Continuous Measurements with Zero Dead-Time in CNT-91

White Paper from Pendulum Instruments

Author: Staffan Johansson

The CNT-91 Timer/Counter/Analyzer introduces the concept of ***continuous data output with zero-time***. This White Paper/Technical note will describe the improved functionality in relation to the CNT-90 Timer/Counter/Analyzer. But first a summary of the different measurement and output data transfer modes in the CNT-90 and CNT-91

1. The CNT-90/CNT-91 Measuring and Output Data Transfer Modes

CNT-90 and CNT-91 can operate in three different modes

- single (individual) measurements
- block measurements
- zero-dead-time single measurements (CNT-91 only)
- zero-dead-time block measurements

Mode	**Local operation**		**Remote operation**	
	CNT-91	CNT-90	CNT-91	CNT-90
single measurements (with dead-time)	Values mode	Values mode	READ?. INIT+FETCH? GET INIT +TALK ONLY	READ?. INIT/FETCH? GET
block measurements (with dead-time)	Stat/plot mode of any measurement	Stat/plot mode of any measurement	READ:ARR? INIT/FETCh:ARRay?	READ ARR? INIT/FETCH ARR?
zero-dead-time single measurements	Freq BtB, Period BtB, TIE	N.A.	READ?. INIT+FETCH? GET INIT +TALK ONLY	N.A.
zero-dead-time block measurements	Freq BtB, Period BtB, TIE in Stat/plot mode	N.A.	READ:ARR? INIT+FETCh:ARRay? of Freq BtB, Period BtB, TIE (1-timestamp/value); Or Raw timestamp (4-timestamp/value)	READ ARR? INIT/FETCH ARR? of Raw timestamp (4-timestamp/value)

Mode	**Max speed (calculated results/second)**	
	CNT-91	**CNT-90**
single measurements (with dead-time)	GET: **650/s** TALK ONLY: **4000/s**	GET: **500/s**
block measurements (with dead-time)	READ:ARR?: **15k/s**	READ ARR? **5k/s**
zero-dead-time block measurements with dead-time between blocks	READ:ARR?: **15k/s** **250k/s to internal memory**	READ:ARR?: **15k/s** **250k/s to internal memory**
Continuous zero-dead-time block measurements	ARM:COUNT INF READ:ARRay:? **10k/s**	N.A.

A: Single Measurements

- READ?
- INIT+FETCH?
- GET

Single measurements mean that you do one measurement at a time and transfer to the display and to the data outputs (GPIB or USB). This mode has a dead time between measurements of approx. 2 ms, when you trigger the counter's individual measurements, one by one, via the GET command (Group Execute Trigger), from the connected controller.
The internal measurement memory buffer is not involved in this mode.

Using the above commands single measurements, you can control exactly when you want to initialize (start) the measurement. The communication process is simplified:

- Controller talker /CNT-9x listener: "start measurement" (GET)
- Controller talker /CNT-9x listener: "send result" (FETCH?)
- CNT-9x talker/ Controller listener: "sending result data"
- Controller talker /CNT-9x listener: "start next measurement" (GET) …

CNT-91: In addition to the CNT-90 modes, the CNT-91 has an additional output mode;
Talker only (SYSTem:TALKonly ON)

Talker only differs from the GET-triggered individual measurements in that the counter will be triggered once and after that the counter will output measurement data at the highest possible rate, without waiting for subsequent trigger commands (GET) from the controller. This reduces the transfer overhead since the controller is not taking control between the measurements. The controller cannot initialize individual measurements and the measurement data continues to be sent until the controller stops the transfer via the Interface Clear command or by pressing the CANCEL key on the counters front panel.

The dead time between measurements in this "real-time" transfer mode is <250 microseconds. To reach this, also

The communication process is simplified:

- Controller talker /CNT-90 listener: "go to Talker only mode and start measurement"
- CNT-90 talker/ Controller listener: "sending result data"
- CNT-90 talker/ Controller listener: "sending result data"
- CNT-90 talker/ Controller listener: "sending result data"
- etc…

B: Block Measurements:

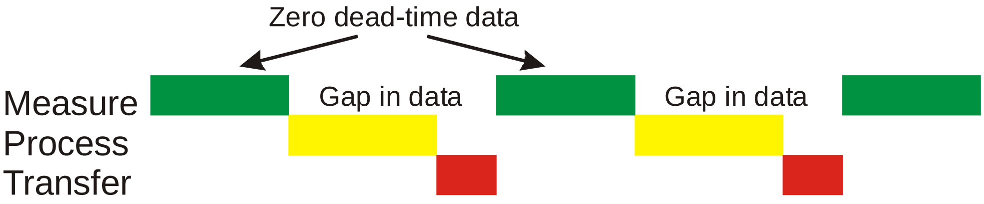
- READ:ARR?
- INIT + FETCH:ARR?

Block measurements mean that you set up a certain measurement function, e.g. frequency, and trigger a block measurement sequence from the controller. Then the individual measurement results are transferred one-by-one to the internal memory and stored, until the set number of samples has been reached. After that the memory contents can be sent to the controller using the FETCH ARRAY command.

The dead-time between measurements is significantly smaller, only 4-8 microseconds. To reduce the dead time to a minimum, you should e.g. disable calibration of interpolators (CAL:INT:AUTO OFF). To maximize through-put you should also turn off the display (DISP:ENAB OFF) and use Packed data format (FORM:PACK).

Using these modes of fetching data, you do a block transfer and cannot see the individual measurements until the whole block is received. You can control via SW exactly when you want to initialize (start) the block.

The procedure for block measurements in CNT-90 is pure sequential. First you measure and store ALL values in the block in internal memory, then all measurements in the block are processed (calculated and formatted), and finally the block is transferred to the PC. The measurement function can be any standard function with dead time (E.g. Frequency, Period, Vp-p) or the raw timestamping function (zero-dead-time). This means there is always a gap of data due to the dead time between blocks in the data output, even if individual blocks contains zero-dead-time data (in raw timestamping mode). In CNT-90, true zero-dead-time measurements (raw timestamping) can only be made up to 750k Samples, after that there must be a gap in the data.



C: Zero Dead-Time Single Measurements

Only CNT-91 features zero-dead-time single measurements, accessible from both the front panel in local operation, as well as in remote bus operation. There are three pre-defined measurements:
Frequency back-to-back, period back to back and TIE (Time Interval Error)

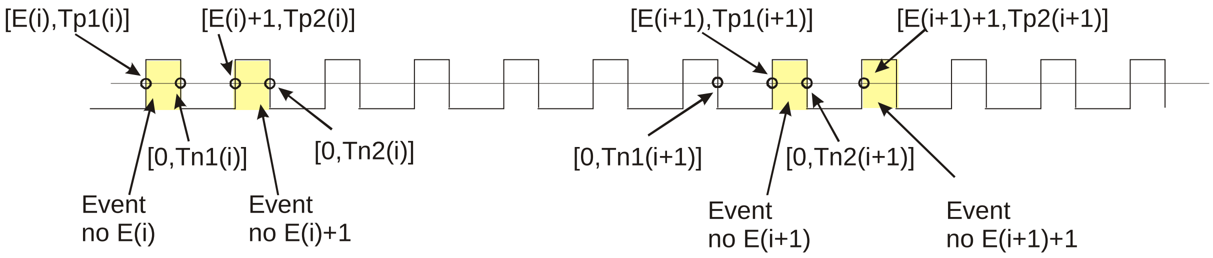
In frequency or period back-to-back you read the number of elapsed input trigger events or input cycles (N_i) and the elapsed time (T_i) at intervals defined via the measuring time setting, every time you read the elapsed events and time, you calculate the Frequency or Period on-the-fly as:

$$Freq(i) = \frac{N_i - N_{i-1}}{T_i - T_{i-1}} \qquad Per(i) = \frac{T_i - T_{i-1}}{N_i - N_{i-1}} \qquad TIE(i) = T_i - T_0 - \frac{N_i - N_0}{F_{REF}}$$

Note that the Frequency value is always, per definition, the *average* number of cycles/s during the measurement time. When measuring frequency, the input signal is divided by 2 before it is time-stamped, which means the minimum frequency measurement time is over 2 cycles.
Period back-to-back and TIE has no input divider, and is a *single* period as long as $N_i = N_{i-1} + 1$. This is the case for input frequencies up to approx. 250 kHz. At higher frequencies, up to 160 MHz, the value is the average period back-to-back.

D: Zero Dead-Time Block Measurements (Raw Timestamping Data)

Both CNT-91 and CNT-90 perform zero-dead-time "Raw" time-stamping measurement as a block measurement function only, which is not accessible from the front panel. From the GPIB or USB bus you set up a certain block size, select a pacing time from 4 microseconds and up, and start the timestamping measurement (FUNC:TSTA). This mode will then store the number of accumulated and time-stamped input trigger events, in groups of 4, without losing any single cycle up to 160 MHz input signals.



E(i) is the i:th sample of accumulated input cycles (ALL trigger events are counted and accumulated up to 160 MHz on input A or B,). E(i)=0 for the negative slopes, and the accumulated number of input cycles for the positive slope

Tp1(i) is the timestamp of the first positive trigger event
Tp2(i) is the timestamp of the second positive trigger event
Tn1(i) is the timestamp of the first negative trigger event
Tn2(i) is the timestamp of the second negative trigger event

Please note that first trigger event that occurs after the pacing time, may be a negative as well as a positive trigger slope, so the sequence of the 4 timestamps may be:

pos – neg – pos – neg OR neg – pos – neg – pos .

The way to identify this, is to look at the Event number which is 0 for negative slopes and is a positive integer for positive slopes

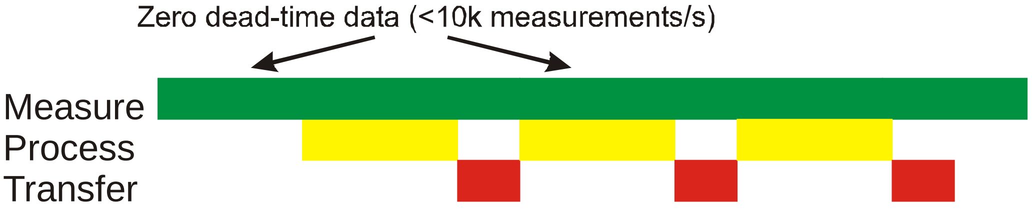
CNT-91: In addition to the CNT-90 raw timestamping mode, the CNT-91 can also measure Frequency and Period Back-to-Back plus TIE, as zero-dead-time block measurements.

Just like normal block array measurements, this measurement can continue at maximum speed a limited time until the internal memory buffer is filled, after 3.5Msamples (CNT-91) respectively 750k Samples (CNT-90). After that the memory must be read and emptied before the next block of zero-dead-time data can be measured again.

However the CNT-91 can also make continuous "streaming" measurements of zero-dead-time data for an unlimited time, as described below.

2. Continuous Streaming Output Mode

This unique CNT-91 output mode is basically a normal block measurement mode, however with the exception that there are parallel processes for storing data in internal memory, processing stored data and outputting data. Furthermore the block can have an unlimited size in CNT-91.



Continuous measurements come in two flavors "Fetch on the Fly" and "Overwrite Mode" as described below.

Fetch on the Fly

This is simply the ability the fetch the results from a normal block measurement that fits in the buffer while it is still running.

One thing to consider though is that while the measurement hasn't finished, you may not know how many samples are actually available for fetching. So, you use the FETC:ARR? MAX functionality (see below).

Overwrite Mode

This is the case when you specify a larger total number of measurements than will fit in the memory, that is >3.5M samples. As the buffer fills up, old values are overwritten with the new ones. This is done in such a way that it should always (with reasonable measurement speed) be safe to fetch on the fly. I.e. the data that is overwritten should never be the data that is in progress of being fetched.

For almost all measurement functions the method used is the following. The memory buffer is divided into six segments, each of which is at any one time dedicated to some particular usage (segments where data is being fetched from and segments where new data is written to). The roles of the segments can be switched around as needed.

As long as the fetching speed more or less keeps up with the measurement speed, this works out much the same as a regular circular buffer would do. But when you do not fetch sufficiently fast, or when you stop the measurement, and the buffer contains a lot of un-fetched data, there is always one or two segment's with valid newest result data remaining in the buffer, i.e. between one sixth and one third of the buffer (600k to 1200k measurements).

If the instrument is set up to measure faster than it can handle in overwrite mode, you will eventually end up with invalid data in the buffer. Most often invalid timestamps, but also invalid measurement results can occur. Internal tests will detect when this happens, and abort the measurement and put a -321, "Storage fault" error in the error queue, indicating that the buffer contents can't be trusted (but data is still available for fetching).

The measurement speed limit in overwrite mode depends on a lot of factors, including e.g. how fast, and in what way, the user fetches data on the fly. For this reason it's hard to set any firm rules for measurement speed recommendation.

Currently, one limit is enforced: In overwrite mode the pacing time used is minimum 50 microseconds, regardless of actual setting. This is to prevent the user from “choking” the instrument.

Practical tests show that data capture rates of >10k Sa/s is likely to cause invalid data at some point during a long measurement. To be on the safe side, set a pacing time of more than 100 microseconds, with some margin. Measurements with more complex calculations such as Phase, requires longer pacing time than simpler measurements such as Period Single.

3. New SCPI Commands or Changed Behavior in CNT-91

FETCh:ARRay? <number of samples> | MAX (CNT-91 only)

When specifying <number of samples>, the instrument will wait until that number is available before responding. This could result in long waits if the measurement doesn't proceed as expected.

When using the "MAX" parameter, the instrument will respond immediately with as many samples as currently possible. The number of samples in the response is limited by the number of un-fetched samples currently remaining in the buffer and the FORMAT:SMAX setting.

Error; -230,"Data corrupt or stale", is generated when there is no valid sample(s) to fetch at all from the buffer.

Error -224,"Illegal parameter value" error is generated if you try to fetch a larger number of samples than remains to fetch in the buffer. This includes also the case of using the "MAX" parameter when already at the end of the buffer, and you have already fetched all samples.

ABORt

ABORT no longer invalidates already finished results when breaking an array measurement. This means you can now fetch the partial result after the abort.

FORMat:SMAX <number>

Where “number” is any integer between 4 and 10000. Default setting is 10000 (not affected by *RST). This command is intended for use with any controller or application program that have problems with reading large amounts of data, but where the FETC:ARR? MAX functionality is wanted. To query the actual setting use:

FORMat:SMAX?

When using the "MAX" parameter, the instrument should always respond within a predictable time. Actual response time depends primarily on the number of samples in the response and the FORMat settings used (ASCii being particularly slow, REAL is fast and PACKed is fastest), and to some extent the actual measurement function.

When no new measurement since the latest fetch is available in the buffer (and not at the end of it), **FETC:ARR? MAX** will give a "zero samples" response, ie empty string in case of format ascii, and an empty binary data packet ("#10") in the case of format real or packed.

ARM:COUNt <number> | INFinity
ARM:COUNt?
The "INF" parameter is available in CNT91 only. It will cause the arm loop to continue endlessly. (Note: The timestamp counter will overflow after approx 107 days of non-aborted operation, something that must be taken care of in the application SW, if it is important to run continuously for more the 3 months.)
When set to INFinity, the query responds with "INF" (without the quotes).

Talker Only Mode

To switch on talker only mode:

SYSTem:TALKonly ON

Since now the CNT-91 will be talker “forever”, and will not listen to any programming command from the controller via the normal handshake lines, there is no command “SYST:TALK OFF” available. Once you have sent the command SYST:TALK ON to the counter, the CNT-91 will no longer respond to normal commands via the data lines in the GPIB interface

To switch off Talker Only mode, you use the **CANCEL key** (C) on the front panel alternatively send an **Interface Clear (IFC)** over GPIB.

Further settings to implement for successful high-speed talker only transfer are:

DISPlay:ENABle OFF.
FORMat REAL or FORMat PACKed.
ARM:COUNt 1
TRIGger:COUNt 1
INIT:CONT ON.

The **measurement function** could be anyone except:

Period average (smart calculation)
Frequency (smart calculation)
Time Interval (smart calculation)
Timestamping
Voltage
Totalize

This means that Frequency BtB and Period BtB is OK, but not “smart” frequency where a regression line fitting between several time-stamped data is used to improve the basic resolution. See further article “Modern Frequency counting principles improve resolution”, which is available for download from www.pendulum-instruments.com.