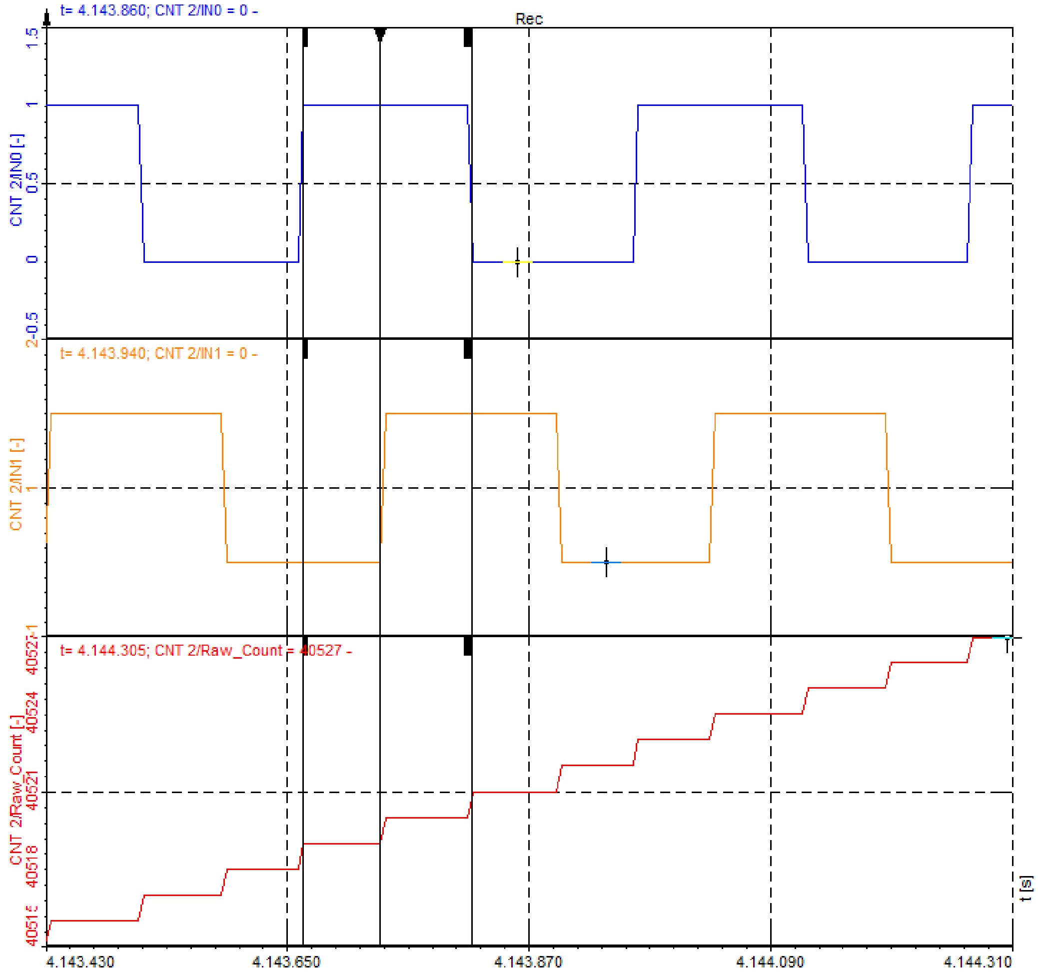


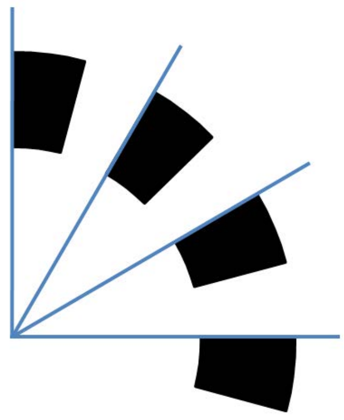
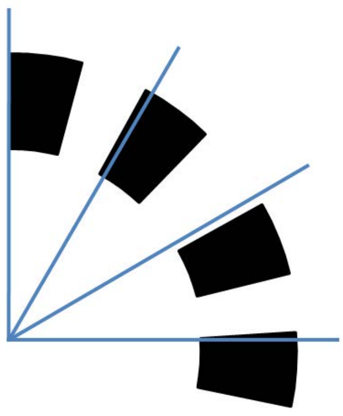
Image 26: An example with X4 mode

Sensor accuracy

An additional error occurs due to mechanical tolerances of the used encoder. Depending on the encoder mode setting used (X1, X2, X4), they have an influence on the measurement.

X1 mode

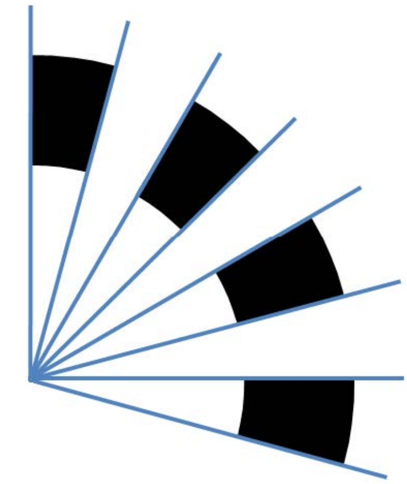
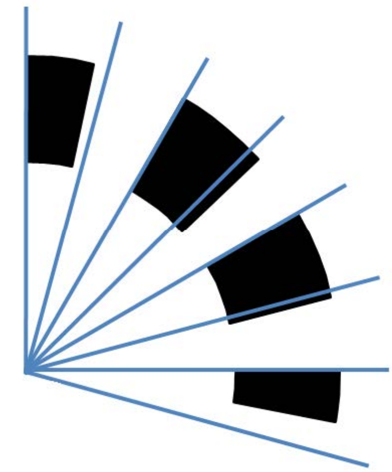
For X1 mode, only the rising edge is important. If the sensor marks are not precisely repeating at constant delta angles (have a constant jitter), this also can be compensated with the reference curve option.

Good:	Bad:
 Image 27: An example of a good sensor X1 mode	 Image 28: An example of a bad sensor X1 mode

X2 mode

For X2 mode both the rising and falling edge of the first encoder track is used, which doubles the resolution. But if the duty cycle is not exactly 50%, another error is introduced.

Good:	Bad:

 Image 29: An example of a good sensor X2 mode	 Image 30: An example of a bad sensor X2 mode

X4 mode

Both falling and rising edges of both encoder tracks are used in X4 mode to get 4 times the precision. The phase shift between the two tracks must be exactly 90Â° and the duty cycle 50%.

Good:	**Bad:**

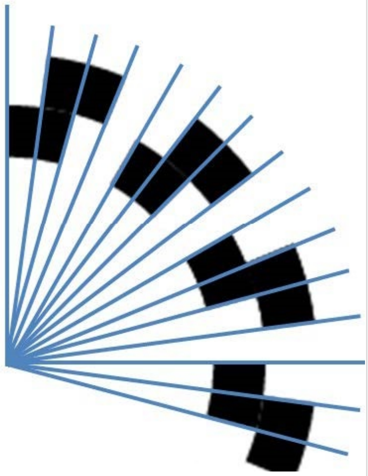


Image 31: An example of a good sensor X4 mode

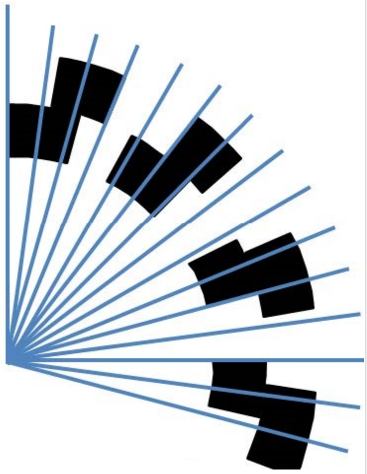


Image 32: An example of a bad sensor X4 mode

The higher the mode used (e.g. X4 compared to X1), the more noise will be in the measurement due to the discussed mechanical tolerances and because all the effects appear together. In a manner of speaking an encoder with a high resolution (e.g. 3600 pulses), is difficult to manufacture precisely, and therefore will have more noise in X4 mode than one with lower resolution (e.g. 360 pulses) in X4 mode.

Of course, if using two encoders (as in torsional vibration), the errors are doubled.

Encoder with zero pulse

The zero pulse is used to reset a measurement when a Z pulse is recognized.

The only change to the setup is to check the Encoder zero checkbox. This will reset the counter value to 0 when a zero pulse is passed.

IMPORTANT: The counter value is reset only when:
- A signal is High AND
- B signal is High AND
- Z signal is High

These signals can also be inverted in Dewesoft.

We also need to set the number of Encoder pulses for internal calculations (360 in this case).

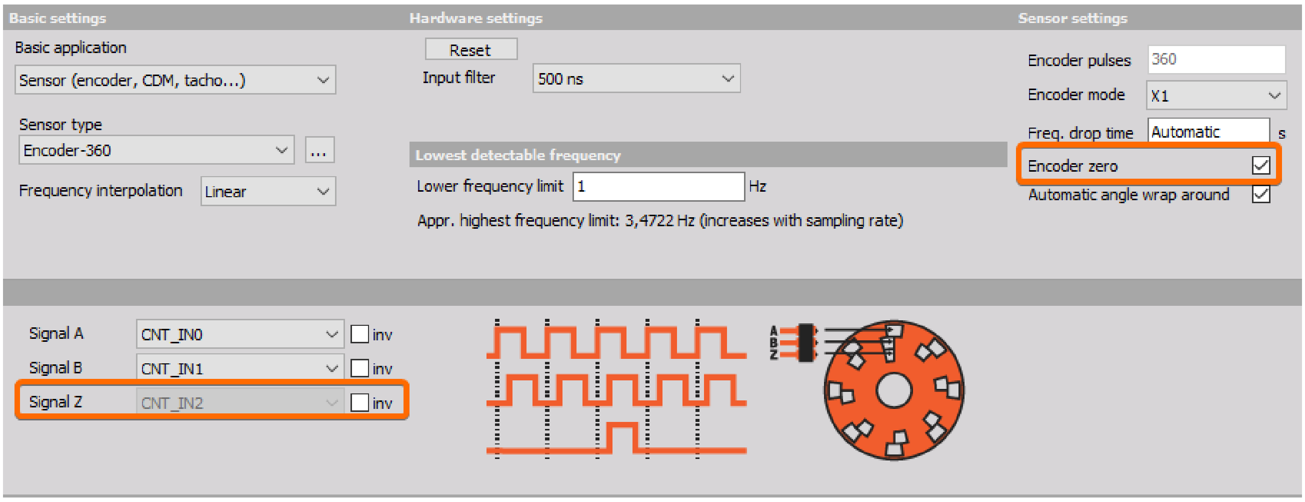


Image 33: Encoder zero option in Dewesoft

The image below shows the operation. The blue curve is the zero signal, and the magenta and orange curves are the encoder output. When a pulse is detected on the zero pulse input, the counter value resets to 0. The red curve shows the counter value.

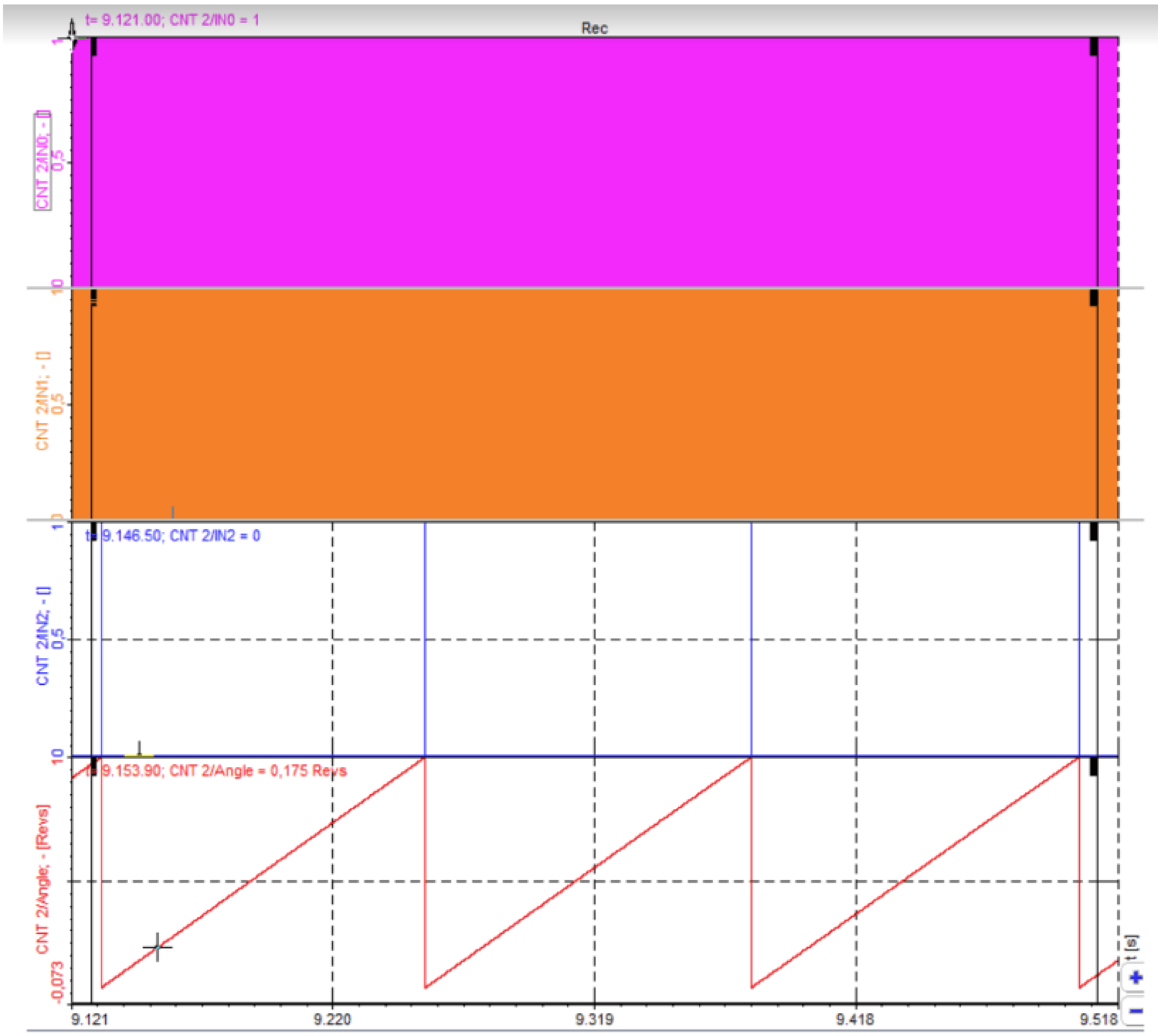


Image 34: An example with encoder zero

The image below is of a zoomed region of the recorder. It shows that the encoder resets the value of the zero pulses and continues to count up on the rising edges of the A signal.

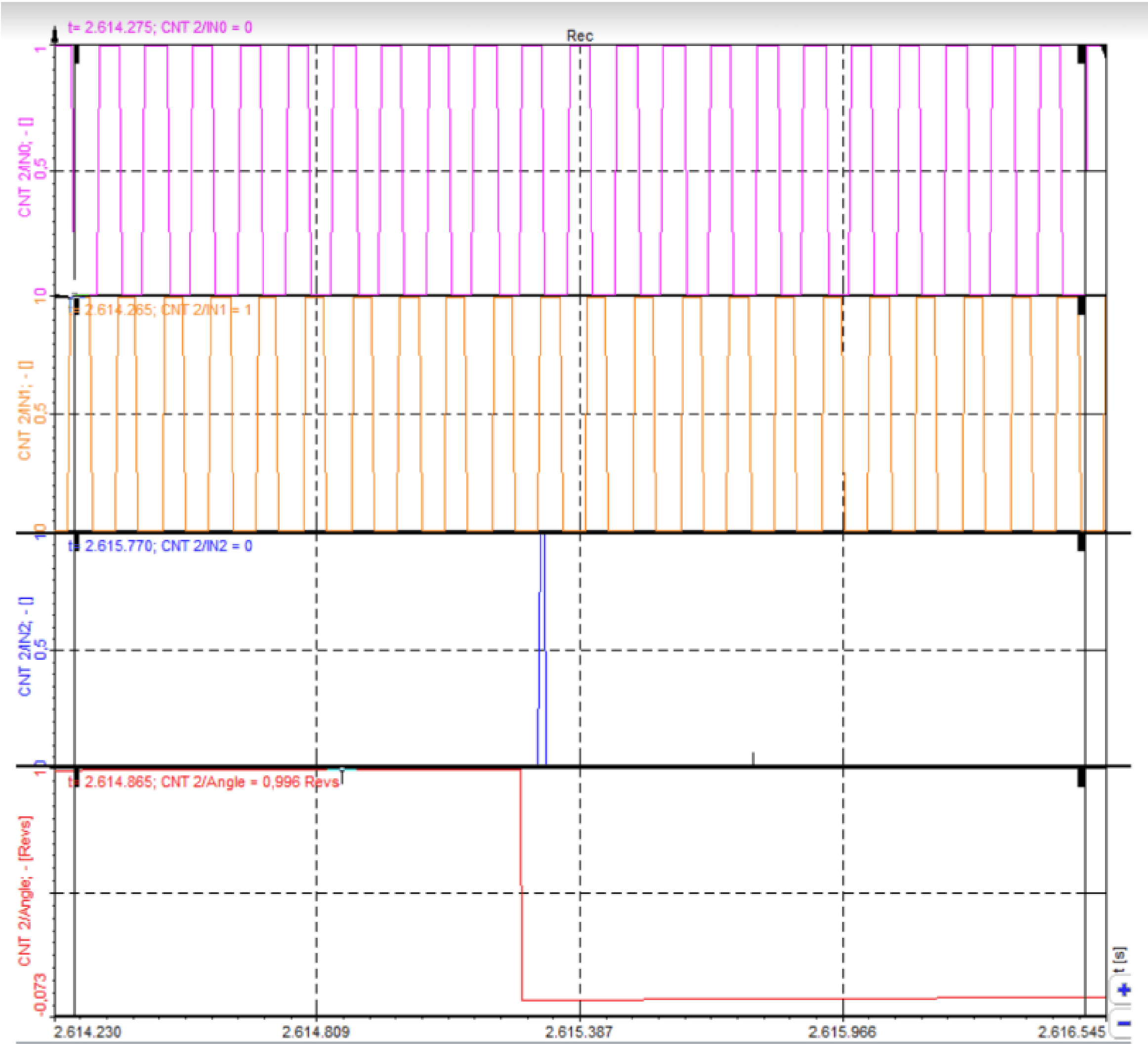


Image 35: Operation of the encoder zero option

Example of a defective encoder

The encoder fatigue life gets shortened by non-centric mounting due to vibrations.

Usually, the Z reset pulse appears when A and B signal are high; in the case shown below we see that the Z pulse stays high for more than one period, so it is faulty and the encoder must be replaced. Usually, you would recognize this by spikes in the frequency signal. To detect, please sample the raw A, B, Z signals with the highest sampling rate (e.g. 200 kHz), while slowly turning the machine.

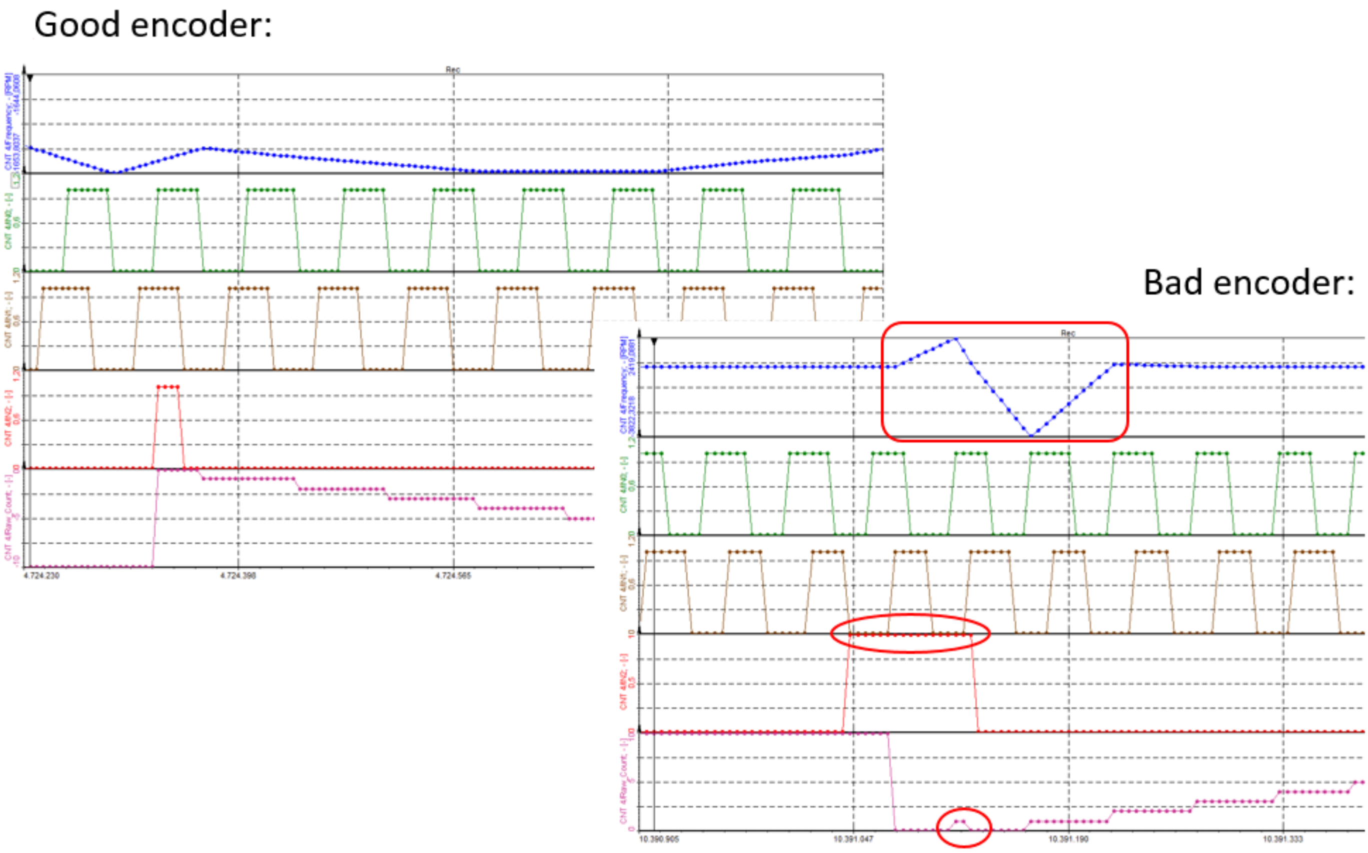


Image 36: An example of a bad encoder with too long Z pulse

Waveform timing

When choosing Waveform timing, timing mode is Period, pulse width, duty cycle.

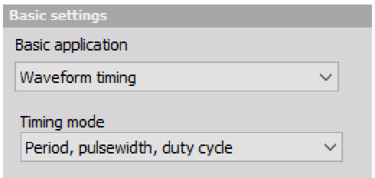


Image 37: Waveform timing application

Period and pulse-width measurements are similar in function. The period measurement measures the time between two consecutive low to high transitions, while the pulse-width measurement measures the time that the signal is high. The duty cycle is a procedure where the ratio between the high (or low) pulse of the signal and the period is measured.

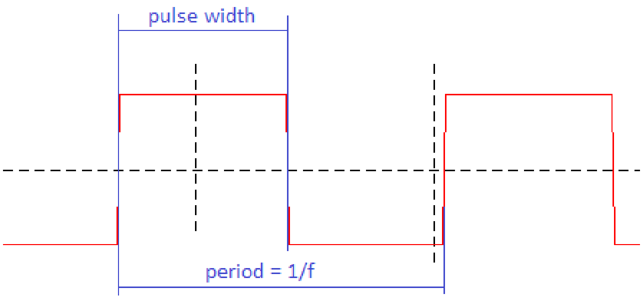
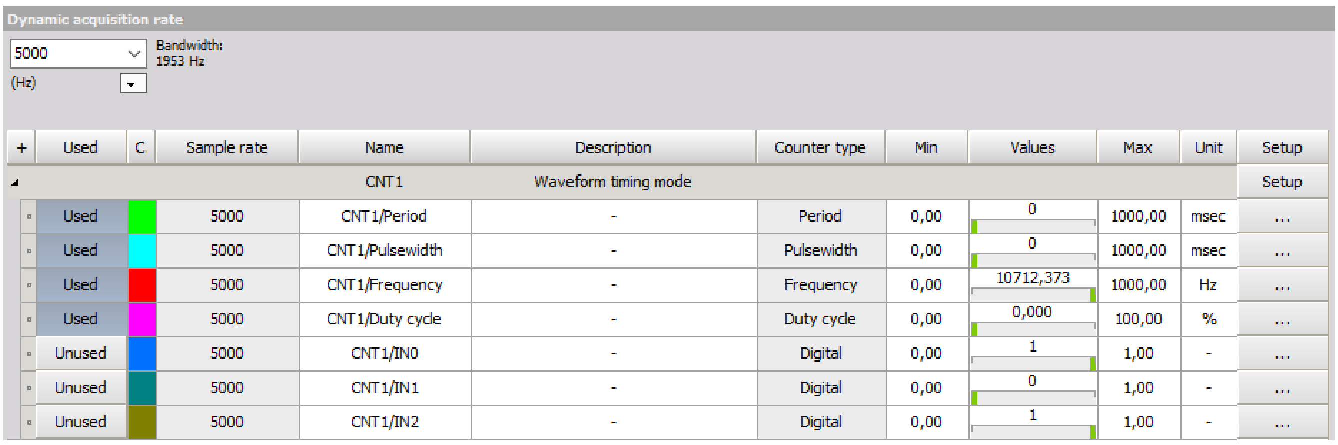


Image 38: Pulse width and period measurement

REQUIRED HARDWARE	DEWE-43, SIRIUS ACC+, MULTI
REQUIRED SOFTWARE	DEWESOFT X
SETUP SAMPLE RATE	AT LEAST 1 kHz

We will use the same configuration with two buttons for this measurement. The hardware connection is simple - we only connect one switch to the CNT_IN0.



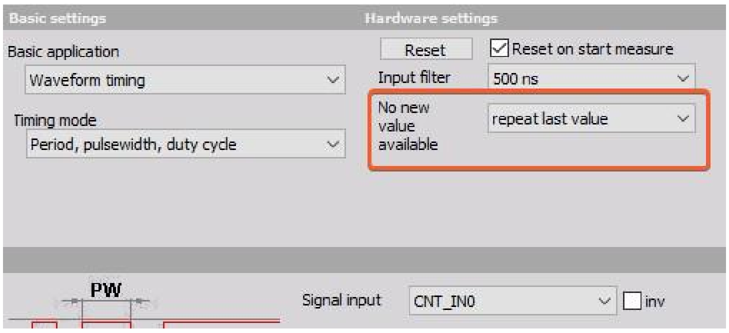
Dynamic acquisition rate

5000 (Hz) Bandwidth: 1953 Hz

+	Used	C	Sample rate	Name	Description	Counter type	Min	Values	Max	Unit	Setup
				CNT1	Waveform timing mode						Setup
	Used		5000	CNT1/Period	-	Period	0,00	0	1000,00	msec	...
	Used		5000	CNT1/Pulsewidth	-	Pulsewidth	0,00	0	1000,00	msec	...
	Used		5000	CNT1/Frequency	-	Frequency	0,00	10712,373	1000,00	Hz	...
	Used		5000	CNT1/Duty cycle	-	Duty cycle	0,00	0,000	100,00	%	...
	Unused		5000	CNT1/IN0	-	Digital	0,00	1	1,00	-	...
	Unused		5000	CNT1/IN1	-	Digital	0,00	0	1,00	-	...
	Unused		5000	CNT1/IN2	-	Digital	0,00	1	1,00	-	...

Image 39: Setup for waveform timing.

IMPORTANT: due to FPGA limitations DEWE-43 does not support the â€Repeat last valueâ€ option. This functionality is exclusive to SIRIUS units.


Basic settings
Hardware settings
Basic application
Waveform timing
Timing mode
Period, pulsewidth, duty cycle
Reset
Reset on start measure
Input filter
500 ns
No new value available
repeat last value
PW
Signal input
CNT_IN0
inv

Period measurements

First, we select the Basic application as Waveform timing and Timing mode as Period, pulse width, duty cycle.

We set the Signal input to CNT_IN0 and invert the signal, by checking the inv checkbox, so it is normally low; we can also set the signal Input filter to prevent glitches.

The new value is calculated only when a signal changes the value from low to high. Therefore, the value can't be calculated most of the time. We can select to output zero value when no new value is available, so we will have only spikes at the points of new data and the rest of the data the value will be zero.

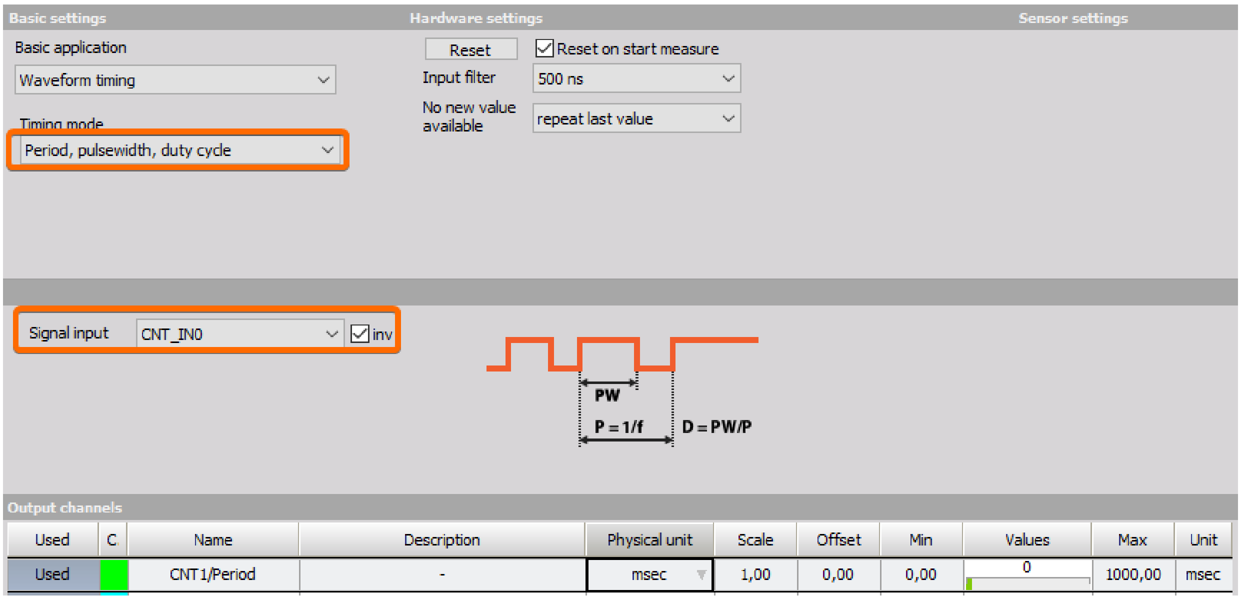


Image 40: Period setup

The example below shows how the measurement is performed. The two white cursors in the recorder show the time difference of the two pulses (it is shown on the left on the recorder setup screen). It reads 201.8 microseconds. The counter value is, of course, much more exact, as it shows 202.7 microseconds. Since the counters are running with an 80 MHz clock, we have a microsecond resolution. Out of the period, the counter also calculates a frequency, which is a simple 1/period. We can set the units of period values and frequency in the channel list on the setup screen.

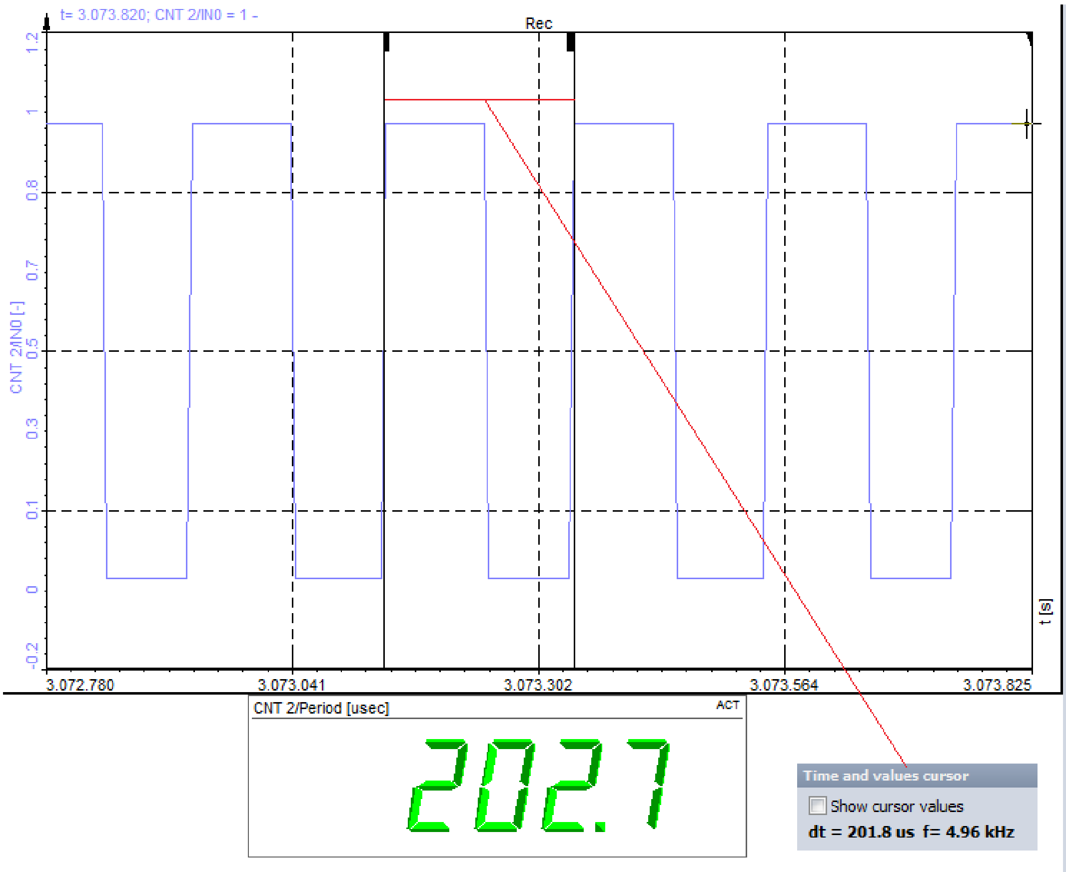


Image 41: An example of a period measurement

Pulse-width measurements

The pulse-width measurement setup is the same as what was previously described in the period measurement sample. We select the Period, pulse width, duty cycle option in the Timing mode section.

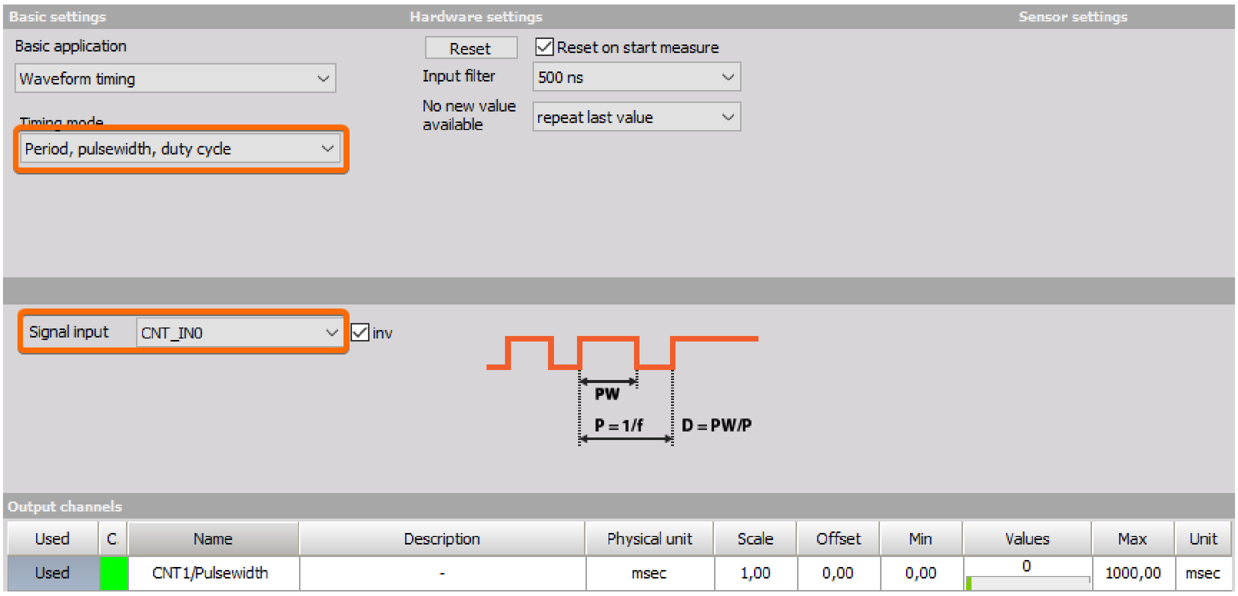


Image 42: Pulsewidth setup

The readout is updated on each high-to-low transition. The recorder cursor readout shows 98.8 microseconds and the counter value shows 0.1029 milliseconds.

We can use the super-counter to measure the low and high time of each counter or to combine the period and pulse width to measure the duty cycle of the signal.

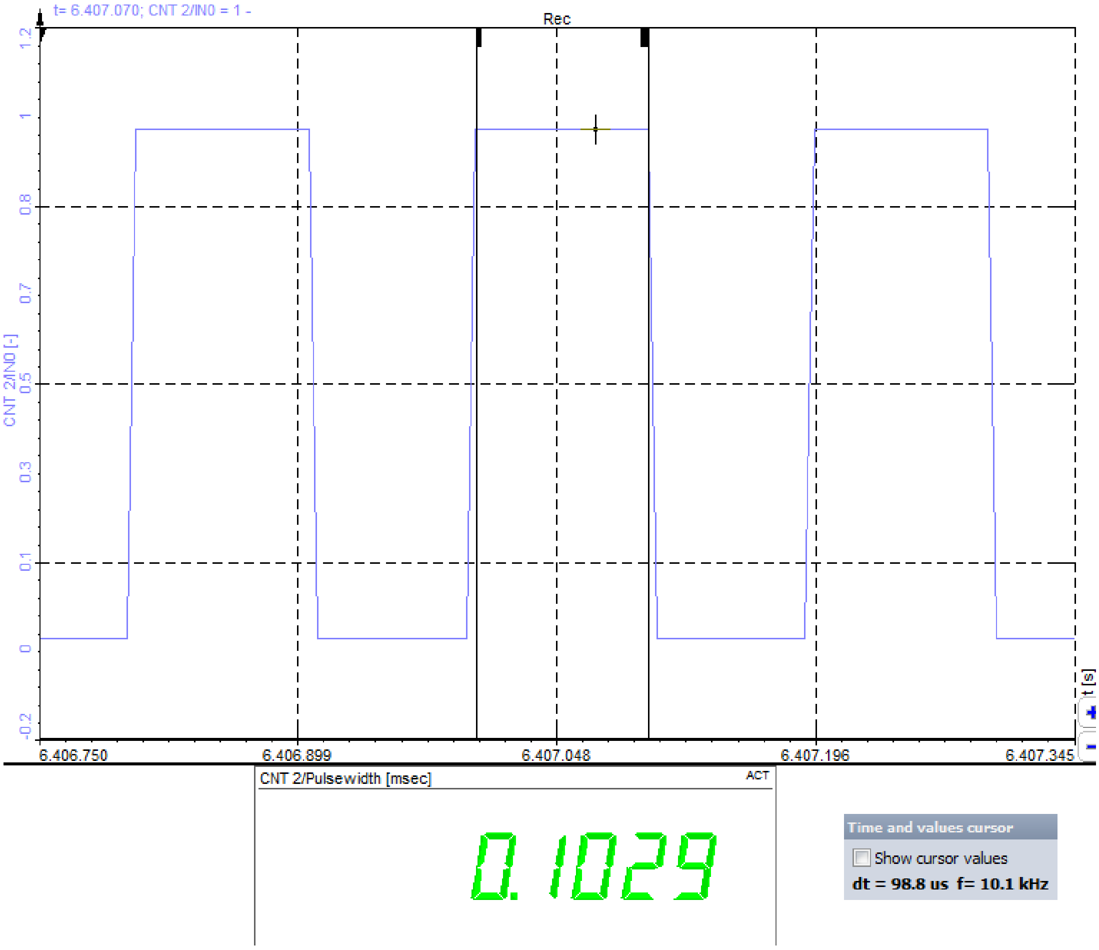


Image 43: Pulsewidth measurement

Duty cycle

The duty cycle measurement is a procedure where the ratio between the high (or low) pulse of the signal and the period is measured.

As was previously stated in the encoder tutorial, the decision to use X1, X2, or X4 mode depends on the quality of the encoder and the encoder electronics. The same encoder, as was used for the Encoder tutorial, will be used to measure its quality. For this measurement, we need to set the Period, pulse width, duty cycle mode as Timing mode. If this mode is not available, then the counter does not support it.

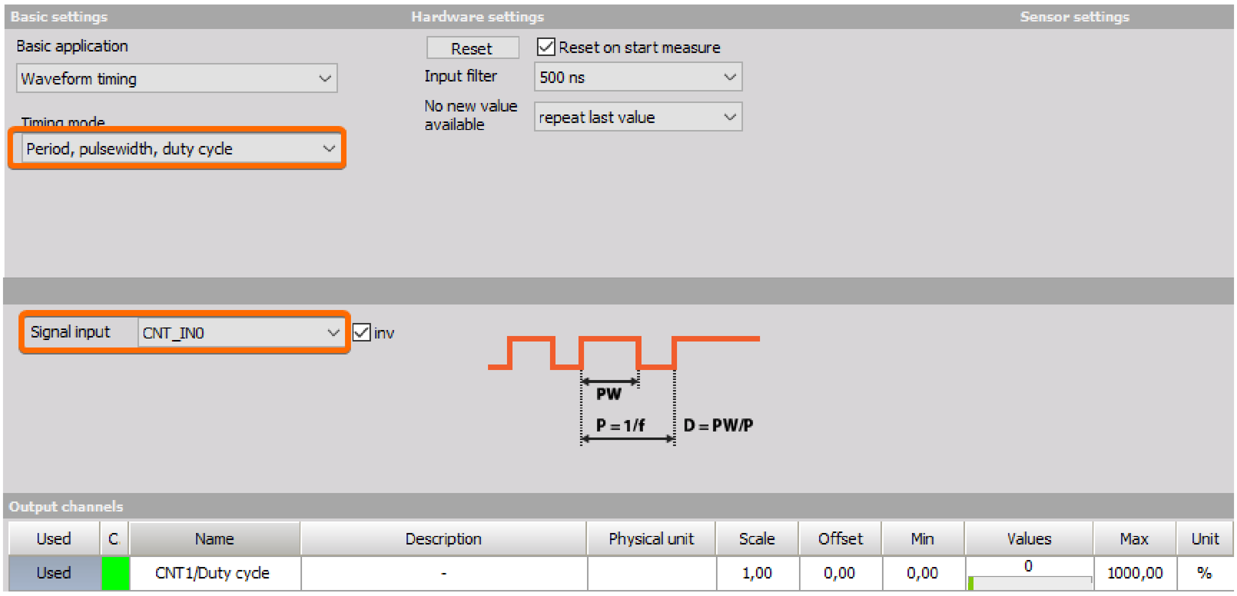


Image 44: Duty cycle setup

Then we can directly select the Period, pulse width, duty cycle, and also the frequency output channels. The counters are set automatically for this operation.

Now let's look at the duty cycle measurements. The upper graph in the picture below shows the period and pulse-width of the signals. In the lower graph, we can see the duty cycle for the few rotations of the encoder. We can observe nicely that there are some points where the encoder has a slightly larger error than in the rest of the data. The value there is approximately 51.978 %, so this means that the encoder mode X2 will have around a 2% of error.

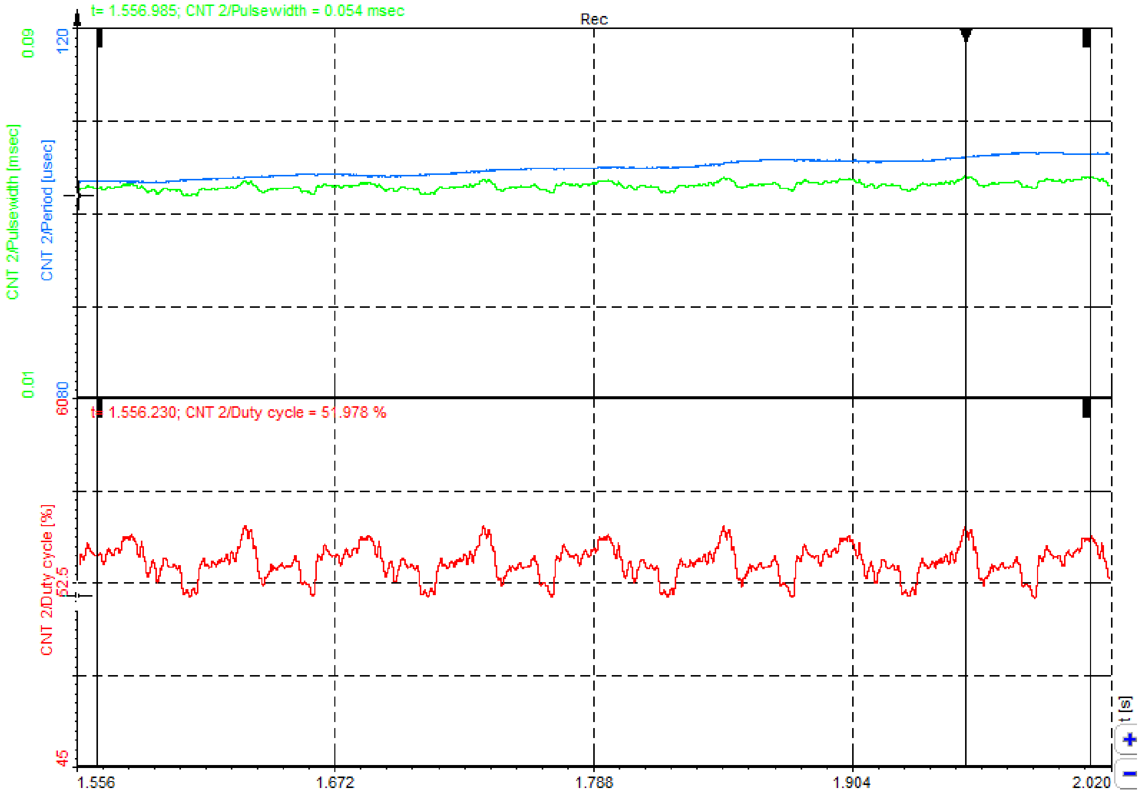


Image 45: Duty cycle measurement

Sensor measurement

REQUIRED HARDWARE	DEWE-43, SIRIUS ACC+, MULTI
REQUIRED SOFTWARE	DEWESOFT X
SETUP SAMPLE RATE	AT LEAST 1kHz

The SuperCounter mode is also used in a special counter mode, called "Sensor" mode (selected from the Basic application drop-down menu). This mode allows the direct use of the digital speed/position sensors as defined in the Counter sensor editor. You can choose rotary encoders, linear encoders, CDM sensors (angle sensors with zero reference), gear tooth with missing or double teeth, and tacho probes.

The only thing needed is to select an appropriate sensor from the Sensor Type drop-down menu. If the sensor is not yet defined, there is a three ellipsis button on the right side which opens the counter sensor editor. This is where sensors can be defined. The sensors will always run in the SuperCounter mode, showing the exact frequency and angle.

The benefit of using sensors is that scaling will be done automatically, so we don't have to worry about that anymore. There are still several options to choose from. For the encoder, we can select the Encoder mode (X1, X2, or X4) and either use the Encoder zero or not. IMAGE 46

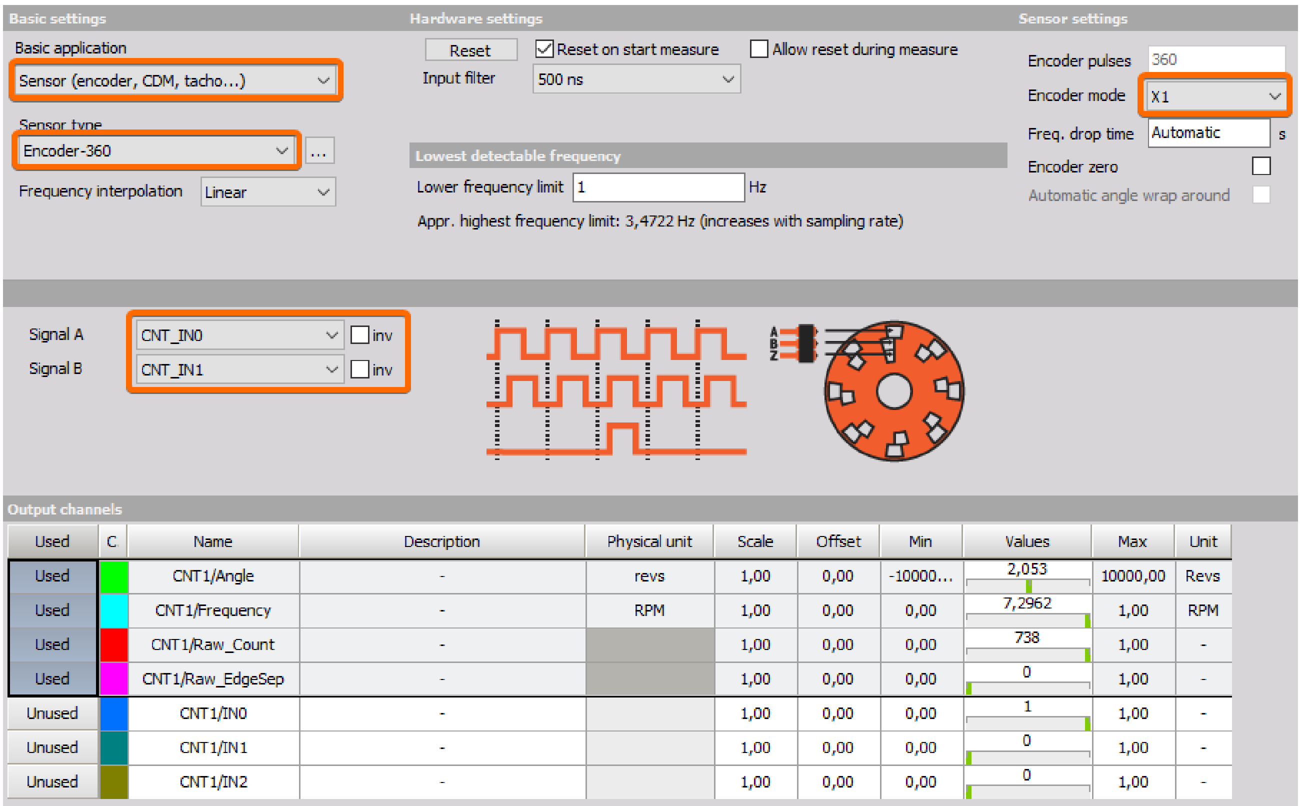


Image 46: Sensor mode settings

The CDM, tacho, and gear tooth have no special settings, so they will depend only on how the sensors are set up in the sensor definition.

DS TACHO 4 - tape sensor

The DS-TACHO4 sensor is a threshold sensor. This is important especially for the proximity detection mode, the most commonly used for rotating: working distance could change with the albedo and/or the form and distance of the target, also, the contrast appears as an important parameter: teeth-no teeth, black and white marks. The recommended distance for encoding application is a few millimeters: put the probe closed to the target to avoid an incorrect reading resulting from rocking and wagging of the turning part (Descartes optical law); on the other hand, the reflective tape allows for much more than 100 mm. It is highly recommended that you use the adhesives encoders for optimal results.

A few phenomena may affect the detection function, such as a drop of liquid on top of the probe, excessive dust covering the top, more generally, a non-transparent environment for our light source such as diesel engine sump film (i.e. carbon is not transparent for the near I.R.). The patented concept implemented in the sensors strongly simplifies mounting and set-ups. Prior to measurement, it is recommended that a detection test is performed, even at low speed, to ensure detection feasibility and determine the detection distance required for the sensor.

If impossible to perform a test due to technical reason or mounting specifics, a theoretical method would be to fix the probe at a distance equivalent to the width of the black and width strips to detect- in any event, without exceeding 4 mm.

Applications:

- Sensor recommended for acyclism, torsional vibrations
- Test bench and embedded measurement
- Measurement on rotating machines: combustion engine, electric and hybrid motors, hydrogen, turbine

Fixing and support of the probe will influence the acquisition of the reading. Please be careful regarding vibration. We recommend that you design your supports including appropriate vibration orders studies. The further the probe will be away from the target, the more the TTL amplitude signal will decrease.

Image 47: DS-TACHO 4 tape sensor

Mounting the probe

- Ensure that you have all items required at your disposal, i.e. the sensor, the probe, and the two hand-pieces for optical fixation
- Put the two hand-pieces down if they are on the optical head of the sensor
- Insert the two optical fibers with their respective rivets
- Screw the first hand-piece on and tighten moderately; a little gap between the rivet head and the optical head is normal
- Remove the two fibers in order to allow for mounting of the second hand-piece
- Make sure that the two fibers and their rivets are assembled correctly
- Hold both probe and sensor simultaneously when inserting the rubber sleeve to avoid damaging the two optical fibers on the level of the rivets.

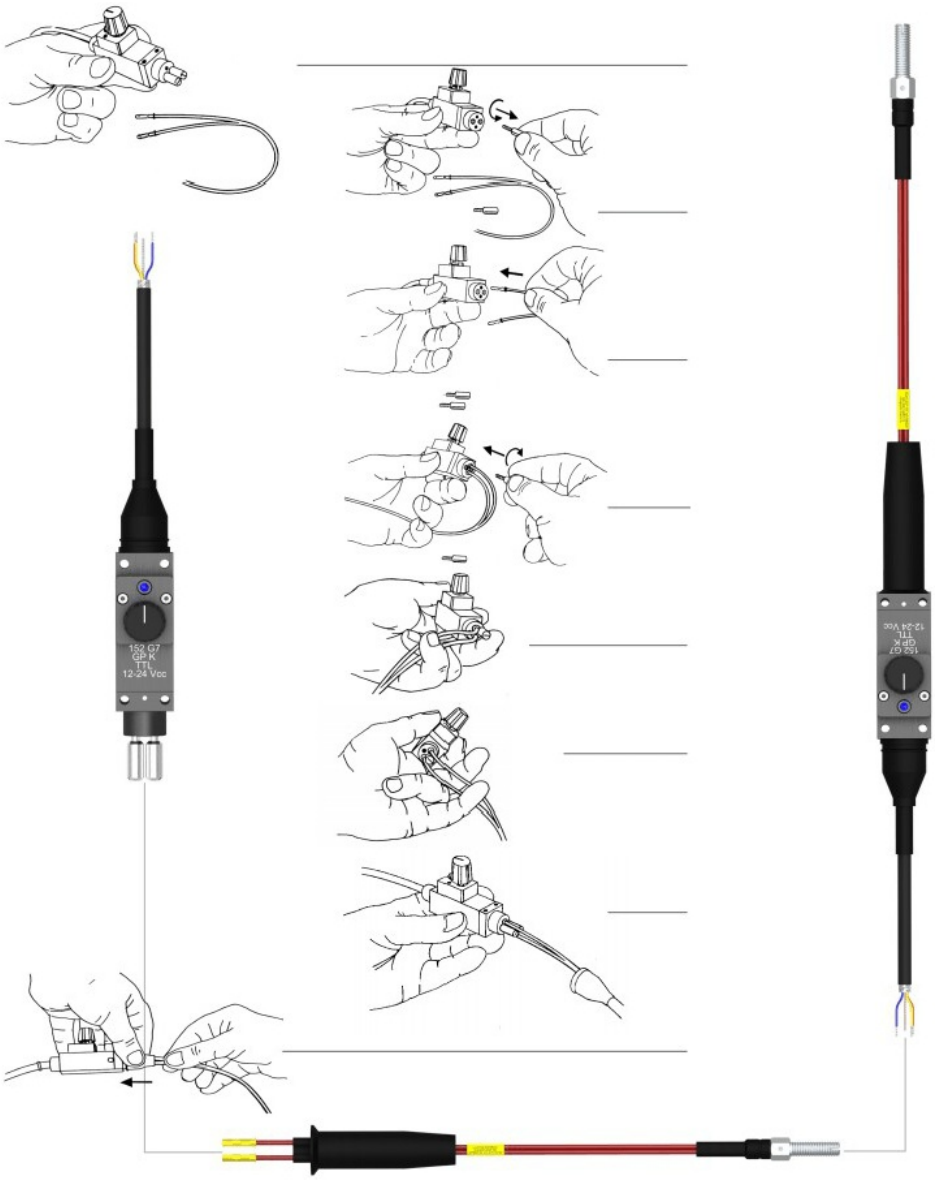


Image 48: Mounting the probe

Adjusting the probes

The operational mode of the sensor can be seen at the end of the optical fiber by a light beam (not dangerous), which is emitted when the sensor is in 1 mode and not emitted when the sensor is in 0 mode. The sensor keeps its wavelength in near Infra-Red to ensure the power and immunity of the detection function. This also gives an indication of the condition of the optical fiber.

The sensor should be placed about 2 to 5mm above the tape. A sensitivity potentiometer is available to adjust the trigger level for reliable pulse output.

First turn the potentiometer in mid position. Bring the probe closer to the target until the indicator at the headlights up, targeting the white mark. Shift the probe, and repeat this operation in order to detect the triggering limits on the black marks of the target. Set up the probe in an average position (length), review this operation to confirm the accurate detection: the set up is finished.

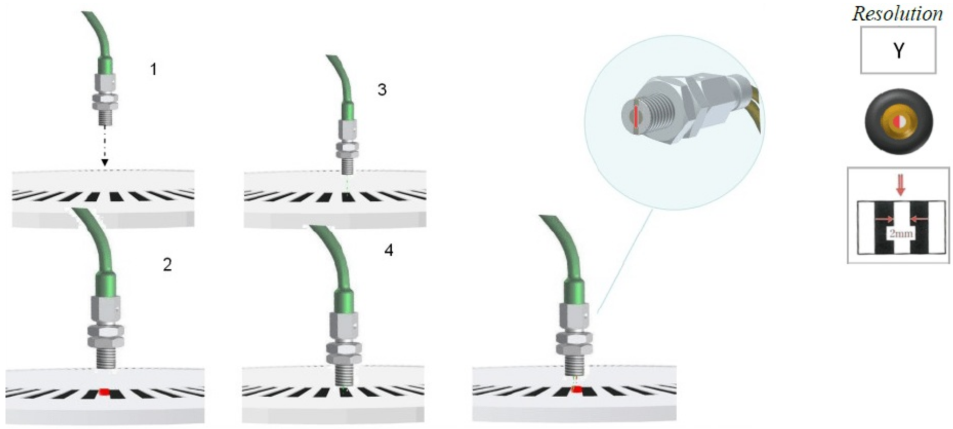


Image 49: The placement of the sensor

Automatic gap detection

When applying the black/white tape to the rotating shaft there will be irregular rasterization at the transition point. This can be used as the zero pulse to indicate a defined start position. On the other hand, this would result in an rpm drop or spike in our rpm measurement.

A software procedure automatically measures the pulses per revolution and also detects the exact gap length to enable robust and high-quality measurement.

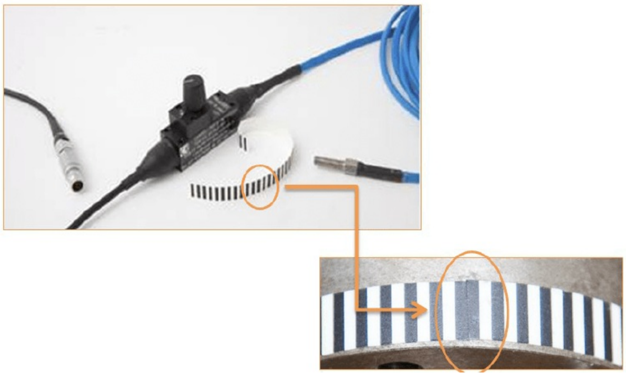


Image 50: Gap length

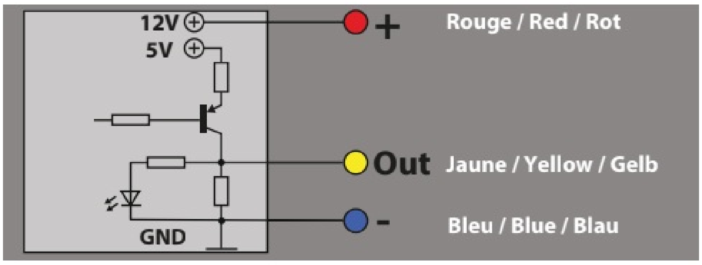
The zero pulse must be at least 3 pulses long!

Sensor set-up

The power supply must be perfectly rectified, filtered, and constantly deliver more than 120mA/12V. This is not an open collector output sensor, but PNP output. 152 G7 can support reverse tension, this tension modifies the signal's Amplitude. 152 G7 TTL Voltage output is 5 Vcc, 152 G7 Voltage output is nominal voltage input -1.5Vcc. If the sensor is connected to the acquisition system the use of dedicated measurement connectors and matching cables is recommended. Please refrain from extending the cable. Otherwise, the sensor's operation may be affected. To confirm that the sensor is live, check if a faint red LED glows on the small light channel in front of the sensor optical head; You can also use a digital camera to see the I.R. Light. The brightness of this small red light is independent of the position of the potentiometer.

Sensor plug-in

- V rating: 12/24 Vcc
- V Min: 10 Vcc
- V Max: 30 Vcc
- I: 120 mA/12Vcc



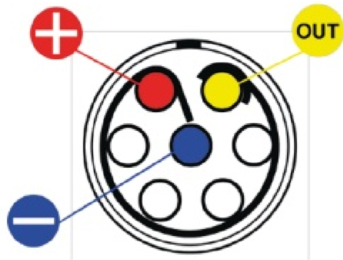


Image 51: Connection of the sensor

Electrical specifications

ELECTRICAL SPECIFICATIONS	
Transmitters	LED near IR+ visible
Supply voltage	From 9 to 30Vcc
Average current consumption (12Vcc)	< 50 mA
Possible inrush current / Duration	>1 Amp. / Less than 10 microsec.
Receiver (sensible to the light)	High-speed Photodiode
Switching Frequency	0 < **F** < 260 kHz
Switching time (10 - 90%)	Rise time: 50 nanosec. Fall time ≤ 0.5 microsec.
Sensor Setting	¾ turn Potentiometer
Connection	1,5 m cable+ Plug & Play Connector
Standard Output	PNP TTL
Accessories Output	Additionnal TTL via SMA / On specification
Output control	Blue LED + top of fibre visible light patented concept
Amplifier Operating temperature	From -10 °C To +70°C
Protection	Temporary short-cut

Image 52: Specifications of the sensor

Lemo connector

Connector type: L1B7f

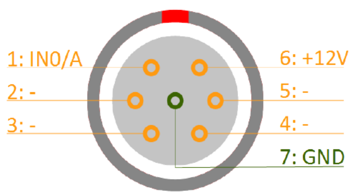


Image 53: LEMO connector

Physical diagram

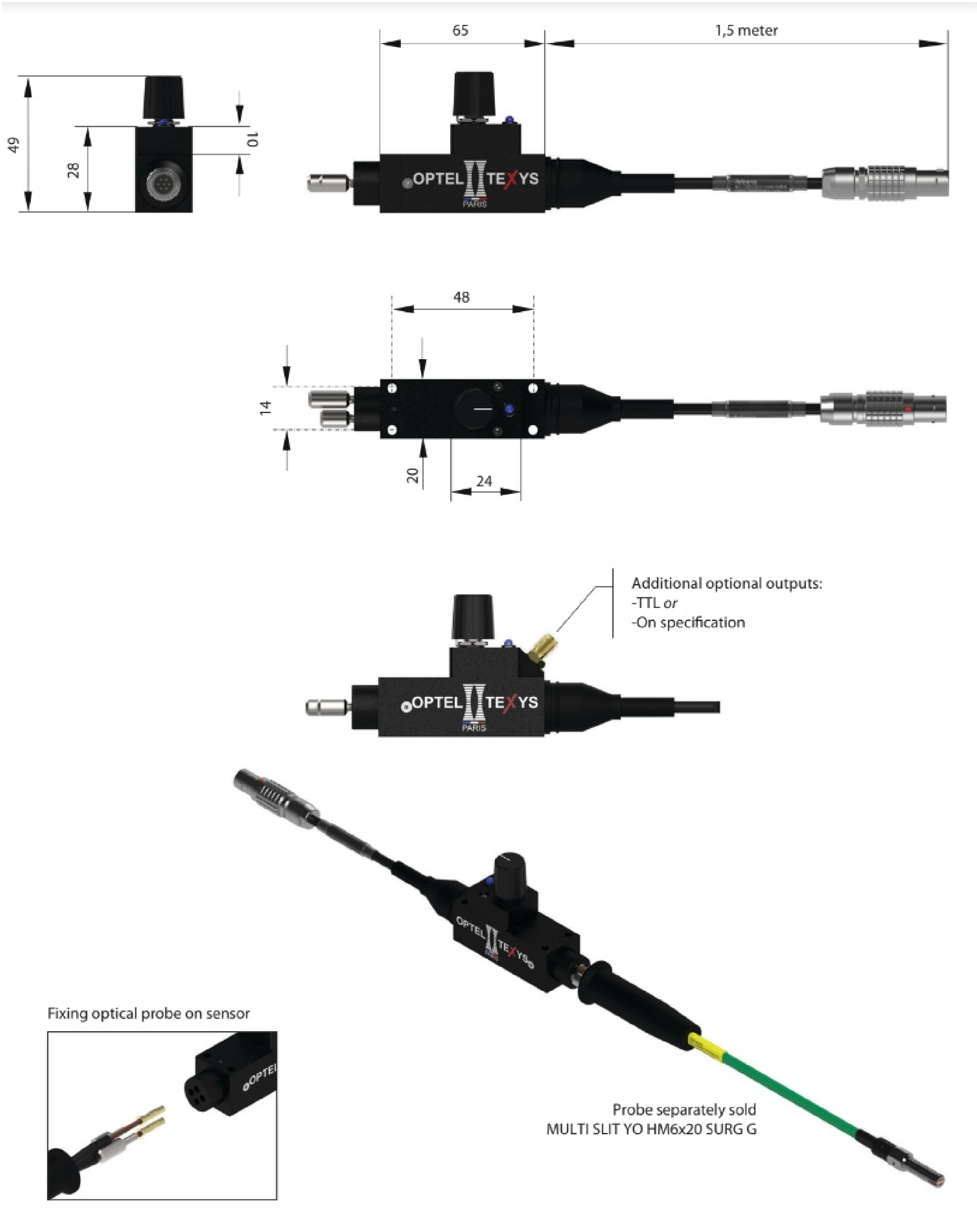


Image 54: Physical diagram of the sensor

Sample rate

We have to detect the frequency drop, so that gap is seen, and software can calculate the start and stop of the angle(0 to 360deg.) So in case, the sampling rate is lower than the input frequency of the TACHO probe the gap could be missed.

Lets assume we have about **64 pulses/rev** the machine is running with 1000 RPM. **1000rpm/60= 16rps = 16Hz**.

Per one second, we would get: **16Hz * 64pulses/rev= 1024Hz** input frequency.

In the example below the sampling rate was set to 1kHz, so we could see that the gap was not recognized at every revolution. In this case the sampling rate must be at least 2 times higher. **1024Hz *2 is about 2kHz**, because the speed of the machine could go up we also have to consider that. We should set it at least to 10kHz.

Sampling rate > Maximum input frequency * 10

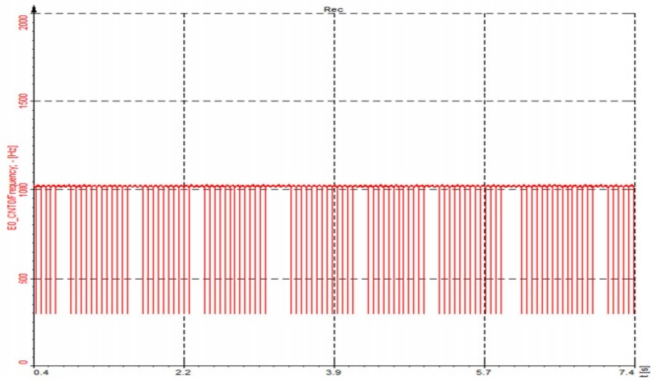


Image 55: Gaps not recognized correctly due to too low sampling rate

For measuring RPMs and angle of rotating machines, we need angle sensors. RPM and angle measurement are important in balancing, order tracking, and rotational and torsional vibration.

We need to choose an rpm sensor that is convenient for our measurement. Not all of the sensors can be installed in our rotating system and sometimes it takes a lot of effort to install them. Also, we have to choose the sensor that has a good resolution for our purpose (e.g.: a sensor with one pulse per revolution is not appropriate for measuring precise angle).

Tape sensor is an optical sensor for measuring speed and angle. It uses black and white tape that is attached to the rotating part of a machine.

Image 56: Black and white tape with the sensor

The sensor is made of optic fibers and should be placed about 5 mm (or less) above the tape. We have to use a sensitivity potentiometer to adjust the trigger level to such a level that gives us steady pulses. The reflection is then converted with an electronic circuit into a TTL signal. The sensor is connected directly to a LEMO counter input.

Image 57: Sensitivity potentiometer with the LEMO input

Tape sensor can be used in many applications:

- RPM measurement,
- angle measurement,
- order tracking,
- rotor balancing,
- rotational and torsional vibration.

Tape sensor setup

First we glue the tape (with black and white stripes) onto our rotating part. If both ends of the tape would come perfectly together we would have no zero pulses per revolution, which are an indication of the start position. If we don't have the information about the start position, the angle would be different at every start of a measurement.

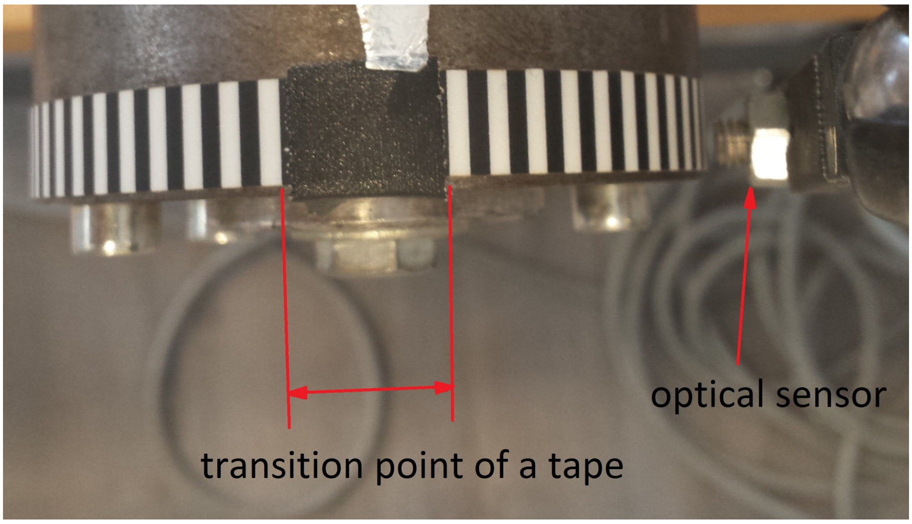


Image 58: Transition point of a tape

In the image above we can see the transition point of tape - we use that as the ZERO pulse. This is an indication of a new revolution so the angle will start all the time at this position - angle information related to the shaft will be the same.

In the image below we can see the drop in frequency where we have the zero pulses. The drop is seen nicely so we could use that to detect the ZERO pulse. The angle will always start at that position. For the software to clearly see this drop or peak, the length of the gap must be more than 3 pulses. So the software will have no problem detecting ZERO pulses because the frequency will drop by 70%.

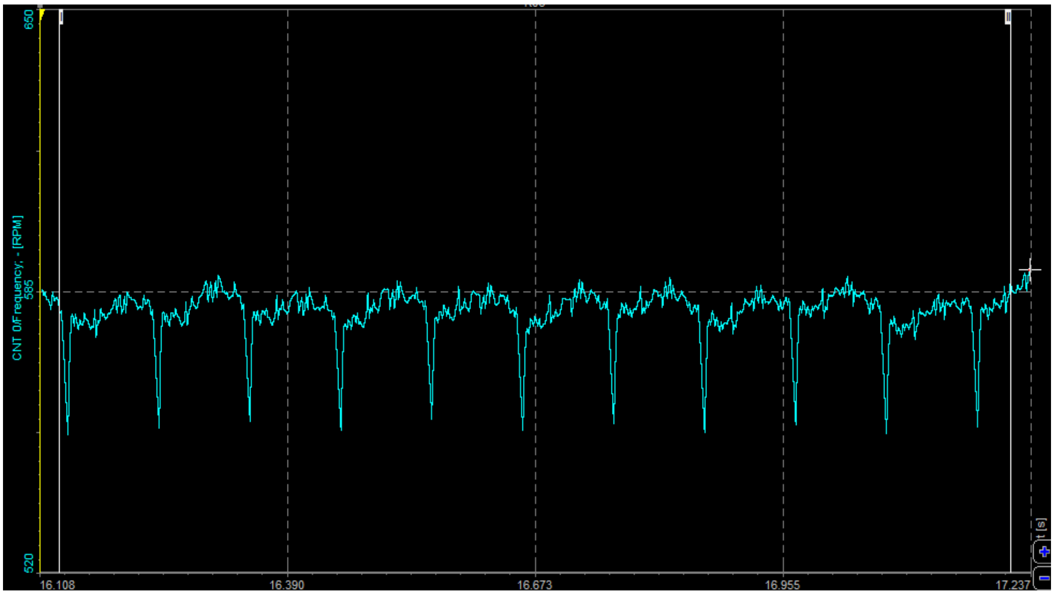


Image 59: Drop of frequency value in time-domain signal

We have to adjust the trigger levels to get reliable pulses from the optical sensor. The trigger level has to be set after the sensor is mounted because it depends on a distance to the tape.

The sample rate must be high enough to detect the frequency drop and that the gap is seen so the software can calculate a start and stop of the angle.

Example: We have 64 pulses per revolution and the machine is running at 1000 rpm - 1000rpm/60 = 16 Hz.

*Input frequency is: 16 Hz * 64 pulses/revolution = 1024 Hz. If the sample rate would be set to 1 kHz, the gap would not be recognized at every revolution.*

The sampling rate must be at least ten times higher than the maximum input frequency.

Defining sensor type

When we do an RPM measurement we have to select Sensor mode in Counter setup in Dewesoft.

When Sensor mode is selected, we select our sensor from the Counter sensor database, where different types of sensors and their setting are already stored.

If we are using a sensor that is not yet in the Counter sensor database we have to define it.

We go to Settings -> Counter sensor editor or just click on the three-dot button to enter Counter sensor editor:

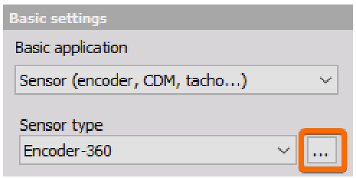


Image 60: Shortcut to Counter sensor editor

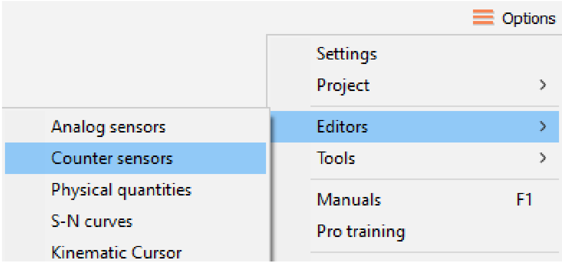


Image 61: Path to Counter sensor editor

In Counter sensor editor, we add a Tape sensor as a sensor type. I renamed it the Tape sensor. When we click Save&Exit the sensor is added to the Counter sensor database and is ready to be used.

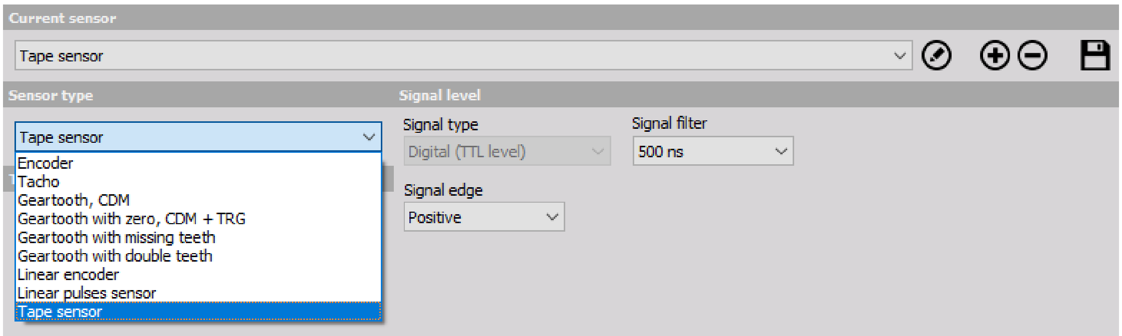


Image 62: Defining the tape sensor in Counter sensor editor

For these sensors, we need to define the number of pulses per revolution (number of lines) and a gap. The last white line defines the start of the gap. For a tape sensor with 200 white lines and a gap which has two missing white lines the number of teeth is 199 (200-1) and the gap is 3 (1 + 2 missing teeth).

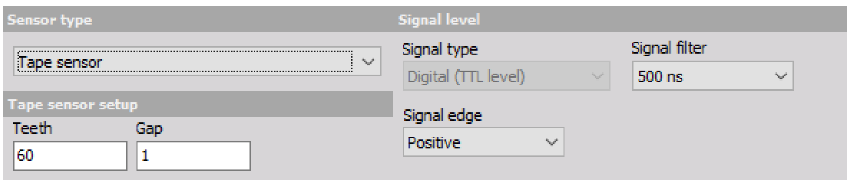


Image: Tape sensor settings

We created a tape sensor that can now be selected from the dropdown menu in the counter channel setup:

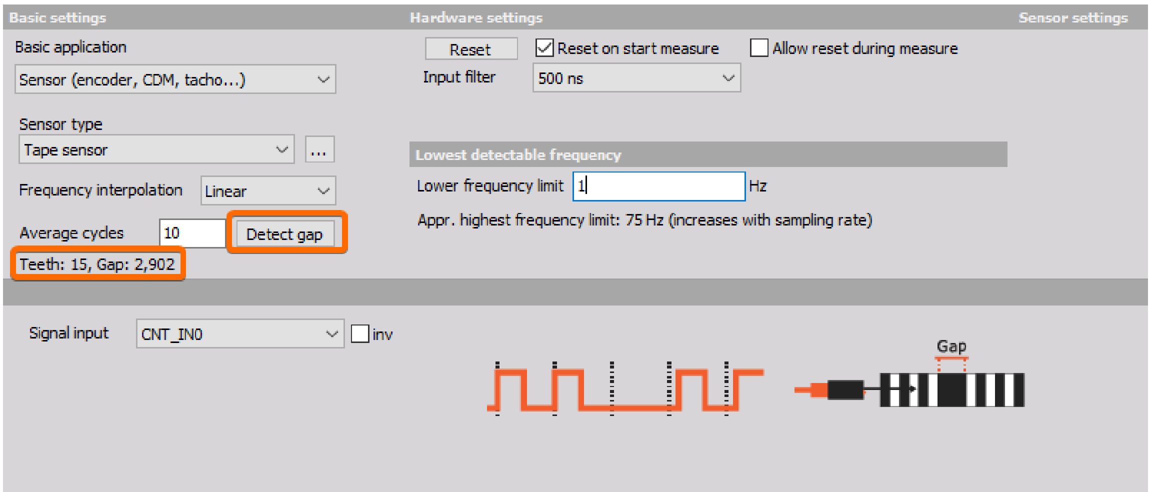


Image 63: Detecting the correct gap of a tape sensor

For a precise measurement, we have to know how many pulses per revolution we get from the tape sensor and how many pulses in the gap wide. We shouldn't count that manually, there is a function called Detect gap - it will automatically measure the pulses per revolution and detect the gap length.

For the gap length calculation the rpm should be as stable as possible. So try to operate the machine in a stable area, so that rotational vibration (rpm deviation) is as small as possible.

The algorithm will average the speed of the machine a few samples before and after the gap, so the average speed around the gap is extracted, and from that we can calculate the missing pulses.

Please be aware that we are in setup, so Dewesoft X is running with the setup sampling rate, and if that is not high enough like described in 3.4 gaps and teeth detection will not work.

Measurement results

Output channels of tape sensors are angle and frequency channels. Angle runs from 0Â° to 360Â°, frequency channel can be seen in RPMs of in Hz.

On recorder, we can see the angle in the range from 0Â° to 360Â° (when the tape is rotating angle values increases when ZERO pulse is passed, the angle value returns to 0) and frequency channel in rpm. The rpm channel (green curve) is not a straight line because our rotor was not balanced. So we can use the tape sensor for balancing rotary parts.

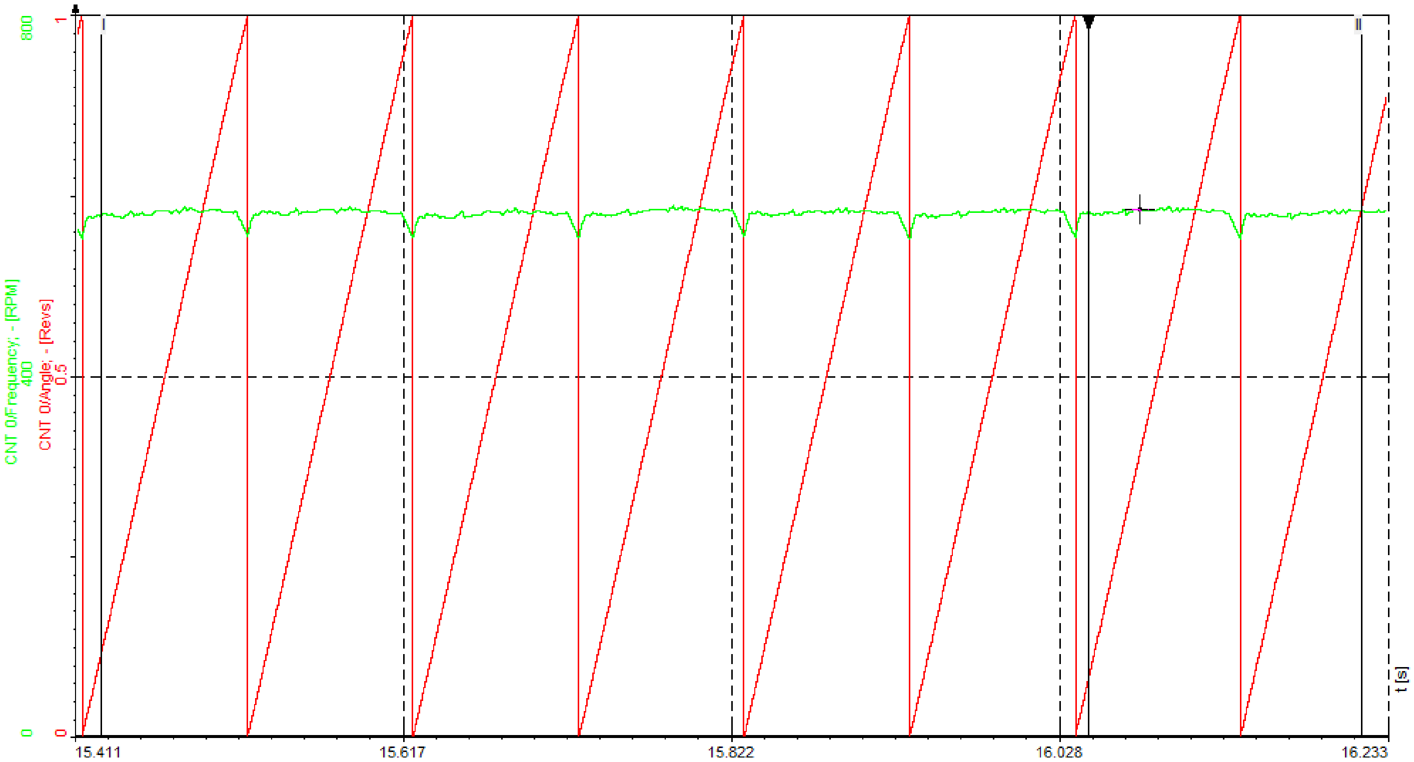


Image 64: The measurement example with tape sensor

Measurement accuracy

The overall error of the measurement has to be split up into errors from the sensor and the counter measurement uncertainty. Usually, the sensor errors make the major amount.

Counter accuracy

Counter architecture

To understand how the angle resolution is determined, it is at first important to understand the internal architecture of the Dewesoft Counters. A combination of main and sub counter is used internally for getting higher precision at the frequency measurement. The main counter is running on event counting (or encoder mode). The sub counter is used for time measurement, it measures exactly the time of the input event with a resolution of 9,77 nsec (= 1/102,4 MHz) relative to the sample clock. At every rising edge on the Counter Source, the counter value of the sub counter is stored in a register. At every Sample Clock, the values of both counters are read.

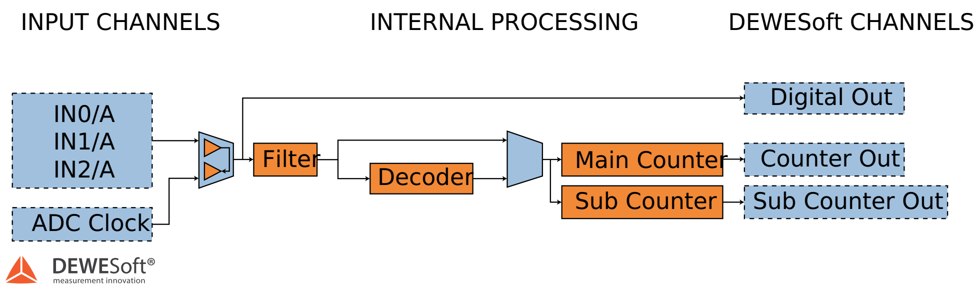


Image 65: Counter architecture

With these, both measurement results not only the frequency can be calculated in a precise way. Also, the event counter result can be shown in fractions because the exact time when the event occurs at the input is known. The event counting result is recalculated with interpolation to the sample point as shown in the diagram below.

Here the improvement of the measurement result is shown. While a standard counter input shows the value up to one sample delayed, the counter input of the Dewesoft instrument calculates the exact counter result at the sample point.

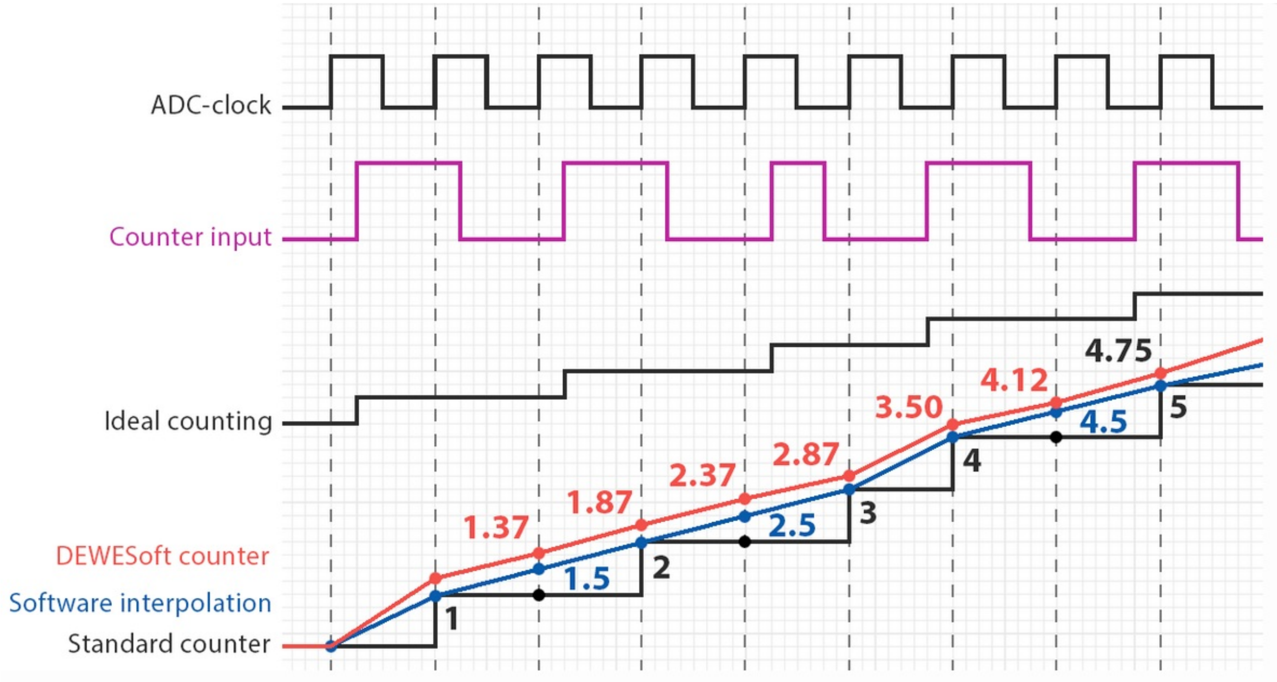


Image 66: Dewesoft SuperCounter

Angle resolution

The counter result is read out with the sample rate, therefore the same update rate applies for the calculated angle.

Furthermore, the angle resolution depends on the rotation speed (RPM).

Below there are two angle-based 2D graphs showing the Sensor angle on the x-axis at the same RPM. The option draw sample points was enabled. On the left side, the sample rate was set to 500 Hz, on the right to 2500 Hz:

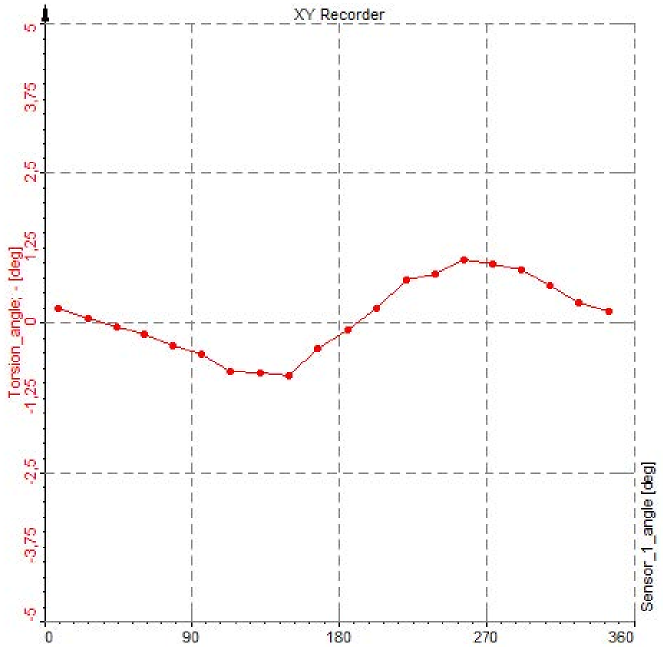


Image 67: Sample rate of 500 Hz

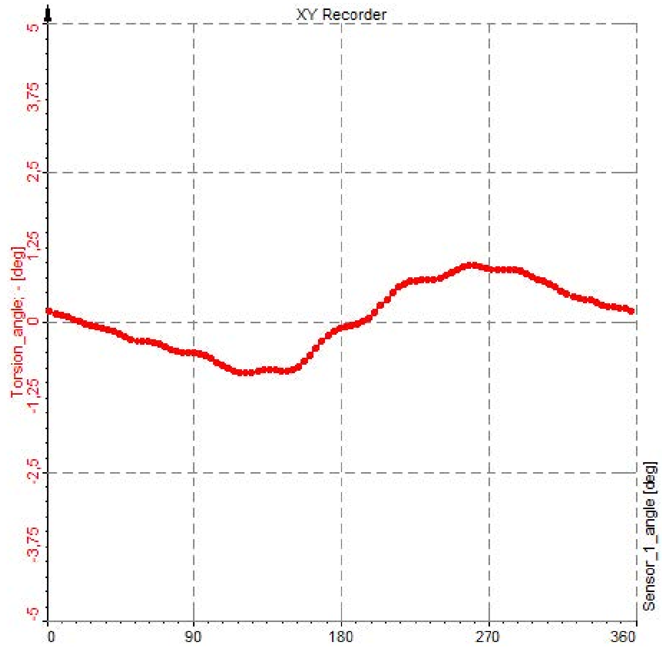


Image 68: Sample rate of 2500 Hz

Since rotational and torsional calculations are all based on the sample rate, the angular resolution is the same.

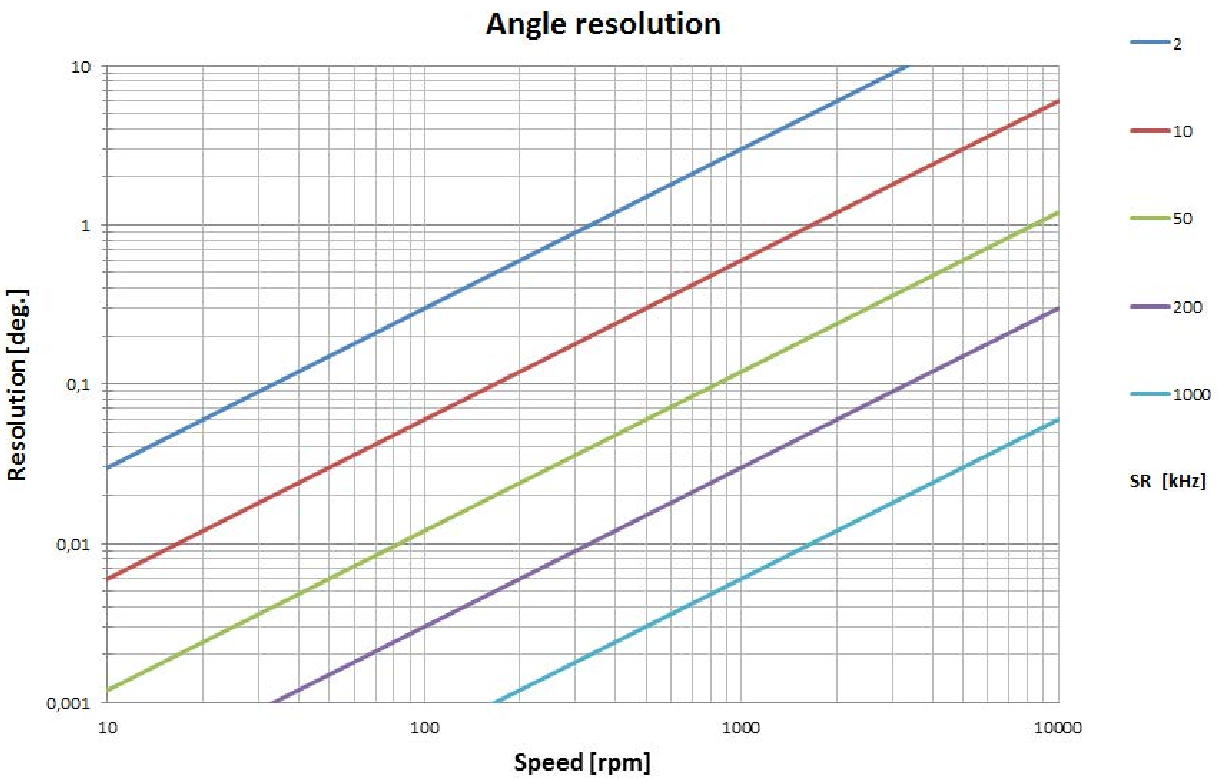


Image 69: Angle resolution is decreasing with increasing RPMs

Angle accuracy

As the counter is working on an internal timebase of 102,4 MHz, the angle accuracy is only depending on the rotation speed (RPM).

The following formula shows how to calculate the angle accuracy:

$$\phi_{\text{accuracy}} = \frac{\frac{RPM}{60} \cdot 360}{102,4 \cdot 10^6}$$

The numerator shows the angle which is passing by in one second, this is then divided by the timebase of the counter (in our case it is 120,4 MHz), in one second we will get 102.4 million samples.

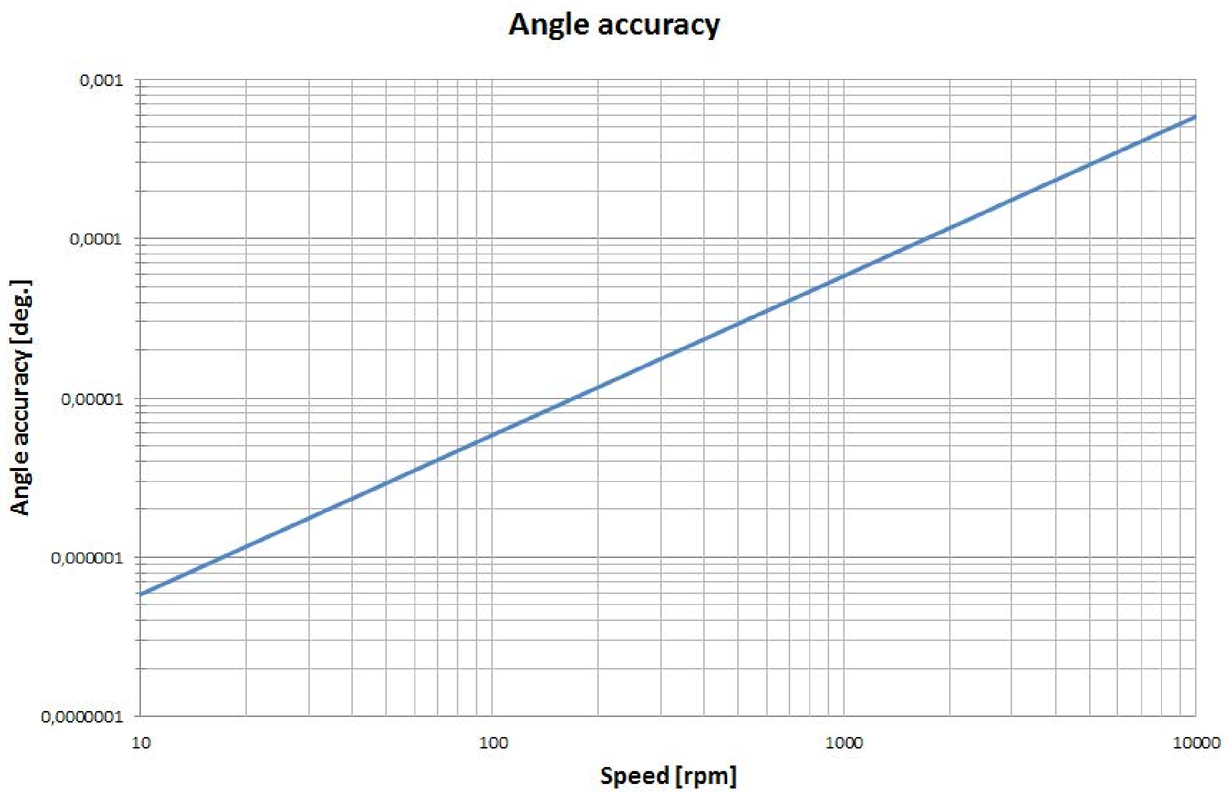


Image 70: Angle accuracy is decreasing with speed

Frequency accuracy

Any digital frequency measurement is based on period time measurement. The time between two edges of the input signal is sampled with the counter timebase of 102.4Mhz. With this simple measurement method the accuracy of the measured frequency is given by the ratio between the input signal frequency and the counter timebase frequency:

$$f_{error} = \frac{f_{in}}{102,4 \cdot 10^6}$$

We can see, the error increases with the input frequency. For example at 10MHz the accuracy goes down to 10%!

As explained above, the advanced counter structure of Dewesoft is using two counters internal counters and the output rate is synchronous with the acquisition rate. With this technology, we can limit the maximum error to the set acquisition rate.

The illustration below shows the accuracy at different input signals between 2 kS/sec to 1000 kS/sec taking also the typical counter time base accuracy of 5 ppm in an account.

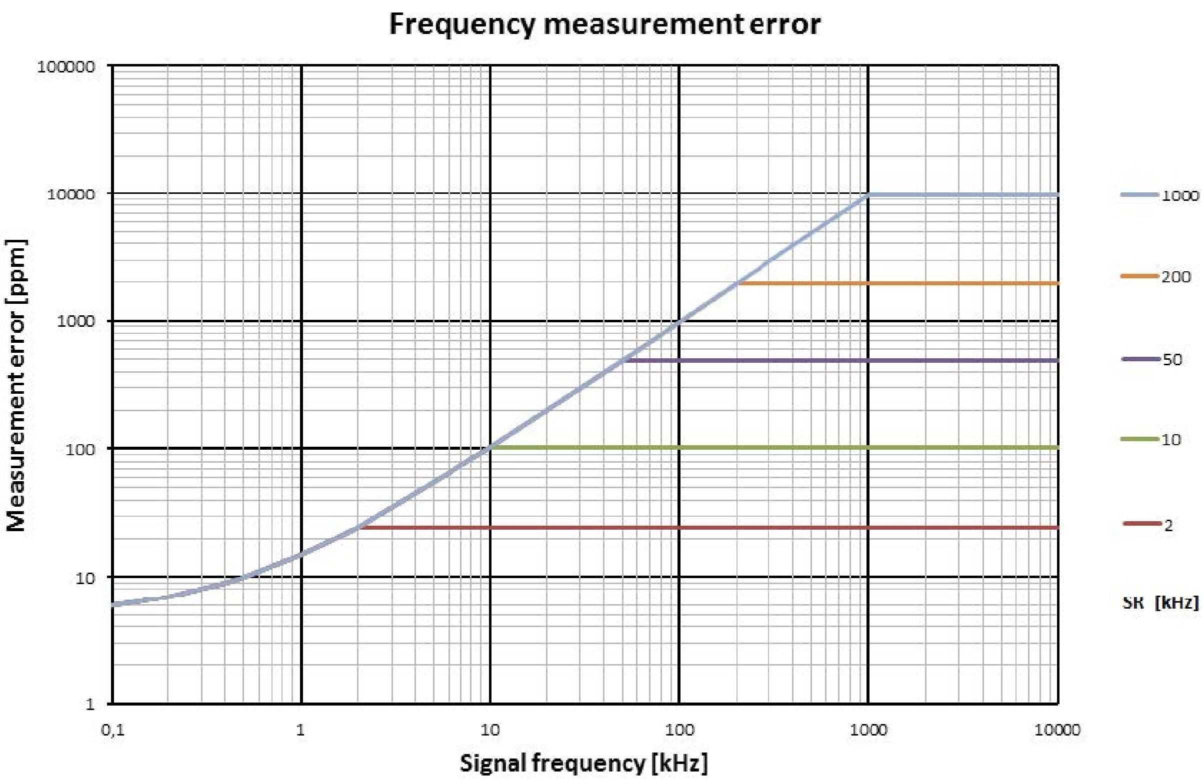


Image 71: Frequency measurement error increases with signal frequency

Signal frequency is shown on the X-axis and on the right side of the graph the acquisition sample rate is displayed.

The blue curve in the graph shows the formula, that is written above.

If we measure with a sample rate, that is higher than the frequency of the signal, we calculate the frequency only on one period. If the sample rate is lower than the frequency of the signal, we calculate the frequency once per sample.

Example: If we measure a signal with a frequency 10MHz with a sample rate of 1MS/s, we will get an average frequency every microsecond for the last 10 periods (and the error will be the same as if we would measure a frequency of 10 times "slower" signal). That is why we get a straight line on the upper right side of the graph.

This inaccuracy depends on the type of sensor that is not taken into account in the formula. The formula assumes the sensor is infinitely accurate. Some sensor types have bad resolution and produce a lot of noise. The graph assumes that this typical inaccuracy is 5pmm and this is also drawn at the beginning of the graph.

Angle sensor comparison

When measuring angle it si critical to select the appropriate sensor to do your measurement correctly.

Encoder	 Image 72: Encoder	
+ best resolution		- mounting is critical
+ useful also for low RPMs		
Tacho sensor	 Image 73: Tacho sensor	
+ easy installation		- bad resolution
+ cheap		- not useful for TV and CA
Tape sensor	 Image 74: Tape sensor	
+ easy installation		- expensive
+ good resolution		
+ zero position		
CA-RIE encoder	 Image 75: CA-RIE encoder	
+ best for high RPM vibrations		- extensive installation

DS-TACHO 1

DS-TACHO1 is a tacho adapter that converts the analog tacho signal to a Dewesoft instrument's CNT / DI input (Lemo 7pin) with an adjustable trigger level (TTL logic signal).

It fits the COUNTER input (Lemo 7pin) on DEWE-43 and SIRIUS Â±100V input isolated, trigger threshold adjustable Â±10mV -> Â±2V.

Image 76: DS-TACHO 1 adapter

Tacho adapter electronic specifications table:

Trigger / Retrigger level:	±10 mV ... ±2 V (adjustable with screwdriver)
Input type	AC-Coupled, 1Hz
Input impedance	150 kOhm
Max input voltage:	±50 Vdc, ±100 Vac
Power supply output	+5Vdc (max current depending on used DEWESoft device: eg. DEWE-43: max 800 mA)

Image 77: DS-TACHO 1 specifications

How does it work?

Connect the DS-TACHO1 with the LEMO 7pin to a DEWE-43 or a SIRIUS Counter input, and on the DSUB 9pin side to your analog tacho probe signal (e.g. magnetic pick-up sensor with a screw, 1 pulse per revolution).

Start the rotating machine, then use a screwdriver to manually adjust the trigger level on the DS-TACHO1, see picture below.

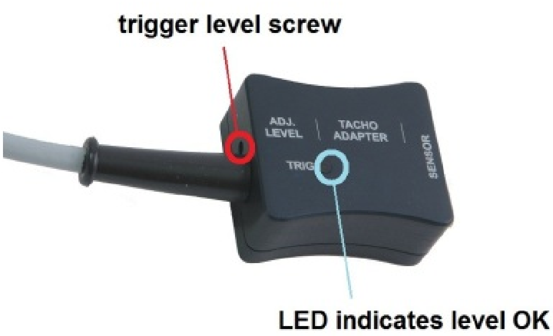


Image 78: Manually adjust the trigger level with a screwdriver

When the trigger is detected correctly, the blue LED will flash. Vary the RPM on the machine to check if the trigger level is ok for the whole RPM range.

The lowest detectable frequency for the counter input on the DEWE-43 / SIRIUS is 5 Hz, therefore if you have 1 pulse/revolution, the lowest RPM is 300. If you need to measure lower RPM, you could increase the number of pulses per revolution (e.g. for inductive probe mount a screw every 90Â° on the rotating disk and then divide a result by 4).

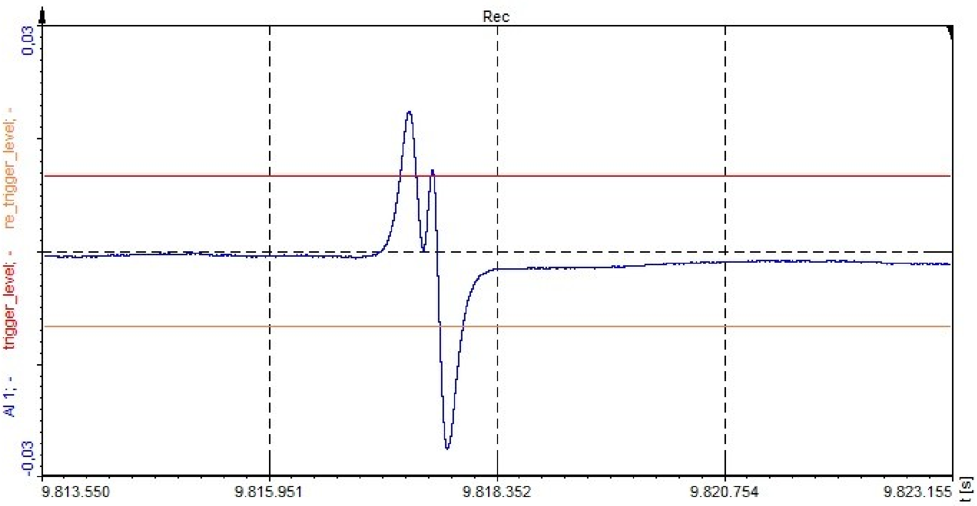


Image 79: Trigger level for a digital signal

In this example, you see the input signal of a magnetic tacho probe (coil), when a screw on the disk is passing by. The higher the RPM, the higher the induced voltage is, so here you have to set the trigger level low (shown in picture Â± 10 mV).

The upper (red) line is the trigger level, the lower (orange) line is the retrigger level. The signal has to fall below the lower line to be armed for the next trigger again. This even makes it possible to correctly detect a bad signal as shown above!

How to calculate the maximum possible RPM for your sensor

To reliably measure gap sensors (gear-tooth or tape) we need to set the right sampling frequency of acquisition. The theoretical Nyquist multiplier factor is 2,56 but we recommend the factor of 5 or even 10 so that a good edge of the signal is acquired. This limits our maximum RPM range in conjunction with the number of pulses per one rotation.

To calculate this limit you can use the following formula:

$$\frac{\frac{Sampling\ rate\ of\ Sirius[Hz]}{Nyquist\ factor}}{Pulses/revolution} \times 60Hz = [RPM]$$

Calculator for maximum Analog RPM acquisition: