Application note

# Triggering for averaging
# ADQ35-FWATD
# ADQ35-PDRX-FWATD

# Table of Content

**1 Document overview ..... 3**
1.1 Scope ..... 3
1.2 Test case ..... 3
1.3 Test setup ..... 3
1.4 Supported digitizer models ..... 3
**2 Averaging ..... 3**
2.1 Motivation: suppress random noise ..... 3
2.2 Problem: systematic noise ..... 3
2.3 Implementation of accumulation in FWATD ..... 4
**3 Systematic noise and its relation to trigger method ..... 4**
3.1 Systematic noise ..... 4
3.2 Trigger method ..... 5
3.3 Large set of accumulated records ..... 6
**4 Trigger method details ..... 8**
4.1 Triggers in the system ..... 8
4.2 Internal trigger ..... 8
4.3 Trigger output ..... 8
4.4 External trigger ..... 9
4.5 Fractional N PLL ..... 9
**5 Comparison and results ..... 10**
5.1 Fractional N PLL vs. internal trigger comparison ..... 10
5.2 Discussion about external trigger source ..... 12
5.3 System design aspects ..... 12
**6 Python Code ..... 13**

# 1 DOCUMENT OVERVIEW

## 1.1 Scope

This application note introduces various internal trigger generation functions in the ADQ35 digitizer, aimed at optimizing performance during record accumulation.

## 1.2 Test case

The study examines noise performance with a terminated input and the accumulation of low-amplitude, short-duration pulses. It compares different trigger methods for these scenarios.

## 1.3 Test setup

The test setup in Figure 1 includes an ADQ35 digitizer running firmware for advanced time domain (FWATD). This firmware enables the accumulation of records within the digitizer. The digitizer's trigger source is connected to the accumulation. Each trigger adds a data record to the accumulator value.

The trigger is available on the output port TRIG to trigger an external pulse source. The pulse source is used to evaluate how triggering of external equipment influences the choice of trigger method.

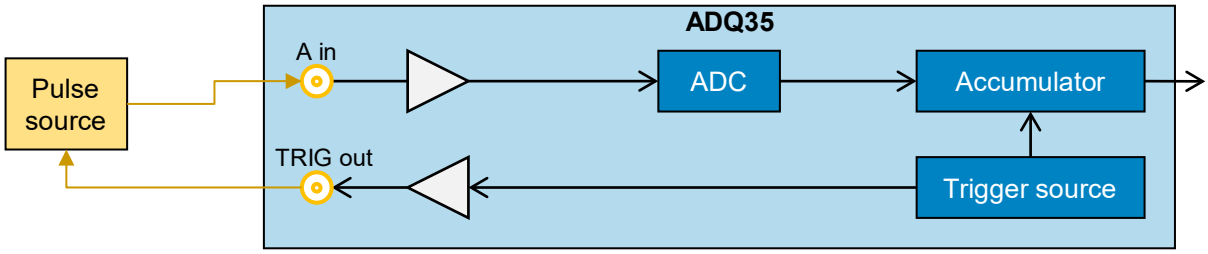


Figure 1 Test setup block diagram.

## 1.4 Supported digitizer models

The tests were conducted on the ADQ35-PDRX using the FWATD firmware, but the results are also applicable to the ADQ35.

# 2 AVERAGING

## 2.1 Motivation: suppress random noise

Using averaging is an effective method to enhance systematic signals. The underlying assumption is that a consistent (periodic) signal is present across multiple records, while the noise remains uncorrelated between these records. By averaging the records, the signal-to-noise ratio (SNR) can improve proportionally to the square root of the number of accumulated records.

## 2.2 Problem: systematic noise

Using averaging can also enhance systematic electrical disturbances in the system. These disturbances, often at the microvolt (µV) scale, are much lower than random noise but become more apparent through averaging. This application note provides guidance on addressing these disturbances, referred to as systematic noise in the text.

### 2.3 Implementation of accumulation in FWATD

FWATD includes an averaging block, which accumulates data in the FPGA of the digitizer. Accumulation means adding up the data points. The division, which happens on the PC, converts these accumulated values into average voltage amplitudes.

Accumulation involves adding corresponding samples from each record. For example, all first samples are summed to one value, all second samples to another, and so on. The resulting series of sums, called the accumulator, has the same length as the original record. The number of accumulated records is denoted N_ACC in the following text.

The effect of the accumulation is that systematic signals grow as N_ACC, whereas random signals grow as SQRT(N_ACC). Therefore, the SNR then increases as:

N_ACC / SQRT(N_ACC) = SQRT(N_ACC).

Converting from codes to volt incorporates N_ACC. FWATD use a 14-bit MSB-aligned representation[1] of the raw samples to save bits for the accumulation (while FWDAQ use 16-bit MSB aligned data output). Voltage conversion is then performed as:

CODE * FULL_SCALE / 2^14 / N_ACC

Figures in this document are presented in Volts after this conversion.

## 3 SYSTEMATIC NOISE AND ITS RELATION TO TRIGGER METHOD

### 3.1 Systematic noise

The system contains numerous systematic noise sources that are weak and often buried in random noise. However, with a large number of accumulated records, the random noise is suppressed, allowing the systematic noise to become apparent. This phenomenon limits the noise reduction benefits of averaging. Figure 2 illustrates that the suppression deviates from the ideal when there is a significant accumulation. Since the systematic signals are very weak, their amplitude has been normalized to a theoretical level. The green and red traces represent 10 and 1,000 accumulated records, respectively, and the signal appears to be noise. For 100,000 accumulations, represented by the blue curve, a systematic signal dominates over the noise. The noise suppression no longer follows the expected pattern, where the SNR increases as the square root of the number of accumulations (SQRT(N_ACC)).

[1] Notice the risk of confusion. 14 bits is only valid for the output data. All other settings, for example, DBS is done with 16 bits representation.

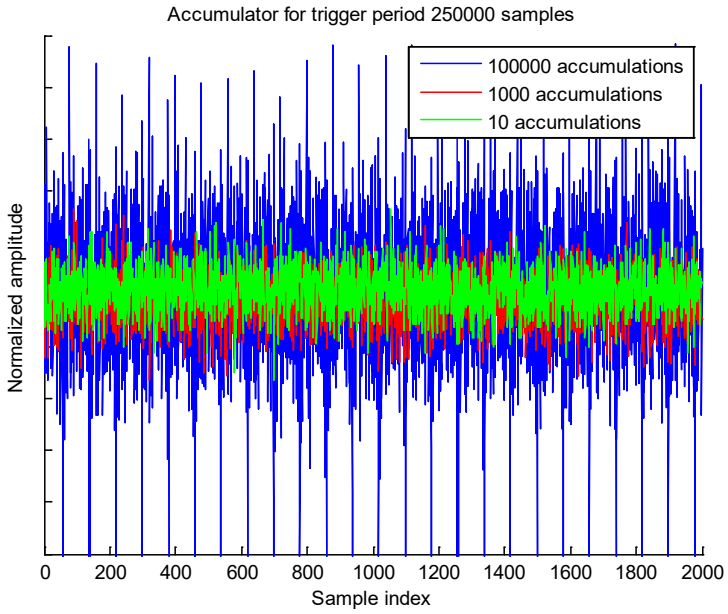


Figure 2 Time-domain accumulator with correlated trigger frequency result in systematic noise.

### 3.2 Trigger method

The choice of trigger method significantly affects how systematic noise influences the accumulator result. Systematic noise can originate from various sources, including all clocks within the digitizer. These noise sources typically operate at frequencies derived from the sampling clock. If the trigger is also synchronized with the sampling clock in a non-optimal way, the accumulation process may reinforce the systematic noise rather than suppress it.

In Figure 2, the trigger is set to 20 kHz, which aligns with harmonics of the 5 GHz sampling clock. This correlation causes the accumulation to amplify the systematic noise.

In contrast, Figure 3 illustrates the same measurement using a trigger period of 250,001 samples. This configuration breaks the correlation with the systematic noise sources, allowing the accumulation to suppress noise as intended. Each measurement's amplitude is normalized to its theoretical value, demonstrating the expected noise reduction.

Further discussion on the importance of de-correlating the trigger from noise sources is provided in section 4.

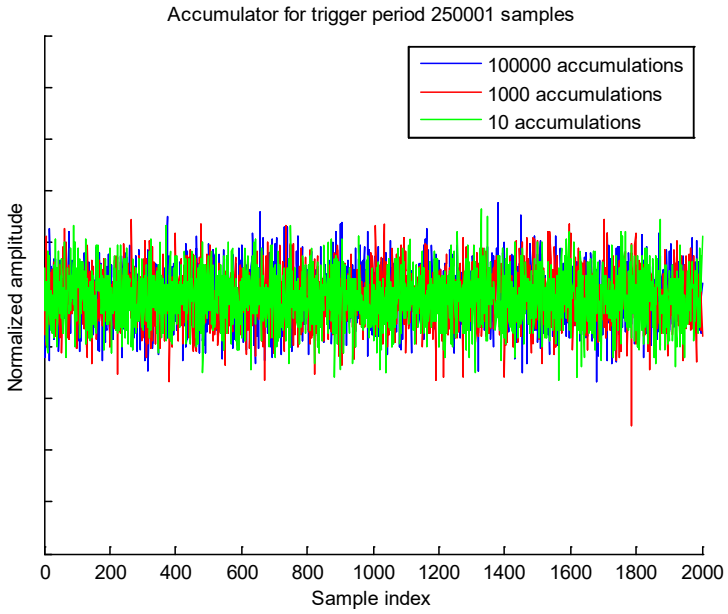


Figure 3 Time-domain accumulator with non-correlated trigger gives expected noise suppression.

### 3.3 Large set of accumulated records

The impact becomes noticeable only when a large number of records are accumulated - that is, for high values of N_ACC. Figure 4 shows the resulting noise levels after accumulation across different trigger periods. At first glance, all methods appear to follow the ideal noise reduction trend of SQRT(N_ACC). However, a closer inspection in Figure 5 reveals significant differences.

Figure 5 illustrates the deviation from the ideal noise level as the number of accumulated records increases. Around 2,000 records, certain methods begin to diverge from the ideal behavior. Notably, trigger periods of 250,000 and 250,016 samples exhibit a marked increase in systematic noise, resulting in a clear deviation from optimal noise suppression. In contrast, the 250,001-sample trigger period maintains performance that is nearly ideal. Additionally, the “Fractional N PLL” method detailed in Section 4.5 demonstrates strong results.

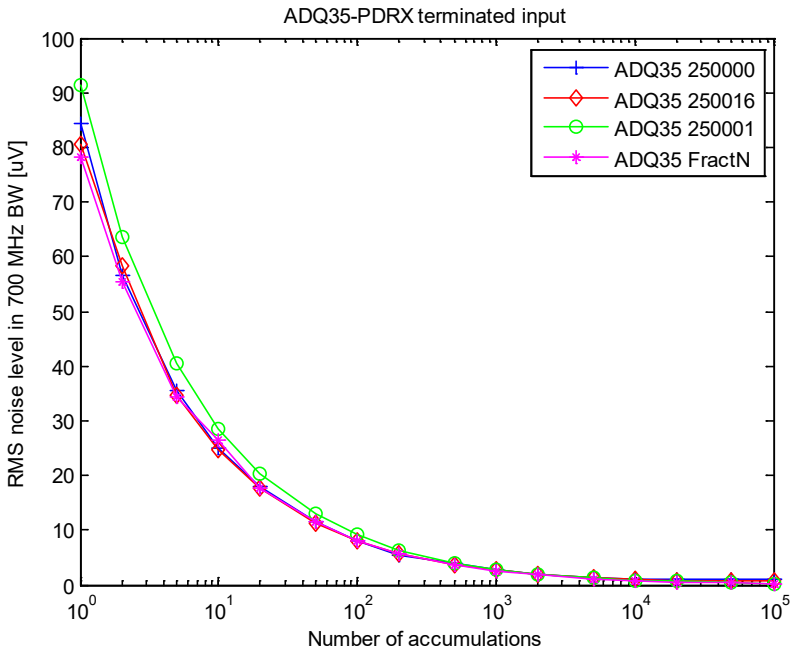


Figure 4 Noise level after accumulation for different trigger periods.

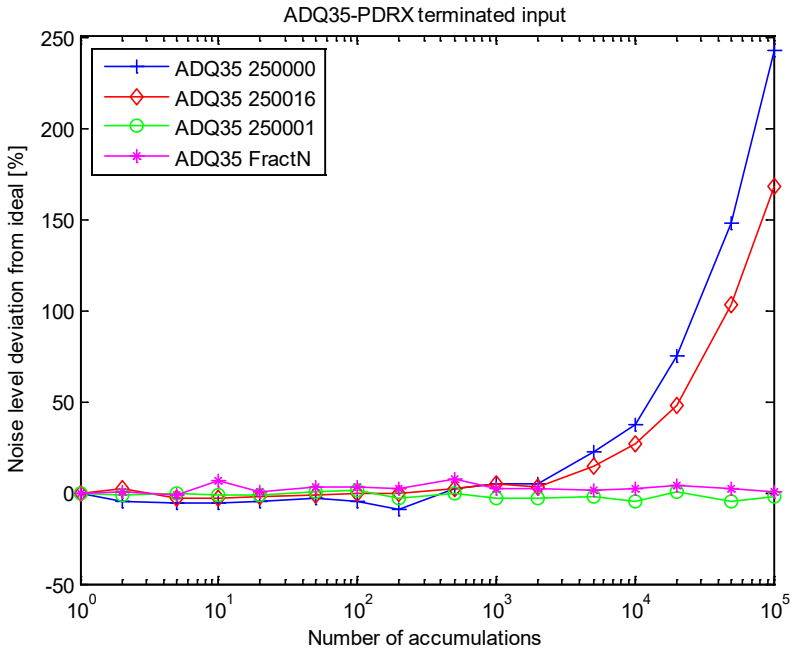


Figure 5 Deviation from ideal noise level for different trigger periods.

# 4 TRIGGER METHOD DETAILS

## 4.1 Triggers in the system

The trigger task is responsible for initiating both the pulse source - such as a pulser in a mass spectrometer - and the digitizer's acquisition process. A block diagram of the system is shown in Figure 1.

Since these two events are tightly coupled, they should be driven by the same trigger source. Any timing jitter between them can cause the recorded pulses to become blurred, which is assumed to be a concern in this context.

The signal source is considered asynchronous, meaning it does not introduce jitter to the trigger. In contrast, the digitizer is clocked and may exhibit jitter at its input ports. This assumption allows the analysis to focus on timing errors originating from the digitizer, making the model applicable to a wide range of pulse sources.

The digitizer operates synchronously using two main clocks:

- The sampling clock, and
- The data clock, which is typically derived as: Data clock = Sampling clock / 16[2]

These clocks are used at different stages of the signal chain and are illustrated in Figure 7.

Note: In the following text, the time unit "samples" refers to 1 / sampling rate, which at 5 GHz equals 200 ps.

## 4.2 Internal trigger

The internal trigger is derived from a periodic function generator implemented within the FPGA. The trigger period can be configured to any number of samples—for example, 250,000, 250,001, or 250,016 samples. This internally generated signal is used to initiate the acquisition process and precisely control the start of each record.

## 4.3 Trigger output

To use the internal trigger for activating the pulse source, it must be routed to an output port. This connection is driven by the Data clock, which imposes a constraint: while trigger periods of 250,000 or 250,016 samples can be generated accurately, a period of 250,001 samples cannot.

If a 250,001-sample internal trigger is used, it effectively breaks the correlation with systematic noise, which is beneficial. However, the trigger signal sent to the pulse source will be quantized to a timing grid defined by the Data clock - specifically, a resolution of 16 samples. This behavior is illustrated in Figure 6.

---

[2] It is necessary to have two different clocks. The FPGA cannot operate at 5 GHz clock. Hence the signal processing is operating at a lower rate and process multiple samples in parallel.

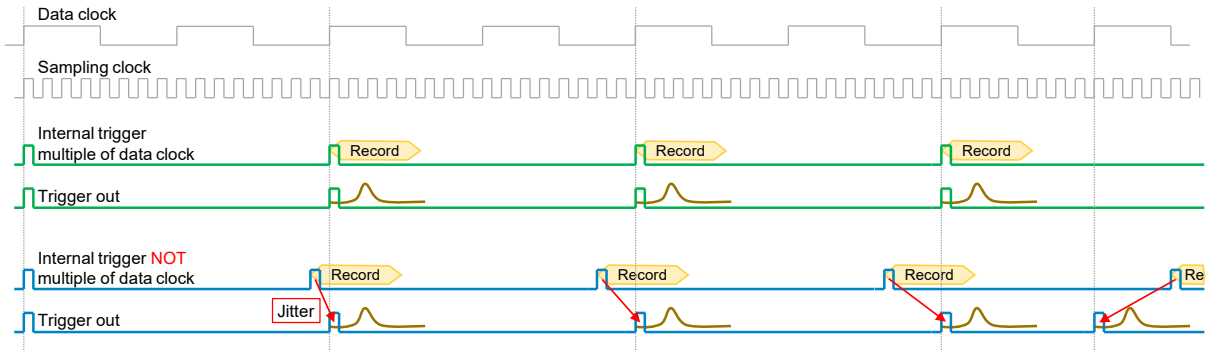


Figure 6 Timing for different periods of internal trigger.

The uncertainty in output timing will cause a deviation of timing between the pulse generation and a start of the recording.

### 4.4 External trigger

The external trigger refers to a trigger source that originates outside the ADQ35 digitizer. This source may or may not be correlated with systematic noise. However, the impact on signal quality remains the same, regardless of the trigger origin. Therefore, the considerations discussed here - particularly regarding trigger frequency selection - also apply to external trigger sources.

### 4.5 Fractional N PLL

The ADQ35 features a dedicated fractional N phase-locked loop (PLL) capable of generating periodic signals at frequencies not constrained by the FPGA clock, as illustrated in Figure 7. This allows the signal to be phase-locked to the sampling clock, ensuring controlled timing behavior. Because the trigger period is not correlated with internal noise sources, the resulting noise appears random and is effectively suppressed through averaging.

Importantly, the output from the fractional N PLL can be routed directly to the pulse source without limitations imposed by the FPGA data clock. This enables precise alignment between the recording process and the pulse source, as shown in Figure 8.

While this method appears optimal, there are practical limitations:

- The ADQ35 must act as the system's trigger source, which may impose constraints on the overall system topology.
- The trigger frequency is not directly tied to the system clock, which make control sequence generation tricky.

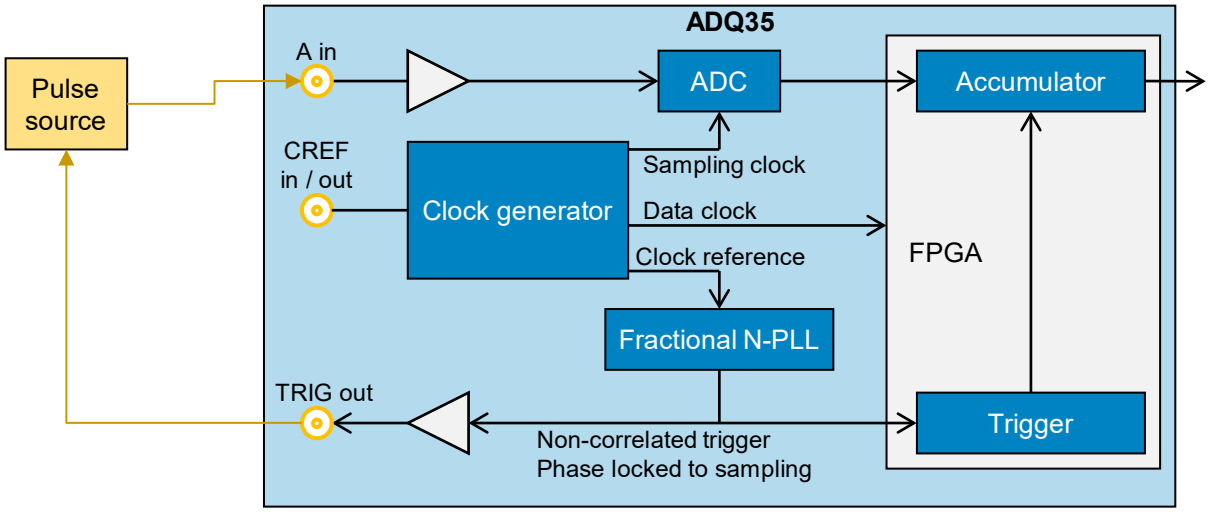


Figure 7 Fractional N PLL for non-correlated trigger rate.

The timing can then be aligned:

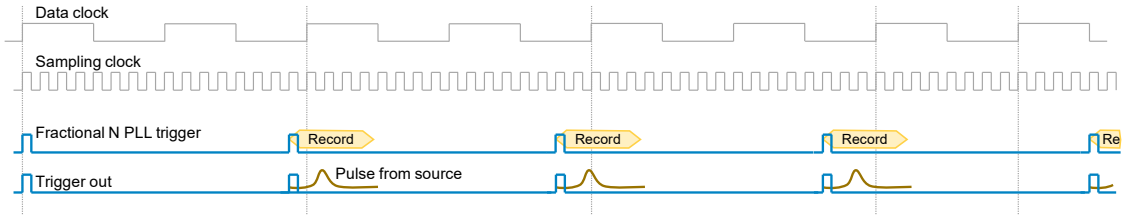


Figure 8 Timing of trigger of ADQ35 and external source.

The connection between the Fractional N PLL and the FPGA is through the external trigger port. So the external trigger event source is selected as trigger for the accumulation.

## 5 COMPARISON AND RESULTS

### 5.1 Fractional N PLL vs. internal trigger comparison

1. The fractional N PLL is configured to a frequency of 10,000.09 Hz, resulting in a trigger period of 500,004.5 samples. This signal is used to trigger both the digitizer and the pulse source.
2. The internal trigger is set up to periods of 500,000 samples for enabling phase-lock of trigger output pulse.
3. The internal trigger is set up to 500,001 samples for enabling suppression of systematic noise.

Figure 9 compares background noise after accumulating 100,000 records using the different trigger methods.

- The 500,000-sample internal trigger shows correlation with internal noise, leading to its accumulation.
- In contrast, both the non-correlated 500,001-sample internal trigger and the fractional N PLL produce low, random noise, indicating effective suppression.

Figure 10 evaluates pulse accumulation.

- The pulse source is triggered via ADQ35-PDRX to ensure timing alignment.
- The pulse has a 1 ns width and 20 mV amplitude.
- Using an odd trigger period (e.g., 500,001 samples) causes pulse broadening, due to jitter on the trigger output (see Figure 6).
- The 500,000-sample internal trigger yields a sharp pulse, and the fractional N PLL achieves equally precise results, matching the performance of the phase-locked internal trigger.

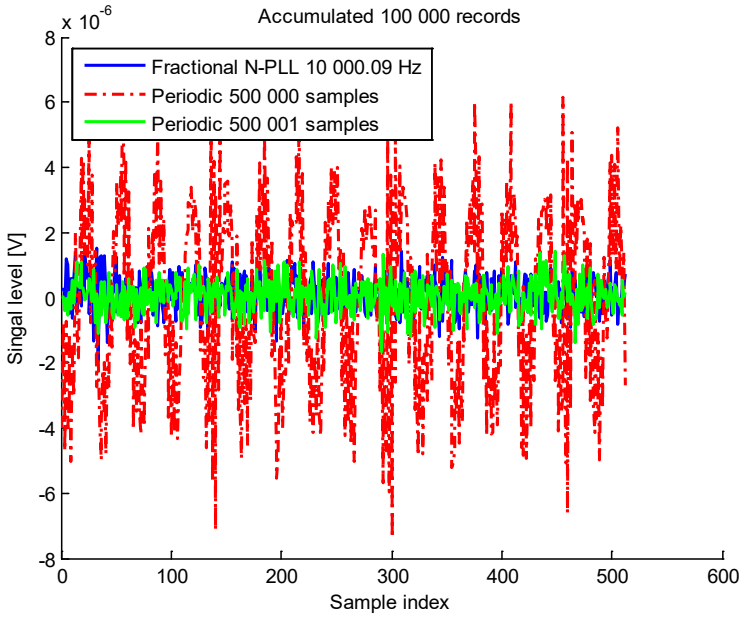


Figure 9 Noise comparison.

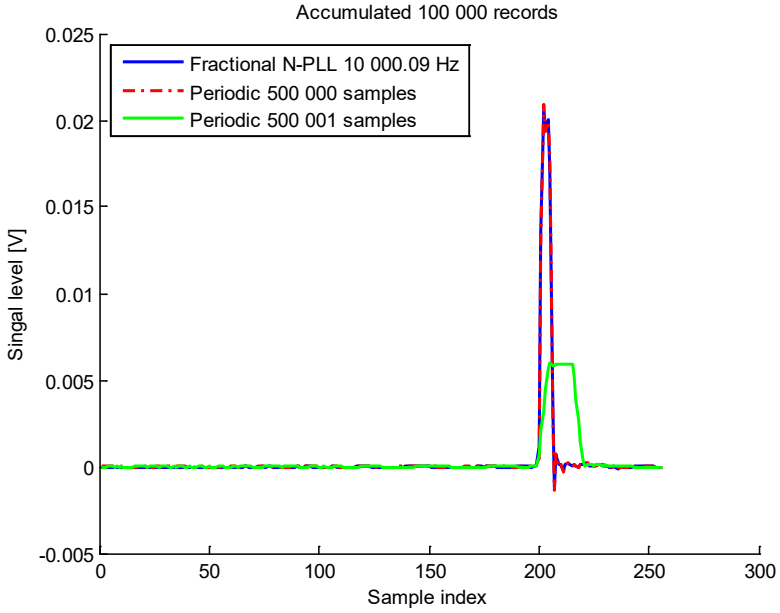


Figure 10 Pulse comparison.

The conclusion is that the fractional N PLL can achieve both accurate triggering of the source and low systematic noise.

### 5.2 Discussion about external trigger source

An external trigger source is not integrated into the ADQ35 digitizer hardware. However, there are two relevant approaches to implementing an external trigger in this context:

1. **Phase-locked Trigger**
   In this configuration, the digitizer and the trigger (and indirectly the pulse source) share a common clock base. This ensures a fixed ratio between the trigger frequency and the sampling rate, maintaining a stable mean phase relationship.
   If the external trigger rate can be configured to an odd number of sampling clock periods—as supported by the ADQ35's fractional-N PLL—systematic noise can be effectively suppressed. Otherwise, there is a risk that accumulation will amplify systematic noise due to phase correlation.
2. **Non-Phase-Locked Trigger**
   In this case, the trigger source operates independently of the sampling clock, meaning the relationship between the trigger rate and the sampling clock is not precisely defined. For crystal-based clock systems, this discrepancy can be on the order of tens of parts per million (PPM), and potentially much higher for other systems. This relationship may also drift over time.
   As a result, some accumulations may achieve good suppression of systematic noise, while others may not—depending on the degree of incidental correlation during each accumulation cycle.

### 5.3 System design aspects

This approach requires the ADQ35-PDRX to act as the timing master of the system. To support this configuration, the digitizer provides several timing interfaces: a clock reference input/output (CLK), an internal fractional N trigger (TRIG), and two additional trigger inputs for timing control - SYNC and GPIO. These interfaces are illustrated in Figure 11.


Figure 11 Connectors on the ADQ35-PDRX.

## 6 PYTHON CODE

```
# Use trigger from TRIG out port
parameters.acquisition.channel[ch].trigger_source =
pyadq.ADQ_EVENT_SOURCE_TRIG
parameters.function.fractional_n_pll.frequency = FRACN_FREQ
# pulse out always on to trigger source
port_id = pyadq.ADQ_PORT_TRIG
parameters.port[port_id].pin[0].direction=pyadq.ADQ_DIRECTION_OUT
parameters.port[port_id].pin[0].function = pyadq.ADQ_FUNCTION_FRACTIONAL_N_PLL

# Read back actual frequency
tmp = dev.GetParameters(pyadq.ADQ_PARAMETER_ID_TOP)
UseTrigF = tmp.function.fractional_n_pll.frequency
print(f"Trigger frequency {UseTrigF}")
```