Manual ADQ8-4X

This manual describes how to get the full potential out of Teledyne SP Devices' digitizer ADQ8-4X. The manual includes these steps:

- *Set up the analog front-end*
- *Master the triggers*
- *Control the acquisition*
- *Manage the sampling clock*
- *Understanding data transfer to host PC*
- *Using GPIO*

Table of content

1 INTRODUCTION 5
1.1 ADQ8-4X Architecture 5
1.2 Fundamental design properties 5
1.2.1 Data format 5
1.2.2 Calibration 6
1.2.3 Data acquisition nomenclature 6
1.2.4 Sampling clock frequency 6
1.2.5 System clocks 6
1.2.6 Analog signal range 7
1.3 Digitizer Studio overview 8

2 SETTING UP THE ANALOG FRONT-END 9
2.1 ADQ8-4X AFE block diagram 9
2.2 Set analog input range 9
2.3 Set analog DC-offset 10
2.4 Adjusting the digital gain and offset 10

3 SIGNAL QUALITY ENHANCEMENT 11
3.1 Digital Baseline Stabilizer 11

4 TRIGGER 12
4.1 Trigger block diagram 12
4.2 Introduction 13
4.3 Position of the trigger in the data 13
4.4 Timestamp 14
4.4.1 Timestamp definitions 14
4.4.2 Timestamp reset 14
4.5 Blocking triggers for synchronization 16
4.5.1 Function overview 16
4.5.2 Block triggers once 18
4.5.3 Windowing triggers 18
4.5.4 Gating and windowing triggers 18
4.5.5 Programming sequence for using trigger blocking 18
4.6 Trigger jitter 19
4.6.1 Trigger jitter definitions 19
4.6.2 Asynchronous triggering 19
4.6.3 Synchronous trigger 20
4.6.4 Extended trigger resolution 20
4.7 Software trigger 20
4.8 External Trigger Inputs 21
4.8.1 External trigger TRIG front panel connector 21
4.8.2 External trigger SYNC connector 22
4.8.3 Driving the external TRIG/SYNC signal by controlling input impedance 22
4.9 External trigger in the backplane 23
4.9.1 PXIe interface 23
4.10 Level trigger 24
4.10.1 Setting the level trigger level 25
4.10.2 Level trigger and DBS 25
4.10.3 Controlling noise sensitivity 25
4.11 Internal trigger 26
4.12 Trigger output 26
4.12.1 Trigger output port selection 26

4.12.2 Frame sync output on SYNC connector ... 26
4.12.3 Triggering external equipment with internal trigger ... 27
4.13 Large scale integration trigger support ... 28
4.13.1 Distributing trigger ... 28
4.13.2 Clock reference ... 28
4.13.3 High precision trigger on analog input ... 28
4.13.4 Calibration ... 28

5 CLOCK ... 30
5.1 Clock domains ... 30
5.2 Flexible clock network ... 30
5.3 ADQ8-4X-PCIe front panel connectors ... 31
5.4 Internal clock reference ... 31
5.5 External clock reference ... 31
5.6 Clock reference phase tuning ... 32
5.7 Internal clock generator ... 32
5.8 External clock ... 32
5.9 Clock reference output ... 32
5.10 Sample skip ... 32

6 GPIO ... 33
6.1 GPIO on TRIG connector ... 33
6.2 ADQ8-4X–PXIe GPIO on SYNC connectors ... 34
6.3 Using GPIO as a trigger ... 34
6.4 Output ... 34
6.5 GPIO in ADQ Development Kit ... 34

7 ACQUISITION CONTROL ... 35
7.1 Multi-thread notice ... 35
7.2 Acquisition memory ... 35
7.3 Triggered streaming acquisition ... 36
7.4 Acquisition mode multi-record ... 36
7.5 Re-arm time ... 37
7.6 User scheduled data transfer mode ... 38
7.6.1 Transfer buffers ... 40
7.6.2 User's buffers ... 40
7.7 Streaming transfer mode ... 40
7.8 Users application software consuming data ... 40
7.9 Record header ... 41
7.9.1 Metadata ... 41
7.9.2 Record Status ... 41
7.9.3 User ID ... 42
7.9.4 Serial number ... 42
7.9.5 Channel ... 42
7.9.6 Record number ... 42
7.9.7 Data format ... 42
7.9.8 Record length ... 42
7.10 Over-range and under-range ... 43

8 HOST PC CONNECTION ... 44
8.1 PCI Express interface ... 44
8.2 Using several units ... 44
8.2.1 Using several digitizers from a single application. ... 44
8.2.2 Using several digitizers from a several applications. ... 44

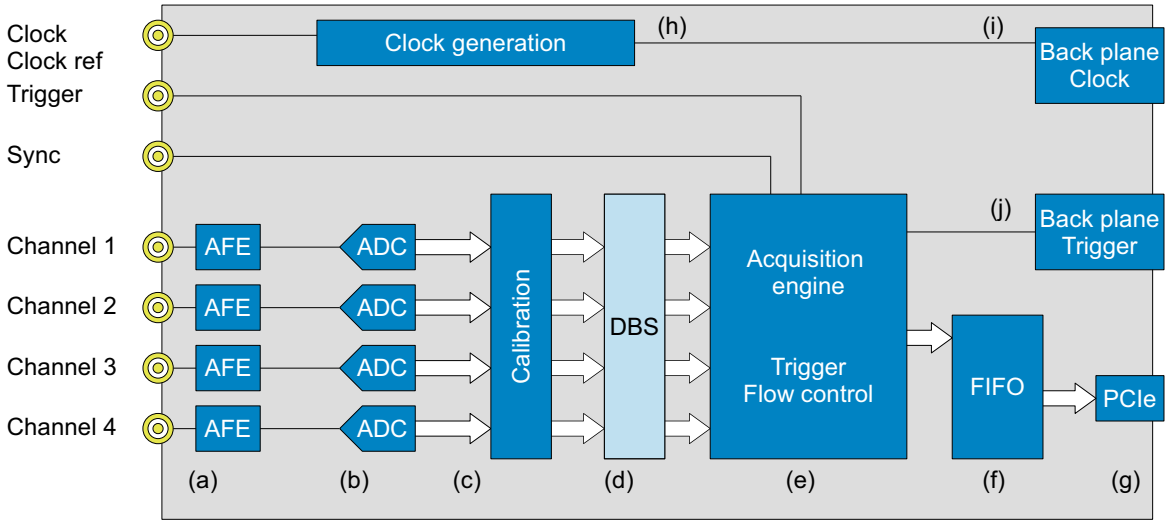
9 REFERENCES 45

1 INTRODUCTION

The purpose of this manual is to explain how the digitizer is operated. The datasheet [1] contain parameters for the specific versions of digitizer. References to software commands are made. In some places, pseudo code is used for description. See [2] for details on how to use the software commands and see [3] for general guidelines on programming the digitizer.

1.1 ADQ8-4X Architecture

The ADQ8-4X architecture is given in Figure 1. References to the corresponding sections with further information are also included.



#	DESCRIPTION	REFERENCE
a	Signal conditioning analog front-end.	2
b	High speed and high resolution A/D converters.	3
c	Calibration of gain and offset.	2
d	Teledyne SP Devices' proprietary technology for signal quality enhancement; DBS for baseline stability in pulse data systems.	3
e	Acquisition engine that handles triggers and controls the data flow.	4, 7
f	Data FIFO to buffer data before transmission to the host PC.	7
g	The data transfer to the host PC is through a PCIe Gen 2 interface.	7
h	Flexible clock generator.	5
i	Backplane clock reference for large scale integration.	6
i	Backplane clock triggers large scale integration.	4

Figure 1: ADQ8-4X architecture.

1.2 Fundamental design properties

There are some fundamental design properties that are necessary to understand before continuing.

1.2.1 Data format

The ADC components of ADQ8-4X has 10 bits resolution, while the data format inside the ADQ8-4X and out to the host PC is 16 bits. The 16 bits from the ADCs are MSB aligned in this 16 bit data word. Thus the 6 LSBs are zero.

The number representation is 2's complement. The full scale maximum code is then 32 704 and the full scale minimum code is –32 768. Overflow or underflow at any position in the signal path will saturate the data and turn on an overflow flag. See **Section 7.10** for more information on over- and under-flow.

1.2.2 Calibration

During the factory calibration procedure the analog properties are measured and parameters for a digital compensation are computed. An analog deviation in the front-end is thus compensated for by the inverse function in the digital signal processing part.

Example 1: With the variable gain –VG option (always included with the ADQ8-4X), the user requests a range. The closest available setting is selected and the actual range is returned to the user for being used in the user's algorithms, see **Section 2.1**.

Example 2: The full scale signal range of the ADQ is measured in production and the ***SetGainAndOffset*** *function is used for adjusting to the correct signal range.*

1.2.3 Data acquisition nomenclature

Table 1 defines some key data acquisition terms.

Table 1: Data recording nomenclature.

PARAMETER	DESCRIPTION	REF
ADQ	Collective name for digitizers from Teledyne SP Devices.	
Analog	Analog signal is the input to the digitizer. This is the signal to be digitized.	
Waveform	Analog signal with a distribution in time. This is digitized into a record.	
Sample	An analog signal level is digitized into a sample, that is a numerical value.	
SYNC	Physical connector on the front panel.	**4.8**, **6.1**
Record	A set of consecutive samples is called a record. An analog waveform is digitized into a record of samples.	**4**
TRIG	Physical connector on the front panel.	**4.8**, **6.1**
Trigger	Trigger is an event that starts acquisition of a record.	**4.3**
Timestamp	Timestamp is a real-time value that identifies when a trigger happened. The timestamp gives timing information for each sample.	**4.4**
GSPS	Giga-samples per second (10^9). Clock frequency [Hz] and sample rate [SPS] are both used to denote speed.	
MSPS	Mega-sample per second (10^6).	
DC-offset	This is an analog DC level which is added to the analog input signal inside the digitizer to vertically move the analog signal to fit within the range of the digitizer. This effectively doubles the ENOB for a unipolar signal.	**2**

1.2.4 Sampling clock frequency

The ADQ8-4X is designed for the specified 2 GHz clock frequency only with the firmware FWDAQ. The firmware FW4GDAQ enables sampling at 4 GSPS. A different sampling rate can be achieved by using the sample skip function, **Section 5.10**.

1.2.5 System clocks

The different parts of the digitizer operate on different clock rates

The sampling of the analog signal is done on the sampling clock of the ADC (see **Section 1.2.4**).

The external trigger input has a trigger clock which is higher than or equal to the sampling clock for high trigger time precision (4 GHz). This clock frequency is the same regardless of selected sampling frequency.

The PCIe host PC connections has its own clock system.

All other interfaces operate on the data processing clock of the FPGA at 250 MHz. This clock is referred to as the Data Clock.

See **Section 5.1** for more details on the clock system.

1.2.6 Analog signal range

The analog signal range (**ACTUAL_ANALOG_RANGE**) is symmetrical around zero. The value of the analog signal range it is depending on the gain setting. With for example a range of, 500 mV_{pp}, the analog input signal can vary from –250 mV to +250 mV and the range can be moved from [–0 mV +500 mV] to [–500 mV +0 mV] by the DC-offset feature, **Section 2.3**.

The maximum digital code 2^15 represents an analog signal with a level **ACTUAL_ANALOG_RANGE / 2** at the input. A specific analog signal **ANALOG_LEVEL** will then be represented by the following digital code:

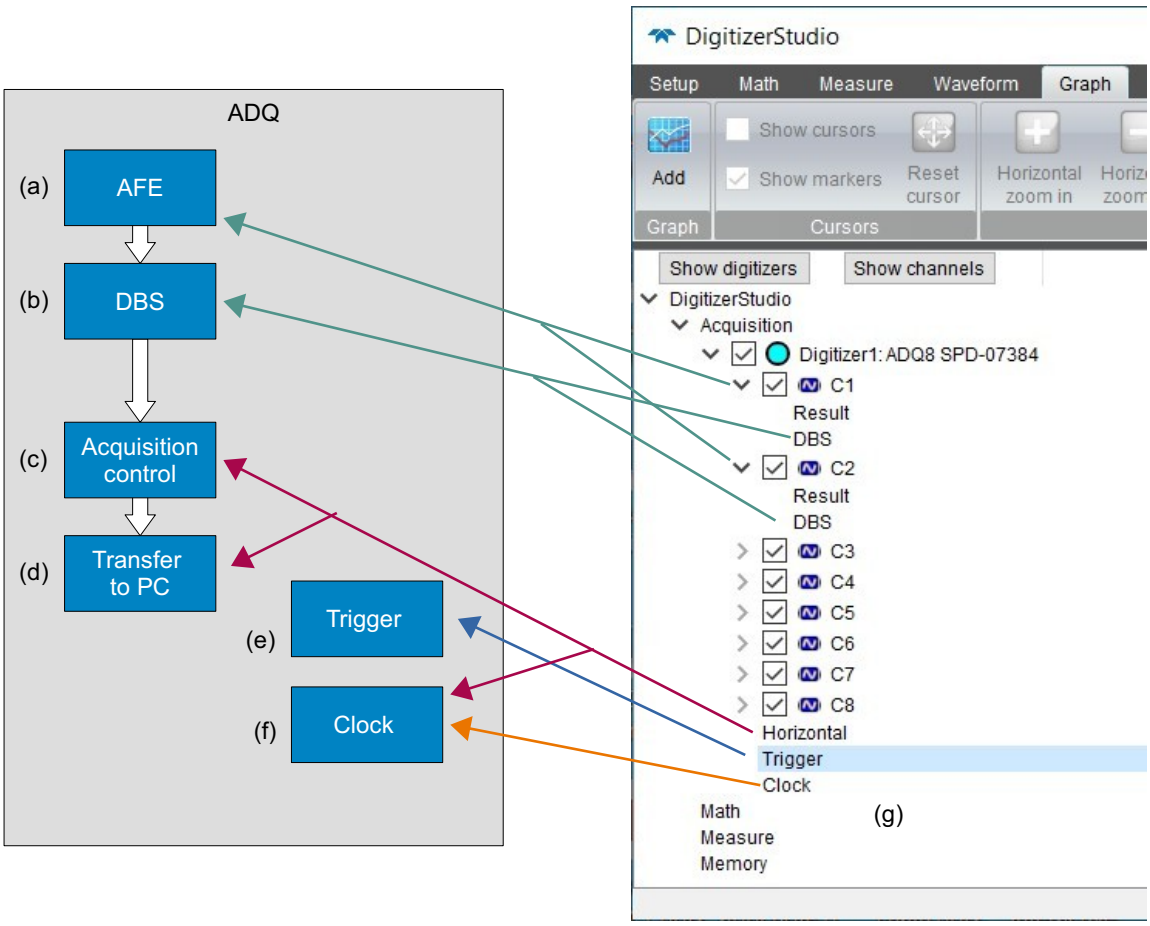
DIGITAL_CODE_LEVEL = ANALOG_LEVEL / (ACTUAL_ANALOG_RANGE / 2) * 2^15 **(1)**

A specific code **DIGITAL_CODE_LEVEL** then represent the analog level as:

ANALOG_LEVEL = (DIGITAL_CODE_LEVEL / 2^15) * (ACTUAL_ANALOG_RANGE / 2) **(2)**

1.3 Digitizer Studio overview

The application software Digitizer Studio is an easy way to operate the ADQ8-4X. Digitizer Studio is a graphical interface. The different blocks of the ADQ digitizer are represented by several views in Digitizer Studio. These blocks are linked as in **Figure 2**. Digitizer Studio is described in **[5]**.



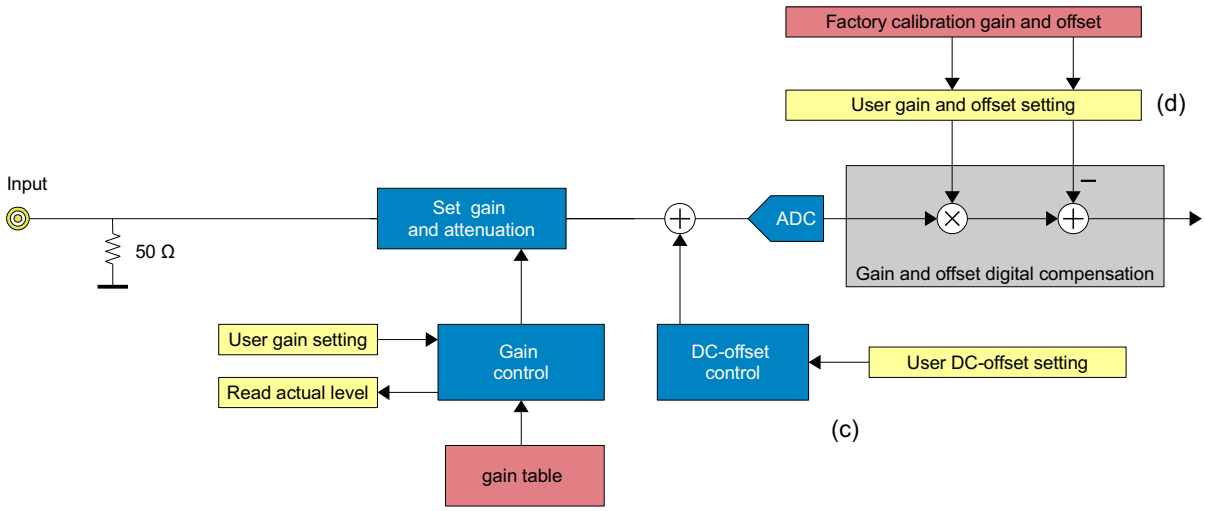
#	BLOCK	DESCRIPTION	REF
a	Analog front-end	Setting up analog input parameters, digital signal transformation	2
b	DBS	Digital Baseline Stabilizer signal conditioning IP	3
c	Acquisition control	Setting up the recording of data	7
d	Transfer to PC	Data transfer methods to move the data to the PC	8
e	Trigger	Trigger, synchronization control	4
f	Clock	Timing control	5
g	Digitizer Studio	Digitizer Studio view	[5]

Figure 2: Digitizer Studio blocks.

2 SETTING UP THE ANALOG FRONT-END

2.1 ADQ8-4X AFE block diagram

The analog front-end setup for ADQ8-4X found in **Figure 3**.



#	DESCRIPTION	USER COMMAND	REF
a	The analog input impedance 50 Ω to GND		
b	On ADQ8-4X-VG, the gain can be set in a discrete number of steps. By requesting a certain range, the closest available setting is activated. The actual range that is set is returned to the user. Note that the compensation has to be done in the software. Changing the input range requires a settling time of 1 s.	*SetInputRange*	2.2
c	Set a DC offset for better using the signal range when the signal is unipolar. Changing the DC offset requires a settling time of 1 s.	*SetAdjustableBias*	2.3
d	The gain and offset is calibrated using a 32 MHz -1dBFS sine wave signal. The digital compensation corrects the offset and the gain at this frequency. The user can access this block to set a different gain and offset. Note that this is a digital gain and offset adjustment and not the same as selecting analog input range in the –VG option.	*SetGainAndOffset*	2.4

Figure 3: AFE control.

2.2 Set analog input range

On ADQ8-4X–VG, the input range is variable. The requested input signal range is sent to the API, which reads available settings and return the best selection. The actual value of each range is available for calculations according to the following:

DESIRED_ANALOG_RANGE is the requested range set *into* *SetInputRange*.
ACTUAL_ANALOG_RANGE is the actual calibrated range of the device returned *from* *SetInputRange*.
The maximum digital code 2^15 represents an analog signal with a level ***ACTUAL_ANALOG_RANGE / 2*** at the input. A specific analog signal ***ANALOG_LEVEL*** will then be represented by the following digital code:

DIGITAL_CODE_LEVEL = ANALOG_LEVEL / (ACTUAL_ANALOG_RANGE / 2) * 2^15 **(3)**

A specific code ***DIGITAL_CODE_LEVEL*** then represent the analog level as:

ANALOG_LEVEL = (DIGITAL_CODE_LEVEL / 2^15) * (ACTUAL_ANALOG_RANGE / 2) **(4)**

2.3 Set analog DC-offset

A user-controlled DC-offset is available. The analog DC-offset is applied to the signal to better adopt to the signal range of the digitizer. The analog range is by default set symmetrical around zero. If the signal is unipolar or heavily unsymmetrical, the DC-offset function can adjust the signal to an optimal vertical position for the A/D converter. In this way, the full 10 bits can be used for representing the unipolar pulse. The DC-offset is set with the command ***SetAdjustableBias***.

The DC-offset is set in digital codes ***DC_OFFSET_CODE*** in the range [-2^15: 2^15-1], which correspond to an analog signal level in the range:

[–ACTUAL_ANALOG_RANGE / 2: ACTUAL_ANALOG_RANGE / 2]. **(5)**

To determine the parameter of ***SetAdjustableBias*** to get a DC-offset at the voltage level ***DC_OFFSET_ANALOG***, use:

DC_OFFSET_CODE = round (DC_OFFSET_ANALOG / (ACTUAL_ANALOG_RANGE / 2) * 2^15) **(6)**

Since the digitizer has higher resolution than the intrinsic accuracy of the DC-offset generator, the actual digital codes read out from the ADQ may differ from the expected level. For accurate baseline measurements, the Digital Baseline Stabilizer (DBS) offers a digital correction of the baseline to an accuracy of 22 bits, **Section 3.1**.

2.4 Adjusting the digital gain and offset

The digital gain and offset block is primarily intended for factory calibration but it may also be accessed by the user, and offers an efficient way of scaling the signal to suit processing in the PC.

The default setting is the calibration parameters ***CAL_GAIN*** and ***CAL_OFFSET***. The raw data from the A/D converter, ***ADC_RAW_CODE***, is corrected with the calibrated values according to:

DIGITAL_OUTPUT_CODE = ADC_RAW_CODE * CAL_GAIN – CAL_OFFSET **(7)**

The user can override these settings by using the software command ***SetGainAndOffset***. The parameter ***USER_GAIN*** and ***USER_OFFSET*** can be applied in two ways; relative to the calibrated value or relative to the raw code.

The normal mode of operation is to apply the gain and offset settings relative to the calibrated data as

DIGITAL_OUTPUT_CODE =
ADC_RAW_CODE * CAL_GAIN * USER_GAIN – CAL_OFFSET – USER_OFFSET. **(8)**

By setting bit 7 in the channels parameter, the calibration data is overridden as:

DIGITAL_OUTPUT_CODE = ADC_RAW_CODE * USER_GAIN – USER_OFFSET **(9)**

To get the raw code, ***ADC_RAW_CODE***, use ***SetGainAndOffset(128+CHANNEL,1024,0)***.

3 SIGNAL QUALITY ENHANCEMENT

3.1 Digital Baseline Stabilizer

The Digital Baseline Stabilizer, DBS, is designed for pulse data measurement where high accuracy relative a known baseline is required. The key features of DBS are:

- Tracks and compensates for baseline variations from, for example, temperature and aging.
- Suppresses pattern noise[1] to 22 bits precision.
- Automatically locks the baseline to a user defined-value.

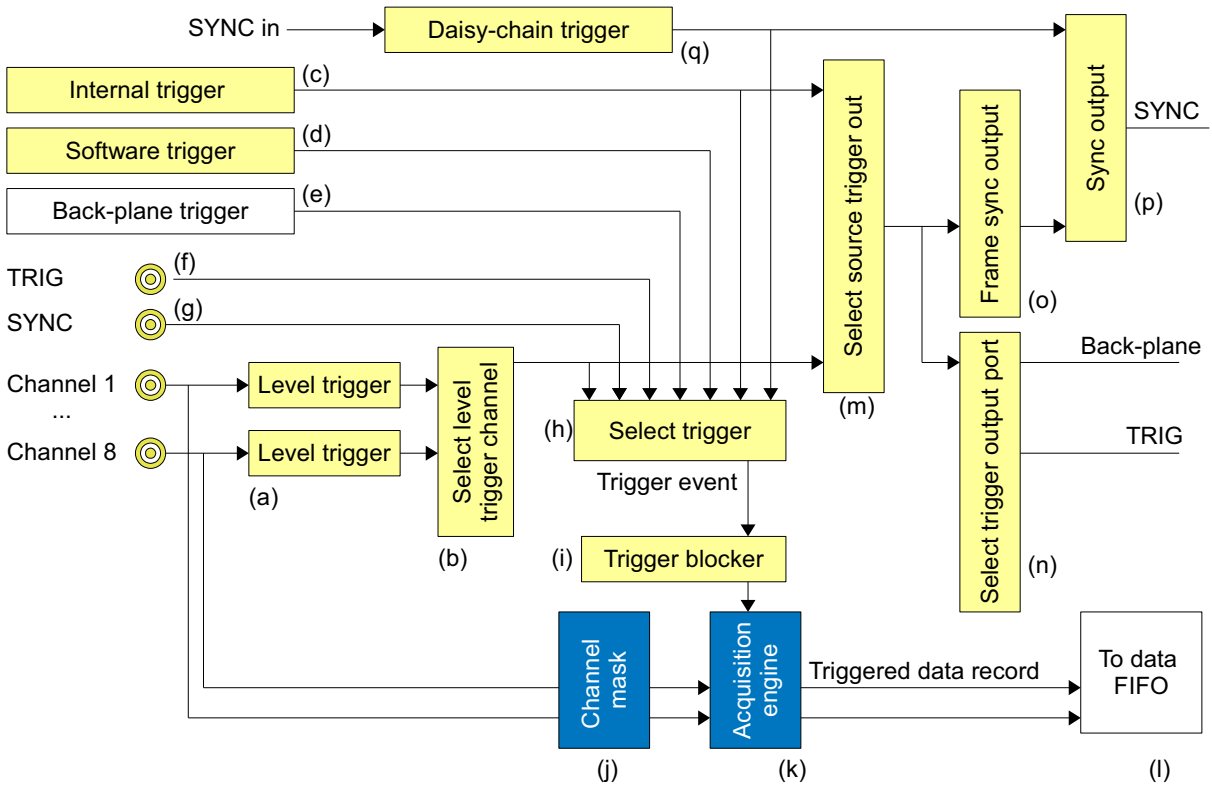
Note that DBS is off at power up. DBS has to be activated by the user's application software. The time when DBS is activated is important. To get a good initial estimate, DBS should be activated when there is very little signal energy present at the input. If there is too much signal power in the initial estimate, the convergence of DBS is slowed down.

Note that DBS is defined for systems with a baseline and distinct short pulses. DBS is not intended for sinusoidal type of signals.

1. Pattern noise is systematic errors that may arise from the actual design of the ADC IC or the board design.

4 TRIGGER

4.1 Trigger block diagram



#	DESCRIPTION	USER COMMAND	REF
a	Each analog input is connected to a level trigger block.	*SetupLevelTrigger*	4.10
b	Select on which channel to trigger (when using level trigger).	*SetupLevelTrigger*	4.10
c	Internal trigger generator.	*SetInternalTriggerPeriod*	4.11
d	A software trigger is available for user control.	*SWTrig*	4.7
e	External trigger input from backplane (different in PXIe and MTCA)		4.9
f	External trigger input on front panel connector TRIG.		4.8.1
g	External trigger input on front panel connector SYNC.		4.8.2
h	Select which type of trigger to activate.	*SetTriggerMode*	
i	The trigger blocking function controls the flow of triggers to the acquisition engine.	*SetupTriggerBlocking*	4.5
j	Select which channels to record data from.	*MultirecordSetChannelMask*	
k	Acquisition engine creates a record from streaming data		7
l	Records are sent to data FIFO for transfer to the host PC		7
m	Activate trigger output and select source.	*SetupTriggerOutput*	4.12
n	Select trigger output connector	*SetupTriggerOutput*	4.12
o	Frame sync is a function that can group triggers.	*SetupFrameSync*	4.12.2
p	Select source for sync output	*SetupTriggerOutput*	
q	Daisy chain trigger for synchronization		4.13

Figure 4: Trigger source selection and setup

4.2 Introduction

The digitizer can be triggered in various ways with a number of different internal and external trigger sources. Selected events in the trigger module can also be output to trigger external equipment. The selection of trigger source is illustrated in **Figure 4**.

4.3 Position of the trigger in the data

The trigger position relative to the data record is controlled by the parameters pretrigger and trigger delay.

The pretrigger buffer enables capturing data prior to the trigger event, **Figure 5**. Use the command ***SetPreTrigSamples*** to define the pretrigger.

The trigger delay postpone the start of the acquisition of the data record specified number of samples after the trigger event, **Figure 6**. Use the command ***SetTriggerHoldOffSamples*** to define the trigger delay.

The timing of the trigger is read from the record header (**Section 7.9**). The parameters ***TIME_STAMP*** and ***RECORD_START*** are explained in **Section 4.4.1**.

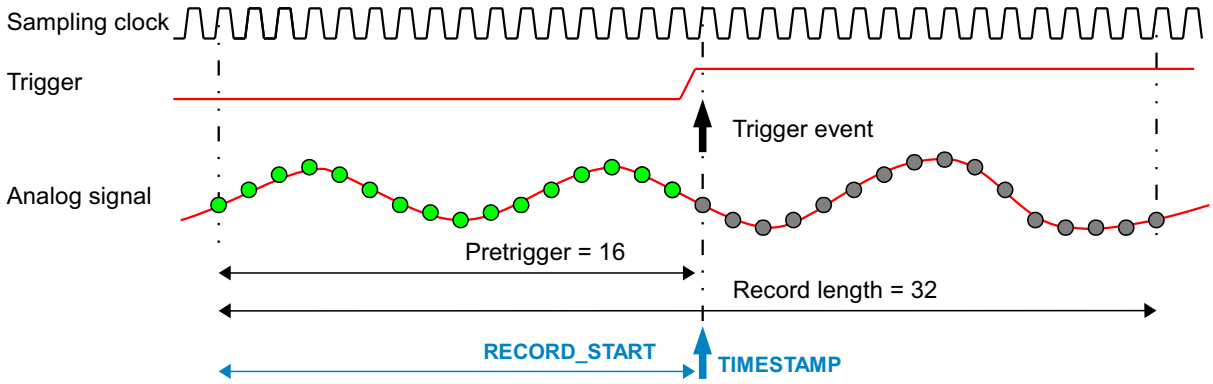


Figure 5: Pretrigger timing.

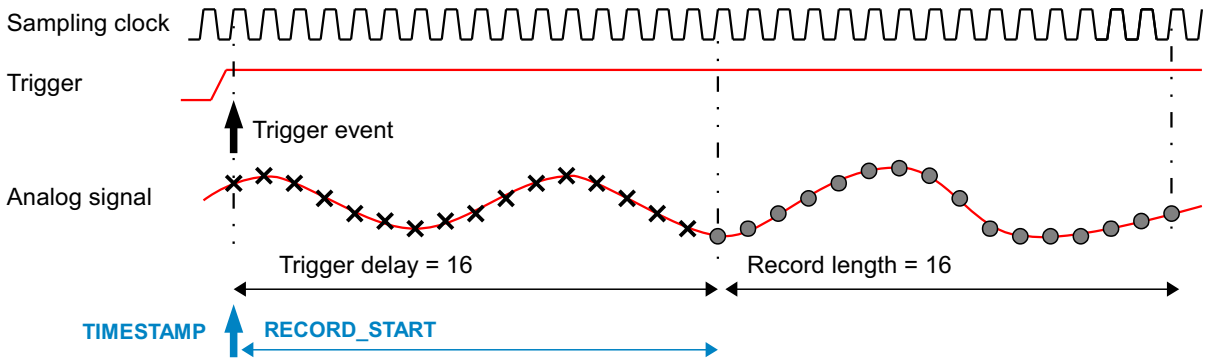


Figure 6: Trigger delay timing.

4.4 Timestamp

4.4.1 Timestamp definitions

The timestamp counter enables real-time measurement of a trigger event. It is used for tagging an event, sorting events in time or comparing timing between events.

The timestamp information consist of three parts, which uniquely defines the timing:

- ***TIME_STAMP*** measures the time of the trigger event relative to other trigger events.
- ***RECORD_START*** is the time between the trigger event and the start of the record. For a pretrigger, this is a negative value. When trigger delay is used, this is a positive value.
- ***SAMPLE_PERIOD*** is the length of a sample period. The sample period may vary with sample skip setting and clock frequency of the digitizer.

The ***TIME_STAMP***, ***RECORD_START***, and ***SAMPLE_PERIOD*** are measured in the unit ***TIME_BASE*** = 25 ps. See **Example 3** on how to use these parameters. These parameters are available in the record header, see **Section 7.9.1**.

The timestamp counter is based on the internal clock of the digitizer. The internal clock is based on the selected clock reference. The timestamp is thus also related to the clock reference. When the clock reference is phase-locked to an external source, the timestamp counter is running synchronized with the external source. On the other hand, if the digitizer is free running, the timestamp counter also free running. (See **Section 5** for all details about the clock system of digitizer.)

The timestamp counter measures the time from a reference time point to the trigger event. The reference time point is when the counter is started or reset. See **Section 4.4.2** for information on how to reset the timestamp counter.

Example 3: Assume an ADQ8-4X sampling with a clock frequency at 2 GSPS. The pretrigger is set to 32 samples and the external trigger is used. The following parameters are returned:
TIME_STAMP *= 5010*
RECORD_START *= –650*
SAMPLE_PERIOD *= 20*
TIME_BASE *= 25 ps*

The time for the trigger was then
TRIGGER_TIME *=* ***TIME_STAMP*** *** ***TIME_BASE*** *= 125250 ps = 125.25 ns*

The time for the first sample in the record is
RECORD_TIME *= (****TIME_STAMP*** *+* ***RECORD_START****) ** ***TIME_BASE*** *= 109.000 ps = 109 ns*

The time between two samples are
SAMPLE_TIME *=* ***SAMPLE_PERIOD*** *** ***TIME_BASE*** *= 0.5 ns*

The time from the record start to the trigger is
RECORD_START *** ***TIME_BASE*** *= –16.25 ns.*

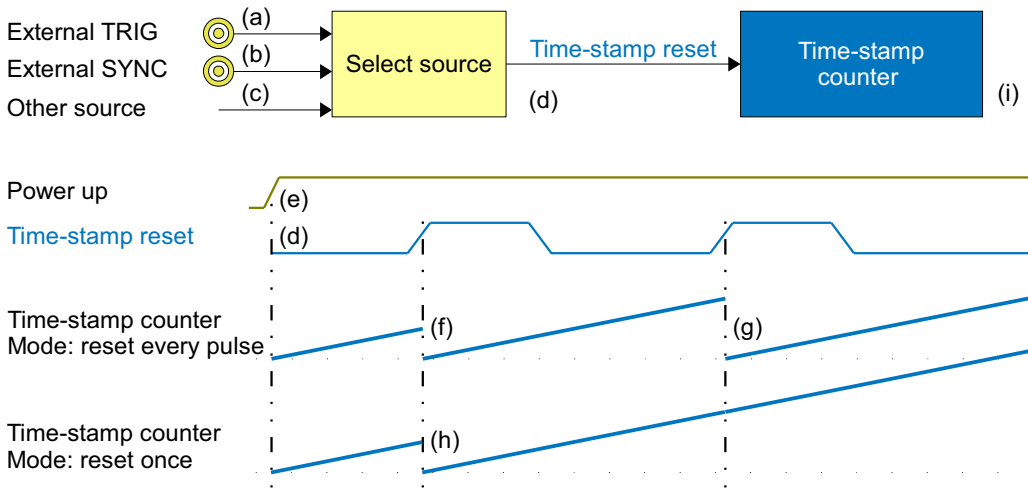
The number of samples between the record start (first sample in the record) and the trigger event is
(***TRIGGER_TIME*** *–* ***RECORD_TIME*** *) /* ***SAMPLE_TIME*** *=*
| ***RECORD_START*** *| /* ***SAMPLE_PERIOD*** *= 32.5 samples*
This is the expected 32 samples set in the pretrigger and 1/2 sample in subsample precision in the external trigger.

4.4.2 Timestamp reset

When powering up a system with many boards, the timestamp counter in each board will start. But the counters start at different times in different physical digitizers. There are four methods for resetting the timestamp and get a common time reference in all the digitizers in the system:

1. The timestamp counter is reset at power-up. This methods does not, however, have absolute precision, since the timing of the power up is not defined. In a multi-board system, the timestamp will differ between the boards.
2. With a software reference reset the user has full control of the reset procedure. A reference time point is created in the users application, which is used for aligning time-stamps in different units. After power-up the user runs a custom timestamp reset sequence including:
   - Apply a reference signal to all boards.
   - Trigger a record on the reference signal.
   - Read the time-stamps from the records and call this reference; ***TIME_STAMP_REFERENCE***.
   - Start the experiment and subtract the timing reference from each record as ***TIME_STAMP*** = ***TIME_STAMP_OF_RECORD*** – ***TIME_STAMP_REFERENCE***.
3. The third method is to apply an external trigger to reset the timestamp reset, **Figure 7**. This method has the possibility to synchronize several boards to full precision of the external trigger. See **Section 4.5**. The sequence of operation is:
   - ***DisarmTimestampSync***
   - ***SetupTimestampSync***
   - ***ArmTimestampSync***

   The number of reset pulses are counted and the information is stored in the record header, **Section 7.9**. However, if there are no triggers accepted, there will be no record headers available. To verify that there is activity going on, the number of reset pulses can also be read from a register via ***GetTriggerBlockingGateCount***.
4. The fourth method is to reset the timestamp with the sync signal, **Figure 7**. The difference between using the external trigger and the sync is that the external trigger has the a sample resolution while the sync timing resolution is controlled by the Data Clock in the FPGA. Note that the backplane triggers work in the same way as the sync signal.



#	DESCRIPTION	USER COMMAND	REF
a	External trigger input signal on front panel connector.		4.8
b	External sync input signal on front panel connector.		4.8
c	Other available sources (see *SetTriggerMode* for a list)		4.8
d	Select source for reseting timestamp.	*SetupTimestampSync* *DisarmTimeStampSync* *ArmTimeStampSync*	4.4.2
e	Timestamp counter value is reset at power-up of the digitizer.		4.4.2
f	Reset the timestamp counter on each pulse of the selected source. Timestamp is then measuring time relative the previous reset signal.		4.4.2
g			4.4.2
h	Reset the timestamp counter only on the first pulse of the selected signal. The external signal is then a systems synchronization signal.		4.4.2
i	The number of times the time stamp has been reset can be read from a register.	*GetTriggerBlockingGateCount*	4.4.2

Figure 7: Timestamp reset from external trigger.

4.5 Blocking triggers for synchronization

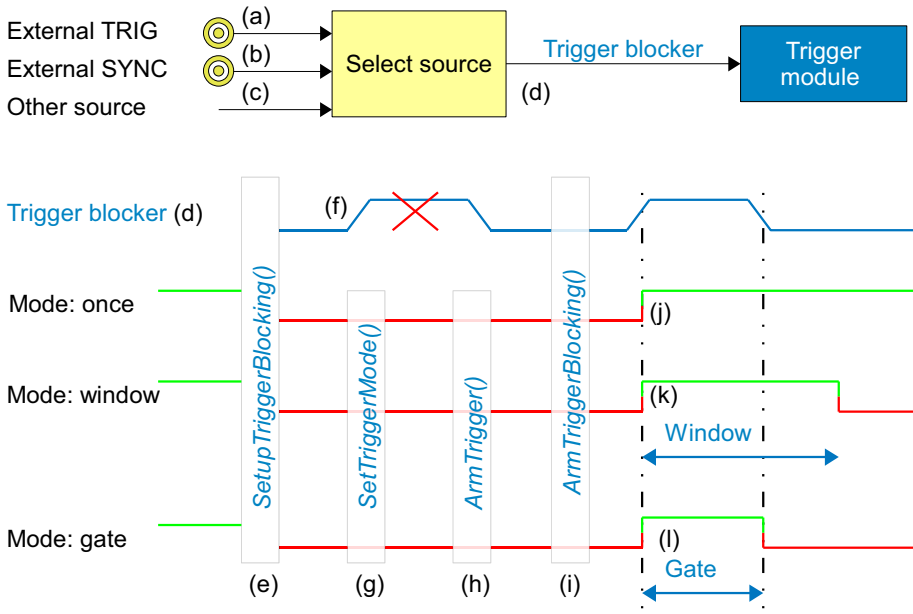
4.5.1 Function overview

In order to synchronize the acquisition to external equipment or to other ADQ digitizers, there is a mechanism for controlling the flow of triggers. The trigger blocking function allows the user to select when to activate incoming triggers, **Figure 8**. The basic function of this block is to use the SYNC signal to frame the trigger signals; for each period of the blocking function, a set of triggers are allowed and and framed by the blocking event. This creates groups of triggers that belong together. The modes of operation for trigger blocking are shown in **Figure 8** (j, k, l).

To avoid that the boards start to produce a large amount of records out of sync, all trigger events can be blocked until the triggers are released by a separate shared signal, **Figure 8** (d). By combining the trigger blocking and the timestamp reset, the timestamp is aligned to the start of the acquisition. The trigger blocker source can be most available trigger sources, **Figure 8** (a, b, c).

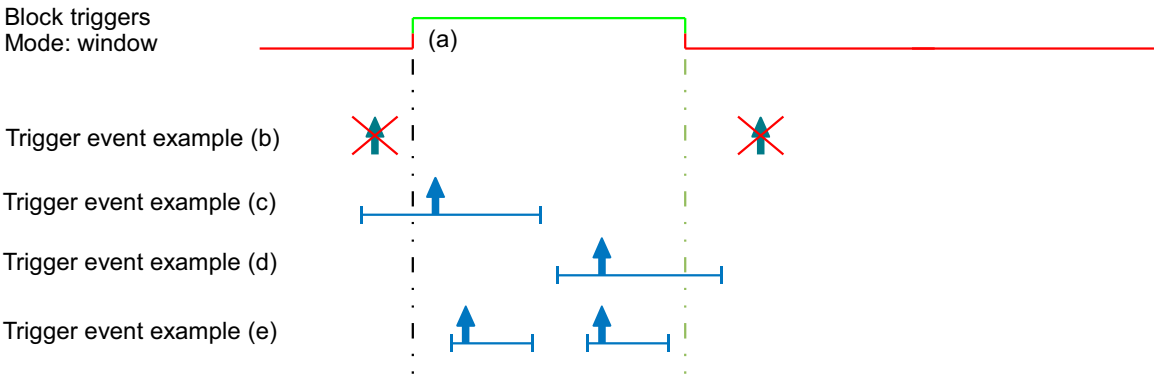
Note the order of the commands for activating triggers and trigger blockers, **Figure 8** (e, g, h, i).

Figure 9 illustrates how the triggers are accepted or rejected in the window mode.



#	DESCRIPTION	USER COMMAND	REF
a	External trigger input signal on front panel connector.		4.8
b	External sync input signal on front panel connector.		4.8
c	Other available sources (see *SetTriggerMode* for a list).		
d	Select source for blocking triggers.	*SetupTriggerBlocking*	4.5
e	Before activating the trigger blocking any selected trigger may pass.	*SetupTriggerBlocking*	
f	This signal is ignored as the trigger blocker is not armed		
g	Select trigger source	*SetTriggerMode*	
h	Start receiving triggers. Note that triggers are still blocked.	*ArmTrigger*	
i	The unblocking of triggers is armed and can be activated by (d).	*ArmTriggerBlocking*	
j	Triggers are blocked until the first accepted blocker signal.	*SetupTriggerBlocking*	4.5
k	The trigger blocker can also be set up with a window function where triggers are accepted or rejected during a user-defined window.	*SetupTriggerBlocking*	4.5.1
l	The trigger blocker can also be set up as a gate where triggers are accepted during a gated time set by signal (d).	*SetupTriggerBlocking*	4.5.1

Figure 8: Blocking and gating of triggers.



#	DESCRIPTION	USER COMMAND	REF
a	The trigger blocker in window or gate mode allows triggers during a certain period.	*SetupTriggerBlocking*	4.5.1
b	Example of rejected triggers outside the window.		
c	Trigger within the window is accepted and a data record is recorded. Note that the pretrigger starts before the window.		
d	Trigger within the window is accepted and a data record is recorded. Note that the record extends after the window.		
e	Several triggers within the same window.		

Figure 9: Trigger blocker examples.

4.5.2 Block triggers once

The mode for blocking triggers once is illustrated in **Figure 8** (j). This mode is used for starting the operation simultaneously in several units. The first time the trigger blocking signal is applied, the triggers are allowed through. Here is the motivation for this mode:

There is no way to broadcast a software command to several units. When setting up acquisition in several units, they will therefore be activated at different times. By using the trigger blocker, an electrical signal to all units can activate them simultaneously. The trigger blocking signal can be external or it can be generated internally using the bussed connections proposed in **Figure 14**.

4.5.3 Windowing triggers

The window mode for blocking triggers is illustrated in **Figure 8** (k). The edge of the trigger blocking signal is activating a window of user-defined length which allows triggers through. There is also a mode where triggers are blocked during the window.

The window mode can be used for two-dimensional triggering where, for example, the trigger signal is a point trigger and a sync signal is a line trigger.

4.5.4 Gating and windowing triggers

The gate mode for blocking triggers is illustrated in **Figure 8** (l). The length of the window where triggers are accepted is equal to the length of the trigger blocking signal.

4.5.5 Programming sequence for using trigger blocking

The order of commands is important when programming the trigger blocking. This is because the ADQ digitizer interact with other external equipment. This external equipment is synchronized to the digitizer through the setup procedure.

The setup of the functions has to be aligned with the expected operation. For example, by asserting the trigger blocking through the ***SetupTriggerBlocking*** command before setting up the acquisition, no triggers are let through before the digitizer is ready.

The function on the digitizer has to be activated (armed) in reverse order compared to the data flow. This means that one stage is set up to be prepared to receive data before the preceding stage is set up to generate data. This is especially important in streaming applications where the DRAM FIFO may overflow if the triggering is activated before the read-out to the host PC has started.

4.6 Trigger jitter

4.6.1 Trigger jitter definitions

The triggering operation is subject to two different types of jitter, **Figure 10**.

1. At the trigger input is a Gaussian distributed jitter which affects the timing of the incoming trigger signal edge. This jitter is called excess jitter and is caused by noise in the input stage. The RMS value of this excess jitter is 25 ps.
2. The actual sampling process causes a timing uncertainty. Since the trigger is sampled with the trigger clock, the time points for reading the trigger are discrete. The difference between the incoming physical trigger signal and the digital representation of the trigger is a stochastic variable with a rectangular distribution. The RMS value of such a process is ***TRIGGER_CLOCK_PERIOD***/sqrt(12). The highest resolution is achieved with an external trigger connected to the TRIG connector. ADQ8-4X has a trigger clock at 4 GSPS, ***TRIGGER_CLOCK_PERIOD*** of 250 ps and a trigger jitter of 72 ps RMS (theoretical value), **Section 4.6.4**.

See **Table 2** for time resolution all the external trigger sources.

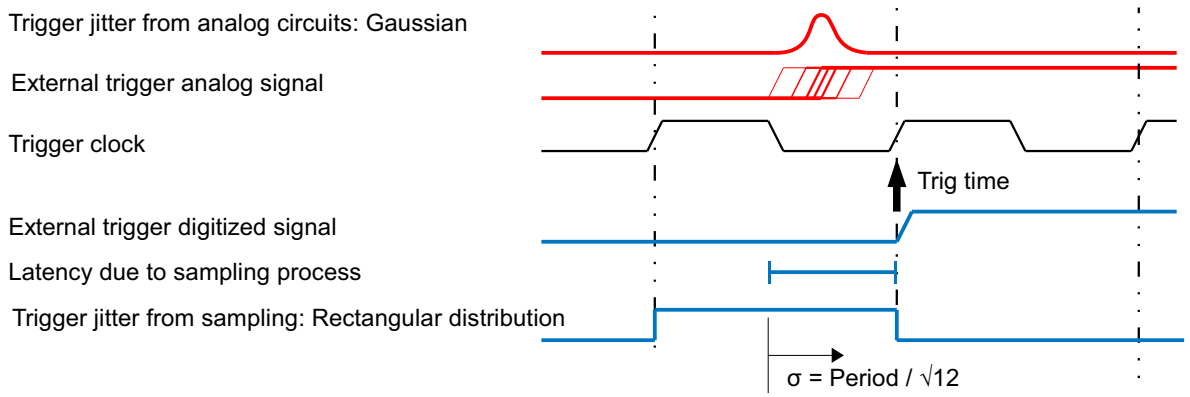


Figure 10: Sources of jitter on the trigger signal.

4.6.2 Asynchronous triggering

If the trigger signal is not phase-locked to the reference clock it is called asynchronous. This trigger does not have a well-determined relation to the sampling clock and will appear at various positions within the sampling period. The time resolution of an asynchronous trigger connected to the TRIG input is set by the Trigger Clock (4 GHz). The time resolution for other triggers is determined by the Data Clock (250 MHz).

The asynchronous trigger will be exposed to both trigger sources from **Section 4.6.1**. These independent stochastic processes are added to 72 ps. See **Table 2** for time resolution of all the external trigger sources.

There are some advantages with the asynchronous trigger:

- Any pattern noise will be reduced in repeated measurements.
- The trigger resolution of 250 ps can be used for accurate timing calculations. The ***TIME_STAMP*** contains the information about the trigger time. See **Section 4.6.4**.

4.6.3 Synchronous trigger

A synchronous trigger is phase locked to the clock of the digitizer. The trigger source needs access to the clock reference of the digitizer. There are three ways to achieve this synchronization:

1. Output the internal clock reference of the ADQ and send it to the trigger source, **Section 5.9**.
2. Use the clock reference of the trigger source as clock reference for the ADQ, **Section 5.5**.
3. Use the internal trigger of the ADQ and output it to trigger the external equipment, **Section 4.11**.

When the trigger is phase-locked to the clock reference the timing is comparable to a digital signal which defines setup and hold time.

4.6.4 Extended trigger resolution

The basic sampling process maps the trigger to the sampling rate of the digitizer. There is also additional trigger time information available; Extended trigger resolution, **Figure 11**.

The Trigger Clock is operating at 4 GHz. This means that the time resolution of the trigger input TRIG is reduced to 250 ps.

Note: The extended trigger resolution is available on triggers connected to TRIG only.

The extended trigger information is included in the timestamp information, **Section 4.4**.

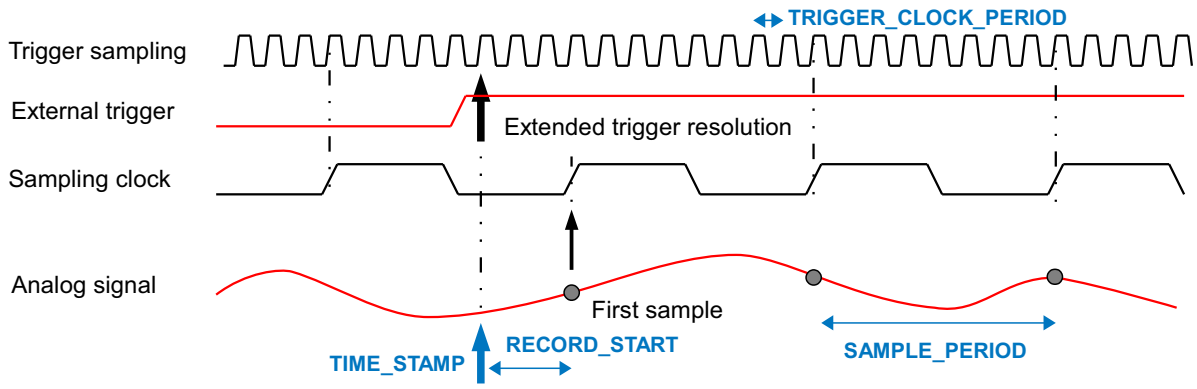


Figure 11: Extended trigger resolution timing for ADQ8-4X

The position of the first sample is rounded up from the trigger position. The parameter ***RECORD_START*** tells where the trigger was. Referring to **Figure 11**, the ***RECORD_START*** parameter can have values in the range –250 ps up to +500 ps. A positive value means that the first sample is after the trigger. The given range is without pretrigger or trigger delay. With pretrigger or trigger delay, the ***RECORD_START*** will have a larger (absolute) value.

4.7 Software trigger

The software trigger is a user command that triggers the ADQ. This is for direct user control of the acquisition and is useful for looking at continuous signals where the timing of the trigger is not critical. The software trigger is sent through several layers of software and the time when it arrives to the digitizer cannot be predicted. However, the time when it actually arrived can be read from the time stamp in the record header, **Section 4.4**.

The software trigger may also be used for time-out function. This is a way to discover faults in the setup. When the device do not trigger for some reason within a certain time frame, a software command sequence may be sent and the data can be analyzed to find out what is wrong, **Example 4**.

Example 4: A time-out function using software trigger can be implemented like this:

1. Time-out occurs

2. *DisArmTrigger*
3. *SetTriggerMode*(“software trigger”)
4. *ArmTrigger*
5. *SWTrig*
6. Read data and analyze the situation

4.8 External Trigger Inputs

An external trigger is a dedicated signal on a dedicated input to the ADQ. There are several inputs for external trigger, **Table 2**.

CONNECTOR	DESCRIPTION	TIME RESOLUTION	TOTAL JITTER	IMPEDANCE	TRIG LEVEL	REF
TRIG	External trigger on front panel.	250 ps	76 ps	50 Ω / 500 Ω	SW contr.	4.8.1
SYNC	Sync signal on front panel.	4 ns	1.2 ns	50 Ω / 500 Ω[1]	SW contr.	4.8.2
STARB	Backplane trigger in PXIe systems. Requires trigger-timing card.	4 ns	1.2 ns	PXIe standard	PXIe standard	4.9.1

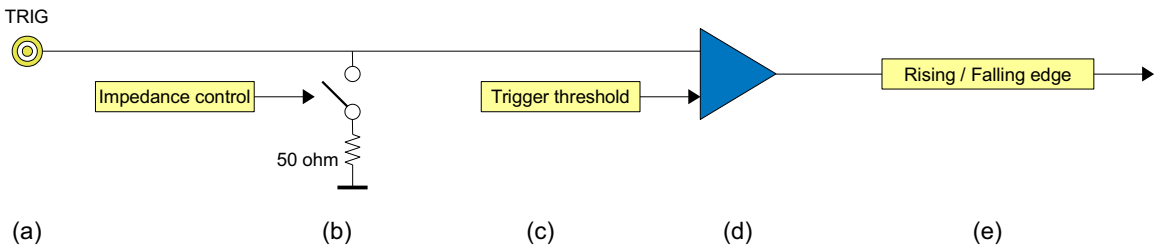
Table 2: External trigger inputs.

1. 500 Ω option available on MTCA form factor only.

4.8.1 External trigger TRIG front panel connector

The block diagram for the TRIG is shown in **Figure 12** and related parameters are listed in **Table 2**. The user can control the external trigger function for adapting it to the system in the following ways:

- The input impedance can be set in 50 Ω (default) or high impedance 500 Ω mode, see **Section 4.8.3**.
- Configure the threshold level.
- Set the trigger edge to rising or falling to adjust to the polarity of the trigger signal.



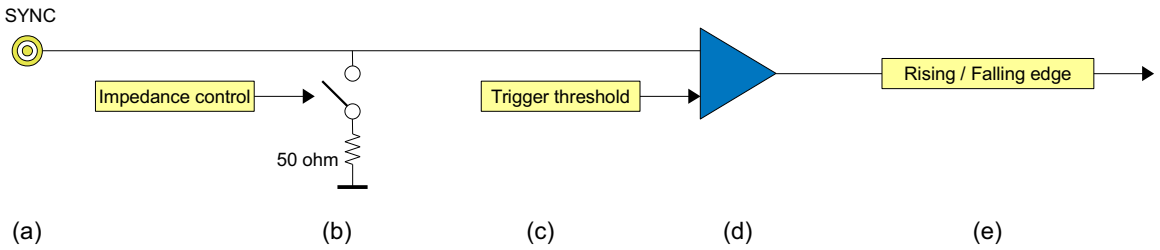
#	DESCRIPTION	USER COMMAND	REF
a	The input is available on an SMA connector on the front panel.		
b	The input impedance can be set as 50 Ω (default) or high impedance 500 Ω.	*SetTriggerInputImpedance*	4.8.3
c	The trigger threshold is 0.5 V (default) and user-controlled.	*SetTriggerThresholdVoltage*	
d	High speed comparator.		
e	Select rising or falling edge.	*SetExternTrigEdge*	

Figure 12: External trigger on front panel.

4.8.2 External trigger SYNC connector

The block diagram for the SYNC input is shown in **Figure 13** and related parameters are listed in **Table 2**. The user can control the SYNC function for adapting it to the system in the following ways:

- The input impedance can be set in 50 Ω (default) or high impedance mode, see **Section 4.8.3**.
- Set the trigger edge to rising or falling to adjust to the polarity of the trigger signal.



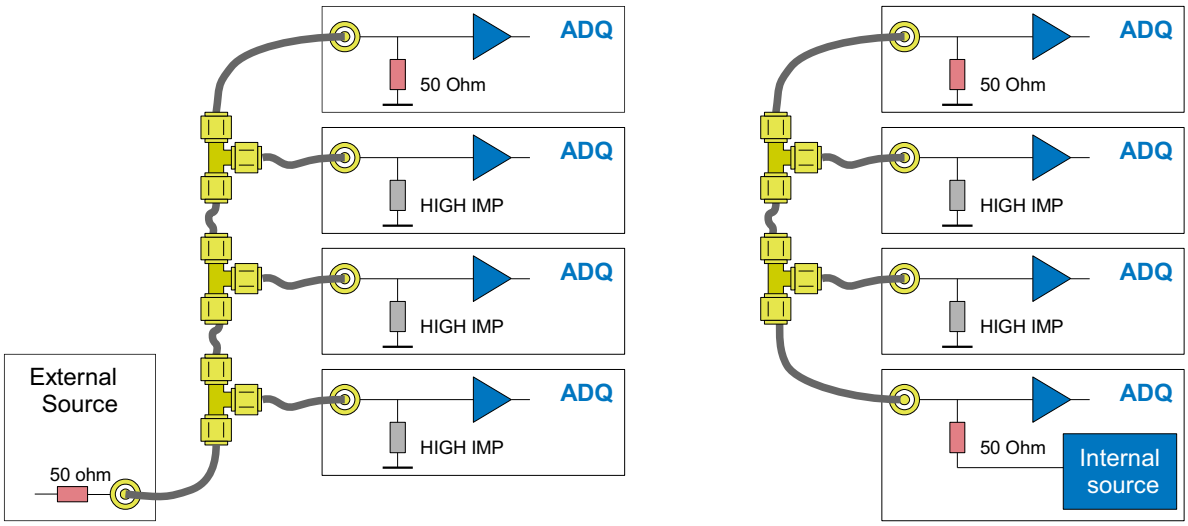
#	DESCRIPTION	USER COMMAND	REF
a	The input is available on an SMA connector on the front panel.		
b	The input impedance can be set as 50 ohm (default) or high impedance.	*SetTriggerInputImpedance*	4.8.3
c	The trigger threshold is fixed at 1.5 V.		
d	High speed comparator.		
e	Select rising or falling edge.	*SetExternTrigEdge*	

Figure 13: Using SYNC as trigger input.

4.8.3 Driving the external TRIG/SYNC signal by controlling input impedance

The TRIG/SYNC input impedance is by default 50 Ω. The trigger is optimized for systems where the trigger source output impedance is 50 Ω and the cables has a characteristic impedance of 50 Ω. This setup results in an optimal high-frequency response and low reflections, which is important for precise timing.

However, in a high fan-out situation, where a trigger source has to drive many nodes, the load can be too high. The trigger input can then be set in a high impedance mode and a bussed connection can be used, **Figure 14**. In **Figure 14** (a) an external source is driving the array of ADQ digitizers. In **Figure 14** (b) one of the ADQ digitizer is used as a master and is driving signals to the array of ADQ digitzers.One has to be careful with the trigger distribution network to handle the reflections. If the trigger is periodic, reflections are less critical and can be handled.



(a) External source

(b) ADQ internal source

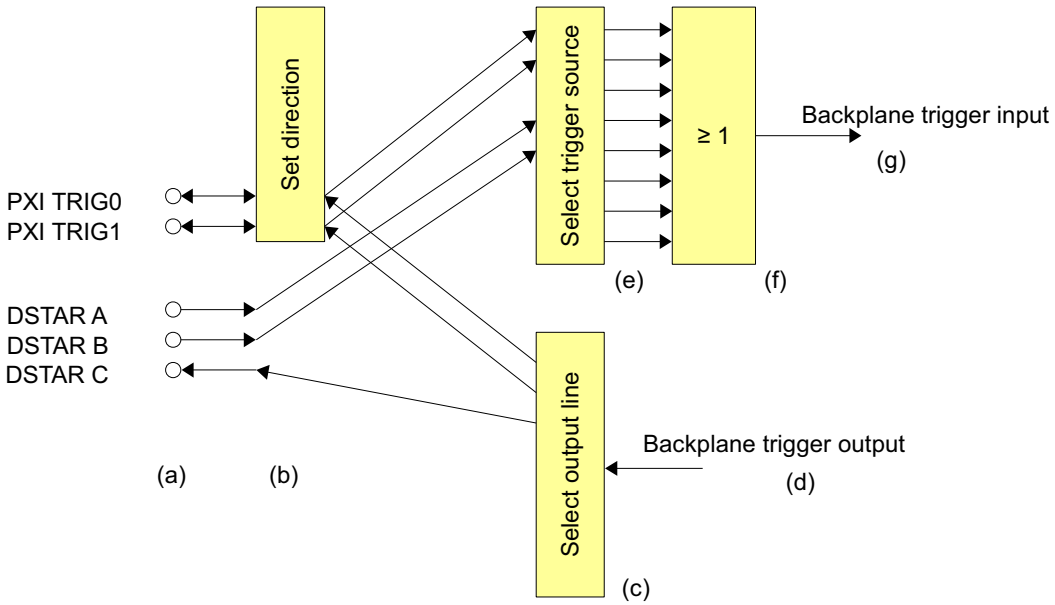
Figure 14: Bussed connections

4.9 External trigger in the backplane

4.9.1 PXIe interface

There are an external trigger in the backplane of the PXIe version. The DSTAR signals are dedicated matched trigger lines from the system timing slot. To use these triggers, a dedicated timing generation board has to be used in the system timing slot. The TRIG bus is a general bus in the backplane which can be used for triggering. The digitizer support connection to port 0 and port 1 of that bus.

The backplane trigger support both input and output triggers. These operations are independent and can be used simultaneously.

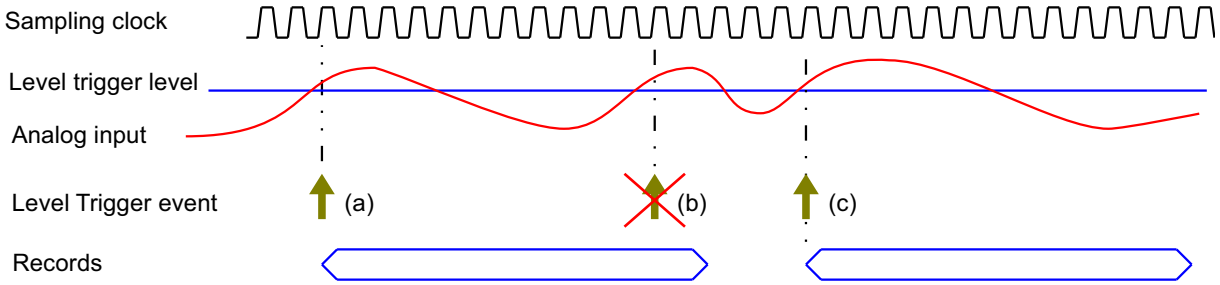


#	DESCRIPTION	USER COMMAND	REF
a	Backplane Trigger bus and DSTAR connections		
b	Set direction for each port in the backplane	*SetDirectionPXI*	
c	Output: Select output port for trigger output signal	*SetupTriggerOutput*	4.12
d	Output: This is the source for the trigger output signal	*SetupTriggerOutput*	4.12
e	Input: Select port for trigger sources	*SetTriggerMaskPXI*	
f	Input: The backplane trigger signal is OR:ed from the selected ports		
g	Input: this is the backplane trigger to the trigger module	*SetTriggerMode*	4.1

Figure 15: Trigger in the PXIe backplane.

4.10 Level trigger

The level trigger allows data-driven acquisition. When the data on a selected channel crosses the trigger level, all channels on the ADQ is triggered, **Figure 16**. The level trigger is set to trigger on rising or falling edge. Here, rising edge is illustrated.



#	DESCRIPTION	USER COMMAND	REF
a	When the signal passes the trigger level, a trigger event is generated and the first record is captured.	*SetupLevelTrigger*	4.10.1
b	During the record, incoming triggers are ignored.		
c	When the signal passes the trigger level, a trigger event is generated and the second record is captured. Re-triggering is also controlled by a hysteresis function.	*SetupLevelTrigger*	4.10.3

Figure 16: Level trigger introduction.

4.10.1 Setting the level trigger level

The level is given in digital codes:

LEVEL_CODE = ANALOG_TRIGGER_LEVEL / (ACTUAL_ANALOG_RANGE / 2) * 2^15 **(10)**

The ***ACTUAL_ANALOG_RANGE*** is the analog full scale range. If a DC-offset is used at the input, add the ***DC_OFFSET_CODE*** from **(6)**.

Note that the trigger level is given in 16 bits MSB aligned format and that the range is from -2^{15} to $2^{15}-1$.

4.10.2 Level trigger and DBS

The Digital Baseline Stabilizer, DBS, compensates for fluctuations in the baseline. The DC level is then set by a parameter ***DBS_CODE*** into DBS. The level trigger is set relative to the analog baseline as:

ANALOG_STEP = ANALOG_TRIGGER_LEVEL - ANALOG_BASELINE **(11)**

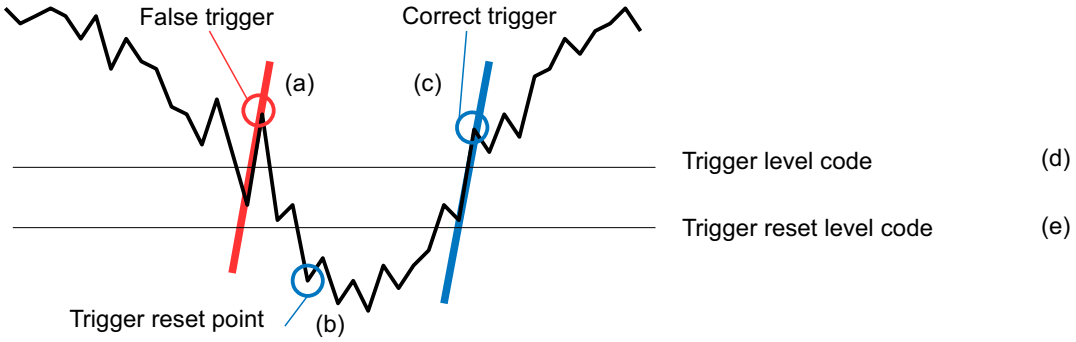
LEVEL_CODE = DBS_CODE + ANALOG_STEP / (ACTUAL_ANALOG_RANGE / 2) * 2^15 **(12)**

4.10.3 Controlling noise sensitivity

The level trigger is sensitive to noise since it can detect a step as small as one digital code. This can cause unwanted triggering. The noise sensitivity is controlled by a hysteresis function, **Figure 17**. After triggering, the signal has to cross a reset level before it can trigger again.

Setting the reset level far from the trigger level will give a robust trigger.

Setting the reset level close to the trigger level will give a sensitive trigger.



#	DESCRIPTION	USER COMMAND	REF
a	The level trigger has a hysteresis function to avoid false triggering on noise.		4.10.3
b	When the signal passes below the ***RESET_LEVEL_CODE***, the ADQ may trigger again.		
c	Trigger position is the first sample above the trigger ***LEVEL_CODE***		
d	Set the trigger ***LEVEL_CODE***.	*SetupLevelTrigger*	4.10.1
e	Set the trigger ***RESET_LEVEL_CODE***.	*SetupLevelTrigger*	4.10.3

Figure 17: Trigger reset level.

4.11 Internal trigger

The internal trigger generates a periodic trigger signal. The internal trigger period is specified in number of samples.

Note that if the internal trigger signal used as a trigger output, the actual output signal is updated at the Data Clock rate. This may appear as jitter on the output. It is recommended to use a period that is a multiple of the Data Clock is the internal trigger shall drive external equipment.

4.12 Trigger output

The trigger output signal is intended for triggering external equipment connected to the ADQ to build a synchronized system. It can also be used for indicating that a trigger event occurred in the digitizer. The trigger output is updated at the rate of the Data Clock.

The trigger output function consists of two parts:

- The first part selects the source of the trigger output signal.
- The second part selects the physical output port for the trigger output signal.

Note that the trigger output is the same physical TRIG connector as the external trigger input.

4.12.1 Trigger output port selection

The trigger output port is selected to one of these ports:

- Trigger connector on the front panel. Note that the trigger output is the same physical TRIG connector as the external trigger input.
- PXIe backplane triggers, **Section 4.9.1**.

4.12.2 Frame sync output on SYNC connector

The frame sync feature enables grouping of trigger signals into frames or blocks or lines. The name for this feature relate to the actual application. This function can, for example, be used in scanning three-dimensional measurements where a record is the first dimension, the trigger is the second and the frame sync is the third dimension.

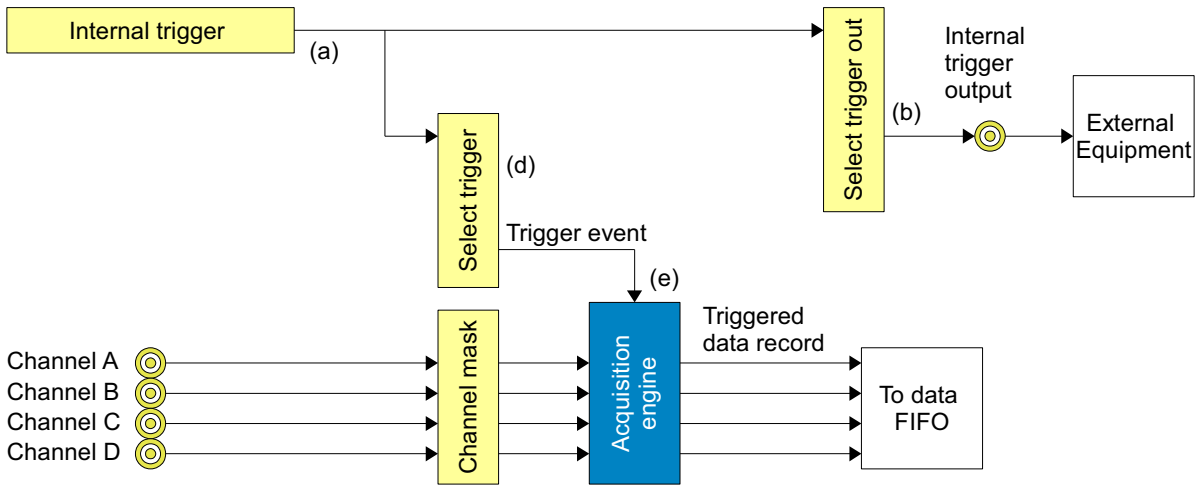
The frame sync count triggers and output a signal at a certain period. The period is measured in trigger events.

4.12.3 Triggering external equipment with internal trigger

Triggering external equipment and the digitizer with the internal trigger may be done in two ways; internally, **Figure 18** and externally, **Figure 19**.

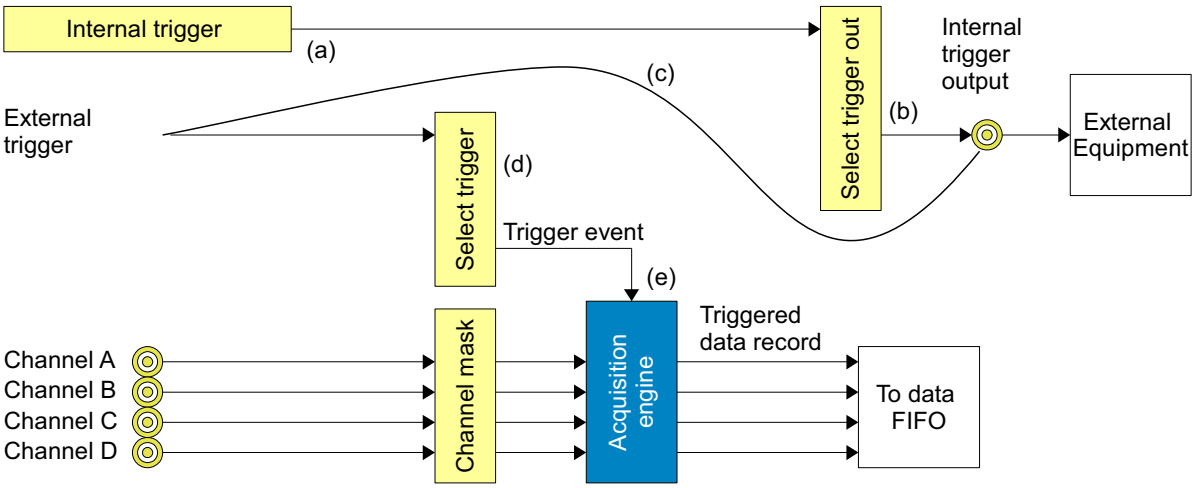
The internal connection is preferred when the trigger is only used for triggering the digitizer. The triggering is then done inside the FPGA as a logical signal and the trigger time is guaranteed to be exact on the expected sample.

The external connection is preferred when the trigger is used for triggering both the digitizer and the external equipment. Then the ADQ will listen to the same physical signal as the external equipment is using.



#	DESCRIPTION	USER COMMAND	REF
a	Internal trigger generator	*SetInternalTriggerPeriod*	4.11
b	Select internal trigger as output	*SetupTriggerOutput*	4.12
c	–	–	–
d	Select **internal** trigger as trigger source	*SetTriggerMode*	
e	Acquisition engine creates a record from streaming data		7

Figure 18: Internal routing of internal trigger.



#	DESCRIPTION	USER COMMAND	REF
a	Internal trigger generator	*SetInternalTriggerPeriod*	4.11
b	Select internal trigger as output	*SetupTriggerOutput*	4.12
c	The trigger output and the external trigger input are electrically connected inside the digitizer		
d	Select **external** trigger as trigger source	*SetTriggerMode*	
e	Acquisition engine creates a record from streaming data		7

Figure 19: External routing of internal trigger.

4.13 Large scale integration trigger support

The ADQ8-4X supports integration into a large scale PXIe chassis for single shot applications, The function is described in the manual [6]. This section is only a short introduction.

4.13.1 Distributing trigger

The trigger is distributed to the boards via the SYNC input and output. The trigger distribution is daisy chained, so there is no limit to the number of boards.

4.13.2 Clock reference

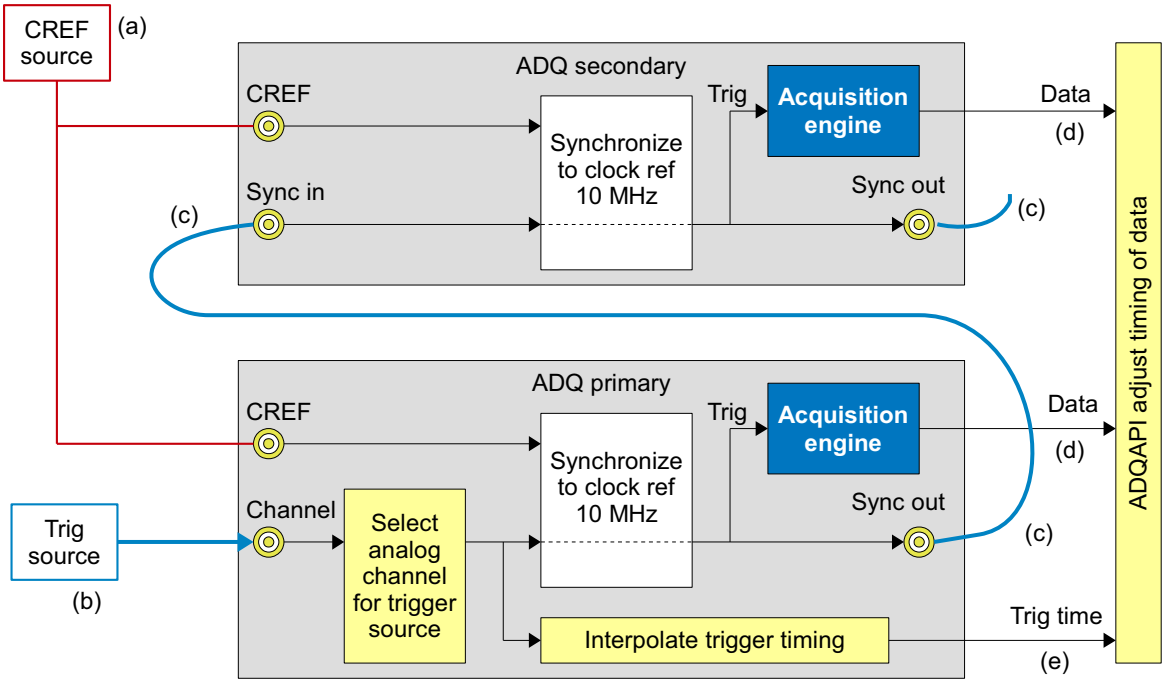
Use the clock reference distribution in the backplane of the chassis for phase alignment of the digitizers in the system.

4.13.3 High precision trigger on analog input

The source for the trigger can be any available trigger. If one of the analog inputs is used as a high precision input, the trigger timing can be computed to 25 ps trigger precision.

4.13.4 Calibration

To reach the trigger accuracy of 200 ps for the entire chassis installation, the system has to be calibrated in the factory [5].



#	DESCRIPTION	USER COMMAND
	Daisy chain block contains of one primary and secondary units. There is a set of general functions to manage the operation.	*DaisyChainReset* *DaisyChainEnable* *DasiyChainSetMode* *DasiyChainSetOutputState*
a	The common clock reference guarantee the phase stability between all units.	*SetClockSource* *AdjustClockReferenceDelay*
b	Select a trigger source in the primary unit. This may be an analog channel using level trigger or any other trigger source	*DaisyChainSetTriggerSource* *DaisyChainSetupLevelTrigger*
c	The trigger signal is synchronized to the clock reference and forwarded to the secondary units. This daisy chain of triggers can include many cards.	*DaisyChainSetupOutput* *DaisyChainEnableOutput*
d	The data is triggered on the synchronized trigger signal.	Data acquisition control commands
e	The trig time is sent to the ADQAPI where the trigger position of all the channels is aligned.	*DaisyChainGetTriggerInformation*

Figure 20: Daisy Chain Trigger.

5 CLOCK

5.1 Clock domains

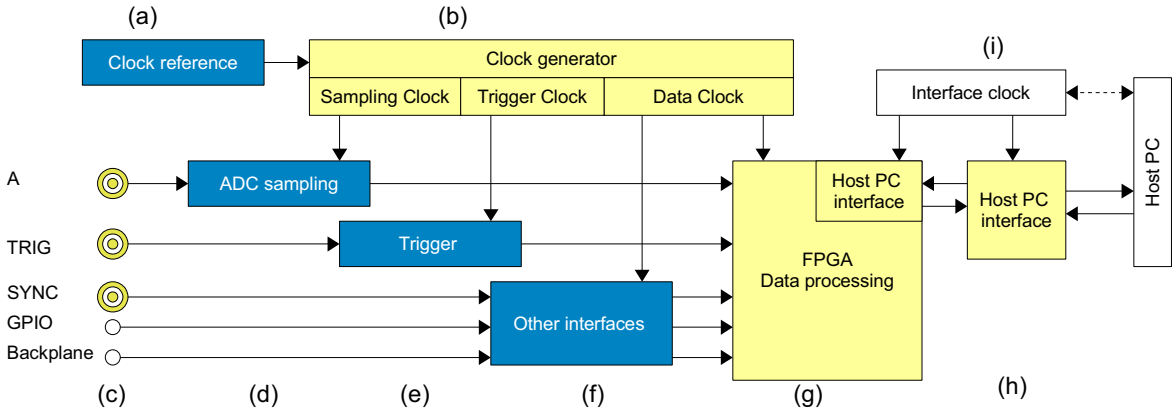
Different parts of the digitizer system operate on different clocks. The core of the clocking system is the clock reference. This is the phase and frequency reference of the digitizer system. It is possible to use different clock reference sources to meet the requirements of different applications.

The sampling clock is generated based on clock reference and drive the ADCs. This clock is important for the signal quality since any timing deviation (jitter) will impact the actual time of the sample.

The trigger signal is sampled by a separate clock at a higher frequency than the sampling clock. This is to achieve a good timing resolution of the trigger. This frequency is the highest in the digitizer and is also the base for the timestamp.

The FPGA cannot operate at the high rate of the sampling clock. Instead the data processing operate on a derived lower frequency denoted Data Clock and several samples are processed in parallel to maintain the throughput. The Data Clock is also synchronized to the clock reference.

Finally the host PC interface also operate on a different clock. The PCIe system clock is provided from the PCIe bus. This part of the design is not phase-locked to the sampling clock.



#	DESCRIPTION	TYPICAL FREQUENCIES	REF
a	Clock reference.	10 MHz	5.2
b	Clock generator that generates several frequencies.		5.7
c	Various interfaces.		4.8
d	The ADC operates on sampling clock.	2 GHz	
e	The trigger operates on the trigger clock.	4 GHz	4
f	Other interfaces operate on the Data Clock.	250 MHz	4 6
g	The FPGA operate on the Data Clock.	250 MHz	
h	The host PC operate on an independent clock.	125 MHz (PCIe)	
i	The clock generation for the host interface is from the backplane.		

Figure 21: Clock system overview

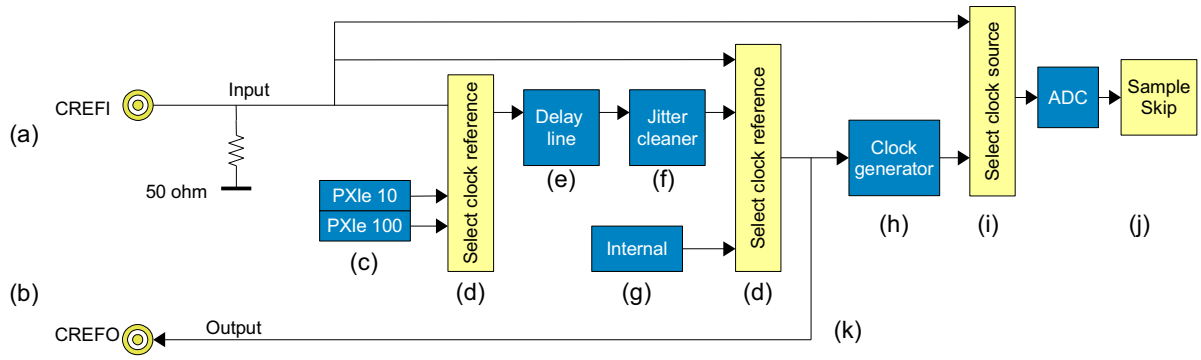
5.2 Flexible clock network

The preferred clock method is a systems design parameter and teh digitizer supports many options. The clock system of the digitizer consist of two key parts; the clock reference and the clock generator. The clock reference is a low frequency (10 MHz) high-precision and high-stability signal that sets the accuracy of the clocks in the digitizer. The clock generator translates the frequency of the clock refer-

ence to the sampling clock rate. The digitizer supports its specified sample rate only. This sample rate can be tuned to allow phase locking to external equipment.

To reduce the sample-rate, a sample skip function is available.

Block diagram of the clock network for ADQ8-4X-PXIe is given in **Figure 22**.



#	DESCRIPTION	USER COMMAND	REF
a	The input SMA connector is common for external clock reference input, external clock input. The impedance is 50 Ω.	*SetClockSource*	5.3
b	The output MCX connector drives the selected clock reference source to enable synchronization of other equipment.	*EnableClockRefOut*	5.3 5.9
c	The backplane clocks are available for synchronization.		
d	Select which clock reference source to use	*SetClockSource* *SetExternalReferenceFrequency*	
e	Delay line to fine tune the phase of the sampling clock	*AdjustClockReferenceDelay*	5.6
f	Jitter reduction circuit.		
g	The internal clock reference is a high performance VCTCXO.		5.4
h	The internal clock generator for the sampling clock		5.7
i	Select which sampling clock source to use; internal or direct external sampling clock.	*SetClockSource*	5.8
j	Reduce sample rate with sample skip.	*SetSampleSkip*	5.10
k	Turn on clock reference output.	*EnableClockRefOut*	5.9

Figure 22: ADQ8-4X-PXIe clock network.

5.3 ADQ8-4X-PCIe front panel connectors

The front panel SMA connector is used for external clock reference input or direct external sampling clock input. The input impedance is 50 Ω to match the cable impedance.

The front panel MCX connector is used for clock reference output.

5.4 Internal clock reference

The internal clock reference is a high accuracy VCTCXO at 10 MHz.

5.5 External clock reference

The free running internal clock reference of the digitizer offers high precision and is suitable for most measurements. However, for some applications an absolute phase-lock to other parts of the system may be necessary. To support that, the ADQ offers several options to accept an external clock reference. A long-term phase stability to other equipment is then guaranteed.

The connector on the front panel accepts a clock reference from external equipment. The clock reference quality is improved in a jitter cleaning circuitry. To match the tuning of the jitter cleaning circuitry the clock reference has to be 10 MHz.

The PXI Express allows clock reference input from the backplane, and can then benefit from the infrastructure of the chassis.

5.6 Clock reference phase tuning

There is a tuning of the clock reference phase. See [6] for more information

5.7 Internal clock generator

There is an internal high quality clock generator that is used for generating the Sampling Clock for the A/D converters. The data and trigger clocks are also generated by this clock generator.

5.8 External clock

If the system is designed with an external high quality signal it may be used for clocking the ADQ. If an external clock source is used, all the internal clocks are generated from that to maintain the phase and frequency ratio.

5.9 Clock reference output

In addition to the synchronization solution with an external clock reference source, the digitizer can also act as master and output its clock reference to external equipment. The selected clock reference source will then be present at the clock connector on the front panel. Note that the connector is shared with clock input.

5.10 Sample skip

The data rate out from the A/D converter is set by the sample rate of the digitizer. The data rate can be reduced by the sample skip function. Setting the sample skip factor to, for example, 4 means that every 4th sample is kept and the others are discarded. This will efficiently reduce the data rate.

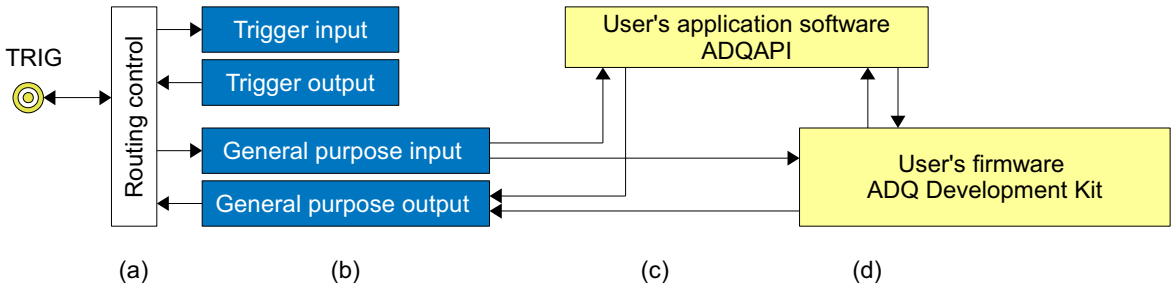
Note that there is no decimation filter in this function.

6 GPIO

The General Purpose Input and Output (GPIO) are digital signals available from the front panel of the digitizer. The GPIO is an optional use of the TRIG and SYNC connectors. The user assigns a function to these pins, either in the firmware through the ADQ Development Kit or from software.

6.1 GPIO on TRIG connector

The connectors for TRIG connector can be used as GPIO. The process is illustrated in the block diagram in **Figure 23**.

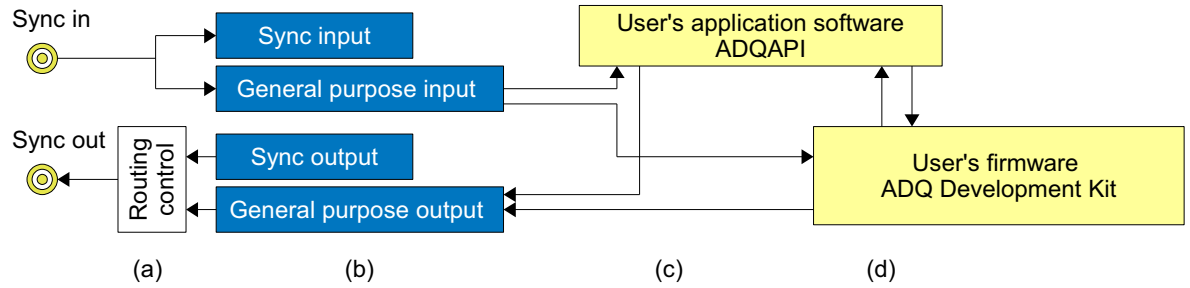


#	DESCRIPTION	USER COMMAND	REF
a	The external pin is automatically connected to the activated function.		6.3
b	The GPIO input function always reads the state of the pin. The GPIO output function is activated by the user.	*SetDirectionGPIOPort*	6.3 6.4
c	The user may access the pin by reading and writing from the software.	*ReadGPIOPort* *WriteGPIOPort*	6.3 6.4
d	The user may build firmware in the ADQ Development Kit for real-time interaction with the GPIO signals.Then GPIO is accessed through register access commands.	*ReadUserRegister* *WriteUserRegister*	6.5

Figure 23: Using front panel connector TRIG as GPIO.

6.2 ADQ8-4X–PXIe GPIO on SYNC connectors

On the ADQ8-4X–PXIe form factor, the sync input pin and output pins are split into two connectors. This means that the sync pin is not bi-directional when used as GPIO, **Figure 24**.



#	DESCRIPTION	USER COMMAND	REF
a	The external pin is automatically connected to the activated function.		6.3
b	The GPIO input function always reads the state of the pin. The GPIO output function is activated by the user.	*SetDirectionGPIOPort*	6.3 6.4
c	The user may access the pin by reading and writing from the software.	*ReadGPIOPort* *WriteGPIOPort*	6.3 6.4
d	The user may build firmware in the ADQ Development Kit for real-time interaction with the GPIO signals.Then GPIO is accessed through register access commands.	*ReadUserRegister* *WriteUserRegister*	6.5

Figure 24: ADQ8-4X–PXIe GPIO on SYNC pin.

6.3 Using GPIO as a trigger

The GPIO can be used for sending a trigger command from the application software to an external device. In such a situation this GPIO signal can also be used for triggering the digitizer itself synchronous to the external device. This is possible since the external trigger input logic always listen to the signal on the TRIG connector. The following example illustrate how to trigger the digitizer and an external device through GPIO function on the TRIG connector.

Example 7: Triggering the digitizer and an external device with GPIO

1. Connect a cable from the TRIG connector to the external device.
2. *SetTriggerMode*(“external trigger”) /* This will activate the trigger module to listen to TRIG”
3. *SetDirectionGPIOPort*(“output”) /* This enables output on TRIG connector*/
4. *ArmTrigger*
5. *WriteGPIOPort*(“1”) /* This sends a signal on the TRIG connector that triggers the devices*/

The GPIO input function always listen to the trigger pin. This means that the external trigger pin value can always be read from the GPIO function *ReadGPIOPort*.

6.4 Output

The output is activated through the software command *SetDirectionGPIOPort* and the signal level is set by *WriteGPIOPort*. The input function *ReadGPIOport* will then return the output level.

6.5 GPIO in ADQ Development Kit

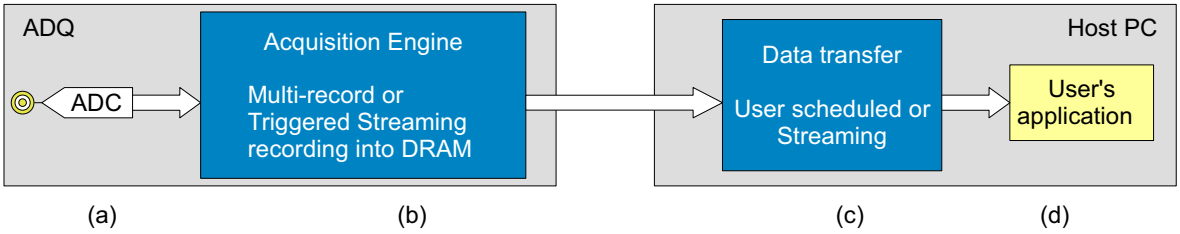
The GPIO signals from TRIG and SYNC are available in the ADQ Development Kit for real-time interaction with the signal flow.

7 ACQUISITION CONTROL

The acquisition control consists of two partly independent parts;

- Acquisition process: acquisition of data in a record int the DRAM of the digitizer
- Transfer process: transfer data from the DRAM of the digitizer to host PC.

Figure 25 shows the flow of data through acquisition and data transfer.



#	DESCRIPTION	USER COMMAND	REF
a	The A/D converter digitizes the analog signal and generate a flow of data.		
b	The acquisition engine manages the data acquisition and builds records of the data. Multi-record recording of data into the DRAM.	*MultiRecordSetup* *ArmTrigger*	7.4
c	The transfer of data to the host PC delivers data in buffers for the user. User schedules the transfer.	*GetDataWHTS*	7.6
d	User's application reads data and headers for further processing and/or storage.		7.1, 7.8,

Figure 25: Acquisition control and data transfer.

7.1 Multi-thread notice

Note that the digitizer does not support multi-threaded applications. In high speed applications, however, a multi-threading programming style has advantages. In such an application, make sure that only one thread communicates with the digitizer at a time.

In the example in **Section 7.6**, one thread handles the control of the digitizer. The other thread only processes data.

Example code available with the digitizer is sometimes written with several threads. Study these examples carefully to see how multi-threading can be used.

7.2 Acquisition memory

The acquisition memory, **Figure 26**, is of size 1 GBytes.

The memory is shared by all activated channels which means that if only one channel is activated, the entire memory is available for that channel.

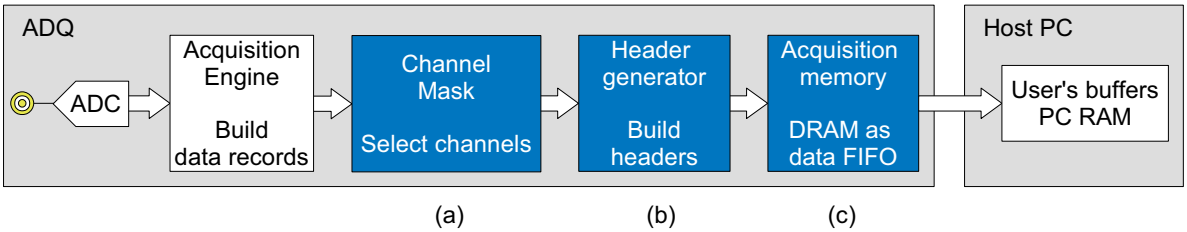
The data memory is also shared between data and headers, **Table 3**. A header contains information about the data record, for example, timestamp and channel number. The headers can be setup in two different modes:

- In the normal mode, the headers are activated. This mode is recommended for all standard acquisition modes.
- In the raw mode, there are no headers and all available memory is used for data. The raw mode is only recommended for custom firmware built in the ADQ Development Kit.

The memory is organized as a FIFO where it is possible to record data and read out data simultaneously. This is called readout-while-recording and is available both in multi-record and streaming mode.

If the readout process is as fast as the acquisition process, the memory size does not impose any limitations.

If readout is scheduled after recording, the total data set from the measurement has to be limited to 1 GBytes to fit in the acquisition memory.



#	DESCRIPTION	USER COMMAND	REF
a	The channel mask selects which channels to record and which to discard.	*MultiRecordSetChannelMask*	
b	The header generator apply header information to each record.		
c	The acquisition memory is a DRAM on the digitizer which operate as a FIFO.		

Figure 26: Acquisition memory.

ACQUISITION MODE	RECORD SIZE	ADQ8-4X
Header	< 256 samples	> 5.9 %
Header	> 256 samples	< 5.9 %
Header	Asymptotic lower limit	3.1 %
Raw mode no headers	Any	0 %

Table 3: Header memory requirement for triggered streaming acquisition mode.

7.3 Triggered streaming acquisition

Triggered streaming is described in [7].

7.4 Acquisition mode multi-record

ADQ8-4X supports acquisition mode multi-record. This mode is suitable for most user-scheduled single shot or multi-shot applications. It is also suitable for control and surveillance applications where the long pre-trigger makes it possible to see what preceded an event.

In the multi-record mode, one or several records are recorded into the acquisition memory and read out on request from the user. The multi-record mode is straight forward to implement and is easy to use if the data set is smaller than the on-board memory. The memory (DRAM) of the digitizers is split into circular buffers where one buffer can hold one record. Data is then written constantly into this buffer until the trigger event arrives. After the trigger event ***Record_length-Pretrigger_samples*** is written into the buffer. The accumulation of the record is then completed. The circular buffer contains the pre-trigger data and the post trigger data. The position of the trigger may be anywhere in the buffer. The API unwraps the data buffer before presenting the record to the user. The multi-record gives the feature of a very long pretrigger.

Readout only can start when the complete record is captured and is always initiated from the user's application.

Multi-record mode requires that all active channels record simultaneously.

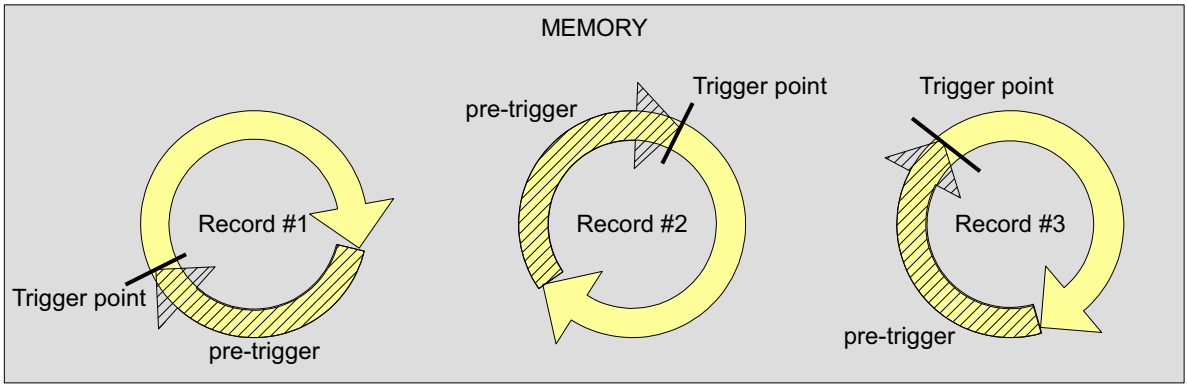


Figure 27: Multi-record organization of memory.

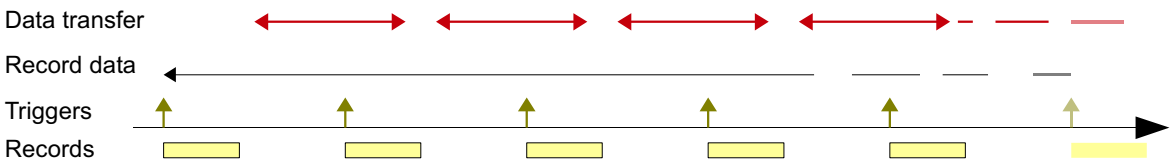


Figure 28: Multi-record timing.

7.5 Re-arm time

The re-arm time a the time after a record has been completely acquired when the digitizer cannot receive a new trigger. During this re-arm time the digitizer prepares the DRAM for receiving the next record. The re-arm time is thus a hardware related fixed timing during which the digitizer cannot record.

The scheduling of triggers depends on the re-arm time but also of the pre-tirgger and the trigger delay. See **Figure 29** for an illustration of the timing relation between trigger and acquisition.

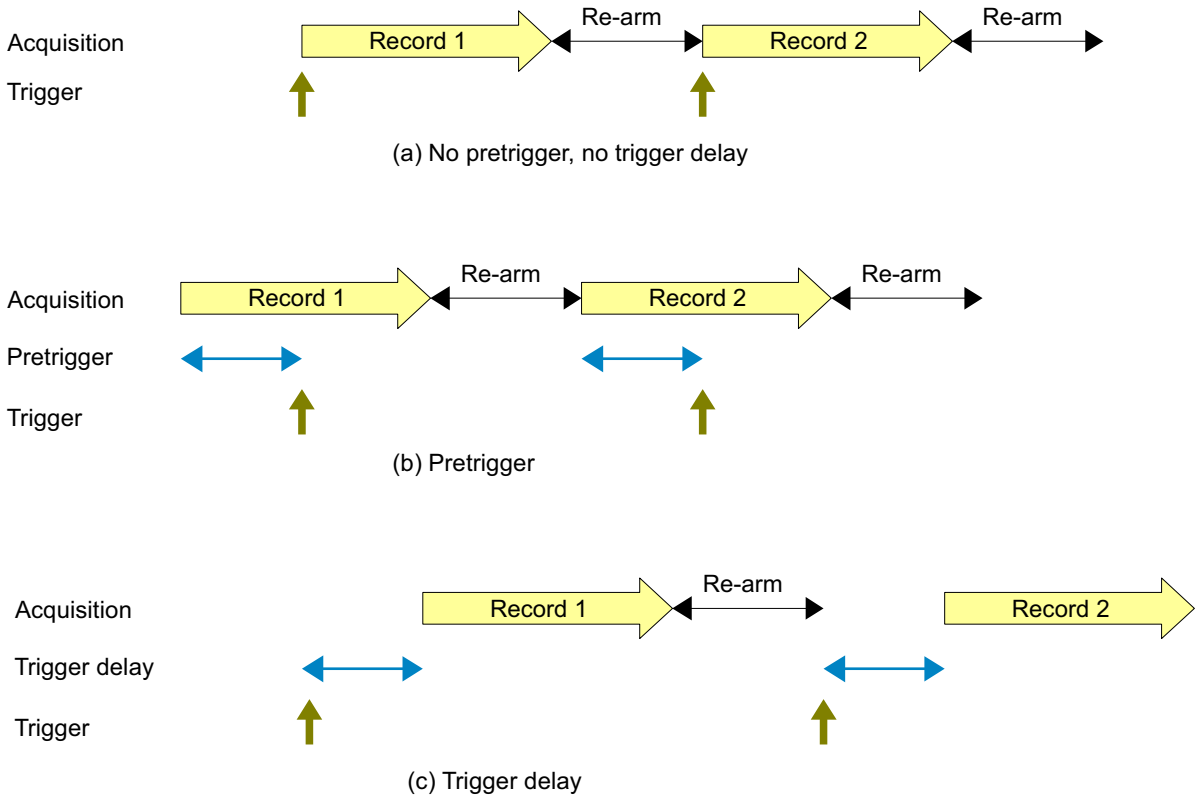


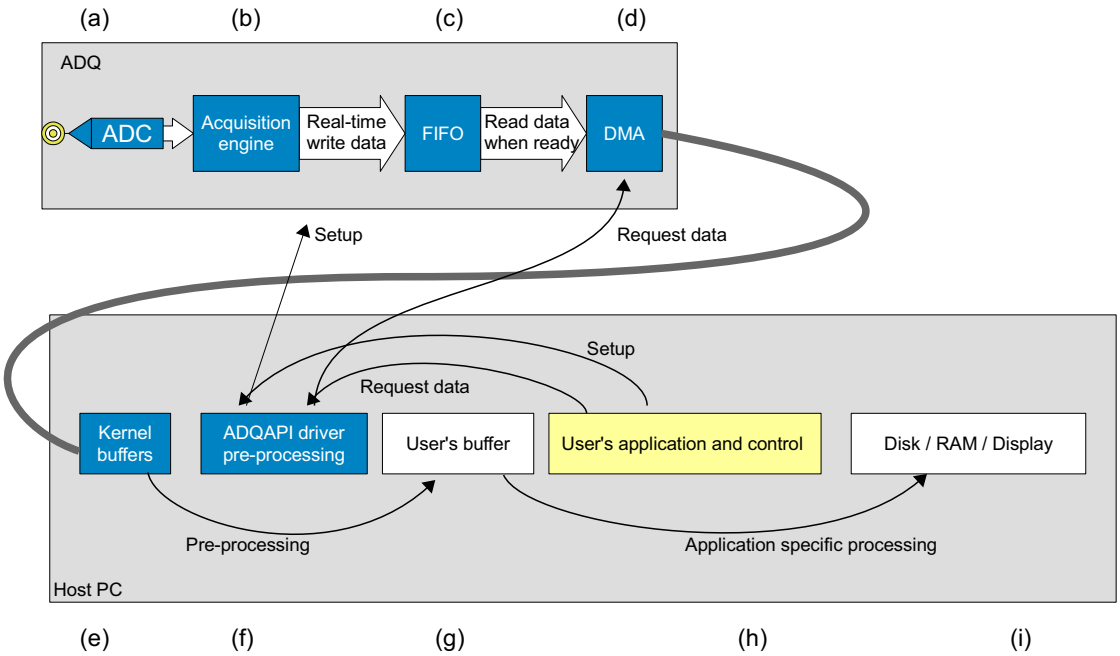
Figure 29: Re-arm timing.

7.6 User scheduled data transfer mode

In the user-scheduled mode the data flow is controlled by the user in a schedule. The user first sets up the digitizer, then starts the acquisition, and after that requests the data. This is the straight forward method if the total amount of data is less than the data buffer size (**Section 7.2**), that is, it can be stored in the buffer of the digitizer.

Note that if using readout while recording, the multi-record is not automatically limited by the data buffer size.

A block diagram of triggered streaming acquisition and user-scheduled data transfer is given in **Figure 30**.



#	DESCRIPTION
a	A/D Converter delivers a stream of data int the acquisition engine.
b	The acquisition engine applies triggers, headers, etc.
c	Data is sent in real-time to the FIFO on the ADQ.
d	The DMA transfers data to the PC when requested by the user.
e	Kernel buffers in the host PC receives the data from the digitizer.
f	The ADQAPI receives incoming data and do necessary pre-processing, for example lost packages, and sends data to the user's buffers.
g	User's buffers in RAM. These are accessed via API commands.
h	The user's application sets up the digitizer for acquiring data. Then the software requests the acquired data and perform application specific processing. This is the user's software.
i	Examples of output devices.

Figure 30: Block diagram of user-scheduled data transfer.

The timing of a user-scheduled multi-record transfer is in **Figure 31**. The acquisition is set up and armed. The user's software checks for available records. The available records can be transfered with the ***GetDataWHTS*** command[1].

1. WHTS means 'with header and timestamp'.

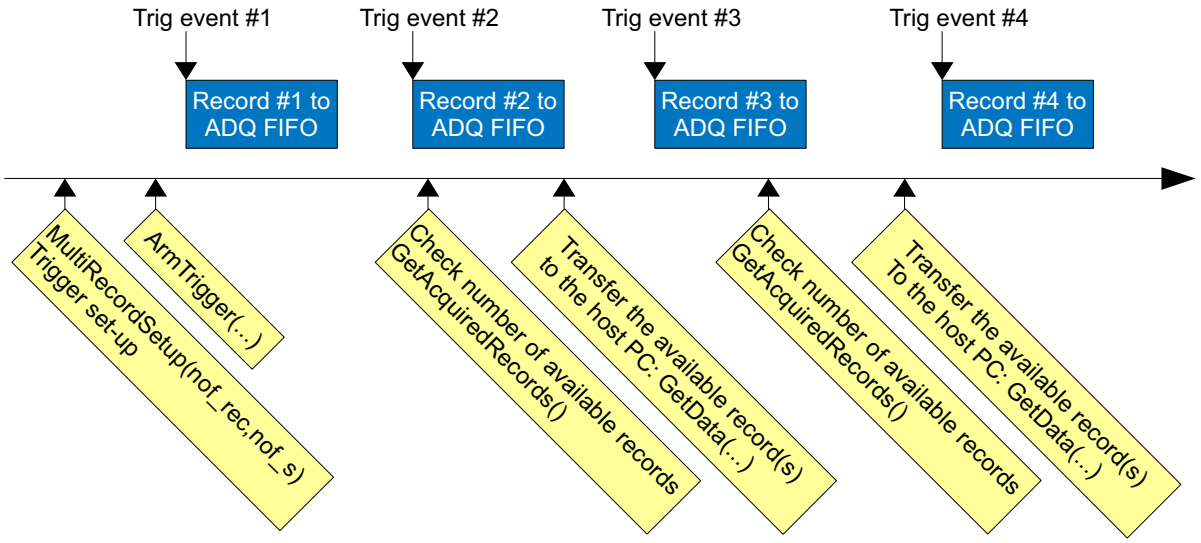


Figure 31: Timing of user-scheduled data transfer.

7.6.1 Transfer buffers

The data transfer buffers are the kernel buffers in **Figure 30**. These buffers are used by the DMA and the ADQAPI to transfer the data from the ADQ digitizer to the host PC.

The transfer buffers are owned and managed by the ADQAPI, but the user sets the size and number of buffers with the command ***SetTransferBuffers***.

7.6.2 User's buffers

The user's buffers in **Figure 30** are allocated and managed by the user. The ADQAPI recreates the data record and puts the result in these buffers.

Create the buffers using ***malloc***. Then provide the pointers of these buffers to the ADQAPI through the command or ***GetDataWHTS***.

The user's buffers consist of two sets of buffers; one for header information and one for data. The header is always 40 bytes per record and the content is described in **Section 7.9**. The data buffer size is depending on the amount of data in each record. For FWDAQ, the record size is always constant and the buffer size can be set to match the record size. The example code in ADQAPI_example[1] illustrate how to handle data buffers in general.

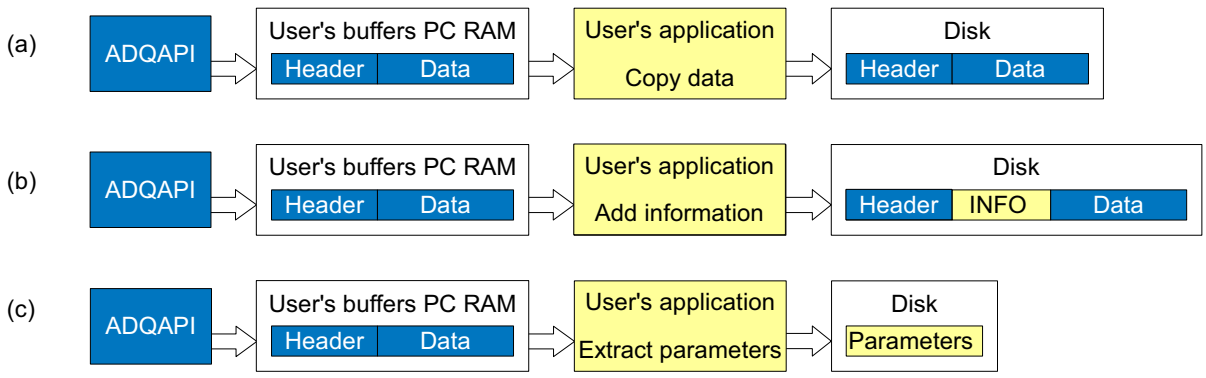
7.7 Streaming transfer mode

Streaming transfer mode is described in **[7]**.

7.8 Users application software consuming data

The users application can consume the data in different ways. Some examples are in **Figure 32**. The component "disk" is used for illustrating the end point for the data. It may also be a display or some other device such as an alarm.

1. The ADQAPI_example is found in the folder installation of the digitizer software.



#	DESCRIPTION	REF
a	Copy the data to disk for offline analysis. Header is in default format.	7.9.1
b	Copy the data to disk for offline analysis. Additional information about e.g. the experiment, is added to the header.	
c	Header and data is analyzed in real-time and only requested parameters are stored.	7.9

Figure 32: Data flow through the system

7.9 Record header

7.9.1 Metadata

The record header contains the information described in Table 4. Example code is available to pack this information to a header and write to disk. The example C-code defines a struct which reads the header information from the header buffer.

PARAMETER	FORMAT	DESCRIPTION	REF
Record Status	Byte	Over/under range, FIFO fill factor and lost data.	7.9.2
User ID	Byte	A user-configurable value to identify the ADQ unit.	7.9.3
Channel	Byte	The channel number.	7.9.5
Data format	Byte	Information about data.	7.9.5
Serial number	uint32	Serial number of the ADQ digitizer.	7.9.4
Record number	uint32	The current record number.	7.9.6
SAMPLE_PERIOD	int32	Time between two samples in steps of 25 ps.	4.4
TIME_STAMP	uint64	Timestamp of trigger event in steps of 25 ps.	4.4
RECORD_START	int64	Time between trigger event and record start in steps of 25 ps.	4.4
Record length	uint32	Length in number of samples of data in record.	7.9.8
General purpose vector	uint16	Usage varies with option.	
Timestamp reset	uint16	Number of times that the timestamp was reset.	4.4.2

Table 4: Header data.

7.9.2 Record Status

The status parameters indicates if the record was transferred correctly, Table 5.

Over-range or under-range condition within the data is indicated. Over-range and under-range can appear at various stages in the signal chain and the result is not easy to predict. A more detailed description of over-range and under-range is in Section 7.10.

FIFO fill factor is indicated. This is useful when tuning a data-driven process to avoid FIFO overflow and still get maximum efficiency from the experiment. For very long records, the maximum fill factor during the record is given.

If there is an overflow in the FIFO on the digitizer, data will be lost. The ***LOST_DATA*** bits inform about this.

OVER RANGE	FIFO FILL			LOST DATA				DESCRIPTION	COMMENT
7	6	5	4	3	2	1	0		
0								No over-range detected	
1								Over/under-range detected	Anywhere in the data.
	X	X	X					FIFO fill factor	Steps of 12.5%. 111 means > 87.5% and 000 is below 12.5%.
				0	0	0	0	No lost data	
							1	Lost record	The software has generated a header with no data to indicate a lost record.
						1		Lost data in beginning of the record	The start of the record is missing. Timestamp information is also missing.
					1			Lost data in the middle of the record	One or many section(s) anywhere inside the data is missing.
				1				Lost data in the end of the record	

Table 5: Status.

7.9.3 User ID

User ID is a custom byte that can be set by the user. Set this parameter by the ***EnableUseOfUserHeaders*** command.

7.9.4 Serial number

The serial number is the serial number of the ADQ hardware. The serial number is printed on a label on the digitizer in the form “S/N SPD-04004”. The serial number field is the number part, that is 04004.

7.9.5 Channel

This is an indication of from which channel the record originates. The first numbers of the channel parameter are reserved for the physical channels. The remaining combinations are available for ADQ Development Kit users to create artificial channels.

7.9.6 Record number

The Record number is counting the number of records captured from the power-up of the digitizer.

7.9.7 Data format

The header data parameter ‘Data format’ (**Table 4**) informs the user on how to interpret data. Allowed values are given in **Table 6**.

VALUE	DESCRIPTION
0	16 bits data word in two’s complement
1	32 bits data word in two’s complement

Table 6: Data formats

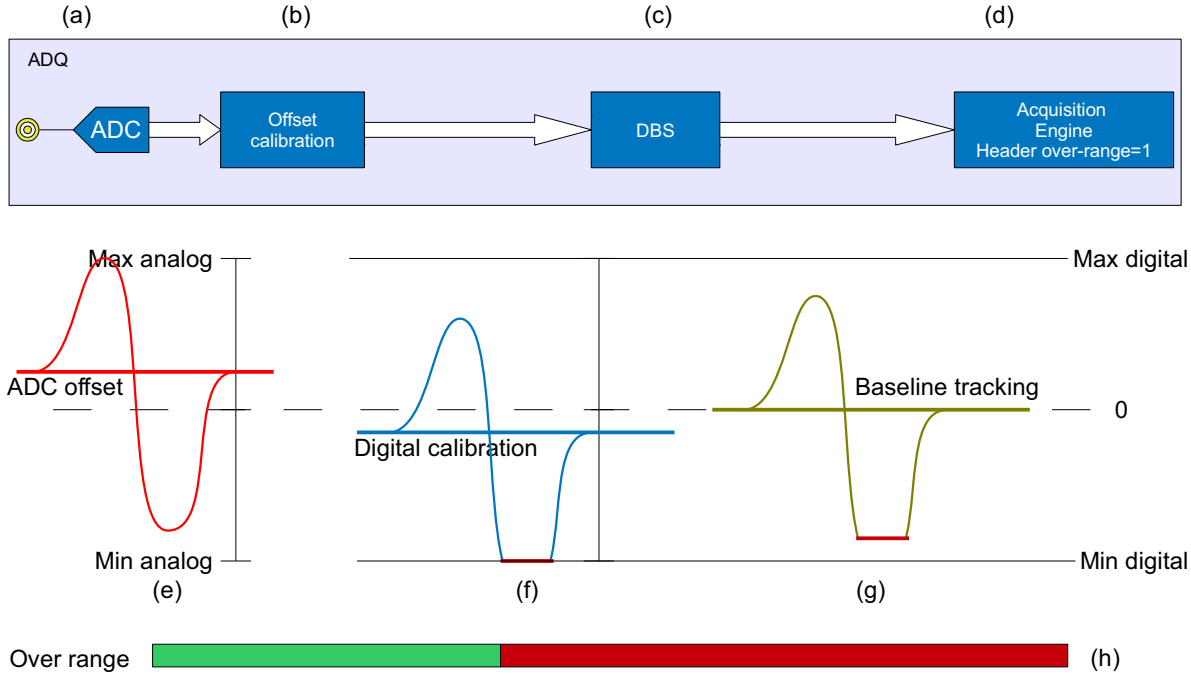
7.9.8 Record length

This is the length of the data record in number of bytes.

7.10 Over-range and under-range

The over/under-range bit in the header indicates that over-range or under-range occurred at one or several samples within the record and at any stage in the signal chain. The result of the over-range is not straightforward to predict, see **Example 6**.

Example 6: **Figure 33** *shows under-range in ADC before the Offset calibration. The result is that the digital code where the under-range occurred is not the maximum code.*



#	DESCRIPTION
a	ADC with offset.
b	Digital offset calibration.
c	Digital baseline stabilizer that dynamically adjust baseline to zero. That is adjust the offset. Under-range has already occurred prior to this stage.
d	Acquisition create a header which contain over/under-range indication.
e	The analog input signal with offset. The ADC also contribute to the offset.
f	Digital calibration cannot account for offset on the analog signal. Under-range occur when signal is out of range on the negative side.
g	The DBS fine-tunes the offset, that is add a value to the signal. Then the under-range is not at the minimum digital code anymore.
h	Overflow indication is set when over/under-range occur and is kept for the record.

Figure 33: Under-range in the ADC calibration.

8 HOST PC CONNECTION

8.1 PCI Express interface

The MTCA, and PXIE versions of the digitizer use PCI Express generation 1 and 2 electrical interface to communicate with the host PC. The PCIe version of the digitizer support up to 8 lanes and the MTCA version support up to 4 lanes.

8.2 Using several units

8.2.1 Using several digitizers from a single application.

Several digitizers can be connected to the same PC.

Each unit is then available and can be accessed individually.

Each software command contains a pointer to the control unit for all ADQ digitizers and an instance number that points out the current ADQ.

To identify a specific unit, read the serial number ***GetBoardSerialNumber***. This gives a mapping between the instance number and a physical unit.

The record header (**Table 4**) contains a byte field (User ID) where the user can set an identifier for each card. This gives a mapping between physical unit and data.

8.2.2 Using several digitizers from a several applications.

When several separate applications or threads are used for talking to different digitizers the commands ***ListDevices*** and ***OpenDevice*** has to be used. ***FindDevices*** will not work since ***FindDevices*** locks all available digitizers to the same application.

9 REFERENCES

[1] 17-1998 ADQ8-4X Datasheet

[2] 14-1351 ADQAPI Reference guide

[3] 08-0214 ADQAPI User guide

[4] 18-2059 ADQUpdater user guide

[5] 20-2382 Digitizer Studio manual

[6] 19-2246 ADQ8 Daisy-Chain Board synchronization

[7] 20-2465 ADQ8 Triggered streaming

Important Information

Teledyne Signal Processing Devices Sweden AB (Teledyne SP Devices) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to Teledyne SP Devices' general terms and conditions supplied at the time of order acknowledgment.

Teledyne SP Devices warrants that each product will be free of defects in materials and workmanship, and conform to specifications set forth in published data sheets, for a period of three (3) years. The warranty commences on the date the product is shipped by Teledyne SP Devices. Teledyne SP Devices' sole liability and responsibility under this warranty is to repair or replace any product which is returned to it by Buyer and which Teledyne SP Devices determines does not conform to the warranty. Product returned to Teledyne SP Devices for warranty service will be shipped to Teledyne SP Devices at Buyer's expense and will be returned to Buyer at Teledyne SP Devices' expense. Teledyne SP Devices will have no obligation under this warranty for any products which (i) has been improperly installed; (ii) has been used other than as recommended in Teledyne SP Devices' installation or operation instructions or specifications; or (iii) has been repaired, altered or modified by entities other than Teledyne SP Devices. The warranty of replacement products shall terminate with the warranty of the product. Buyer shall not return any products for any reason without the prior written authorization of Teledyne SP Devices.

In no event shall Teledyne SP Devices be liable for any damages arising out of or related to this document or the information contained in it.

TELEDYNE SP DEVICES' EXPRESS WARRANTY TO BUYER CONSTITUTES TELEDYNE SP DEVICES' SOLE LIABILITY AND THE BUYER'S SOLE REMEDY WITH RESPECT TO THE PRODUCTS AND IS IN LIEU OF ALL OTHER WARRANTIES, LIABILITIES AND REMEDIES. EXCEPT AS THUS PROVIDED, TELEDYNE SP DEVICES DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

TELEDYNE SP DEVICES DOES NOT INDEMNIFY, NOR HOLD THE BUYER HARMLESS, AGAINST ANY LIABILITIES, LOSSES, DAMAGES AND EXPENSES (INCLUDING ATTORNEY'S FEES) RELATING TO ANY CLAIMS WHATSOEVER. IN NO EVENT SHALL TELEDYNE SP DEVICES BE LIABLE FOR SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFIT, LOST DATA AND THE LIKE, DUE TO ANY CAUSE WHATSOEVER. NO SUIT OR ACTION SHALL BE BROUGHT AGAINST TELEDYNE SP DEVICES MORE THAN ONE YEAR AFTER THE RELATED CAUSE OF ACTION HAS ACCRUED. IN NO EVENT SHALL THE ACCRUED TOTAL LIABILITY OF TELEDYNE SP DEVICES FROM ANY LAWSUIT, CLAIM, WARRANTY OR INDEMNITY EXCEED THE AGGREGATE SUM PAID TO SP BY BUYER UNDER THE ORDER THAT GIVES RISE TO SUCH LAWSUIT, CLAIM, WARRANTY OR INDEMNITY.

Worldwide Sales and Technical Support

www.spdevices.com

Teledyne SP Devices Corporate Headquarters

Teknikringen 6
SE-583 30 Linköping
Sweden

Phone: +46 (0)13 465 0600
Fax: +46 (0)13 991 3044
Email: info@spdevices.com