Acqiris SA217P Acquisition Card

1 channel, 14-bit, 2 GS/s,
DC up to 1.2 GHz bandwidth

User's Manual

Copyright Statement

Version

March 2024

Contact us

Acqiris Americas

contact-americas@acqiris.com

Acqiris Europe

contact-emea@acqiris.com

Acqiris Asia-Pacific

contact-asia-pacific@acqiris.com

Acqiris Japan

contact-japan@acqiris.com

Worldwide contact for support

support@acqiris.com

Acqiris SA

Chemin des Aulx, 12

1228 Plan-Les-Ouates

Switzerland

www.acqiris.com

SA217P Acquisition Card User's Manual

This help document is intended to provide in-depth information and reference material specific to your ADC card.

For information about installation and about getting started with your ADC card, please refer to the Startup Guide which can be downloaded from https://extranet.acqiris.com/ or which is installed with your software.

Content

Introduction

New generation of Signal Acquisition Cards

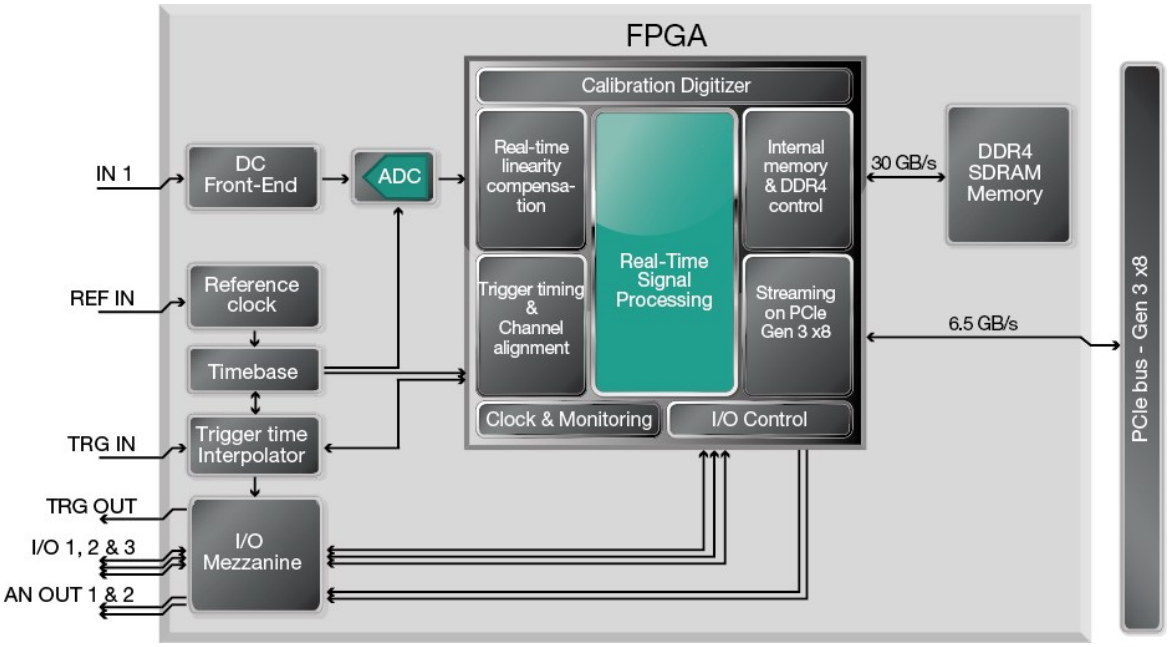
The Acqiris SA2 is a high-performance 14-bit ADC card platform, performing signal acquisitions from 1 GS/s up to 10 GS/s, with excellent signal fidelity across a wide bandwidth. This new card and module generation with advanced real-time processing capabilities is designed for embedded OEM applications in a variety of challenging measurements, imaging and processing systems.

Product description

The SA217PADC card is the 2 GS/s single channel version of the SA2 product family. This unique DC-coupled 14-bit digitizer captures waveforms from DC up to 1.2 GHz.

Featuring very long acquisition memory up to 8 GB, the SA217P includes a powerful Xilinx Kintex UltraScale FPGA offering real-time signal processing capability such as waveform averaging or peak listing. The PCIe Gen 3 interface enables high data transfer rate and streaming capabilities to the host computer at up to 6.5 GB/s. This ADC card occupies a single PCIe slot, offering high performance in a small footprint.

All the ADC cards and modules from the SA2 family implement a proprietary low noise front-end. With undisputed spurious-free dynamic range (SFDR) and signal noise ratio (SNR) performances in high frequencies, it is ideal for OEM applications requiring digitizer sampling at wide bandwidth and very high dynamic range, especially at 500 mV full scale range (FSR). Moreover, optimized response allows few hundred picoseconds pulse analysis. Overall performance enables final products to measure deeper, faster and more precisely.



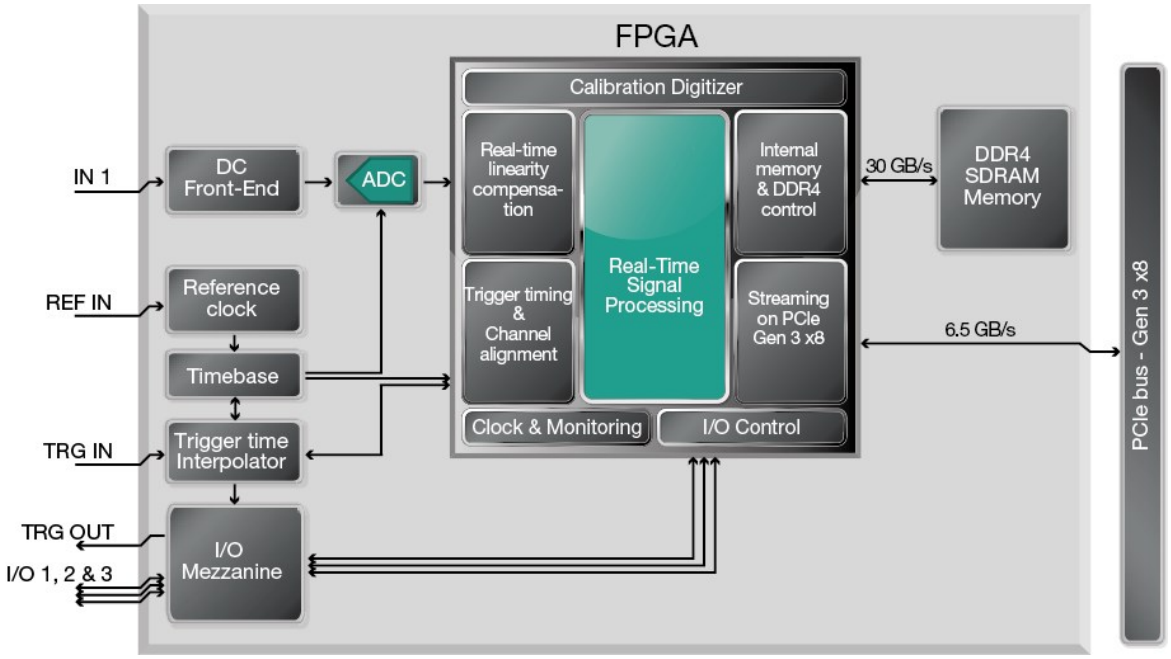


Figure 1.1 - SA217P block diagram

Most of the technical specifications concerning your particular ADC card are covered in this manual, however for the complete specifications please refer to the SA217P datasheet.

Product configurable options

The SA217P comes with several options:

Additional memory:

- 4 GB (MEA option)
- 8 GB (MEB option)

ADC Card modes:

- Digitizer mode (DGT)
- Real-time averaging (AVG option)

Optional features:

- Simultaneous acquisition and readout - Streaming records (CST option)
- Zero Supress - Thresholding (ZS1 option)
- Custom firmware capability (CFW option)

Other internal references:

- I/O ports for SA217P: 3 I/Os,1 trigger output (EXC)

Chapter 1

Main Card Features

1.1 SA217P front panel features

Front panel connectors

Connector	Type	Description
TRG IN	MMCX female	External trigger input, 50 Ω DC terminated, ± 5 V range.
IN 1	SMA female	Analog signal inputs, DC-coupled and 50 Ω terminated. The input full scale ranges are selectable:
TRG OUT[1]	MMCX	Trigger Out signal (programmable). 50 Ω source, LVCMOS 3.3 V
I/O 1, 2, 3	MMCX	User configurable digital Input / Output signal. DC coupling, LVCMOS 3.3 V. Output: 50 Ω source, Input: +5 V max.
REF IN	MMCX	External reference clock input, AC coupled and 50 Ω terminated. It can accept a 10 MHz or a 100 MHz signal from -3 to +3 dBm.

Voltage	500 mV FSR	2.5 V FSR
Recommended maximum operating voltage	± 600 mVpk	±3 Vpk

Table 1.1 - List of SA217P front-panel IOs.

Note The ADC card can usually work with signal present at the external reference input (REF IN). However, to ensure the best performance, or if the calibration is found to be unreliable, it is recommended to remove such signals when working with internal clock.

Warning All the inputs shall not be driver until the SA217P is powered on.

[1] The trigger out connector depends on the product version.

1.2 Channel input specifications

This section provides information and specifications regarding the input characteristics of the ADC card.

The SA217P provides one 14-bit DC-coupled channel at the sampling rate of up to 2 GS/s.

Channel input

The SA217P has the following front end capabilities:

Coupling / Impedance	Full Scale Ranges (FSR)	Maximum operating voltage	Input voltage offset
DC / 50 Ω	500 mV	± 600 mVpk	± FSR/2
	2.5 V	± 3 Vpk	± FSR/2

Table 1.2 - Channel input specifications.

Impedance & coupling

The input channel termination is 50 Ω. The input coupling is DC.

Input protection

The input amplifiers are designed to accept signals within the absolute maximum operating voltages shown in the table.

Bandwidth and rise time

The bandwidth specification indicates the frequency at which an input signal will be attenuated by 3 dB (approximately 30% loss of amplitude).

The bandwidth of the SA217P is from DC to 1.2 GHz *(typical)*.

The bandwidth also has an impact on the minimum rise and fall times that can be passed through the front-end electronics. A pulse with a very sharp edge will be observed to have a minimum rise time T_{min} determined by the front-end electronics. In general a pulse with a given 10-90% rise time $T_{10\text{-}90real}$ will be observed with a lower value given by:

$$T_{10\text{-}90}^2 = T_{10\text{-}90real}^2 + T_{min}^2$$

where $T_{min}(\text{ns}) \approx 0.35/BW(\text{GHz})$.

Bandwidth limitation

Not yet supported.

Vertical resolution

The SA217P uses 14-bit ADCs giving 16384 levels at each input full scale range i.e. 16384 level of ~152 µV average width when using the 2.5 V FSR, or 30 µV using the 500 mV FSR. See Acquired data format (page 21) for more details.

1.3 Sampling and data acquisition

The ADC card acquires waveforms in association with triggers. Each waveform is made of a series of measured voltage values (sample points) coming from the ADC at a uniform sampling rate.

Sampling rate

The SA217P acquisition card contains an analog-to-digital conversion (ADC) system that can sample waveforms, in a real time sampling mode, at the maximum rates shown in the table below.

Model	Max. Sampling Rate	Available Channels	Resolution	Acquisition Modes
SA217P	2 GS/s (default),	1	14 bits	Single or multi-record (up to 131'072 records) or continuous steaming with CST option.

Table 1.3 - Acquisition sampling rate and resolution per channel.

Data acquisition modes and functions

The SA217P ADC card supports several acquisition modes and optional functions for real-time signal processing in FPGA. You can refer to the corresponding section for details.

ADC card modes:

- Digitizer acquisition mode (page 19)
- Real-time averaging mode (AVG option) (page 1)

Available signal processing features:

- Sampling rate reduction (decimation) (page 25)
- Data inversion (page 26)

Read out modes:

- One shot with single waveform
- One shot with multiple waveforms
- Continuous Simultaneous acquisition and readout, with Trigger (CST option) (page 1)

1.4 Calibration

The SA217P is factory calibrated and shipped with a calibration certificate.

The internal calibration refers to the adjustment of ADC card internal parameters, corresponding to user selected parameters and required before starting acquisition.

Internal calibration

The internal calibration (or self-calibration) measures and adjusts the internal timing, gain and offset parameters between the ADCs and against a precise reference.

The ADC card includes a high precision voltage source and a 16-bit DAC, used to perform the input voltage and offset calibration.

The supplied software drivers include self-calibration function which can be executed upon user request. The ADC cards are never self-calibrated in an automatic way, (i.e. as a consequence of another operation). This ensures programmers have full control of all calibration operations performed through software in order to maintain proper event synchronization within automated test applications.

Note For accurate time and voltage measurements it is recommended to perform a self-calibration once the module has attained a stable operating temperature (usually reached after 20 minutes of ADC card operation after power on).

A full internal calibration of a ADC card can be time consuming because of the many possible configuration states. Therefore, the self-calibration is performed only for the current configuration state, and is mandatory before making the first acquisition with given settings. Indeed the AqMD3 driver prevents an acquisition from being performed unless a self-calibration has first been completed. Note that some configuration changes do not require a new self-calibration. To avoid unnecessary self-calibrations, the `IAqMD3Calibration.IsRequired` IVI.NET property or the `AQMD3_ATTR_CALIBRATION_IS_REQUIRED` IVI-C attribute should be queried.

Caution ADC card can usually work with signals present at the channel input, or trigger input. However, to ensure the best performance, or if the calibration is found to be unreliable (as shown by a calibration failure status), it is recommended to remove such signals.
Similarly, when working with internal clock, it is recommended to remove external reference input during calibration to avoid parasitic effects.

Caution It is not recommended to perform multiple successive calibrations. If a recurrent calibration failure occurs, in case of specific application, please contact support for advice.

TIP Switching from one operational modes to another one (e.g. from Averager mode to Zero-Suppress mode) does not need to recalibrate the card.

Smart-calibration

The smart calibration implemented in MD3 drivers allows to save time by automatically keeping in memory the calibration information from any self-calibration performed since the beginning of the session. When the acquisition parameters are changed, no re-calibration of the card is necessary if a self-calibration has already been performed with the same acquisition conditions (i.e. the same set of parameters), unless the clock mode parameters are changed.

Indeed, any change in the clock mode parameters (i.e. **External clock frequency** or **Reference mode** parameters), induces a restart of the clocks which requires a new self-calibration.

For details, see Parameter change requiring a new self calibration (page 45).

Guidelines

The input cables can be connected at any time on the channel input connectors.

However:

- Do not drive any active signal on the input channels until the DAQ Module is powered on and correctly initialized.
- Avoid any active signal driving the input channel during the calibration.
- For SA2 DAQ modules that provide multiple full-scale ranges, it should be ensured that the maximum DC input voltages are met, especially while switching from larger full-scale range to the lower one.

Special care should be taken for applications that drive the DAQ modules with an equivalent current source. During power-up and power-down, calibration or configuration changes, the channel input impedance changes would induce an input voltage that might exceed the user's measured voltage at the channel input.

Factory calibration

Factory calibration is the process of measuring the actual performance of a device-under-test (DUT) using laboratory instruments that have significantly better performance than the DUT. Laboratory instrument performance must be traceable to the International System (SI) Units via a national metrology institute (NIST, NPL, NRC, PTB, CENAM, INMETRO, BIPM, etc.)

The measured performance is then compared to published datasheet specifications. For each factory calibration, Acqiris tests the performance corresponding to all datasheet specifications, for every installed option. If needed, the DUT is adjusted and re-qualified ; ensuring it is in line with full specifications.

Our ADC cards are calibrated at factory during the production phase. There is no need to systematically calibrate each year.

Firstly, the cards include a self-calibration function providing a good degree of confidence that your card is operating within its specifications on a day-to-day basis, and triggering an error message if out of calibration relative to the internal calibration signal.

Secondly, our cards are warranted to stay within specification over the standard 5-year warranty. They usually stay within specification much longer and we rarely have to effectively recalibrate the cards.

Lastly, a onetime calibration can be ordered in case customer detects a deviation in the measure of its final product that appears to be caused by the ADC card. The onetime calibration consists in processing the card through production test to determine if it is still within specification:

- If yes, the ADC card is returned with the certificate of calibration which certifies it is within specification.
- If not, the required calibration is performed, and another production test is done to provide the certificate of calibration.
- If repair is required, and the card is out of warranty, a repair quote will be provided.

For more information, or to request for a calibration, please contact technical support support@acqiris.com.

1.5 Trigger

The trigger settings applied to the ADC card are used to determine at which time the device will start recording data. The various trigger settings are outlined below.

Trigger source

The trigger source can be:

- the signal applied to an input channel (internal triggering)
- an external signal applied to the TRG IN front panel input connector (external triggering)
- a software trigger (See How to generate a software trigger? (page 47)).

The different trigger modes are detailed in section Trigger modes and time-stamps (page 28)

Trigger impedance & coupling

The SA217P has a fixed 50 Ω termination impedance with DC coupling.

Trigger input bandwidths

The bandwidth depends on the trigger source.

Channel trigger

The -3 dB bandwidth of the comparator of the channel triggers is from DC to 2.5 GHz.

External trigger

The external trigger input has a bandwidth from DC to 3 GHz.

Refer to section How to set the external trigger? (page 49) for additional information.

Trigger level

The trigger level specifies the voltage at which the selected trigger source will produce a valid trigger. All trigger circuits have sensitivity levels that must be exceeded in order for reliable trigger to occur.

Both the external trigger input and channel triggers have a hysteresis of 1% of FSR (Full Scale Range).

On the external trigger, the Full Scale Range FSR is ±5 V, therefore the ADC card will trigger on signals with a peak-to-peak amplitude > 0.5 V.

On the internal trigger, the ADC card will trigger on signals with a peak-to-peak amplitude > 5% full scale range.

When using the channel triggers, the trigger level must be set within Offset ± FSR .

Edge trigger slope

The trigger slope defines which one of the two possible transitions will be used to initiate the trigger when it passes through the specified trigger level. Positive slope indicates that the signal is transitioning from a lower voltage to a higher voltage. Negative slope indicates the signal is transitioning from a higher voltage to a lower voltage.

Trigger delays

For more details about triggers modes, post/pre-trigger delays and time-stamps, see Trigger modes and time-stamps (page 28).

Chapter 2

Signal Acquisition Modes and Real-Time Processing

Thanks to the powerful Xilinx Kintex Ultrascale FPGA, the SA217P enables real-time signal pre-processing on the sampled data e.g. data compression or noise reduction. This allows to save analysis time and make the user application running faster. Acqiris proposes various firmware and real-time processing features in the FPGA, so on-board processing can be optimized for each application and eventually for each of your system.

This section details the acquisition modes.

TIP

The modes available with your product depends on the firmware options ordered with your products. To check which options and mode are present on your ADC card you can use the MD3 Software Front Panel from the: **Windows Start Menu > Acqiris > MD3 > Acqiris MD3 SFP**. Then use the menu **Help > About**. The field **System Options** gives the option list.

TIP

Easy mode switch

A simple change of acquisition mode allows to automatically switch, for instance, from the digitizer to the average mode. The corresponding firmware switches automatically.

2.1 Digitizer acquisition mode

The digitizer mode (normal mode) allows standard data acquisition, including: ADC card initialization, setting of the acquisition, management of channel triggering for best synchronization, storing data in the internal memory and/or transferring them to the host computer.

The digitizer mode is the acquisition mode by default.

Note For units ordered without the Digitizer mode (DGT option), a basic digitizer function is still available with some limitations. For the list of limitation, please refer to section Using basic digitizer function (without DGT option) (page 22).

Single and multi-record acquisition modes

To maximize sampling rates and utilize memory as efficiently as possible the ADC cards include both single and multi-record modes. For both of these modes the data of all of the active channels is acquired synchronously; all of the ADC's are acquiring data at the same time, to within a small fraction of the maximum sampling rate.

The **single record acquisition mode** is the normal operation of most ADC card products. In this mode an acquisition consists of a waveform recorded with a single trigger. The user selects the sampling rate and record size, and sets the number of records to 1 (default value). For details about the trigger sources, see Trigger (page 16).

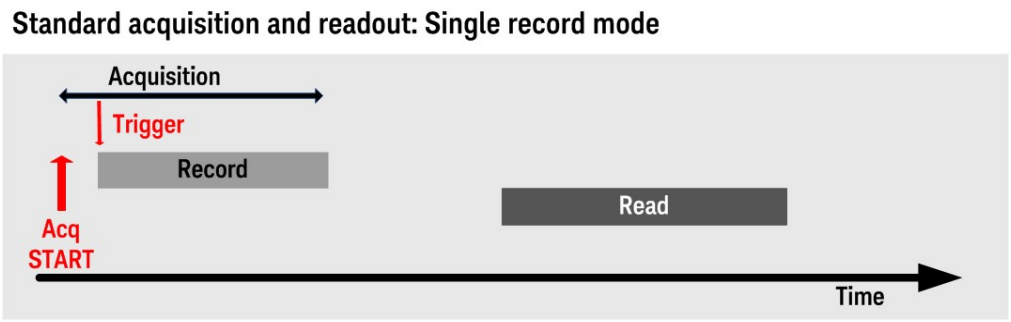


Figure 2.1 - Acquisition sequence using a single record.

The ADC cards also feature a **multi-record acquisition mode**. This mode allows the capture and storage of consecutive single waveforms. Multi-record acquisition mode is useful as it can optimize the ADC card's sampling rate and memory requirements for applications where only portions of the signal being analyzed are important. The mode is extremely useful in almost all impulse-response type applications (RADAR, SONAR, LIDAR, Time-of-Flight, Ultrasonics, Medical and Biomedical Research, etc.).

In multi-record acquisition mode the acquisition memory is divided into a pre-selected number of records. Waveforms are stored in successive memory records as they arrive. Each waveform requires its own individual trigger.

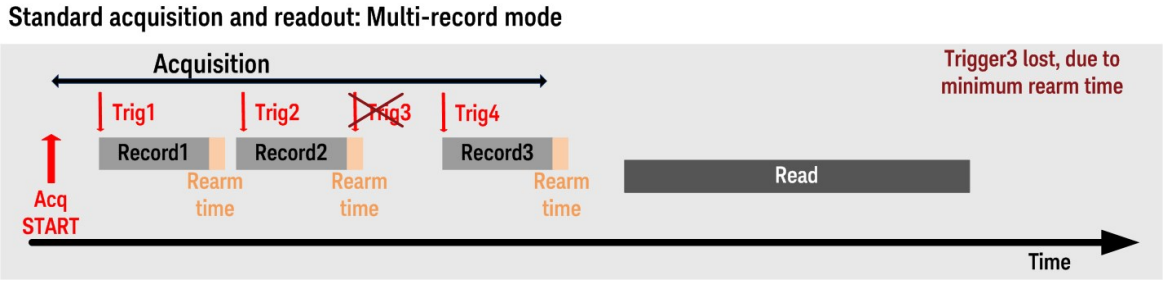


Figure 2.2 - Acquisition sequence using a multi-records. It is possible to miss a trigger at high trigger rate, as illustrated with trigger 3.

The **multi-record acquisition mode** enables successive events, occurring within a very short time, to be captured and stored without loss. A very fast trigger rearm time is a crucial feature for multi-record acquisitions. Thanks to fast trigger rearm, the SA217P achieves very low “dead time” between the records of a multi-record acquisition. The “dead time” is the period after the end of an event when the card cannot accept a new trigger event. The re-arm time is provided in the SA217P datasheet.

The maximum number of records is 128k.

TIP Program examples for single record or multi-records acquisitions are available:
C:\Program Files\IVI Foundation\IVI\Drivers\AqMD3\Examples

Acquisition memory

Data from the ADC is stored in on-board acquisition memory. The amount of memory in use for acquisition can be programmed and is selectable from 1 point to the full amount of acquisition memory available.

Model	Memory option ordered	Acquisition memory ordered	Max samples/channel
SA217P	-MEA (default)	4 GB	2 GSamples
	-MEB (optional)	8 GB	4 GSamples

Table 2.1 - Maximum number of samples which can be recorded per channel, depending on ordered memory option.

For technical reasons, a certain acquisition memory overhead is required for each waveform, reducing the available memory by a small amount.

TIP

The effective maximum memory available for acquisition depends on several parameters, such as the acquisition mode (single / multi-record / streaming), sampling rate, record size, number of records, trigger delay, etc.... This maximum is determined by the driver for each specific configuration. The **`AQMD3_ATTR_MAX_SAMPLES_PER_CHANNEL`** attribute in IVI-C or **`IAqMD3Acquisition.MaxSamplesPerChannel`** property in IVI.NET can be used to retrieve the maximum number of samples per channel that can be acquired for a specific configuration. When using the Soft Front Panel, the **Max Samples** per channel parameter is given on the **Acquisition panel**.

Acquisition time (Timebase range)

The timebase range defines the time period over which data is being acquired.

For example, the SA217P has a standard acquisition memory of 4 GB, i.e. 2 GSample and a sampling rate of 2 GS/s. Therefore, at the maximum sampling rate, the ADC card can record a signal over a time window of up to 1 s.

Model	Memory option ordered	Acquisition memory	Max sampling rate	Max recording time window at higher sampling rate
SA217	-MEA (default)	4 GB	2 GS/s	1s
	-MEB (optional)	8 GB		2s

Table 2.2 - Maximum recorded time at maximum sampling rate, depending on ordered memory option.

Maximum acquisition time

There is a limit on the acquisition time / acquisition length in digitizer mode depending on the record size, post trigger delay and binary decimation factor. Above this limit, the driver returns a post-trigger overflow.

Acquired data format

The raw 14-bit data is subjected to post-calibration processing, which compensates for gain and offset errors in the internal ADCs, The result of this post-processing is then stored and read-out as a 16-bit value. For this reason the returned data will not always be divisible by 4 as may be expected, neither equally spaced.

Signed left-aligned 16-bit ADC code

The signed, raw ADC code is shifted to the left to align to 16 bits. The result is then converted, with the sign, to the final format (16, 32 or 64 bits).

For the 14-bit SA217P, signed left-aligned 16-bit ADC code means that the signed raw ADC code is shifted to the left by 2 bits, and coded in 2’s complement to the final number of bits (16, 32 or 64) as illustrated below.

	Signed raw (binary)	Signed left-aligned 16-bit (binary)
max ADC code	8191 (01 1111 1111 1111)	32764 (0111 1111 1111 1100)
min ADC code	-8192 (10 0000 0000 0000)	-32768 (1000 0000 0000 0000)
	-1 (11 1111 1111 1111)	-4 (1111 1111 1111 1100)
	1 (00 0000 0000 0001)	4 (0000 0000 0000 0100)

Table 2.3 - Structure of the signed left-aligned 16-bit ADC code

Note that because of ADC linearity corrections, all 16 bits should be considered when using the samples values in ADC code format.

Using basic digitizer function (without DGT option)

If the ADC card had been ordered with a specific application mode and without the digitizer mode (.i.e. without DGT firmware option), user can perfom the basic digitizer functions described in previous sections but with restriction for the following features:

- Limitation of the maximum record size
- Limitation of the maximum number of records
- Decimation capability
- No baseline correction
- No self-trigger capability
- No data inversion capability.
- No pre-trigger
- Limitation of the post-trigger range.

Chapter 3

Readout Modes

3.1 Standard readout modes

As presented in the Digitizer acquisition mode (page 19), the two standard acquisition and readout modes are:

- Single record mode: one shot with single waveform, with a single trigger
- Multi-record mode: one shot with multiple waveforms, with multiple triggers

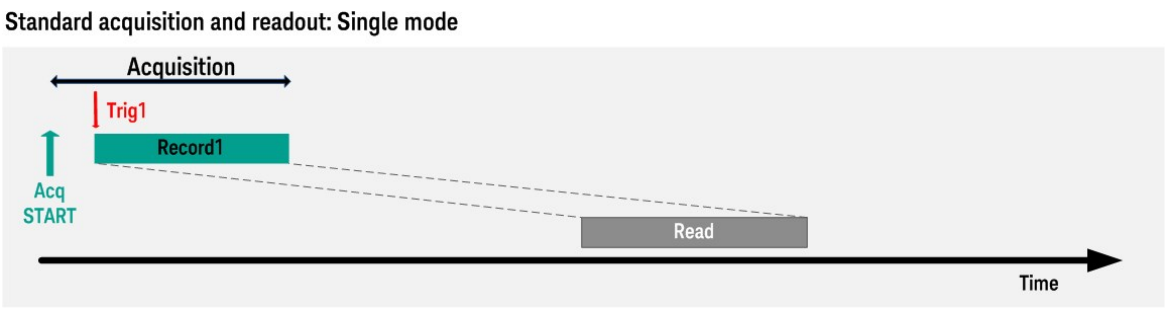


Figure 3.1 - Acquisition sequence using a single record.

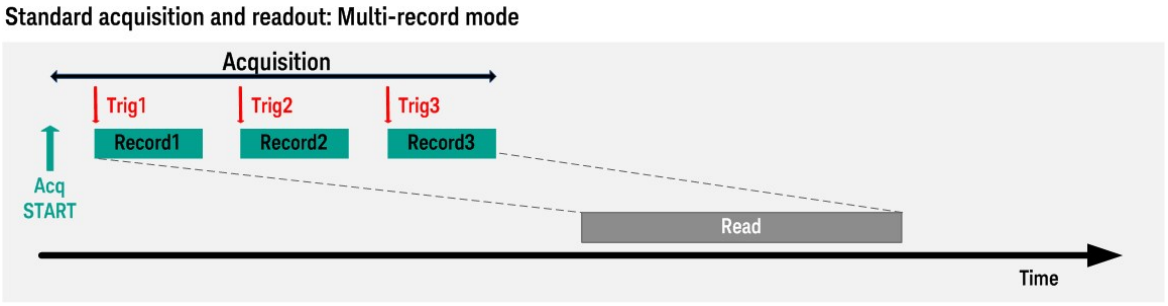


Figure 3.2 - Acquisition sequence using multi-records.

The specific readout mode(s) detailed in the following depends on your product version and ordered options.

Chapter 4

Other Signal Processing Features

This sections presents the on-board signal processing features that can be enable e.g. to optimize signal performance or reduce data volume, depending on each application.

These features are common to the acquisition modes, excepted when specified differently.

4.1 Sampling rate reduction (decimation)

A programmable decimation can be used to lower the sample rate (See table below). Sampling rate reduction can be used in both digitizer and averager acquisition mode.

Decimation Ratio	Resulting sampling rate
No	2 GS/s
2	1 GS/s
4	500 MS/s

Table 4.1 - List of selectable sampling rates.

$$Decimated\ sampling\ rate = \frac{Sampling\ rate}{Decimation\ ratio}$$

Note The decimation can be used in digitizer or averaging mode. The zero-supress mode is not compatible with the sampling rate reduction (decimation).

To enable decimation, user should set the sampling rate to required decimated sampling rate.

4.2 Data inversion

By enabling the channel data inversion capability, the signal will be inverted.

This feature is available with both digitizer or real-time averaging modes.

When enabled, the signal inversion is applied before the NSA settings or the signal thresholding.

Configuration

IVI-C

Attribute	Description
AQMD3_ATTR_CHANNEL_ DATA_INVERSION_ ENABLED	Specifies whether the data acquired is inverted.

IVI.NET

Interface	Method / Property name	Description
IAqMD3Channel	DataInversionEnabled	Specifies whether the data acquired is inverted.

Chapter 5

Control and Synchronization

5.1 External reference

For applications for which the user wants to replace the internal clock of the acquisition card and drives the ADC with an external source, an external reference signal can be used. The reference signal can be entered into the ADC card by the dedicated **REF IN** connector.

External reference (REF IN)

For applications that require greater timing precision and long-term stability than is obtainable from the internal clock, a 10 MHz or a 100 MHz reference signal can be used.

The external reference is nominally at 10 MHz or 100 MHz. However, frequencies in a range will be accepted. If your input is not at exactly the specified value, you must remember to compensate for the difference in your application since the ADC card and the driver have no way to know about such deviations.

Parameter	Value	Tolerance
Nominal Frequency	100 MHz or 10 MHz	±1 kHz
Signal level	-3 dBm to +3 dBm	
Impedance	50 Ω	
Coupling	AC	
Minimum amplitude	440 mVpp (sinus)	
Maximum power	2 mW	
Maximum voltage	900 mVpp (sinus)	

Table 5.1 - External reference specifications.

If synchronization between several ADC cards is required, the reference signal should be applied to all of them.

5.2 Trigger modes and time-stamps

Trigger modes

As listed previously the trigger source can be:

- the signal applied to an input channel (internal triggering)
- an external signal applied to the TRG IN front panel input connector (external triggering)
- a software trigger (See How to generate a software trigger? (page 47))

Pre- and post-trigger delay

Description

To increase trigger flexibility, a pre- or post-trigger delay can be applied to the trigger position.

Note The pre-trigger is not supported in Averager mode or when in combination with one of the following features: baseline stabilization, zero suppress (ZS1), data inversion capability or simultaneous acquisition and readout (CST).

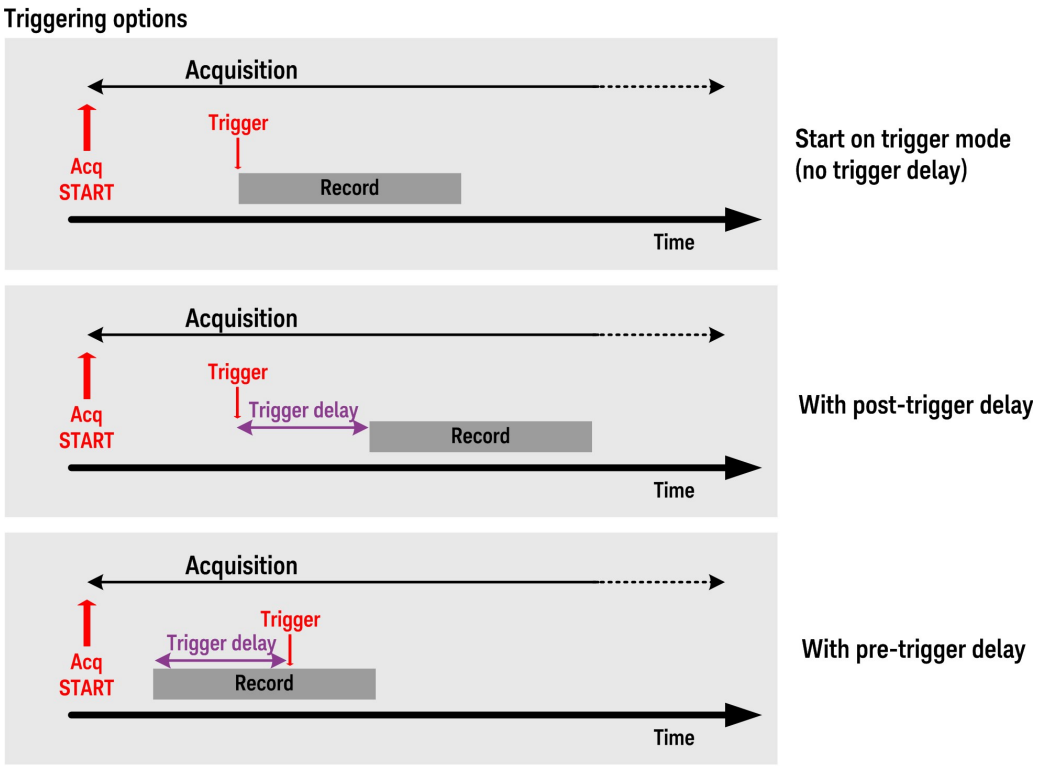


Figure 5.1 - Acquisition timeline depending on the trigger delay defined.

Trigger delay parameter

Digitizer acquisition mode

The amount of pre-trigger delay can be adjusted between 0 and 100% of the acquisition time window, thus the minimum trigger delay is given by:

```
-1 * Record size / SamplingRate
```

The maximum post-trigger delay is given, respectively in sample or seconds by:

$(2^{24} - 1) * 8$ (in samples)

$(2^{24} - 1) * 8 / SamplingRate$ (in seconds[1])

Note When using a reduced sampling rate, the decimation changes the `SamplingRate` to apply in previous formulas.

Pre- or post-trigger delays are just different aspects of the same trigger positioning parameter:

- The condition of 100% pre-trigger indicates that all data points are acquired prior to the trigger, i.e. the trigger point is at the end of the acquired waveform.
- The condition of 0% pre-trigger (which is identical to a post-trigger of 0%) indicates that all data points are acquired immediately after the trigger, i.e. the trigger point is at the beginning of the acquired waveform.
- The condition of a non-zero post-trigger delay indicates that the data points are acquired after the trigger occurs, at a time that corresponds to the post-trigger delay, i.e. the trigger point is before the acquired waveform.

By definition post-trigger settings are a positive number and pre-trigger settings are a negative number.

The trigger delay granularity is less than one sample interval (~ 0s).

Averager acquisition mode

In this mode the pre-trigger delay is not supported. Thus the minimum trigger delay is 0s.

The maximum trigger delay is given by:

$(2^{24}-1)*816 / SamplingRate$

The trigger delay granularity is given by:

$816 / SamplingRate$

Trigger time interpolator and time-stamps

The trigger time-stamp is the trigger arrival time.

The ADC card accurately measures and stores these time-stamps using the information from the on board Trigger Time Interpolator (TTI). This information is essential for determining the precise relation between the trigger and the digitized samples of the signal. The TTI resolution determines the resolution of the trigger time-stamps.

Please refer to Trigger section of your SA217P datasheet for the relevant specifications.

Managing time-stamps

The ADC card features a time-stamp counter. In multi-record acquisitions, each acquired record has a precise time-stamp, given by the time-stamp counter.

The accurate time-stamp of each trigger is given by the sum of the time reference and the time-stamp counter.

The time reference is configured:

[1] Actual limit might be slightly smaller.

- at the first card initialization,
- at the last initialization with reset flag = 1,
- or when using `Set Time` function (IVI-C) or writing `Time` (IVI.NET).

The time-stamp counter is reset each time the time reference is configured.

User can control the reset of the time-stamp counter thanks to the `TimeResetMode` property:

- `Immediate`: The time-stamp counter is reset upon software, using `Set Time` function (IVI-C) or writting `Time` (IVI.NET), then it continues counting freely.
- `OnFirstTrigger`: The time-stamp counter is reset by the first trigger of a (multi-record) acquisition.

Parameters

IVI-C

`InitialXTimeSeconds`: Specifies the seconds portion of the absolute time at which the first data point was acquired.

`InitialXTimeFraction`: Specifies the fractional portion of the absolute time at which the first data point was acquired.

The actual time is the sum of `InitialXTimeSeconds` and `InitialXTimeFraction`.

Note Adding these values in a variable of type double implies a loss of precision.

IVI.NET

`StartTime`: StartTime is the time between the first valid data point (that is the data point at index FirstValidPoint) in the waveform and the trigger. Positive values indicate that the StartTime occurred after the trigger. If StartTime is zero, the waveform is not relative to anything, or the relative measure is zero, or the waveform is empty.

`EndTime`: EndTime is the time between the last valid data point in the waveform and the TriggerTime. Positive values of EndTime indicate that it occurred after the trigger. If EndTime is zero, there is exactly one data point and the StartTime is zero, or the waveform is empty.

`TotalTime`: TotalTime is the timespan represented by the valid points in the waveform. Numerically, it is equivalent to the IntervalPerPoint * (ValidPointCount - 1). It is also numerically the EndTime – StartTime. TotalTime is zero if there is exactly one data point in the waveform, or the waveform is empty.

`TriggerTime`: TriggerTime is the absolute time at which this measurement was triggered. Note that this differs from Start Time in that the trigger may have occurred at some time other than when the first data point was captured, as in pre-trigger or post-trigger applications. TriggerTime is an absolute time and cannot be set to zero. If it is set to NotATime, the waveform is empty or there is no absolute reference for the waveform.

5.3 Trigger output

A trigger output pulse can be generated for external synchronization.

When the ADC card is ready to be triggered and a valid trigger signal occurs, a trigger output pulse is generated. This signal is available on the front panel TRG OUT MMCX connector, and may be enabled or disabled as required.

In idle state, Trigger Out signal is low. When a trigger is accepted a high level pulse occurs.

There are several trigger sources or signals which may be assigned to the trigger out connector (See table below).

Trigger out sources	Description of signal
TriggerAccepted (default)	A pulse signal is sent directly to the Trigger Out.
TriggerAcceptedResync	A pulse signal resynchronized to a sub-multiple of sample clock is sent to the Trigger Out.
TriggerCompare	The trigger input condition has been satisfied, but not necessarily triggered, e.g. the trigger enable was not asserted.

Table 5.2 - List of supported trigger out signals.

Trigger output signal behavior

By default, the trigger output is **LowLevel**.

If selecting a trigger output source, e.g. **TriggerAcceptedResync** or **TriggerAccepted** as trigger output source, when the ADC card is ready to be triggered and a valid trigger signal occurs, a trigger output pulse is generated.

Selecting the trigger output source

The trigger output can be selected using following properties / attributes:

Driver	Attribute / Property	Available Instance Value
IVI-C	AQMD3_ATTR_TRIGGER_OUTPUT_ENABLED	Boolean
	AQMD3_ATTR_TRIGGER_OUTPUT_SOURCE	TriggerAccepted, TriggerAcceptedResync, TriggerCompare, SelfTrigger
IVI.NET	IAqMD3TriggerOutput.Source	
	IAqMD3TriggerOutput.Enabled	Boolean

Specifications

In the default software configuration, the output swing is 3.3 V, when unloaded and 1.6 V when terminated on 50 Ω.

The maximum output current capability is ± 15 mA. As the output is retro-terminated, it is possible to drive a 50 Ω line un-terminated (HiZ) without loss of performance.

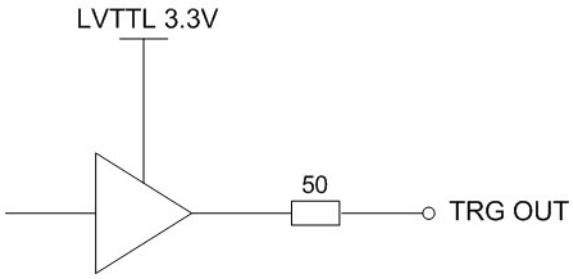


Figure 5.2 - Trigger output block diagram.

Note The external trigger output functionality is implemented in the hardware. No trigger out signal occurs for software-generated triggers.

5.4 Multi-purpose inputs and outputs

The multi-purpose I/O connectors may be used for any of the functions shown in the following table:

IO Connector Functions	Type	Description of signal	Mode / Option	Notes
Inputs				
Disabled	-	IO connector is disabled.		
In-BScanSync	Pulse	SS-OCT application: B-scan Trigger Input.	-SS4/SS5	IO 2 only
In-CScanSync	Pulse	SS-OCT application: C-scan Trigger Input.	-SS4/SS5	IO 3 only
Outputs				
Out-LowLevel	Level	Fixed 'low' level signal for debug purposes.		
Out-HighLevel	Level	Fixed 'high' level signal for debug purposes.		
Out-AScanSync	Pulse	SS-OCT application: A-scan Trigger Output (Requires -SS4 option).		IO 1 only.
Out-AScanEna	Level	SS-OCT application: A-scan Trigger Enable (Requires -SS4 option). If High, the card is processing A-scans		IO 1, 2 or 3.
Out-AScanUp	Level	SS-OCT application: A-scan Trigger Up (Requires -SS4 option). Used to distinguish source up/down (forward/backward sweeps).		IO 1, 2 or 3.
Out-BScanSync	Pulse	SS-OCT application: B-scan Trigger Output (Requires -SS4 option).		IO 2 only.
Out-CScanSync	Pulse	SS-OCT application: C-scan Trigger Output (Requires -SS4 option).		IO 3 only.

Table 5.3 - List of signals selectable for the programmables I/Os .

The list of Available signals is indicated (as a comma separated list) by member **`IAqMD3ControlIO.AvailableSignals`** (IVI.NET) or attribute **`AQMD3_ATTR_CONTROL_IO_AVAILABLE_SIGNALS`** (IVI-C).

Signal Logic Levels

The multi-purpose IO signals are 3.3 V CMOS compatible (5V Tolerant buffer). The levels shown in the table below should be observed.

Direction	Low level	High level
Input	< 0.8 V	> 2.0 to 3.45 V
Output	In the range 0 to 0.8 V	In the range 1.6 to 3.3 V

Table 5.4 - Logic levels.

As an Input

The input is high-impedance and will be pulled high if unconnected via an internal weak pull-up (10 k pull-up resistor).

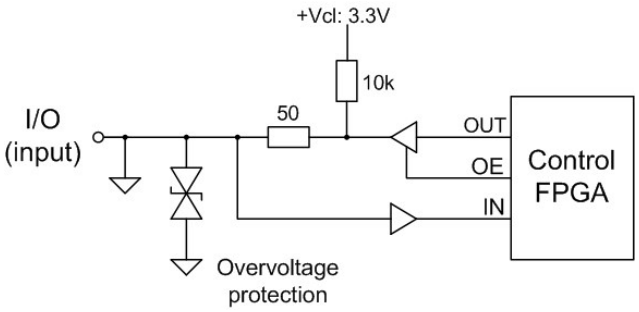


Figure 5.3 - Programmable IO schematic

As an Output

The high level output will typically give 1.6 V into 50 Ω. As can be seen in the diagram below, the 3.3 V output buffer has a 50 Ω resistor in series. Therefore the available output high level voltage will depend on the load applied. In the example below a 50 Ω termination will result in a nominal high level of 1.6 V.

(Vo = ($Rload$/(50 + $Rload$)) * 3.3).

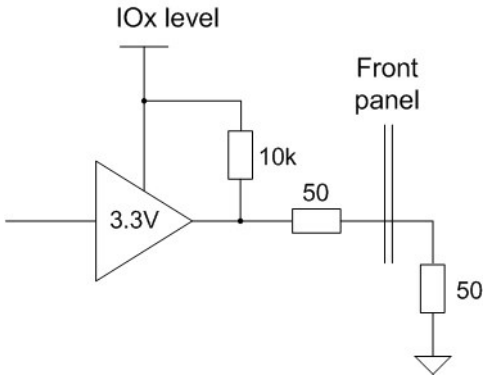


Figure 5.4 - Output equivalent circuit.

Chapter 6

Programming Information

This section provides general programming information regarding the use of the Acqiris drivers.

The AqMD3 IVI driver provides access to the functionality of AqMD3 ADC cards through a .NET or ANSI C API which also complies with the IVI specifications.

6.1 Overview of the AqMD3 Driver

Development environments

IVI-C Driver

The **AqMD3 IVI-C** driver can be used in the following development environments: Visual C++, LabWindow/CVI, LabVIEW, MATLAB.

IVI.NET Driver

The **AqMD3 IVI.NET** driver can be used in the following development environments: Visual C#, Visual C++/CLI, Visual Basic.NET

Driver API documentation

The AqMD3 APIdocumentation can also be accessed from:

IVI-C: **Start > Acqiris > MD3 > Documentation > AqMD3-C IVI Driver** *Version#* **Documentation**

IVI.NET: **Start > Acqiris > MD3 > Documentation > AqMD3.NET IVI Driver** *Version#* **Documentation**

or from:

IVI-C : C:\Program Files[1]\IVI Foundation\IVI\Drivers\AqMD3\AqMD3.chm

IVI.NET: C:\Program Files\IVI Foundation\IVI\Drivers\AqMD3\Acqiris.AqMD3.Fx40.chm.lnk

You can also found more information about IVI at http://www.ivifoundation.org/resources.

[1](Or your installation path)

Program examples

You may build and run the example programs installed with the driver. They demonstrate driver usage in a variety of application development environments.

Program examples can be found in C:\Program Files\IVI Foundation\IVI\Drivers\AqMD3\Examples

6.2 Programming with the IVI-C Driver in various development environments

IVI-C drivers are implemented using standard Windows DLL technology. Consequently, IVI-C drivers can be used in a wide variety of development environments. The topics in this section provide detailed instructions on how to access and use IVI-C drivers in a variety of popular development environments. Each topic includes a complete example of IVI-C driver usage.

Using Visual C++

Explains how to use the IVI-C driver from Visual C++ .

Referencing the Driver

In order to access any of the driver functions, the proper header file (.h) must be included in the project and the proper import library (.lib) must be referenced. This section demonstrates usage of the driver using instrument-specific references.

All IVI-C driver programs must do the following:

- #include <AqMD3.h>
- Link to AqMD3.lib
- Prefix function calls with "AqMD3_"

To use the AqMD3 specific driver, perform the following steps in Visual Studio.

1. In solution explorer, right-click on the project and choose **Properties**.
2. In the property pages dialog, expand the **Linker** node and select **Input**.
3. In the **Additional Dependencies** field, enter "AqMD3.lib".
4. Click **OK**.
5. In the main application source file (.cpp), add the following line to the top of the file:

```
#include <AqMD3.h>
```

Visual Studio must know the path to your driver's library (.lib) and header (.h) files. You can enter the following paths using the Tools, Options, Projects and Solutions, VC++ Directories dialog.

- For Include Files (Win32 & x64) add: C:\Program Files (x86)\IVI Foundation\IVI\Include
- For Library Files (Win32) add: C:\Program Files (x86)\IVI Foundation\IVI\Lib_\msc
- For Library Files (x64) add: C:\Program Files (x86)\IVI Foundation\IVI\Lib_x64\msc

Note: For 32 bit operating systems paths start with: C:\Program Files\

Alternately, these paths may be entered in the Project Properties dialog, Configuration Properties, C++ and Linker panes.

Initializing the Driver

Calling **AqMD3_InitWithOptions** will establish an I/O connection to an instrument (often referred to as an "I/O session") or setup the driver to work in simulation mode. Calling **AqMD3_close** at the end of your program is required by the IVI specifications, else unpredictable driver behavior could result. Any resources held by the driver will not be properly released if **close** is not called.

For more details on initializing the driver, see **AqMD3.chm**, section **Initializing the Driver**.

```
ViSession session;

ViStatus status;

status = AqMD3_InitWithOptions("PXI40::0::0::INSTR", VI_TRUE, VI_TRUE, "", &session);

status = AqMD3_close(session);
```

Initializing Using Options

This example shows how IVI-defined initialization options and driver-specific options can be passed to the Initialize function.

```
// If true, this will query the instrument model and fail initialization
// if the model is not supported by the driver
ViBoolean idQuery = VI_FALSE;
// If true, the instrument is reset at initialization
ViBoolean reset = VI_FALSE;
// Setup IVI-defined initialization options
ViConstString standardInitOptions =
"Cache=true, InterchangeCheck=false, QueryInstrStatus=true, RangeCheck=true, RecordCoercions=false,
Simulate=false";
status = AqMD3_InitWithOptions("PXI40::0::0::INSTR", idQuery, reset, standardInitOptions, &session);
status = AqMD3_close(vi);
```

Accessing Attributes

Accessing attributes in an IVI-C driver is accomplished via a set of IVI-defined accessor functions. There are two accessor functions for each attribute type -- one accessor for reading attribute values and another accessor for writing attribute values.

The standard attribute accessors for reading attribute values are:

- GetAttributeViInt32
- GetAttributeViInt64
- GetAttributeViReal64
- GetAttributeViBoolean
- GetAttributeViString

Correspondingly, the standard attribute accessors for writing attribute values are:

- SetAttributeViInt32
- SetAttributeViInt64
- SetAttributeViReal64
- SetAttributeViBoolean
- SetAttributeViString

Each attribute accessor takes an attribute ID that uniquely identifies the attribute to access. These attribute IDs are #define'd constants listed in the AqMD3.h header file and documented in the "Attributes by Name" section of the help file.

The following example demonstrates basic usage of attribute accessors to read and write IVI-C driver attribute values.

6.3 Migrating from MD2 2.x to MD3 3.x

Please refer to the following documents for guidelines, accessible from: **Start > Acqiris > MD3 > Documentation** or from: **C:\Program Files\Acqiris\MD3\Documentation**

- AgMD2 to AqMD3 (IVI-C) Software Migration Note.pdf
- AgMD2 IVI.COM to AqMD3 IVI.NET Software Migration Note.pdf

6.4 Initial configuration

At initialization, the driver uses the pre-defined defaults values. The following table details the initial configuration of the ADC card.

Property	Default value	Comment
Channel1	Enable	
Input filter	Bypass = false	By default, the bandwidth limiter is disable
Vertical range	2.5 (Volts)	
Vertical offset	0 (Volts)	
Vertical coupling	DC	
Trigger source	Internal1	
Trigger delay	0 (ns)	
Trigger type	Edge	
Trigger coupling	DC	
Trigger level	0 (Volts)	
Trigger slope	Positive	
Interleave	Disable	
Mode	Normal (DGT)	
Sampling rate		
Sample clock	Internal	
Reference oscillator	Internal	
Reference oscillator frequency	10 MHz	Fixed
Record size	1024	
Number of records to acquire	1	
Number of averages	1	for AVG mode only

6.5 Apply setup

The MD3 driver implements the following consistent behavior: ***No configuration change is applied immediately to the ADC card hardware.*** Specifically, this means that setting any property/attribute only changes the 'setup' in the driver, and an explicit call to **ApplySetup** is required to implement the change in the card hardware.

There are some exceptions for *'actions'*: being methods/functions that perform an action which e.g. modifies also the ADC card's state.

The following methods WILL perform an implicit **ApplySetup** before the actual action.

Actions with implicit ApplySetup

Method name	Description
SelfTest	To insure the card is actually in the desired state before doing the self test.
SelfCalibrate	To insure the card is actually in the desired state before self-calibrating.
Initiate	To apply the configured setup to the card hardware before starting the measurement.
Read	All Read methods start by performing an Initiate followed by Wait and then Fetch.
Reset	Places the card in a known state and configures card options on which the IVI specific driver depends.
ResetWithDefaults	Does the equivalent of Reset and then, (1) disables class extension capability groups, (2) sets attributes to initial values defined by class specs, and (3) configures the driver to option string settings used when Initialize was last executed.

Chapter 7

How To ... ?

7.1 How to discover the PXI Instrument?

User has to link the AqLio.lib installed by MD3 in his project.

C:\Program Files (x86)\IVI Foundation\IVI\Lib_x64\msc

The C/C++ code below can be used to discover the PXI instruments on user system and get their VISA addresses.

```
#include <stdio.h>
#include <visa.h>
int main()
{
        ViSession rm = VI_NULL;
        viOpenDefaultRM( &rm );
        ViChar search[] = "PXI?*::INSTR";
        ViFindList find = VI_NULL;
        ViUInt32 count = 0;
        ViChar rsrc[256];
        ViStatus status = viFindRsrc( rm, search, &find, &count, rsrc );
        if( status==VI_SUCCESS && count>0 )
        {
                  do
                  {
                           printf( "Found: \"%s\"\n", rsrc );
                           status = viFindNext( find, rsrc );
                  } while( status==VI_SUCCESS );
                  viClose( find );
         }
        else if( count==0 )
        {
        printf( "No PXI instrument found\n" );
        }
        viClose( rm );
        return 0;
}
```

7.2 How to calibrate the card?

Calibration principle

The card is initialized without calibration.

A calibration is mandatory before making any acquisition to guarantee measurement accuracy. The MD3 driver prevents an acquisition from being performed unless a self-calibration has first been completed.

Since a full internal calibration of an ADC card can be time consuming because of the many possible configuration states, the self-calibration is performed only for the current configuration state. A new self-calibration is required after every change of configuration of the card.

See Calibration (page 13) section.

Running fast calibration

The function **`SelfCalibrate`** should be used to perform a fast calibration.

Note

As explained above, a calibration is required after every acquisition parameter modification (e. g. full scale range, filter, sample rate, ...). The **`IAqMD3Calibration.IsRequired`** IVI.NET property or the **`AQMD3_ATTR_CALIBRATION_IS_REQUIRED`** IVI-C attribute can be used to check if a new self calibration is required .

MD3 Smart-calibration

The smart calibration implemented in MD3 drivers allows to save time by automatically keeping in memory the calibration information from any self-calibration performed since the beginning of the session. When the acquisition parameters are changed, no re-calibration of the card is necessary if a self-calibration has already been performed with the same acquisition conditions (i.e. the same set of parameters), unless the clock mode parameters are changed.

Indeed, any change in the clock mode parameters (i.e. **External clock frequency** or **Reference mode** parameters), induces a restart of the clocks which requires a new self-calibration.

Calibration optimized for a target voltage

(Not yet supported)

It is possible to optimize the calibration for a specified target voltage. In this case, the self-calibration will minimize the noise level at the channel input value equal to this specified target voltage.

When setting to true the **`Calibration.TargetVoltageEnabled`** property , the self-calibration will minimize the noise level at the channel input value equal to **`Channel.CalibrationTargetVoltage`**. Note that when set to true, the **`Channel.Offset`** value is taken into account in the self-calibration, and therefore changing the **`Channel.CalibrationTargetVoltage`** value, or changing the **`Channel.Offset`**, will require a new self-calibration.

The **`Channel.CalibrationTargetVoltage`** property allows to define, in volts, the channel input level at which the noise level will be minimized by the self-calibration. This value is only used if the property **`Calibration.TargetVoltageEnabled`** is true.

Parameter change requiring a new self calibration

The table below lists the parameters that require a new self calibration of the card when changed.

Group	Category	Parameter	Calibration required
A	Acquisition	Sampling rate	Only the 1st time, for an identical set of parameter values
	Channel parameters	Vertical range	
		Input filter Bypass (Yes/No)	
		CalibrationTargetVoltage	Only if *Calibration.TargetVoltageEnabled* is true. And each time the Offset parameter changes by more than 5% FSR.
		Offset	
B	Trigger	Trigger source	Only the 1st time
C	External reference	Reference mode (External or Internal Reference)	Each time this parameter changes

Driver interfaces and functions

The interfaces/methods/properties (functions/attributes) listed below are provided by the Acqiris MD3 driver. Please refer to **AqMD3.chm** (IVI-C) or **Acqiris.AqMD3.Fx40.chm** (IVI.NET) for detailed help.

IVI-C

Functions

AqMD3_SelfCalibrate

Attributes

AQMD3_ATTR_CALIBRATION_IS_REQUIRED

IVI.NET

Interface	Method / Property name
IAqMD3Calibration	IsRequired
	SelfCalibrate

7.3 How to access repeated capabilities?

For SA217P, the AqMD3 driver supports the following main repeated capabilities with pre-defined values detailed in following table.

Repeated capability	Available instance name
Channel	"Channel1"
TriggerSource	"Internal1", "External1", "Software", "Immediate", "SelfTrigger"
ControlIO	"ControlIO1", "ControlIO2", "ControlIO3"
MonitoringValue	*These parameters are for information only or can be used for debugging purpose. There are accessible through the MD3 SFP or the command below.Please refer to* ***AqMD3.chm*** *(IVI-C) or* ***Acqiris.AqMD3.Fx40.chm*** *(IVI.NET) .*
Stream	"StreamCh1", "MarkersCh1" "PeaksCh1"

The number of instances and their names however can be queried from the driver:

- Using the AqMD3 IVI.NET driver:
  Collection Interfaces have a **`Count`** property. Instances interface have **`Name`** property. User can iterate over all collection instances using .NET **`foreach`** loop control.

- Using the AqMD3 IVI-C driver:
  For each repeated capability **`XXX`**, there is a **`AqMD3_ATTR_XXX_COUNT`** attribute and a **`AqMD3_GetXxxName`** function (e.g. **`AqMD3_ATTR_CHANNEL_COUNT`** attribute and **`AqMD3_GetChannelName`** function).

7.4 How to generate a software trigger?

A call to function **`AqMD3_SendSoftwareTrigger`** (IVI-C) or to method **`IAqMD3Trigger.SendSoftwareTrigger`** (IVI.NET) sends a single software trigger.

`SendSoftwareTrigger()` must be called as many times as required.

Multi-record acquisitions required a trigger per record. Accumulated records require a trigger per accumulation. **`SendSoftwareTrigger()`** needs to be called for each trigger event.

7.5 How to enable or bypass the bandwidth limiter?

The ADC supports several bandwidth filter at different frequencies. User can chose to enable disable the bandwidth limiter.

Enabling the filter

The following commands allow to enable the filter.

- Using the AqMD3 IVI-C driver:

```
AqMD3_SetAttributeViBoolean(session, "Channel1", AQMD3_ATTR_INPUT_FILTER_BYPASS, VI_FALSE);
```

- Using the AqMD3 IVI.NET driver:

```
driver.Channels[L"Channel1"].Filter.Bypass = false;
```

Disabling the filter

The following commands allow to bypass the filter.

- Using the AqMD3 IVI-C driver:

```
AqMD3_SetAttributeViBoolean(session, "Channel1", AQMD3_ATTR_INPUT_FILTER_BYPASS, VI_TRUE);
```

- Using the AqMD3 IVI.NET driver:

```
driver.Channels[L"Channel1"].Filter.Bypass = true;
```

Selecting the filter frequency

User can select the desired the Max frequency by setting “**Channels[].Filter.MaxFrequency**”.

Both Channel1 and Channel2 must have the exact same filter configuration.

- Using the AqMD3 IVI-C driver:

```
ins.Channels["Channel1"].Filter.Bypass = false;
ins.Channels["Channel2"].Filter.Bypass = false;
ins.Channels["Channel1"].Filter.MaxFrequency = 20e6; //20MHz
ins.Channels["Channel2"].Filter.MaxFrequency = 20e6; //20MHz
```

- Using the AqMD3 IVI.NET driver:

```
AqMD3_SetAttributeViBoolean(vi, "Channel1", AQMD3_ATTR_INPUT_FILTER_BYPASS, VI_FALSE);
AqMD3_SetAttributeViBoolean(vi, "Channel2", AQMD3_ATTR_INPUT_FILTER_BYPASS, VI_FALSE);
AqMD3_SetAttributeViReal64(vi, "Channel1", AQMD3_ATTR_INPUT_FILTER_MAX_FREQUENCY, 20e6);
AqMD3_SetAttributeViReal64(vi, "Channel2", AQMD3_ATTR_INPUT_FILTER_MAX_FREQUENCY, 20e6);
```

7.6 How to set the external trigger?

To set the trigger source to **External1** and configure the trigger level, the following commands can be used.

IVI-C:

```
AqMD3_SetAttributeViString(session, "", AQMD3_ATTR_ACTIVE_TRIGGER_SOURCE, "External1");
AqMD3_SetAttributeViReal64(session, "External1", AQMD3_ATTR_TRIGGER_LEVEL, level);
```

IVI.NET:

```
spDriver->Trigger->ActiveSource = "External";
IAqMD3TriggerSourcePtr spTrigSrc = spDriver->Trigger->Sources->Item[L"External1"];
spTrigSrc->Level = level; //in volts
```

The different trigger sources are listed in the section How to access repeated capabilities? (page 46).

7.7 How to perform binary decimation? (depending on firmware)

The binary decimation is not supported for all combinations of firmware, channel configuration and sampling rate.

Please refer to section for more information.

- Using the AqMD3 IVI-C driver:

To use the binary decimation and set the sample rate to a lower value use the **AQMD3_ATTR_SAMPLE_RATE** attribute.

```
sampleRate = 200e6;
status=AqMD3_SetAttributeViReal64(session,"", AQMD3_ATTR_SAMPLE_RATE,sampleRate);
```

- Using the AqMD3 IVI.NET driver:

To use the binary decimation and set the sample rate to a lower value use the **SampleRate** property.

```
sampleRate = 200e6;
driver.Acquisition.SampleRate = sampleRate;
```

Chapter 8

Software utilities

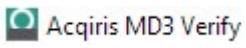
This section describes supplied programs which may be used to configure various aspects of your cards.

8.1 ADC card Verification Utility (AqMD3Verify)

The **AqMD3Verify** utility verifies the card status:

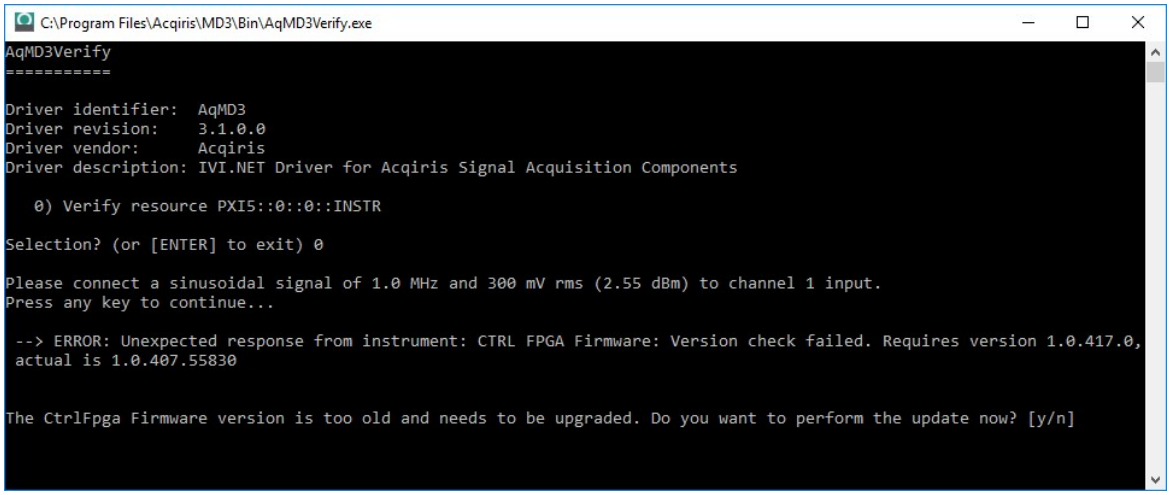
- This utility proposes the user to connect a precise simple reference signal, and then, it compares the ADC card acquisition of this reference signal with reference signal expected values.
- This utility checks the version of control FPGA firmware already loaded. If necessary, it proposes to update the firmware using the **Firmware Update Utility**.

You can launch **AqMD3Verify** from the start menu.



AqMD3Verify requests the user to connect a reference signal and then to press any key to continue (as shown in the window below).

AqMD3Verify checks the version of control FPGA firmware already loaded, and if necessary, proposes the user to update the firmware, automatically using the **Firmware Update Utility** (As shown in the window below : Accept the FPGA update answering "y").



When the version of control FPGA firmware is updated and successful, please power off your computer, restart it again for the update to take effect, and process AqMD3Verify tool as described in this section.

Chapter 9

FAQ

9.1 Q. What is coherent sampling?

A. Coherent Sampling refers to the relationship between the input frequency F_{in}, sampling frequency F_s, number of cycles in the sampled set and the number of samples. With coherent sampling one is assured that the signal power in an FFT is contained within one FFT bin (assuming a single input tone).

The condition for coherent sampling is given by:

$F_{in}/F_s = N_{cycles}/N_{samples}$

For example if we have $N_{samples} = 2^{11}$, and F_s = 100e6, and we expect and input frequency close to $F_s/2$, let's say F_{in} = 44 MHz, then N_{cycles} = 901.12 which is close to an integer. We could therefore round down to N_{cycles} = 901 and we would get F_{in} = 43.994140625 MHz, which is an input frequency that satisfies coherent sampling.

The integer number should be chosen carefully. We have three possible types of integers, even, odd, and prime. Even is not a good idea since we would hit the same code every $M_{samples}$, where M can be much less than N. Odd is a better idea since it takes longer to hit the same code. According to some sources a prime number of cycles is the best (with the exception of the prime 2) because it takes a long time before the same code repeats.

9.2 Q. How to manage the internal temperature?

A. The operating temperature of the SA217P as specified in the SA217P datasheet, is the workstation internal ambient temperature at intake of the ADC card's fan.

The effective temperature limit is fixed by the maximum internal DPU temperature which should stay below 100°C to guarantee FPGA proper operating.

The ADC card's fan speed is automatically controlled with the internal temperature, it maintains the FPGA core temperature below 100°C.

This DPU FPGA core temperature (or junction temperature Tj) can be monitored from the MD3 SFP from **Interfaces > Temperature**, or using the function **BoardTemperature (IVI.NET) / AqMD3_ QueryBoardTemperature (IVI-C)**.

Note that the channel temperatures (given by **ChannelTemperature (IVI.NET) / AqMD3_ ChannelTemperature (IVI-C)**) can reach 100°C in standard operating mode, which is within the components operating conditions. This parameter is provided for information only.

9.3 Q. What happens if the host processor goes in hibernation mode?

A. Hibernation while the ADC card is in operation is not supported. Recommendation is to close the ADC card before the host computer is allowed to go into hibernation.

There are many situations where equipment should go into hibernation mode. If the system allows hibernation for saving power, then the ADC card should also be powered down.

If user system is capable of managing hibernation, then when the application code decides/detects that the system should go into hibernation, it can close the ADC card, and re-initialize it when it wakes up from hibernation.

After being powered off, the ADC card must reload of the FPGA (several seconds) upon power on, and a self-calibration is required.

Chapter 10

General information

10.1 Safety notes

The following safety precautions should be observed before using this product and any associated instrumentation.

This product is intended for use by qualified personnel who recognize shock hazards and are familiar with the safety precautions required to avoid possible injury. Read and follow all installation, operation, and maintenance information carefully before using the product.

Warning If this product is not used as specified, the protection provided by the equipment could be impaired. This product must be used in a normal condition (in which all means for protection are intact) only.

The types of product users are:

- **Responsible body** is the individual or group responsible for the use and maintenance of equipment, for ensuring that the equipment is operated within its specifications and operating limits, and for ensuring operators are adequately trained.
- **Operators** use the product for its intended function. They must be trained in electrical safety procedures and proper use of the card. They must be protected from electric shock and contact with hazardous live circuits.
- **Service personnel** are trained to work on live circuits, perform safe installations, and repair products. Only properly trained service personnel may perform installation and service procedures.

Operator is responsible to maintain safe operating conditions. To ensure safe operating conditions, cards should not be operated beyond the full temperature range specified in the datasheet. Exceeding safe operating conditions can result in shorter lifespans, improper card performance and user safety issues. When the cards are in use and operation within the specified full temperature range is not maintained, card surface temperatures may exceed safe handling conditions which can cause discomfort or burns if touched. In the event of a card exceeding the full temperature range, always allow the card to cool before touching or removing cards from host computer or chassis.

Exercise extreme caution when a shock hazard is present. Lethal voltage may be present on cable connector jacks or test fixtures. The American National Standards Institute (ANSI) states that a shock hazard exists when voltage levels greater than 30 V RMS, 42.4 V peak, or 60 V DC are present. A good safety practice is to expect that hazardous voltage is present in any unknown circuit before measuring.

Operators of this product must be protected from electric shock at all times. The responsible body must ensure that operators are prevented access and/or insulated from every connection point. In some cases, connections must be exposed to potential human contact. Product operators in these circumstances must be trained to protect themselves from the risk of electric shock. If the circuit is capable of operating at or above 1000 V, no conductive part of the circuit may be exposed.

Do not connect cards directly to unlimited power circuits. They are intended to be used with impedance-limited sources. NEVER connect cards directly to AC mains. When connecting sources to cards, install protective devices to limit fault current and voltage to the card.

Before operating a card, ensure that the line cord is connected to a properly-grounded power receptacle. Inspect the connecting cables, test leads, and jumpers for possible wear, cracks, or breaks before each use.

When installing equipment where access to the main power cord is restricted, such as rack mounting, a separate main input power disconnect device must be provided in close proximity to the equipment and within easy reach of the operator.

For maximum safety, do not touch the product, test cables, or any other instruments while power is applied to the circuit under test. ALWAYS remove power from the entire test system and discharge any capacitors before: connecting or disconnecting cables or jumpers, installing or removing ADC cards, or making internal changes, such as installing or removing jumpers.

Do not touch any object that could provide a current path to the common side of the circuit under test or power line (earth) ground. Always make measurements with dry hands while standing on a dry, insulated surface capable of withstanding the voltage being measured.

The card and accessories must be used in accordance with its specifications and operating instructions, or the safety of the equipment may be impaired.

Do not exceed the maximum signal levels of the cards and accessories, as defined in the specifications and operating information, and as shown on the card or test fixture panels, or ADC card.

If you are using a test fixture, keep the lid closed while power is applied to the device under test. Safe operation requires the use of a lid interlock.

Cards and accessories shall not be connected to humans.

Before performing any maintenance, disconnect the line cord and all test cables.

Any part or component replacement must be done by Acqiris.

Warning No operator serviceable parts inside. Refer servicing to qualified personnel. To prevent electrical shock do not remove covers.

10.2 Cleaning precautions

Warning To prevent electrical shock, disconnect the card from mains before cleaning. Use a dry cloth or one slightly dampened with water to clean the external case parts. Do not attempt to clean internally. To clean the connectors, use alcohol in a well-ventilated area. Allow all residual alcohol moisture to evaporate, and the fumes to dissipate prior to energizing the card.

10.3 Product markings

The CE mark is a registered trademark of the European Community.

Australian Communication and Media Authority mark to indicate regulatory compliance as a registered supplier.

This symbol indicates product compliance with the Canadian Interference-Causing Equipment Standard (ICES-001). It also identifies the product is an Industrial Scientific and Medical Group 1 Class A product (CISPR 11, Clause 4).

The FCC certification mark certifies that the electromagnetic interference from the device is under limits approved by the Federal Communications Commission.

This symbol on an card means caution, risk of danger. You should refer to the operating instructions located in the user documentation in all cases where the symbol is marked on the card.

This product complies with the WEEE Directive marketing requirement. The affixed product label (above) indicates that you must not discard this electrical/electronic product in domestic household waste. **Product Category**: With reference to the equipment types in the WEEE directive Annex 1, this product is classified as “Monitoring and Control instrumentation” product. To return unwanted products, contact your local Acqiris office.

This symbol indicates the time period during which no hazardous or toxic substance elements are expected to leak or deteriorate during normal use. Forty years is the expected useful life of the product.

This symbol indicates the card is sensitive to electrostatic discharge (ESD). ESD can damage the highly sensitive components in your card. ESD damage is most likely to occur as the module is being installed or when cables are connected or disconnected. Protect the circuits from ESD damage by wearing a grounding strap that provides a low resistance path to ground. Alternatively, ground yourself to discharge any built-up static charge by touching the outer shell of any grounded instrument chassis before touching the port connectors.

This symbol denotes a hot surface. The side cover of the module will be hot after use and should be allowed to cool for several minutes.

10.4 Electrical & environmental specifications

For full specifications, please refer to the SA217P datasheet.

10.5 Related documentation

All documentation relating to your ADC card may be found from https://extranet.acqiris.com/.

If you have run the Acqiris MD3 software installer on your PC, the related product documentation has been installed to your hard drive.

Document	Description
Startup Guide	Includes procedures to help you to unpack, inspect, install (software and hardware), perform card connections, verify operation, and troubleshoot your product.
User Manual	Provides in- depth information and reference material specific to your ADC card product
Data Sheet	In addition to a detailed product introduction, the data sheet supplies full product specifications.
Soft Front Panel (help system)	Provides information on the use of the Soft Front Panel.
IVI Driver reference (help system)	Provides detailed documentation of the IVI.NET and IVI-C driver API functions, as well as information to help you get started with using the IVI drivers in your application development environment.

Monday, March 11, 2024,
Switzerland

www.acqiris.com