Ultra High Speed, High SNR/DR Gen 3 PCIe Data Acquisition Boards with 8GB RAM and Kintex Ultrascale FPGA

Product Specification – September 18, 2019

AD12-2000 Series: 12-bit 2/4 GSPS 2/1-Channel

- AD12-2000x2-8GB 2-Chan 2GSPS / 1-Chan 4GSPS A/D, 8GB RAM
- Includes Zero Dead-Time Hardware Averaging and analog/digital triggering

AD14-400 Series: 14-bit 400MSPS 2-Channel

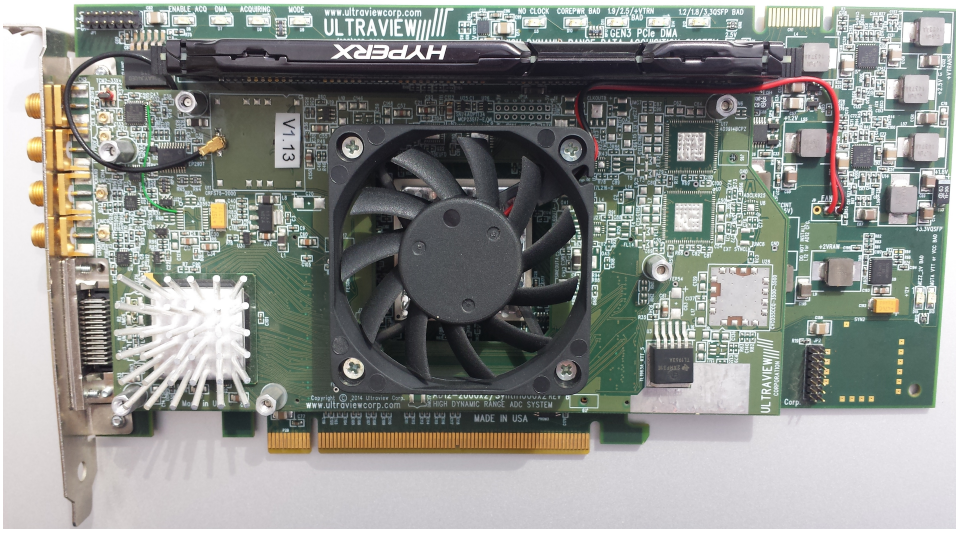
- AD14-400x2-8GB 2 Chan 400MSPS A/D, 8GB RAM
  Includes Zero Dead-Time Hardware Averaging and analog/digital triggering

AD14-500 Series: 14-bit 500MSPS 2-Channel

- AD14-500x2-8GB 2 Chan 500MSPS A/D, 8GB RAM
  Includes Zero Dead-Time Hardware Averaging and analog/digital triggering

AD16-250 Series: 16-bit 250MSPS 4/2-Channel

- AD16-250x2-8GB; AD16-250x4-8GB 2 or 4-Chan 250MSPS A/D, 8 GB RAM
  Includes Zero Dead-Time Hardware Averaging and analog/digital triggering



Ultraview Corporation
808 Gilman Street, Berkeley, CA 94710
PH:(925) 253-2960
Fax (925) 253-4894
www.ultraviewcorp.com

TABLE OF CONTENTS

1. Warranty

Ultraview Corporation hardware, software and firmware products are warranted against defects in materials and workmanship for a period of two (2) years from the date of shipment of the product. During the warranty period Ultraview Corporation shall at its option, either repair or replace hardware, software or firmware products which prove to be defective. Ultraview products are only supported with Ultraview provided firmware and software, any modifications made by customers are not supported and are not covered under warranty. This limited warranty does not cover damage caused by misuse or abuse by customer, and specifically excludes damage caused by dropping the unit or by the application of excessive voltages to the inputs and/or outputs of data acquisition boards. Due to the complex nature of computer systems, Ultraview boards operation should be verified in the desired host computer system prior to purchasing multiple units of host system. For example in some systems the reset time is too short to allow certain Ultraview FPGA-based boards to fully configure before being accessed by the system, therefore requiring a warm boot before operation is possible in these systems.

While Ultraview Corporation hardware, software and firmware products are designed to function in a reliable manner, Ultraview Corporation does not warrant that the operation of the hardware, software or firmware will be uninterrupted or error free. Ultraview products are not intended for use as critical components in life support systems, aircraft, military systems or other systems whose failure to perform can reasonably be expected to cause significant injury to humans. Ultraview expressly disclaims liability for loss of profits and other consequential damages caused by the failure of any product, and recommends that customer purchase spare units for applications in which the failure of any product would cause interruption of work or loss of profits, such as industrial, shipboard or military equipment. In no way will Ultraview Corporation's liability exceed the amount paid by the customer for the product.

This limited warranty is in lieu of all other warranties expressed or implied. The warranties provided herein are buyer's sole remedies. In no event shall Ultraview Corporation be liable for direct, special, indirect, incidental or consequential damages suffered or incurred as a result of the use of, or inability to use these products. This limitation of liability remains in force even if Ultraview Corporation is informed of the possibility of such damages.

Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation and exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

WARNING! To avoid overheating, all Ultraview boards must be installed in a well-cooled workstation or server chassis, or alternatively in an industrial chassis PC. Installation in a PC or workstation without fans at the front end of the card cage may cause the board to overheat, and resulting damage is not covered by warranty.

2. Series Descriptions

Ultraview's PCIe data acquisition boards are complete high-speed data acquisition systems on a single full-size PCI-express (PCIe) card. Designed for low jitter operation in military, scientific, medical and industrial applications these boards function in PCIe systems having at least one free Gen3 x16 PCIe slot. Drivers and ready-to-use user programs for Linux (Centos 7.0/6.8/6.6/6.4/6.3/6.2 64-bit) and 64-bit Windows 7 and 10 are provided. Windows 7/10 64-bit drivers are fully signed**.**

The digital section of the these boards includes a Xilinx Kintex 7™ FPGA, which queues up the A/D data in its FIFO registers, and outputs bursts of 128-bit wide LVDS data vectors at up to 500 MWPS (8GB/sec) that are transferred into one 8GB DDR4 DIMM module. Between these forward bursts of A/D data, the FPGA can burst read data to the PCIe bus interface, allowing for uninterrupted high speed data acquisition. These devices require x16 PCIe slots but are only wired for x8 lane operation. Sustained transfer rate to the host is up to 7.0 GB/s, and is determined by the speed of this 8 lane Gen 3 PCIe interface, which is dependent on the motherboard, the operating system and applications running. All 14-bit and 16-bit boards in this line have a selective recording input (Trigger Input) that allows the user to start and stop acquisition in response to an external TTL input. Additionally all 12, 14 and 16-bit boards have zero-dead-time hardware averaging and TTL and analog waveform-based triggering.

The FPGA on all boards is an 1156 pin Xilinx Kintex 7™. Most of the 1156 pin Virtex 7™ FPGA's may be specified by the customer at the time the board is ordered, these include the XCKU035, XCKU040, and XCKU060. **The default FPGA is an XCKU040™** , whose registers and BRAMS are approximately 30%/50% for the AD12-2000, 30%/70% for the AD14-400/AD14-500, and TBD %/TBD% for the AD16-250x2/AD16-250x4 occupied when the board is shipped. The larger XCKU060 FPGA can be substituted and uses significantly less of these resources or the smaller XCKU035 FPGA can be substituted for a small cost savings. With the purchase of any Ultraview data acquisition system, and under the protection of a signed Non-Disclosure Agreement, Ultraview Corporation will provide the VHDL firmware source code to allow its OEM users the option of loading their own firmware into the on-board FPGA. Please request a copy of Ultraview Corporation's NDA for specifics by e-mailing sales@ultraviewcorp.com.

All boards are full-size PCIe boards (4.2” x 8.625”). **Due to the excess height of the memory modules on the board each board occupies the space of two slots.**

To avoid overheating, all boards must be installed either in a well-cooled workstation or a server chassis. Installation in a standard PC chassis is feasible as long as a minimum of 100 linear feet per minute of airflow is provided to cool the board. Further, the filler bracket for the slot in front of the Ultraview board (the slot into which the Ultraview's DIMMs protrude) must be removed to allow additional airflow to the board.

2.1 AD12-2000x2 12-Bit Dual Channel 2GSPS / 1 Channel 4GSPS A/D w/Hardware Averaging and Digital and Analog Waveform Triggering

Models in the AD12-2000 series contain a dual 2GSPS 12-bit A/D converter, 8GB of on-board DDR4 DRAM memory and the ability to transfer data directly into the computer system's memory up to approximately 7 GB/s. A/D sampling may either be controlled by an external clock input between 300 MHz and 2GHz (the A/D sampling rate is equal to the clock frequency) or the on-board programmable internal clock up to 2000MHz. Multiple boards may be configured to acquire either concurrently, for more simultaneous acquisition channels, or sequentially, for longer record length. Two TTL inputs are also sampled concurrently with each analog input and stored along with the A/D data. A 26-pin connector and TTL Input Panel will be provided which the user can drive from an SMA. TTL triggering is available on GC_P and specifying TTL trigger. Selective recording is available on GC_P by specifying no trigger. See Appendix for more details. Decimation up to x8 is available.

ECL triggering is available for edge-triggering the start of data acquisition using the Trigger/Sync SMA jack. A suitable ECL trigger signal is a positive-going edge with a rise time less than 2ns and a swing of approximately 1V (can be 0.7V to 1.2V). Selective recording / acquire disable can be used to record data on either the high or low logic level of a user specified signal. A proper selective recording signal is a TTL signal with levels describe in Section 5.

The AD12-2000x2-8GB-K7US may also be configured to acquire a single 12-bit channel at any clock frequency between 300MSPS and 4 GSPS. In single channel mode, the sampling rate will be twice the input clock rate.

Hardware averaging, with near-zero dead-time, is implemented in the board's Xilinx™ FPGA. The AD12-2000x2-8GB-K7US can average repetitive signal strings up to 1 million times with averaged record lengths to 131072-samples (single channel) or 65536-samples (dual channel) uninterrupted. The precise repetitive summing of each new string of samples onto a running 32-bit average can be triggered by either of the two software-selectable triggering mechanisms:

- A TTL input, with selectable –/+ slope, causes waveforms to be acquired or added to a running average.
- A software slider-adjustable level on the incoming signal waveform on any of the 2 channels, with + or – slope, enabling scope-like triggering, with pre-trigger, on a given place on a repeating waveform.

2.2 AD14-400x2 Dual Channel 14-bit 400MSPS A/D w/Hardware Averaging and Digital and Analog Waveform Triggering

Models in the AD14-400 series contain two 400 MSPS 14-bit A/D converters, 8GB of on-board DDR4 DRAM memory and the ability to transfer data directly into the computer system's memory up to approximately 7 GB/s. A/D sampling may either be controlled by an external clock input between 20 MHz and 400 MHz (the A/D sampling rate is equal to the clock frequency) or the 400MHz on-board internal clock (available via an SMA jack). Multiple boards may be configured to acquire either concurrently, for more simultaneous acquisition channels, or sequentially, for longer record length. Four TTL inputs are also sampled concurrently with the analog input and stored along with the A/D data. Both channels on the AD14-400x2-8GB sample concurrently at a rate equal to the input clock frequency. Selective recording / acquire disable can be used to record data on either the high or low of a user specified signal. A proper selective recording signal is a TTL signal with levels describe in Section 5. Specifications - AD14-400. Selective recording is defaulted as an active LOW (board has an internal pull-down resistor, leave unconnected if unused) and can be specified to an active high. Fiducial marks can be added to easily mark the samples recorded during each high or low phase.

Hardware averaging, with near-zero dead-time, is implemented in the board's Xilinx™ FPGA, the upgraded AD14-400-8GB-K7US can average repetitive signal strings up to 1 million times with record lengths to 131072-samples uninterrupted. The precise repetitive summing of each new string of samples onto a running 32-bit average can be triggered by one of two software-selectable triggering mechanisms:

- A TTL input, with selectable –/+ slope, causes waveforms to be acquired or added to a running average.

- A software slider-adjustable level on the incoming signal waveform on any of the 2 channels, with + or – slope, enabling scope-like triggering, with pre-trigger, on a given place on a repeating waveform.

2.3 AD14-500x2 Dual Channel 14-bit 500MSPS A/D w/Hardware Averaging and Digital and Analog Waveform Triggering

Models in the AD14-500 are rated up to 500MHZ external clock and are otherwise identical to the AD14-400x2 series.

2.4 AD16-250 Dual or Quad channel 16-bit 250MSPS A/D with Hardware Averaging and Digital and Analog Waveform Triggering

Models in the AD16-250 series contain two or four 250 MSPS 16-bit A/D converters, 8GB of on-board DDR4 DRAM memory and the ability to transfer data directly into the computer system's memory at up to approximately 7 GB/s. A/D sampling may either be controlled by an external clock input between 40 MHz and 250 MHz (the A/D sampling rate is equal to the supplied external clock frequency) or the 250MHz on-board internal clock. Multiple boards may be configured to acquire either concurrently, for more simultaneous acquisition channels, or sequentially, for longer record length. All channels on the AD16-250x2-8GB sample concurrently at a rate equal to the input clock frequency (250MSPS when internal clock is specified, or 40-250MSPS when a 40MHz to 250 MHz external clock is supplied). Selective recording / acquire disable can be used to record data on either the high or low of a user specified signal. A proper selective recording signal is a TTL signal 0V to 3V. Selective recording is defaulted as an active low, but can alternatively be specified to an active high.

Hardware averaging, with near-zero dead-time, is implemented in the board's Xilinx™ FPGA. The AD16-250x2-8GB-K7US can average repetitive signal strings up to 1 million times with averaged record lengths to 131072-samples uninterrupted. The precise repetitive summing of each new string of samples onto a running 32-bit average can be triggered by any one of three software-selectable triggering mechanisms:

- A TTL input, with selectable –/+ slope, causes waveforms to be acquired or added to a running average.
- A software slider-adjustable level on the incoming signal waveform on any of the 4 channels, with + or – slope, enabling scope-like triggering, with pre-trigger, on a given place on a repeating waveform.
- Heterodyning trigger input - Triggering will occur on the difference frequency between this input and the sampling clock frequency. This is useful for time-of-flight imaging systems, RADAR and pulsed spectroscopy systems, in which transmit or stimulus waveforms are repeated M-times/second and the A/D samples data at a rate of N samples per sec. The result is that the AD16-250 will automatically acquire and/or average complete waveforms that repeat M minus N times per second.

On AD16-250x4-8GB-K7US models, the averager length is 131072 averaged samples in single channel mode, 65536 samples on each channel in dual channel mode, or 32768 averaged samples on each sample in 4-channel mode.

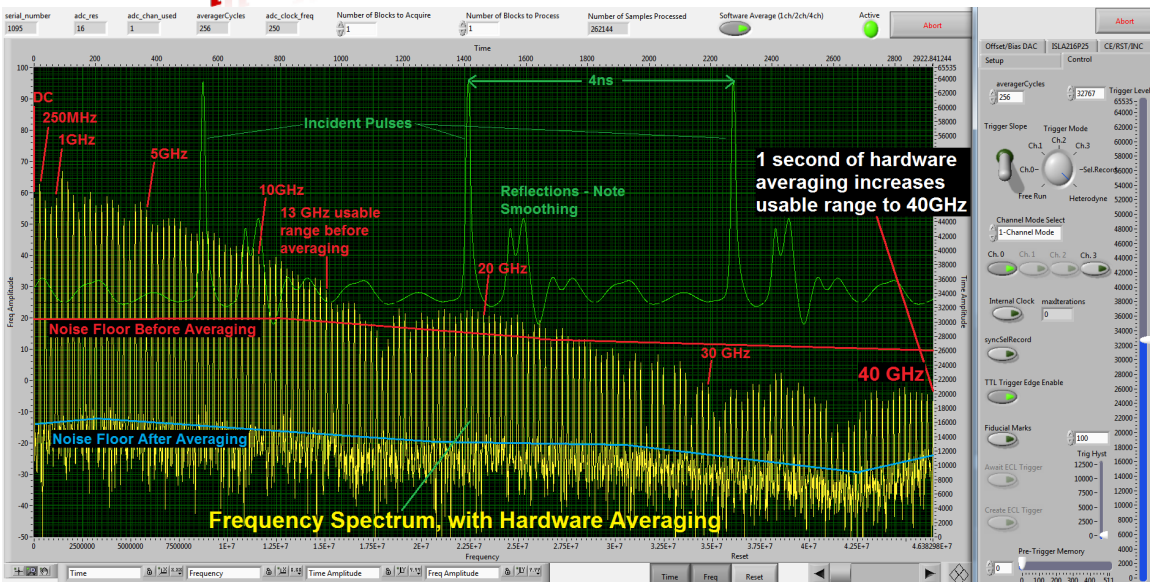


Figure 2.1. Ultradyne16-250x2 fed by a Libove-Chacko 40GHz microwave sampler/pulser TDR spectrometer, displaying the reflection spectrum of a microwave bowtie antenna excited by a repeating 15 picosecond 250 million pulse/second train. Entire DC-40GHz spectrum (yellow trace) is captured with 120dB dynamic range, using under 1 second of hardware averaging, at nearly 100% collection efficiency**.** Averaged waveforms, despite containing antenna pickup of ambient RFI are so stable that they appear static. Heterodyne trigger **automatically triggers the board at the difference frequency (250.290 MPPS pulse frequency sampled at 250MSPS) to coherently average 290,000 waveforms per second**.

3. Specifications - AD12-2000x2

A/D Converter Resolution:	12 Bits
Number of Channels:	2 at up to 2GSPS or 1 at up to 4GSPS
Signal-to-Noise Ratio – At 4GSPS:	>85 dB up to 115dB w/ 65536x Hardware Averaging
Analog Input Range	
Single Chan Mode, at At 2.4GSPS:	+/-317mV (634mV p-p = 0 dBm) nom (rev C)
Single Chan Mode, at 4GSPS:	+/-292mV (584mVpp = -1.6dBm) nom. (rev C) **Do not exceed +/-1V**
Analog Input Impedance:	50 ohms nominal || 2pF
Analog Input Bandwidth (-3dB BW)	
Single Chan Mode, at 2.4-4.0GSPS	DC to 1.8 GHz
Dual Chan Mode, at 1.2-2GSPS	DC to >2GHz
Sampling Rate into on-board RAM:	
Maximum:	2 GSPS (4GSPS into single channel)
Minimum:	200 MSPS (dual channel mode) 400 MSPS (single channel mode) (rates <400MSPS / 200MSPS use decimation) Decimation: x1 (none), x2, x4,x8
Clock Input (AC Coupled)	
Frequency	200MHz – 2GHz
Input Impedance:	50 ohms nominal in series with 0.01uF
AC Voltage Minimum:	0.7V Peak-to-Peak
AC Voltage Maximum:	3.0V Peak-to-Peak **DC voltage must not exceed +/- 5V**
Optional Trigger ECL Input (AC coupled) : (Trigger/Sync or Sync A/D SMA jack)	Positive-going edge - rise time must be <2ns 0.5V p-p min, 1.2V p-p max.
Optional Selective Recording Input:	0V - 3.3V **Do not exceed 3.3V**
Trigger Modes (software selectable):	1) Start immediately upon software command 2) Wait for externally supplied ECL trigger 3) Wait for externally supplied TTL trigger 4) Analog Waveform Triggering on either channel with selectable slope and level.
DMA Transfer Rate:	6-7 GB/s (host system dependent)
Sample rate for continuous stream to host:	>3GSPS (12bit 1 ch mode, >1.5GSPS 12-bit 2ch)
Operating Temperature Range:	0 to +50 Degrees Celsius
Storage Temperature Range:	-25 to +85 Degrees Celsius
Power Requirements (board occupies 2 slots):	+3.3V +/-5% at 3.3A Max (2.6 A typical) +12V +/-5% at 3.0A Max (2.0 A typical)

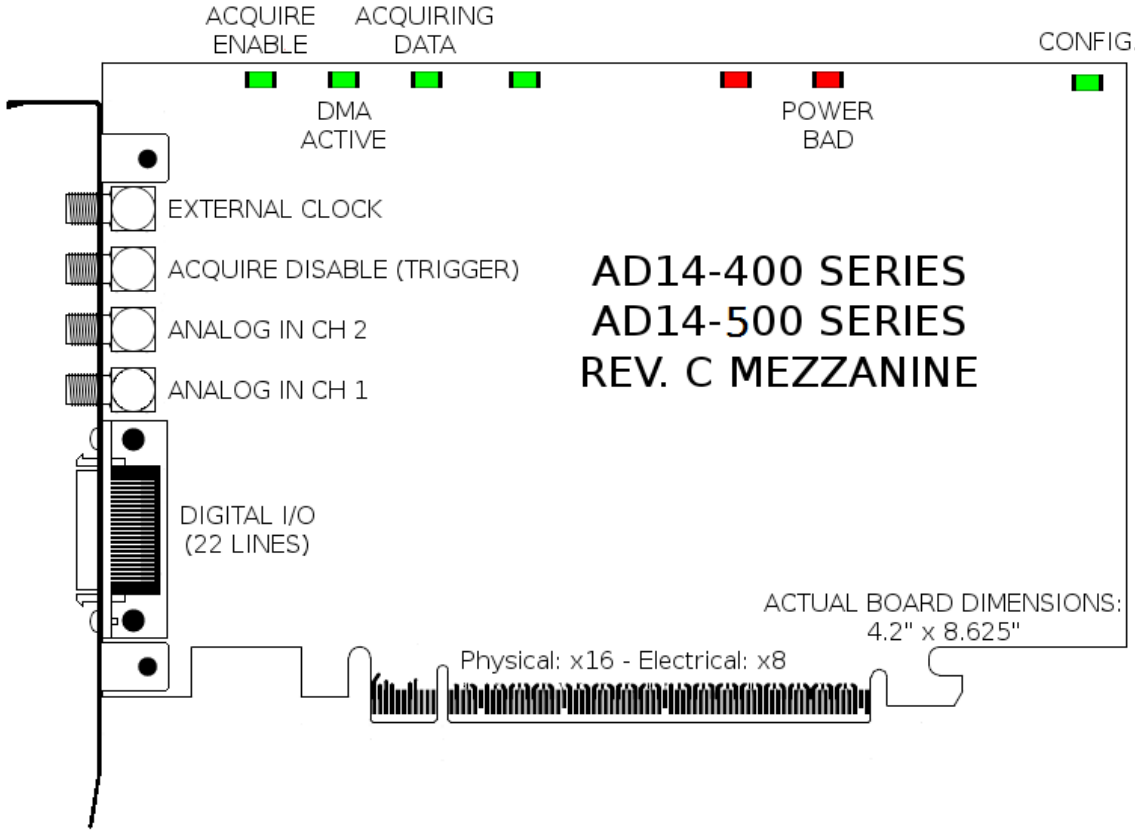
4. Specifications – AD14-400x2 / AD14-500x2

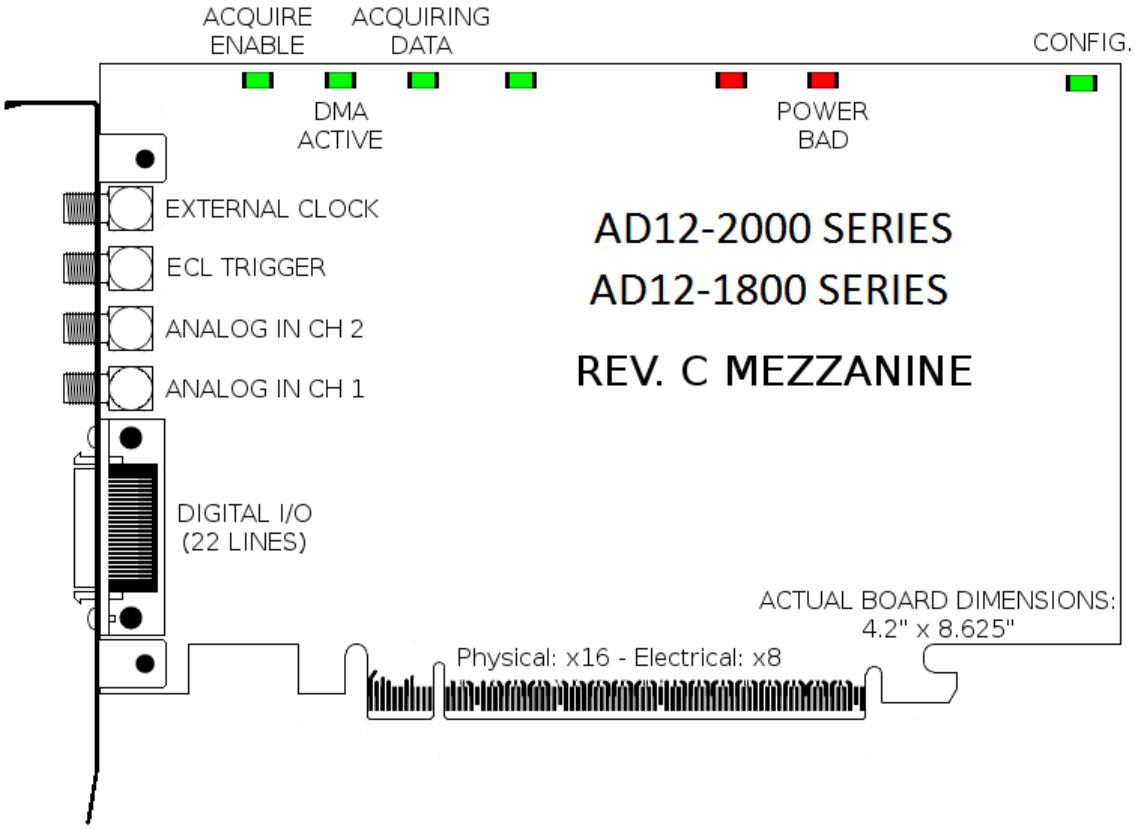
A/D Converter Resolution:	14 Bits
Number channels:	2, Simultaneously sampled
Signal-to-Noise Ratio	70 dB (70MHz Input, 400 MSPS, 1MB FFT)
SFDR	74 dB (70MHz Input, 400 MSPS, 1MB FFT)
Analog Input Range:	+/-375mV (750mVpp) **Do not exceed +/-1V**
Analog Input Impedance:	50 ohms || 2pF
Analog Input Bandwidth	DC to 1.4 GHz (-3dB BW)
Sampling Rate into on-board RAM:	
Maximum:	400 MSPS / 500 MSPS
Minimum:	20 MSPS
Clock Input (AC Coupled)	
Frequency	20MHz – 400MHz / 500MHz
Input Impedance:	50 ohms in series with 0.01uF
AC Voltage Minimum:	0.7V Peak-to-Peak
AC Voltage Maximum:	4.0V Peak-to-Peak **DC voltage must not exceed +/- 5V**
Optional Trigger (Selective Recording) Input:	0V - 3.3V **Do not exceed 3.3V**
Trigger Modes (software selectable):	1) Start immediately upon software command 2) Wait for externally supplied TTL trigger 3) Analog Waveform Triggering on either channel with selectable slope and level.
DMA Transfer Rate:	6-7 GB/s (host system dependent)
Operating Temperature Range:	0 to +50 Degrees Celsius
Storage Temperature Range:	-25 to +85 Degrees Celsius
Power Requirements (board occupies 2 slots):	+3.3V +/-5% at (2.6A Typ.) (3.0A Max) +12V +/-5% at (1.2A Typ.) (1.5A Max) .

5. Specifications – AD16-250x2, AD16-250x4

A/D Converter Resolution:	16 Bits
Number of Channels:	2, Concurrently sampled (Model: AD16-250x2) 4, Concurrently sampled (Model: AD16-250x4)
Signal-to-Noise Ratio	Approx. 88 dB
Dynamic Range:	Approx. 88 dB (w/o Hardware Averaging) Up to 160 dB (w/ 2sec. of Hardware Averaging)
SFDR	Approx. 84 dB @30MHz
Analog Input Range:	+/-300mV (600mVpp) **Do not exceed +/-1.5V**
Analog Input Impedance:	50 ohms || 2pF
Analog Input Bandwidth	DC to 200 MHz (-3dB BW)
Sampling rate into on-board RAM:	
Maximum:	250 MSPS
Minimum:	40 MSPS
Clock Input (AC Coupled)	
Frequency	40MHz – 250MHz
Input Impedance:	50 ohms in series with 0.01uF
AC Voltage Minimum:	0.5V Peak-to-Peak
AC Voltage Maximum:	1.2V Peak-to-Peak **DC voltage must not exceed +/- 5V**
Optional Trigger (Selective Recording) Input:	0V – 2.5V **Do not exceed 3.3V**
Trigger Modes (software selectable):	1) Start immediately upon software command 2) Wait for externally supplied TTL trigger 3) Analog Waveform Triggering on any channel with selectable slope and level. 4) Heterodyne mode – trigger on difference frequency between sampling clock and external stimulus clock inputted at H jack.
Input connectors:	7 MMCX female jacks. Board is shipped with seven 4-foot MMCX-male to SMA-male cables.
DMA Transfer Rate:	6-7 GB/s (host system dependent)
Operating Temperature Range:	0 to +50 Degrees Celsius
Storage Temperature Range:	-25 to +85 Degrees Celsius
Power Requirements (board occupies 2 slots):	+3.3V +/-5% at 3.3A Max (2.6 A typical) +12V +/-5% at 2.0A Max for 4GB

6. AD12-2000x2 / AD14-400x2 / AD14-500x2 Cable I/O and LEDs





6.1 AD12-2000x2 / AD14-400x2 / AD14-500x2 Cable Description

External Clock A single frequency clock must be continuously supplied before any user programs are run. This external clock is useful if acquisition is to be synchronized to an external source. For applications in which an external clock is not desired, the board's internal clock can be used instead, the software must select internal clock mode, and the external clock input must be left unconnected.

Analog In 2, Analog In 1 The SMA analog input connectors accept a single-ended voltage in the range specified in the specifications section, under no circumstances should the signal supplied to the analog input exceed the specified range as damage may occur that is not covered by the warranty. Both Analog In 2 and Analog In 1 are concurrently sampled, and their precise time-alignment make them well suited for I/Q sampling.

Acquire Disable (Trigger) (AD14-400x2 / AD14-500x2) This is the trigger or selective recording TTL input for the AD14-400x2 and AD14-500x2. The board will store data to the on-board RAM only when the acquire disable input is LOW (board has an internal pull-down resistor, leave unconnected if unused). The timing in this mode is not precise to a single sample and can vary from 0 to 3 samples if the input is not synchronized with the A/D clock.

GC_P on TTL Board (section 16) (AD12-2000x2) This is the selective recording TTL input for the AD12-2000x2. The board will store data to the on-board RAM only when the acquire disable input is LOW (board has an internal pull-down resistor, leave unconnected if unused). The timing in this mode is not precise to a single sample and can vary from 0 to 3 samples if the input is not synchronized with the A/D clock. Selecting TTL edge trigger will use this input as a trigger.

TTL Input 0-3 These TTL inputs are sampled and stored along with the analog inputs.

6.2 AD12-2000x2 / AD14-400x2 / AD14-500x2 LED Description

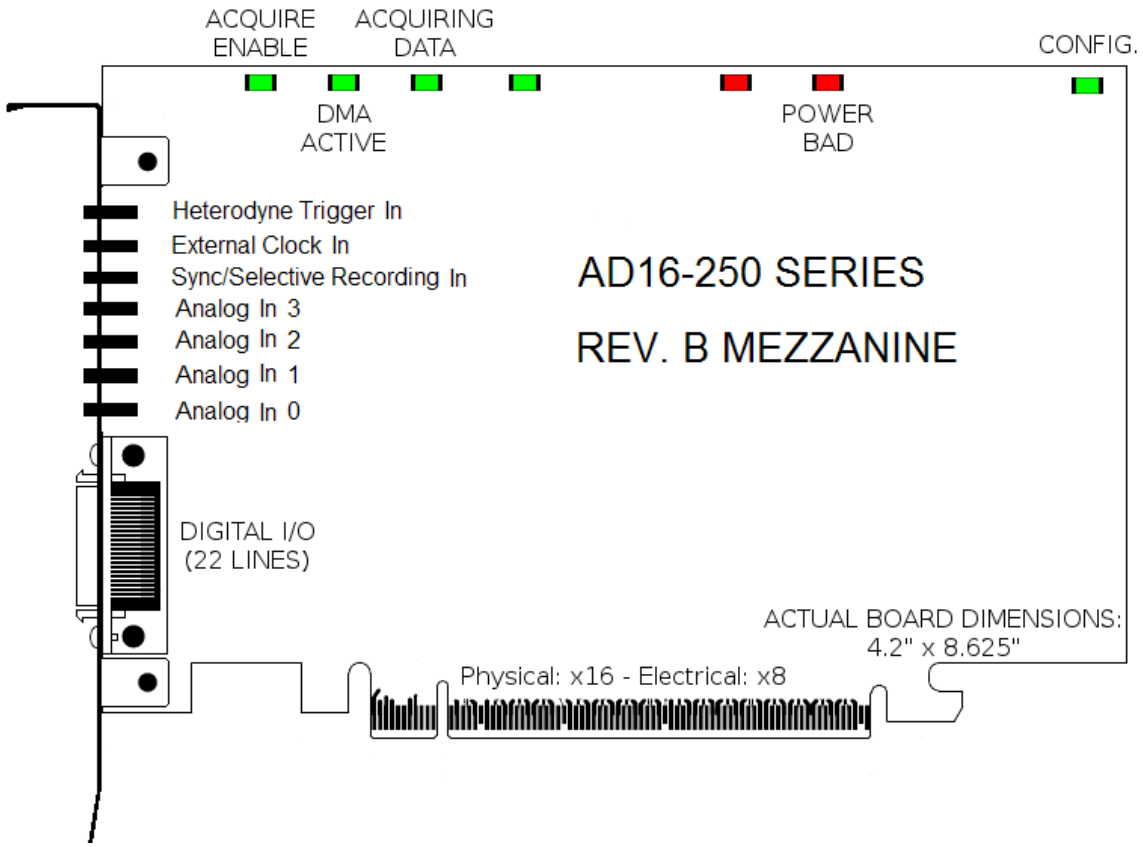
Clock bad Indicates that the board sees the external oscillator for DRAM when this LED is flashing.

DMA Indicates that the board sees an input clock when this LED is flashing, the rate of flashing is proportional to the clock source frequency.

Power Bad Indicates the DC power to the DRAM and ADCs is out of spec. This may occur if a fault develops in the board's power supply, or if the slot in which it is installed is incapable of supplying sufficient current to power the board.

Mezz_2V_bad Indicates that one of the power supplies has failed. This should briefly light up and then disappear when the power is powered on.

7. AD16-250 Series Cable I/O and LEDs



7.1 AD16-250 Cable Description

Heterodyne Trigger In A heterodyning sine wave input. Triggering will automatically occur on the difference frequency between this input and the sampling clock frequency. This is useful for time-of-flight imaging systems, RADAR/LIDAR and pulsed spectroscopy systems, in which transmit or stimulus waveforms are repeated M-times/second and the A/D samples data at a rate of N samples per second. The result is that the ULTRADYNE board will automatically acquire and/or average complete waveforms that repeat M minus N times per second.

External Clock In A single frequency clock must be continuously supplied before any user programs are run. This external clock is useful if acquisition is to be synchronized to an external source. For applications in which an external clock is not desired, the board's internal clock can be used instead, the software must select internal clock mode, and the external clock input must be left unconnected.

Sync/ Selective Recording In This is the trigger or selective recording TTL input. The board will store data to the on-board RAM only when this “acquire disable” input is LOW (board has an internal pull-down resistor, leave unconnected if unused). The timing in this mode is not precise to a single sample and can vary from 0 to 3 samples if the input is not synchronized with the A/D clock.

Analog In 3, Analog In 2, **Analog In 1, Analog In 0** The SMA analog input connectors accept a single-ended voltage in the range specified in the specifications section, under no circumstances should the signal supplied to the analog input exceed the specified range as damage may occur that is not covered by the warranty. All Analog Ins are concurrently sampled, and their precise time-alignment makes them well suited for I/Q sampling.

7.2 AD16-250 LED Description

Acquire Enable Indicates that the board is armed for acquisition and will store data when the user requests this by means a software command and optional external trigger.

Acquiring Data Indicates the board is currently storing data to its local memory.

DMA Active Indicates the board is performing DMA transfers to the host system. As the boards can concurrently acquire to the on-board dual-ported memory, and DMA to host, both this and the Acquiring Data LED will often be lit concurrently.

Power Bad Indicates the DC power to the DRAM and ADCs is out of spec. This may occur if a fault develops in the board's power supply, or if the slot in which it is installed is incapable of supplying sufficient current to power the board.

Config Indicates the board's FPGA is properly configured.

8. Hardware Installation and Setup

16GB of system memory is required (with >8GB free) to properly run 8GB acquisitions. 32GB is recommended. Handle the board carefully – mechanical damage to the board is not covered by warranty. **To avoid overheating, the board must be installed in a well-cooled workstation, server or industrial chassis PC**. Installation in a PC or workstation without **fans at the front end of the card cage will cause the board to overheat, and resulting damage is not covered by warranty. If after 5 minutes of operation, any component on the board feels too hot to comfortably touch, a system with better cooling is required**.

1. Use the shutdown command, turn off the system power, and disconnect the power cord.

BEFORE REMOVING THE COMPUTER SYSTEM COVER OR REMOVING ANY BOARD, BE SURE THAT THE POWER TO THE COMPUTER, AS WELL AS TO ALL PERIPHERAL DEVICES IS OFF. WEAR A STATIC-DISSIPATING WRISTBAND WHICH IS GROUNDED TO THE SYSTEM CHASSIS WHILE OPENING OR WORKING ON YOUR SYSTEM.

2. Remove any screws that attach the computer system cover and remove the cover.

3. Remove the filler bracket from the PCIe slots the board will occupy **and the filler bracket from the PCIe slot below the board, to allow adequate air flow across the board's heatsink**.

4. Hold the board by the top of the metal bracket and the back of the board (**Do NOT EVER hold or exert any force on the DIMM memory modules**). Carefully slide the board in so its PCIe connector mates with the motherboard PCIe connector. **Do not force the board. If there is any resistance, rock the board slightly when inserting it.** Be sure the board is seated firmly into the motherboard PCIe connector. Be sure no other PCIe/PCI boards have become unseated.

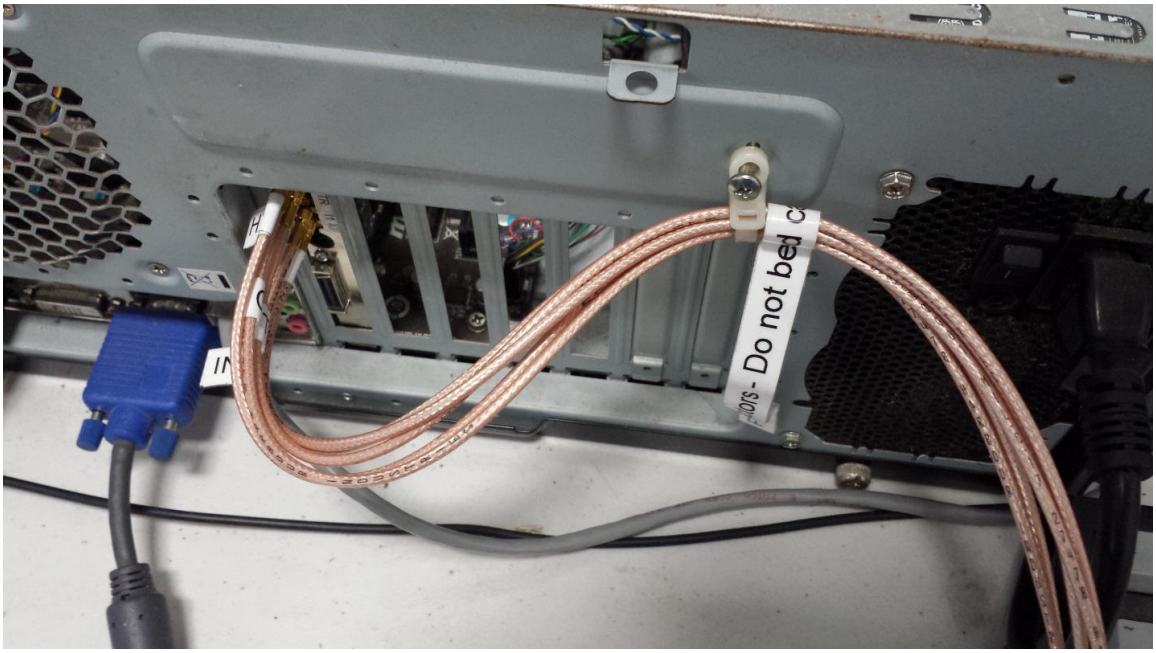
5. Plug coaxial I/O cables for the analog inputs and/or outputs into the appropriate SMA connectors on the board. Connect the free ends of the analog input cable to the signal sources to be digitized, and connect the clock input cable to a suitable clock source, if using an external clock. **WARNING: The MMCX connectors on the AD16-250x2/x4 are very fragile. Do not exert any side force on connectors – push cables straight on and pull them straight out with only very slight rocking (<10 degrees**). Clamp cable assemblies to back of chassis to provide strain relief, and do not exert any sideways force on cables.

6. Replace the computer system cover. Reconnect power cord to the system. Power up and reboot the system. The system will then be ready for software installation.

In some systems the configuration time of the on-board FPGA may be long enough that the FPGA is not fully configured before the system begins accessing the board. If the board is not recognized by the system (e.g. not present in the Device Manager (Windows) or not shown using lspci (Linux)), or the system hangs when accessing the board, or other errors occur, then this may be the problem. If any of these issues are seen the system will need to be "warm restarted" before the board can be used. The easiest way to do this is to boot the system to the OS boot menu, then restart the system without powering it off, then let the system start normally. This allows the FPGA sufficient time to fully configure (a result of powering on) before it is assessed. This can be easily done by adding a "reboot" option to the GRUB boot loader in Linux, hitting this reboot option after the first cold boot, and then allowing the system to continue to reboot again with its normal default boot option. Another way to do this is issue a ctrl-alt-delete during the bios post screen.

WARNING!

MMCX INPUT JACKS USED ON AD16-250 BOARDS ARE FRAGILE. GENTLY EASE CABLE CONNECTORS STRAIGHT ONTO MATING JACKS (WIGGLING VERY SLIGHTLY). **DO NOT BEND OR APPLY SIDEWAYS MOMENT TO CABLES NEAR BOARD INPUTS**. AFTER PLUGGING IN CABLES, CLAMP CABLE ASSEMBLY TO CHASSIS TO RELIEVE STRAIN AS SHOWN BELOW.



9. Software Installation and Setup

"Newer boards may have FPGA firmware that is different from earlier firmware versions shipped with prior boards of this model. To ensure correct operation of your new board, **please download the current software package from Ultraview's website. Do not install older versions of drivers or user software that you may have used with previously purchased boards. You may also need to recompile custom user software you may have written**, so it will correctly run with the current driver and board firmware."

For users who modified a previously LabVIEW™ release and wish to update their projects to work with the latest software, see "APPENDEX – How to Update Your LabVIEW™ Project".

9.1 Software Installation for Windows 7 and 10.

The Windows software release can be downloaded from the following URL:

http://www.ultraviewcorp.com/downloads

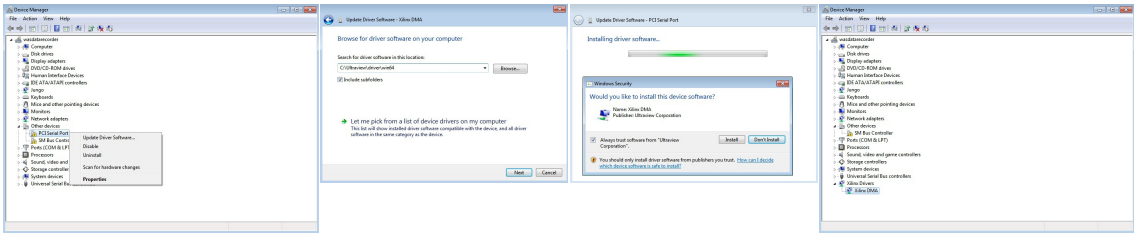
The Microsoft Visual Studio C++ (64 bit) runtime can be downloaded from the following URL:

http://www.microsoft.com/en-us/download/confirmation.aspx?id=30679

To install the software, place the Ultraview Windows release into a directory on your local drive. Run the Microsoft executable to install the runtime. Please read the README file provided in the root of the release for an understanding of the files provided.

To run the LabVIEW executables, the LabVIEW Run-Time Engine 2014 (64-bit) must be installed. If it is not installed, when running the LabVIEW programs a pop-up will direct you to the appropriate download location.

Navigate to the device manager, and find the device "pci serial port" in unknown devices. Right click and select update driver. Select browse for driver software on your computer and browse to the driver/win64 directory of the Ultraview installation files. Click next, and when it asks if you'd like to install the device software with name: "Xilinx DMA", Publisher: "Ultraview Corporation", select install. When successful the device will show up under category "Xilinx Drivers", device "Xilinx DMA".



To avoid data overruns, interruptions in data acquisition, and hanging of user programs, **turn off all system power management options, screen savers, etc. The system must not be allowed to go into sleep mode when the board is running.** To run the example programs for Windows 7 or 10, go to the section below, "Running the Example Programs under Windows 7, 10".

9.2 Software Installation for Linux (64-bit)

Centos 6.2, 6.3, 6.4, 6.6, 6.8, and 7.0 are the only officially supported Linux operating system versions, but many other similar flavors should work.

To avoid data overruns, interruptions in acquisition, and hanging of user programs, **turn off all system power management options, screen savers, etc. The system must not be allowed to go into sleep mode when the board is running.**

After installing the board, download the software from the website and log in as **root**, copy the file **uvdma-x-y.z.x86_64.rpm** to any directory (x,y,z are integers denoting the current revision of the Linux release, below the revision is shown as 1-1.10), and type in the following line at the prompt (shown here as #):

 # rpm –ivh uvdma-1-1.10.x86_64.rpm

The installation script will automatically create a directory /uvdma, and install the software in this directory. Precompiled driver modules are supplied and must be loaded as follows:

 # unzip linux_driver.zip
 # run "make" in the driver directory that was just extracted
 # run /tests/load_driver.sh

To uninstall the RPM, as required prior to installing a newer version of software, the following command is used:

 # rpm –e uvdma

To insure correct operation of the device driver with your kernel version you may be required to recompile the driver module. **On Centos 7, rebuilding the driver is required for proper operation.** Before recompiling the driver module a link to the kernel source must be setup. Ensure that the gcc package is installed by running "yum install gcc". The driver module can now be built, **you must then reboot** before going further.

10. Running the Example Programs

The software provided allows the A/D boards to be run under either a Command Prompt under both Windows and Linux, as well as in LabVIEW for Windows.

See Section 11.2 for how to run the LabVIEW project.

See Section 11.3 and 11.4 for how to run the Command Prompt project

10.1 Preparing to Run/Modify the Example Programs

The software release has both source and executables for the various example programs, which can immediately be run to demonstrate the use of the board, and form an excellent basis for developing your own custom software. The source code for both the command line Linux example programs and the command line Windows example programs share an identical section of cross platform source code; this common section of code, located in directories “AppSource” and “_DllSource” in the release, and can be compiled under Linux or Windows. There is also a supplied multi-threaded high performance Qt based project in the “Qt” directory, however custom application support for this project is not supported without support contract due to the complexity. There is both a Qt based GUI and a Qt command line project, both located in the “Qt” directory.

Usage information for the applications is available by simply running the executable without any arguments. The on board prom contains calibration settings (offset/gain/bias) of each board. If the user would like to change these, please see appendix 16.

For Linux:

To insure correct operation of the device driver with your kernel version you are required to recompile the driver module. On Centos 6 and 7, rebuilding the driver, on the actual host machine in which it is to be run, is required for proper operation. Ensure that the gcc package is installed by running "yum install gcc". The driver can be rebuilt by navigating to /uvdma/driver/src, opening a terminal, typing make clean, then make. After the driver recompiles successfully, enter the commands: rmmod uvdma.ko, then insmod uvdma.ko to reload the module. The driver module is now built and loaded, you must then reboot before going further.

The example programs can be rebuilt under Linux by typing “**make**” from “/uvdma/src/_DllSource” and “/uvdma/src”. Typing “**make clean**” clears all executables so that “make” can write new files. The gcc-c++ package must be installed; under CentOS, the command "yum install gcc-c++" will install gcc-c++.

For Windows:

For users who modified a previously LabVIEW™ release and wish to update their projects to work with the latest software, see “APPENDEX – How to Update Your LabVIEW™ Project”.

The DLL was compiled using Microsoft Visual Studio 2012 Professional and the user acquire programs were compiled using Microsoft Visual Studio 2017 Community.

The LabVIEW project was developed with LabVIEW 2014 64-bit Professional Development System.

To recompile the DLL:

1) Open "src/AcqSynth_dll/**AcqSynth.sln**" in MSVS 2012 professional.

2) In the "Build" tab, select "Rebuild Solution"

To recompile the Acquire program:

1) Open "src/command_line_utilities/Acquire/**acquire.sln**" in MSVS 2017 community.

2) In the "Build" tab, select "Rebuild Solution"

Do not open any ".vcproj" as they are not updated and point to old directories. The quickest way to undo is to unzip a new folder and open the ".sln" files.

The resulting executables are built into the "utils" and the "command_line_utilities" directories in the Windows release.

Other versions of Microsoft Visual studio will work as well, but may require some additional steps.

10.2 LabVIEW™ The Graphical Waveform Viewer under Windows 7™ and Windows 10 For AD12, AD14 and AD16 – series boards.

This complex but versatile LabVIEW project incorporates most of the same features as the 'acquire.exe' command but has the advantage of displaying the data acquired in real time with a powerful user interface. To run the LabVIEW executables, the user must first download the LabVIEW 2014 64-bit Run-Time Engine, depending on the description of the software package.

The examples in this section are specific to the AD12 boards, however the AD16 has the same LabVIEW interface and the AD14 has a similar LabVIEW interface.

Please Note: All directory references are within the current software directory that the ultraview software was extracted to.

Quick Start in Run And Display:

1) In the "complete_daq_utilities\LabVIEW_2014_64-bit" directory, double-click either **LabVIEW_Acquire_Data_x64.exe** or **LabVIEW_Acquire_Data_x86.bat** depending on the system OS: Windows 7: exe, Windows 10: bat.
2) In **ReadandDisplay.vi**, press **Run** under the **RunAndDisplay** tab.
3) At this point, the board should be running in Continuous mode using the internal clock with no triggering (i.e. free-run mode). Data is simply being read from the board and displayed, and is **NOT being stored to disk.**

Quick Start in Acquire And Save:

1) In the "complete_daq_utilities\LabVIEW_2014_64-bit" directory, double-click either **LabVIEW_Acquire_Data_x64.exe** or **LabVIEW_Acquire_Data_x86.bat** depending on the system OS: Windows 7: exe, Windows 10: bat.
2) In **ReadandDisplay.vi**, select the **AcquireAndSave** tab, browse to select or create the file where you wish to save the data, and press **Acquire Data.** At this point, the board should be running in Acquire Once mode using the internal clock with no triggering. Data is simply being read from the board and displayed, and **IS being stored to disc** in the default directory with the file name **uvdma.dat** with a size of **1MB** when the **# of Blocks to Acquire** is "1".

Quick Start in Read Data File:

1) In the "complete_daq_utilities\LabVIEW_2014_64-bit" directory, double-click either **LabVIEW_Acquire_Data_x64.exe** or **LabVIEW_Acquire_Data_x86.bat** depending on the system OS: Windows 7: exe, Windows 10: bat.
2) In **ReadandDisplay.vi**, select the **ReadDataFile** tab, browse to select the file where you wish to read data from, and press **Play Whole File.** At this point, the file should be read and displayed in the graphical windows. Data is simply being read from the file and displayed, nothing is being stored or read from any external source. If the bit resolution of the file does not match the bit resolution of the board currently in the system, use the ADC res readback control at the bottom. This viewer can be used to view the amplitude of any raw binary file with 8,12,14,16, or 32 bits per sample.

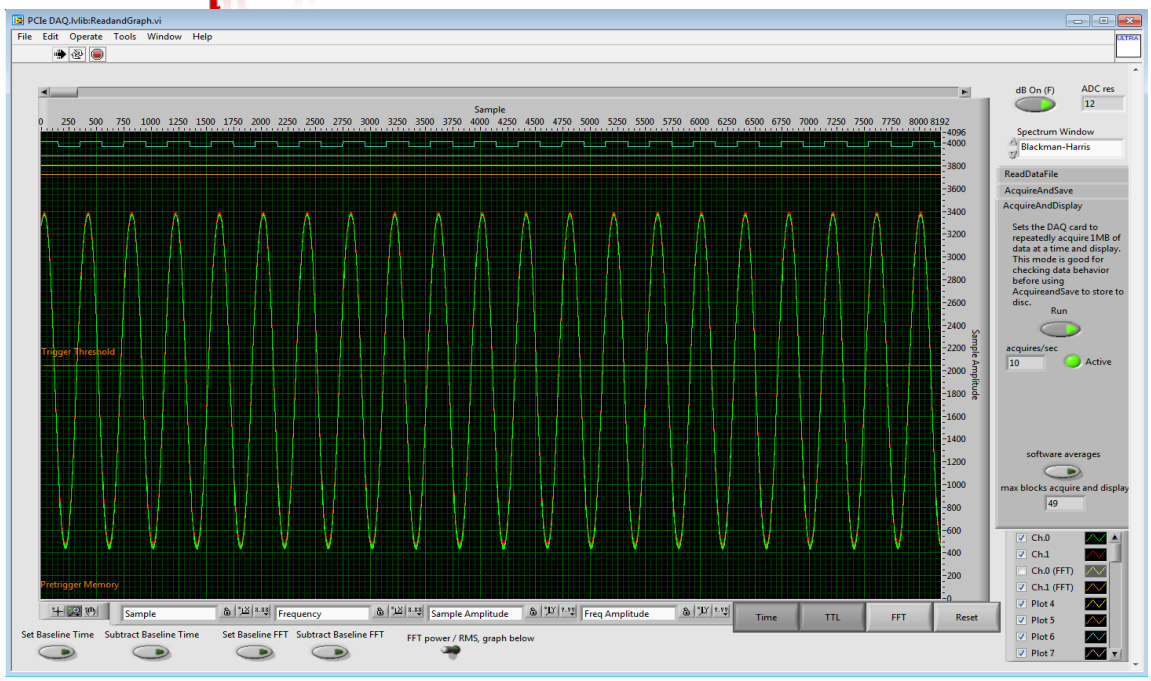


Figure 11.2.1 ReadandGraph.exe

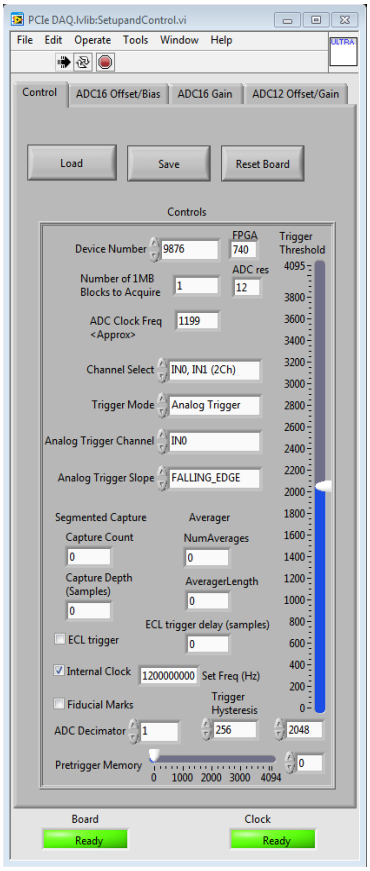


Figure 11.2.2 SetupandControl.exe

Two executables communicate over TCP/IP to share important board configurations.

ReadandDisplay.exe acts as the client, and SetupandControl.exe acts as the server. For the user's convenience, **LabVIEW_Acquire_data_x64.**exe and **LabVIEW_Acquire_Data_x86.bat** are simple batch files to open both programs so that TCP/IP connects properly. To do this manually, the user must open SetupandControl.exe FIRST and ReadandGraph.exe SECOND, located in the "build" directory. All boards are set up with default parameters when the Labview program is opened and only if there is a lockup does the "Reset Board" button need to be pressed.

When running **SetupandControl.vi**, the first tab displayed in the program is the "**Control**" tab. The "**Load**" button can be used to load the device and configuration settings from the previous running of the program. Modify these settings and press the "**Save**" button to save configuration settings. Next, to change boards, if multiple boards are in the system, simply change the "**Device Number**" to the proper serial number and click **"Setup Board"** to initialize the default configuration settings. **Note: These configuration settings are the same as used for the acquire program. Triggering and select recording options are explained in Section 13.4.1.**

Note: The board has not yet begun acquiring.

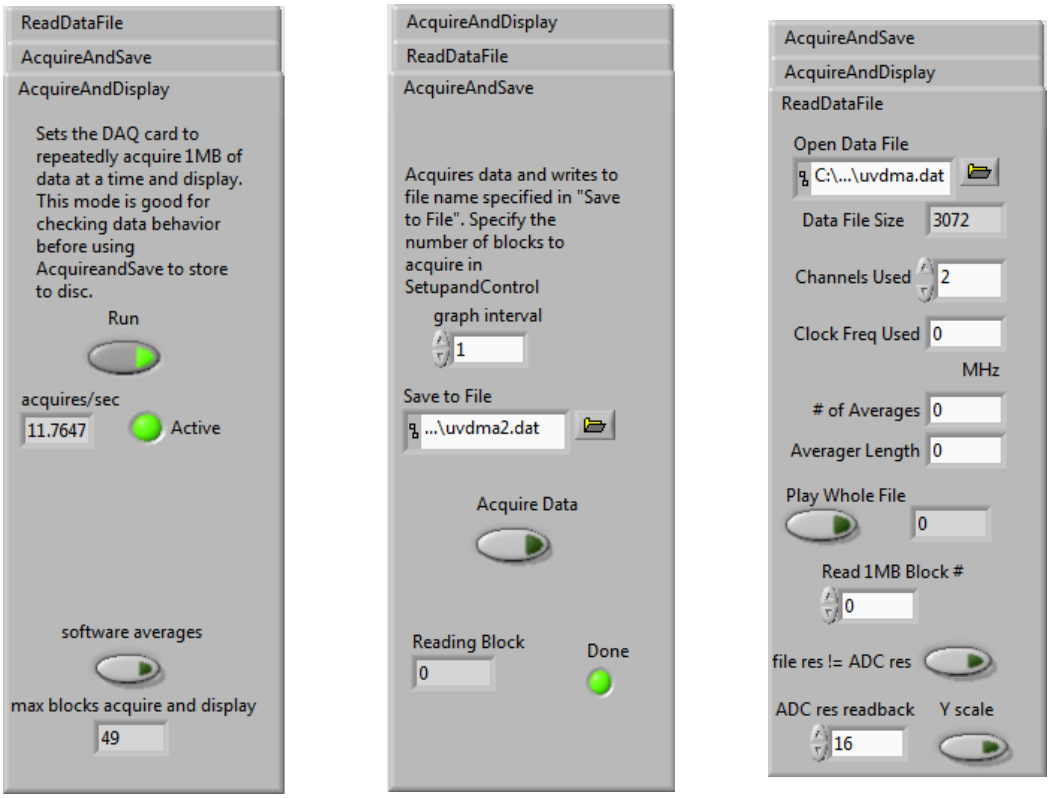


Figure 11.2.3 AcquireAndDisplay – AcquireAndSave – ReadDataFile tabs

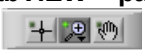
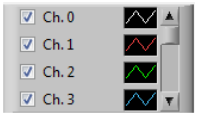
For all triggered modes of acquisition the AD12, AD14, and AD16 adds programmable capture length. For each recognized trigger (TTL edge or analog threshold), the board will acquire "Capture Depth" number of samples. Note that if "Capture Depth" or "Capture Count" are zero then once triggered the data will be recorded until the acquisition is complete (normal mode). There is an added "Capture Count" modifier, which programs how many trigger events to record before resuming normal capture operation. Since our minimum block size to acquire is 1Mb, and

a user may have a situation where only a limited number of triggers are going to be produced, using the “Capture Count” flushes out the rest of the buffer after a programmed number of triggers.

When running **ReadandGraph.vi**, the time and frequency graphs are **overlayed** onto a single graph. The **left and bottom axes** are for the **frequency domain response**, while the **top and right axes** are for the **time domain response**. Simple **“Time”** and **“Freq”** buttons are available to easily hide Time or Frequency Plots. The **“Reset”** button resets the X and Y scales to default values for both Time and Frequency.

In the “**AcquireAndDisplay**” mode, use the "**Run**" button to start and stop acquisition of data. In this mode the board is used to acquire and display a small amount of data. **AcquireAndDisplay does not store data to disk.** In this mode the board starts acquisition, then after accumulating the desired number of blocks the data is displayed. The board starts again, and this cycle repeats as long as the “Run/Stop” button is pressed. This mode is useful for quick observation/verification of real-time data. When triggering, convenient cursors are shown to indicate the waveform trigger threshold and pre-trigger value. Additional cursors can be made and customized by right-clicking the cursor legend. In SetupandControl.exe the user can adjust the number of 1MB blocks for LabVIEW™ to process by increasing the “**# Blocks to Acquire**”. The RunAndDisplay mode sets the board up to acquire 1 to 1+X (AD12 and AD16) blocks at a time, where X is the number of additional blocks supported by available memory at run time. Use of the hardware averager is accomplished by specifying a number of averages, an averager length, and a trigger. When using the hardware averager with a valid trigger on supported boards, the software averager may also be used by pushing the “software averages” button. This will sum and divide the entire record (up to many blocks) into the length specified by AveragerLength and give a very smooth, low noise waveform for a consistent signal with incredibly high SNR.

The "**AcquireAndSave**" tab can be used to acquire the previously specified number of 1MB blocks of data to a file and then view the data. Change the “graph interval” value to acquire data without displaying every waveform to speed up acquisition. Specify the number of blocks to acquire by changing the “# **Blocks to Acquire**” before pressing “Acquire Data”. When using AcquireAndSave, the user can specify 8192 blocks guaranteed without overruns and data is stored to “uvdma.dat”. **Specifying more than 8192 blocks may result in overruns if the input clock is too fast.** When the Acquire button is pressed, each 1MB of data is displayed and stored in user defined filename in the default directory. AD12 unaveraged is 12 bit data with 2 TTL bits and 2 blank bits. AD12 averaged is 32bit data. AD14 unaveraged data is 14bit data with 2 TTL bits. AD14 averaged data is 32bit data. AD16 unaveraged data is 16bit data. AD16 averaged data is 32bit data.
AD12/16 only: For looking at the entire record without software averaging, select the appropriate bit resolution in the ADC res readback control and push the “file res != adc res” button. The scale can be quickly adjusted to the correct scale by pushing the Y scale button.

Built-in LabVIEW™ panning and zoom utilities are available on the **bottom and bottom-left** of the graph  (zoom palette)  (axes palette). Right-clicking the graph itself allows for more graph options. **Built-in LabVIEW™ plot legend** are made available to the user on the **bottom-right** of the graph 

(plot legend).

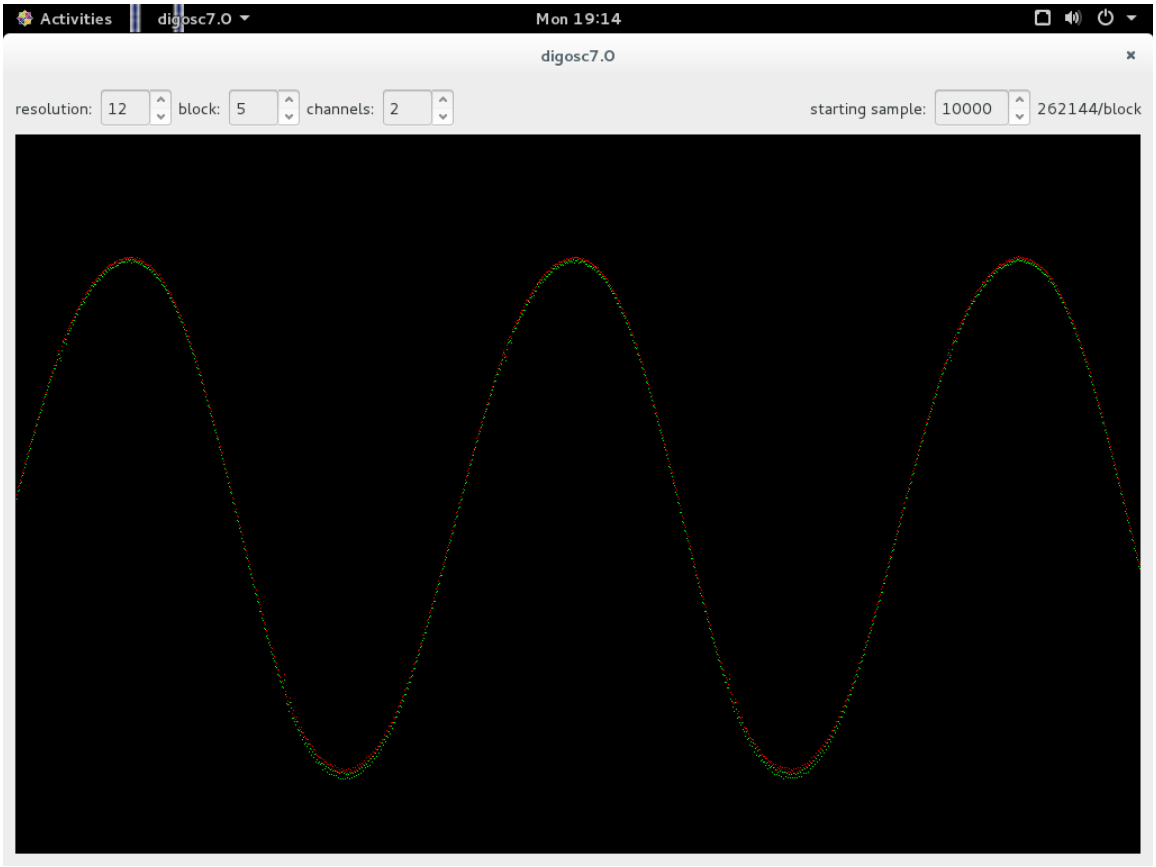
Note: There is a LabVIEW™ bug where the checkbox for the first channel does not work. Instead, left-click on the plot icon and deselect “Plot Visible”.
Note: There is a LabVIEW™ bug where the plot legend retains all plots in the list, even if there is no data in those channels. So ignore the long list.

At any time the user wishes to return to default window settings, the "**Reset Board"** button will do just that.

To properly close the LabVIEW software without any warnings popping up about TCP/IP error, close SetupControl.exe FIRST, then close ReadGraph.exe SECOND.

10.3 The Graphical Waveform Viewer under Linux

The application "digosc7" displays, in waveform format, the A/D data acquired by the board. This graphical waveform viewer can be used to read data from a previously stored record on the hard disk.



When executed the program will ask for a binary file to read in, then ask for the specifications of that file. After specifications are entered, digosc will display the data. The data can be scrolled through by changing the sample number in the upper right, and the block number in the upper left. Only the current 1MB block, the previous 1MB block, and the next 1MB block are loaded into memory at a time for fast access to large files that may be larger than system memory. As such, changing the block number re-enters data from the file specified, and if the file is changed the new file will be read in instead.

10.4 Cross Platform Command Line Acquisition & Synthesis

Both the Windows and Linux software releases include the same C source code files that implement a program to acquire data to disk and a program to synthesize data from disk. The Windows release includes these files as Visual Studio projects and the Linux release includes these files in /uvdma/src/AppSource and /uvdma/src/_DllSource, each with an included makefile. These example programs are mostly single threaded, with small sections of multithreading for DMA transfer, this is the simplest project to modify to get up and running with the board for custom applications.

There is a Qt based project that is available on both Windows and Linux, that shows how to operate a high performance multi threaded application for controlling the board. There is both a command line and graphical Qt program. Due to the complexity, only moderately experienced or advanced programmers should attempt to make large changes to this program. This multi threaded program achieves the highest sustained transfer rates due to complex architecture designed to keep the board running as fast as possible. For anything but the highest AD12-2000 data rates combined with a state of the art RAID system, the non-Qt programs will transfer data sufficiently fast to keep up with acquisition.

10.4.1 acquire - acquire data into on-board DRAM, and then store the buffer to disk.

The program “acquire” acquires a selectable number of DMA blocks of data into the board's onboard memory and then move the data to a file on disk.
The program acquire can be used to acquire data to the board’s on-board RAM, followed by an automatic storage of the board’s buffer to a disk file. **For usage of acquire, perform the following commands and read all statements printed, further details can be found in the comments in acquire.cpp.** A minimum of 1 1MB blocks of data can be acquired, there is no maximum, but for the data to be valid through the entire record the sampling rate cannot exceed the saving to disk rate for longer than the 8GB on board memory buffer allows.

 # cd /uvdma/example_programs
 # ./acquire

Ex. ./acquire //Linux
acquire //Windows

// This will prompt a list of options the user can use to configure the board when acquiring

Ex. ./acquire 1 –ic //Linux
// acquires 1MB using internal clock with no triggering and stores to disc in the build directory with file name uvdma.dat

Note: In Windows, the acquire program is accessed through the “Command Prompt” in the “build” directory. Simply type “acquire” in the Command Prompt for a list of operations.

Ex. acquire 1 //Windows
// acquire 1MB using external clock with no triggering and stores to disc in the build directory with file name uvdma.dat

The acquire program has several different modes of operation. In the simplest case, the board acquires to onboard DRAM while continuously reading from the board into a small bounce buffer allocated in host memory and then writing from the bounce buffer to disk. In this case the board can acquire up to the size of the board's memory without any limitation on incoming data rate, or for a very long time if the disk DMA rate is sufficient relative to the incoming date rate. Alternatively, the acquire program can be compiled to use a large amount of system memory (typically limited to the amount of system memory minus a couple gigabytes) to increase the capture length. In this case writing to disk is delayed until the capture is completed, data is acquired into onboard DRAM and DMA'ed to the allocated system memory until this system memory is filled then the board's memory is filled until it contains only new data (data not yet transfered to the host). In this case the acquisition rate is limited to approximately the DMA rate of the system.

The acquire program offers both triggering and on 12-bit, 14-bit and 16-bit models, selective recording. The board defaults to active LOW, selective recording. For example, if the board is prepared such that there is no trigger signal connected to the “Acquire Disable (Trigger)” SMA jack with the command “acquire 1 -ic”, the board will continuously acquire. However with the same command, if the board has the proper trigger input (see board specifications under “Optional (Selective Recording) Input”), the board will disable acquisition when the trigger signal is HIGH and will enable acquisition when the trigger signal is LOW.

Special arguments for AD12-2000 boards:

1) “-ecltrig” Waits for ECL trigger from the Trigger Sync SMA Jack
2) “-ecldelay” Sets the ECL trigger delay.

64us * (1clk cycle)/(8 samples) * clk_rate (samples /s)

Example: clk_rate = 1GHz=> ecldelay = 8000

Default: ecldelay = 8000

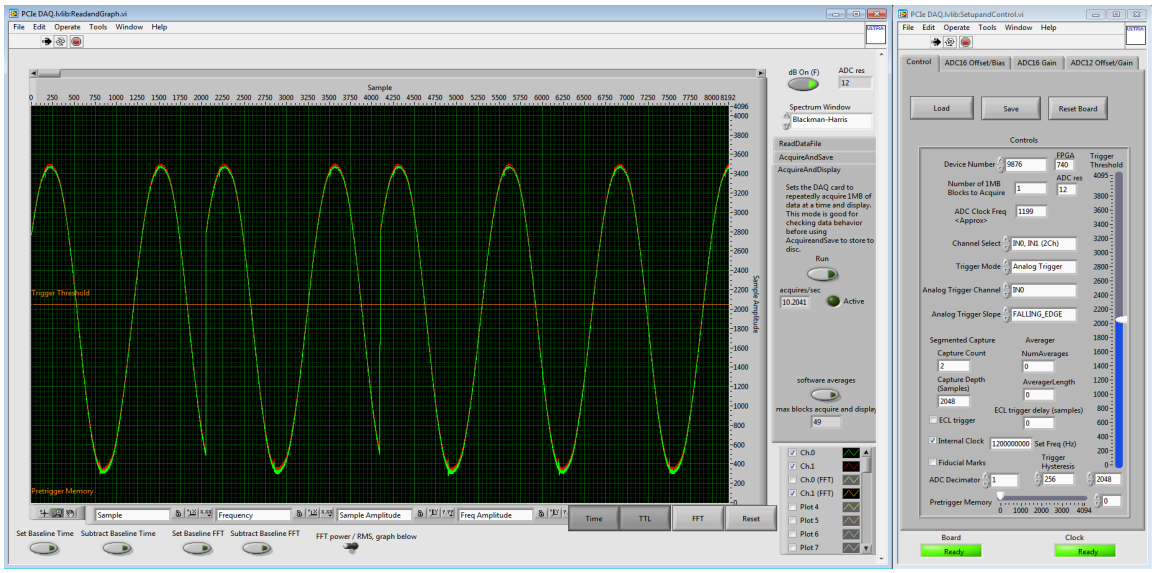


Figure 14.4.1 Shows capture_count = 2, capture_depth = 2048. The triggers start at 0 and 2048, data is captured continuously after the second trigger.

Special arguments for AD12-2000, AD14-400, AD14-500 and AD16-250 boards:

1) “-syncSelRecord” Waits to record until the “Acquire Disable” input transitions from “disabled” to “enabled. For example the command : “acquire 1 –syncSelRecord” will wait until the selective recording input transitions from High to Low to begin recording. Whereas the command: “acquire 1 –syncSelRecord -ttlinv” will wait until the selective recording input transitions from Low to High to begin recording. This enables the user to be certain of the timing of the first block of data relative to the “Acquire Disable” input. The recorded data is shifted by 13 samples relative to the “Acquire Disable” input, such that there are 13 samples of pre-trigger memory. The user should be aware that this also means that when “Acquire Disable” goes to disabled, the last 13 samples are not recorded.

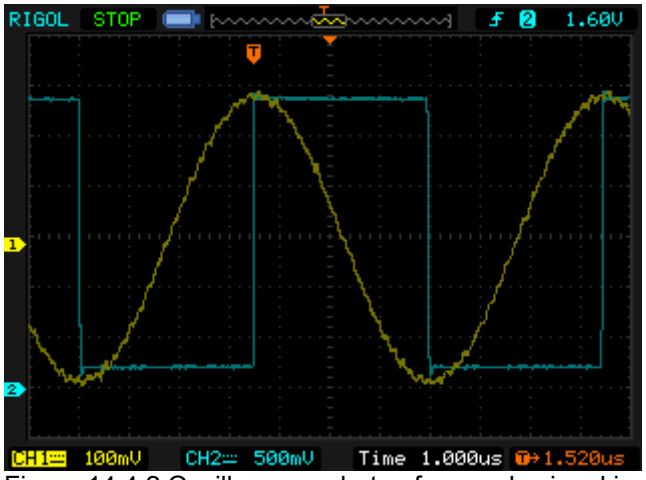


Figure 14.4.2 Oscilloscope photo of example signal input (yellow) and Acquire Disable input (blue) to the AD16-250.

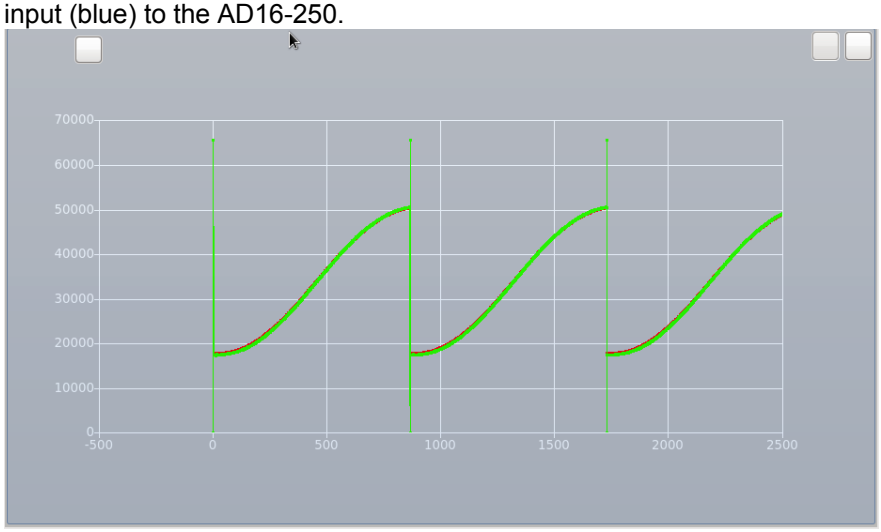


Figure 14.4.3 Plot showing data captured as in 11.1 using the command:
./acquire 1 –ic –syncselrecord –fiducial

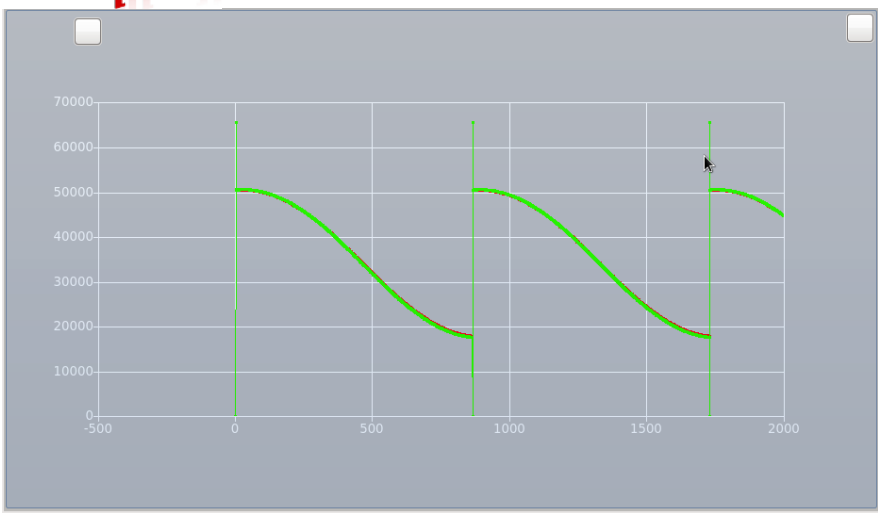


Figure 14.4.4 Plot showing data captured as in 14.4.1 using the command: ./acquire 1 –ic –syncSelRecord –fiducial –ttlinv

2) “-fiducial” places fiducial marks at the beginning of each selective recording. Fiducial marks are as follows:

a. 1 channel mode: chA 0x0000 , 0x0001, 0xFFFF, 0xFFFE, 9 pre-trigger samples, Aligned data.
b. 2 channel mode: chA 0x0000, 0xFFFF, 11 pre-trigger samples, Aligned data. chB 0x0001, 0xFFFE, 11 pre-trigger samples, Aligned data.

3) “-avg <N>” enables averaging of the waveform where N = [0,65535] (0 = no averaging).

a. When averaging 1 channel of data, the maximum averaging buffer size is 131072 samples.
b. When averaging 2 channels of data, the maximum averaging buffer size is 65536 samples.
c. When averaging 4 channels of data, the maximum averaging buffer size is 32768 samples.

4) “-avg_len <N>” specifies the averaging buffer size

The averager waits for a valid trigger to occur (analog waveform, TTL, or heterodyne). Once a valid trigger is found, the average collects the averaging buffer size worth of data, holds it in memory, and evaluates whether it has accumulated enough times (N times). If it has not accumulated enough times, it will wait for the next valid trigger and add it into its previous stored data. Once it has accumulated enough data, it will start the next buffer. This process will repeat until all requested blocks of data has been acquired

5) “-ttledge” allows for TTL triggering

6) “-analog” allows for analog waveform triggering. (Default falling edge)

“-ttlinv” for rising edge trigger

7) “-analog_ch <Chan N>” to select which channel to analog trigger from. (Chan N) must be either 0,1,2, or 3.

8) “-thresh_a <N>” for analog trigger threshold. N=[0,65535]

9) “-thresh_b <N>” for analog trigger hysteresis. N=[0,65535]

thresh_b should always be greater than thresh_a whether rising edge or falling edge.

10) “-pretrigger <N>” for pretrigger memory after trigger is acquired. N=[0,4095] (AD12/AD16). N=[0,511] (AD14). Increments of 8/16 samples for AD12 in dual/single channel mode, increments of 2 samples in all modes for AD14/AD16.

Appending **“-ttlinv”** to the original acquire command will modify the trigger condition to an active high. For example, “acquire 1 -ic -ttlinv” will acquire 1 blocks with the internal 400MHz clock only when the selective recording trigger signal is HIGH.

Triggering allows for continuous acquisition, beginning at the first rising or falling edge (depending if “-ttlinv” has been asserted) of the trigger signal and stopping only when all data has been acquired.

To use an external clock instead of the internal clock, simply leave out **“-ic”** in the command. For example, “acquire 1” will acquire 1 blocks with an external clock and active low, select recording. When using the internal clock, the external clock SMA must remain unconnected.

For multiple channel boards, the **“-scm”** extension enables single-channel mode. When using this option, also use **“-scs”** to specify which channel to acquire from (single-channel select) starting from “0”. For the channel not used, make sure to leave the input unconnected. For boards with four channels, the **“-dcm”** extension enables dual-channel mode, and **“-dcs”** is used to specify which two channels to acquire from (dual-channel select). Valid entries for “-dcs” are “01”/“10” for channel 0 and channel 1, “02”/“20” for channel 0 and channel 2, “03”/“30” for channel 0 and channel 3, “12”/“21” for channel 1 and channel 2, “13”/“31” for channel 1 and channel 3, and “23”/“32” for channel 2 and channel 3. For example, “acquire 1 –ic –dcm –dcs 10 will acquire 1MB of data from channel 0 and channel 1 using the internal clock. Data will output the first sample of channel 0 first, then channel 1 regardless when specifying “10” or “01”.

Note: Both triggering and selective recording use the “Acquire Disable (Trigger)” SMA jack as its input.

Multiple boards can be armed to acquire data in response to the same external trigger by arming them using multiple instances of the acquire program. Run the acquire program with no options and view the output for usage on how to select which board is targeted when running the program. To target a specific board, append the extension **“-b 0”** to target device 0, and **“-b 1”** to target device 1. **Device 0 will output data into udvma.dat, while device 1 will output data into uvdma1.dat, and so on.** In LabVIEW™, multiple boards will not automatically set. So the user will need to “Reset Board” for the selected device prior to acquiring data.

10.5 Cross Platform Command Line Options

acquire (N blocks) [OPTIONS]

Acquires specified number of blocks. Number of blocks option must always be first.

The following [OPTIONS] may follow the number of blocks option in any order:

For all boards:

"-ic" Board will use the internal clock, do not connect an external clock. If not specified board uses external clock.

"-scm" Run multi-channel boards in single channel mode. Not required for single channel boards.

"-scs (Chan N)" When running multi-channel boards in single channel mode use channel (Chan N).

"-dcm" Run multi-channel boards in dual channel mode. Not required for dual channel boards.

"-dcs (Chan NM)" When running multi-channel boards in dual channel mode. (Chan N) must be either 01,10,02,20,03,30,13,31,23, or 32. "-dcm" must be asserted.

"-b (BoardNum)" Acquires from the specified board. Do not specify the same board from two separate consoles. BoardNum starts at 0.

"-ttlinv" Inverts the TTL trigger. Default is active low TTL trigger.

"-capture_depth <N>" Fixed sample-size capture after each trigger. Sizes are in exact multiples of 8 up to N samples. Must be used with valid triggers. N=0 is normal acquisition. Default N=0.

1-channel mode:
N=131072(Default),65536,32768,16384,8192,4096,2048,1024,512,256,128,64,32,16
2-channel mode:
N=65536(Default),32768,16384,8192,4096,2048,1024,512,256,128,64,32,16,8
4-channel mode:
N=32768(Default),16384,8192,4096,2048,1024,512,256,128,64,32,16,8,4

"-capture_count <N>" Specifies the number of triggers to acquire with size capture_depth. N=0 is infinite triggers until desired number of blocks is acquired. Default N=0.

"-dec (factor)" Enables input sample decimation. (factor) can be 1,2,4,or 8.

"-ttledge" Sets acquisition to await TTL trigger edge.

"-forceCal" Forces calibration

"-syncSelRecord" Waits for trig/sel = FALSE (follows ttlinv) to arm selective recording

"-capturedelay N" (delay N=0-63) Overrides adc clock to data capture delay (14 bit boards only)

"-avg <N>" (N=0-65535) Averages N+1 cycles, N=0 is flow through mode

"-avg_len <N>"
1-channel mode:
N=131072(Default),65536,32768,16384,8192,4096,2048,1024,512,256,128,64,32,16
2-channel mode:
N=65536(Default),32768,16384,8192,4096,2048,1024,512,256,128,64,32,16,8
4-channel mode:
N=32768(Default),16384,8192,4096,2048,1024,512,256,128,64,32,16,8,4

"-analog" for analog waveform trigger. Use -ttlinv to specify rising or falling edge.

"-analog_ch (Chan N)" for analog waveform trigger Used with -ttledge and -averager. Must be either 0,1,2, or 3.

"-thresh_a <N>" for analog waveform trigger threshold. <N> = [0,65535].

"-hysteresis <N>" for analog waveform trigger hysteresis. <N> = [0,65535].

"-pretrigger" for pretrigger memory after trigger is acquired. pretrigger must be 0 to 4095 (AD12/AD16) or 0 to 511 (AD14)

For AD12-2000:

"-ecltrig" Sets acquisition to await ECL trigger.

"-ecldelay" Set the ECL trigger delay. Default = 4000. See manual for details.

"-desiq" Sets the AD12-2000 to DESIQ single channel mode. Both inputs must be externally driven. In DESIQ, the I- and Q- inputs are shorted together. "-scm" must be asserted.

"desclkiq" Sets the AD12-2000 to DESCLKIQ single channel mode. Both inputs must be externally driven. In DESCLKIQ, the I- and Q- inputs remain electrically separate, increasing input bandwidth. "-scm" must be asserted.

For AD16-250:
"-hDiode" for heterodyne triggering.

For AD14-400, AD14-500, AD16-250:
"-fiducial" Places Fiducial marks in data when trig/sel record starts recording

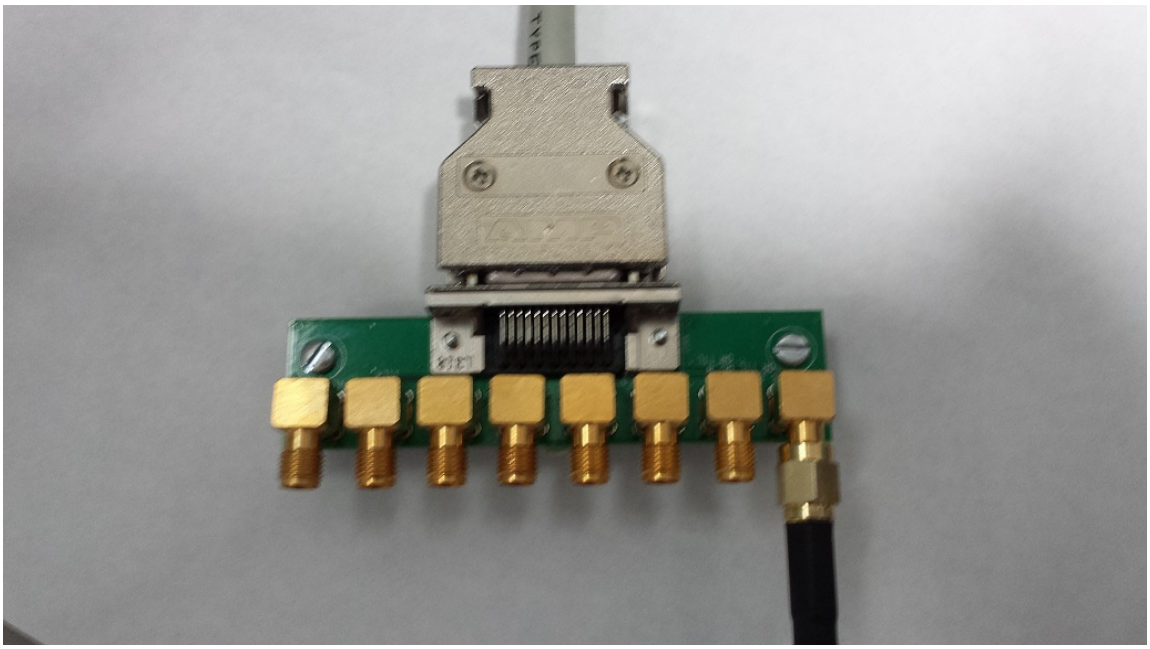


Figure 14.1

11. TTL Input Panel for AD14-400x2 and AD14-500x2 models

Models AD14-400x2 and AD14-500x2 allow concurrent high speed acquisition of up to Four TTL signals, at the same rate as, and time aligned with, the A/D sampling. This feature is useful for correlation of analog/RF signals and the TTL control signals in the system. For example, a rotating RADAR antenna's position may be monitored using the TTL inputs, while the received RF signal is concurrently acquired by the two high-bandwidth analog inputs. These TTL bits are easily inputted to the AD14 board via the small TTL boardlet and uDB26 cable provided with these boards. One end of the cable should be connected to the installed data acquisition card, and the other end should be connected to the TTL boardlet shown below.

Figure 14.1. Location of inputs on the TTL boardlet. Signals labeled TTL0, TTL1, TTL2 and TTL3 are the four TTL bits that are acquired synchronously with the A/D data and are stored in the two most significant bits in each 16-bit sample word. TTL4, TTL5, GC_P, and GC_N are reserved for OEM use and are normally not used**. Standard 0V - 3.3V input levels should be used for the TTL inputs. Do not exceed 3.3V**

	Sample0	**Sample1**	**Sample2**	**Sample 3**	**….. etc**
Data bit 31:	TTL 0(s=0)	TTL 0(s=1)	TTL 0(s=2)	TTL 0(s=3)	
Data bit 30:	TTL 1(0)	TTL 1(1)	TTL 1(2)	TTL 1(3)	
Data bits 29-16:	Analog input IN0(0)	Analog input IN0(1)	Analog input IN0(2)	Analog input IN0(3)	
Data bit 15:	TTL 2(0)	TTL 2(1)	TTL 2(2)	TTL 2(3)	
Data bit 14:	TTL 3(0)	TTL 3(1)	TTL 3(2)	TTL 3(3)	
Data bits 13-0:	Analog input IN1(0)	Analog input IN1(1)	Analog input IN1(2)	Analog input IN1(3)	

Figure 14.2. In dual channel mode, the four TTL bits are shown, above the MS bit of analog data in each sample. The two sampled channels, along with the two TTL bits above each are contained in each 32-bit longword of data sent to the host system, as shown above.

Data bit 31:	TTL 0(s=0)	TTL 0(s=2)	TTL 0(s=4)	TTL 0(s=6)	
Data bit 30:	TTL 1(0)	TTL 1(2)	TTL 1(4)	TTL 1(6)	
Data bits 29-16:	Analog input IN0(0)	Analog input IN0(2)	Analog input IN0(4)	Analog input IN0(6)	
Data bit 15:	TTL 0(s=1)	TTL 0(3)	TTL 0(5)	TTL 0(7)	
Data bit 14:	TTL 1(1)	TTL 1(3)	TTL 1(5)	TTL 1(7)	
Data bits 13-0:	Analog input IN0(1)	Analog input IN0(3)	Analog input IN0(5)	Analog input IN0(7)	

Figure 14.3. In single channel mode, in which only IN0 is specified to be acquired, the two TTL bits TTL3 and TTL2, are shown, above the MS bit of analog data for each of the two successive analog samples of IN0 contained in each 32-bit longword of data sent to the host system.

Data bit 31:	TTL 2(s=0)	TTL 2(s=2)	TTL 2(s=4)	TTL 2(s=6)	
Data bit 30:	TTL 3(0)	TTL 3(2)	TTL 3(4)	TTL 3(6)	
Data bits 29-16:	Analog input IN1(0)	Analog input IN1(2)	Analog input IN1(4)	Analog input IN1(6)	
Data bit 15:	TTL 2(s=1)	TTL 2(3)	TTL 2(5)	TTL 2(7)	
Data bit 14:	TTL 3(1)	TTL 3(3)	TTL 3(5)	TTL 3(7)	
Data bits 13-0:	Analog input IN1(1)	Analog input IN1(3)	Analog input IN1(5)	Analog input IN1(7)	

Figure 14.4. In single channel mode, in which only IN1 is specified to be acquired, the two TTL bits TTL1 and TTL0, are shown, above the MS bit of analog data for each of the two successive analog samples of IN1 contained in each 32-bit longword of data sent to the host system.

TTL bits are captured only during non-averaging modes for AD14-400x2 and AD14-500x2. If averaging, leave the TTL bits unconnected.

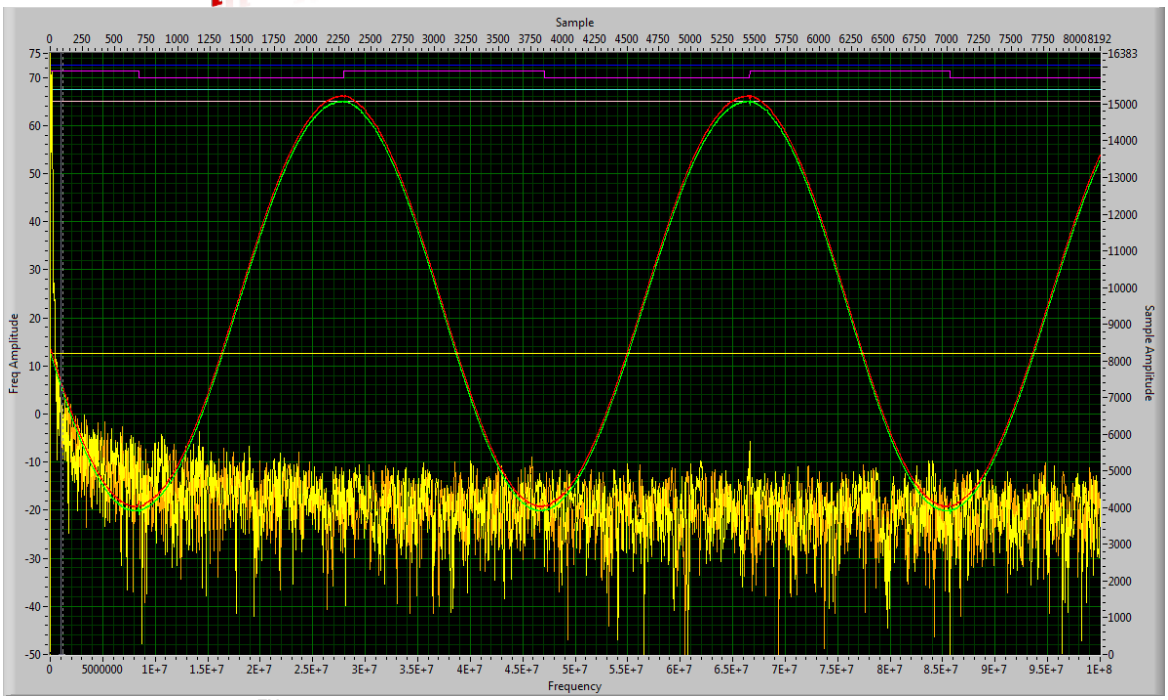


Figure 14.5 LabVIEW™'s graph panel showing an AD14-400x2 in dual channel mode with the TTL signals embedded the top of the graph.

The standard program will keep these TTL signals on the top of the graph, assuming the Sample Amplitude range is large enough to display them. The TTL signals on the graph should be considered arbitrary high and low values.

12. TTL Input Panel for AD12-2000 model

The AD12-2000x2 model allow concurrent high speed acquisition of up to Four TTL signals (when not in averaging mode), at the same rate as, and time aligned with, the A/D sampling. This feature is useful for correlation of analog/RF signals and the TTL control signals in the system. For example, a rotating RADAR antenna's position may be monitored using the TTL inputs, while the received RF signal is concurrently acquired by the two high-bandwidth analog inputs. These TTL bits are easily inputted via the small TTL boardlet and uDB26 cable provided. One end of the cable should be connected to the installed data acquisition card, and the other end should be connected to the TTL boardlet shown in Figure 14.1 above.
The selective recording input is the left-most SMA jack on Figure 14.1 with label "GC_P". The least significant bits are low and the most significant bits are high.

	Sample0	Sample1	Sample2	Sample 3	….. etc
Data bit 31:	TTL 0(s=0)	TTL 0(s=1)	TTL 0(s=2)	TTL 0(s=3)	
Data bit 30:	TTL 1(0)	TTL 1(1)	TTL 1(2)	TTL 1(3)	
Data bits 27-16:	Analog input IN0(0)	Analog input IN0(1)	Analog input IN0(2)	Analog input IN0(3)	
Data bit 15:	TTL 2(0)	TTL 2(1)	TTL 2(2)	TTL 2(3)	
Data bit 14:	TTL 3(0)	TTL 3(1)	TTL 3(2)	TTL 3(3)	
Data bits 11-0:	Analog input IN1(0)	Analog input IN1(1)	Analog input IN1(2)	Analog input IN1(3)	

Figure 14.2. In dual channel mode, the four TTL bits are shown, above the MS bit of analog data in each sample. The two sampled channels, along with the two TTL bits above each are contained in each 32-bit longword of data sent to the host system, as shown above.

Data bit 31:	TTL 0(s=0)	TTL 0(s=2)	TTL 0(s=4)	TTL 0(s=6)	
Data bit 30:	TTL 1(0)	TTL 1(2)	TTL 1(4)	TTL 1(6)	
Data bits 27-16:	Analog input IN0(0)	Analog input IN0(2)	Analog input IN0(4)	Analog input IN0(6)	
Data bit 15:	TTL 0(s=1)	TTL 0(3)	TTL 0(5)	TTL 0(7)	
Data bit 14:	TTL 1(1)	TTL 1(3)	TTL 1(5)	TTL 1(7)	
Data bits 11-0:	Analog input IN0(1)	Analog input IN0(3)	Analog input IN0(5)	Analog input IN0(7)	

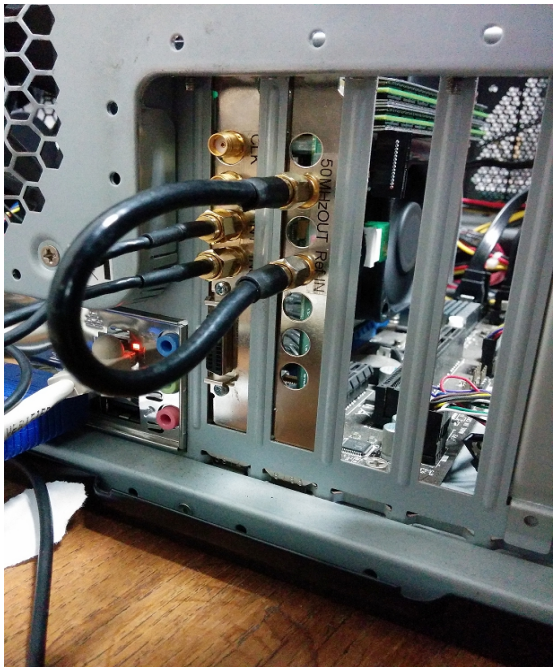
Figure 14.3. In single channel mode, in which only IN0 is specified to be acquired, the two TTL bits TTL3 and TTL2, are shown, above the MS bit of analog data for each of the two successive analog samples of IN0 contained in each 32-bit longword of data sent to the host system.

Data bit 31:	TTL 2(s=0)	TTL 2(s=2)	TTL 2(s=4)	TTL 2(s=6)	
Data bit 30:	TTL 3(0)	TTL 3(2)	TTL 3(4)	TTL 3(6)	
Data bits 27-16:	Analog input IN1(0)	Analog input IN1(2)	Analog input IN1(4)	Analog input IN1(6)	
Data bit 15:	TTL 2(s=1)	TTL 2(3)	TTL 2(5)	TTL 2(7)	
Data bit 14:	TTL 3(1)	TTL 3(3)	TTL 3(5)	TTL 3(7)	
Data bits 11-0:	Analog input IN1(1)	Analog input IN1(3)	Analog input IN1(5)	Analog input IN1(7)	

Figure 14.4. In single channel mode, in which only IN1 is specified to be acquired, the two TTL bits TTL1 and TTL0, are shown, above the MS bit of analog data for each of the two successive analog samples of IN1 contained in each 32-bit longword of data sent to the host system.

13. Microsynth programmable internal clock

On most boards shipped after 8/17/15, the internal clock is a TI lmx2581 which is programmable to any acceptable frequency for the board. This can be done by setting -freq (frequency in Hz) as an acquire flag, or by typing in the desired frequency in the Labview control panel internal clock frequency control and pressing return. If there are additional SMA jacks occupying a second board slot, a reference oscillator **must** be connected to correctly generate an internal clock frequency. This reference oscillator should have an RMS voltage level of 150-400mV and should ideally be a square wave. This frequency must be specified in the prom for your board if switching to another frequency other than 50MHz. If there are no additional SMA jacks, the reference oscillator for the internal clock is hard wired, and is nominally 50MHz.



13.1 Hardware Averager

The hardware averager uses a repetitive trigger to acquire and lay down (sum) successive repeating waveforms on top of each other, to improve SNR. Samples 0,1,2,etc after trigger 0 is summed with samples 0,1,2, etc after trigger 1 and samples 0,1,2, etc after trigger 2 in the case where 3 hardware averages are specified. After the number of hardware averages is reached, the summed 32-bit data is pushed into board memory, and the next set of hardware averages starts. After a sufficient number of hardware average cycles have completed to fill a 1MB buffer (the default transfer size), the data is sent to the host.

Under certain experimental conditions, a faster response time may be required, since hardware averaging requires that many waveforms and triggers occur before data is has filled each 1MB block and becomes available for transfer to the host. The solution in this case is to choose fewer hardware averages (so that each 1MB block fills up quickly) and then use software to perform successive averaging of these pre-averaged waveforms. This improves response time, albeit with the tradeoff of the system CPU being used more heavily (instead of all averaging being done by the Xilinx FPGA). Labview provides a simple example to run both hardware and software averages simultaneously, and different amounts of each may be ideal for different situations:

The examples below are for the case where the user specifies 100 hardware averages and a 2048-sample averager length. In figure 13.1 (all averaging done in hardware) one can see that at

each 2048/4096/6144/8192 boundary the waveform repeats. Since the record length in 2 channel mode is 131072 samples, by clicking the software averages button, as shown in Figure 13.2, the 131072/2048=64 waveforms (each containing the results of the hardware averaging) within each 1MB record are summed and divided 64 times. This provides 64X faster response time but relies on the CPU for the 64x post averaging instead of having the the FPGA average all 6400 times. At some point (application specific) the CPU will become a bottleneck so the number of hardware averages specified should be tuned for optimal response in each individual application.

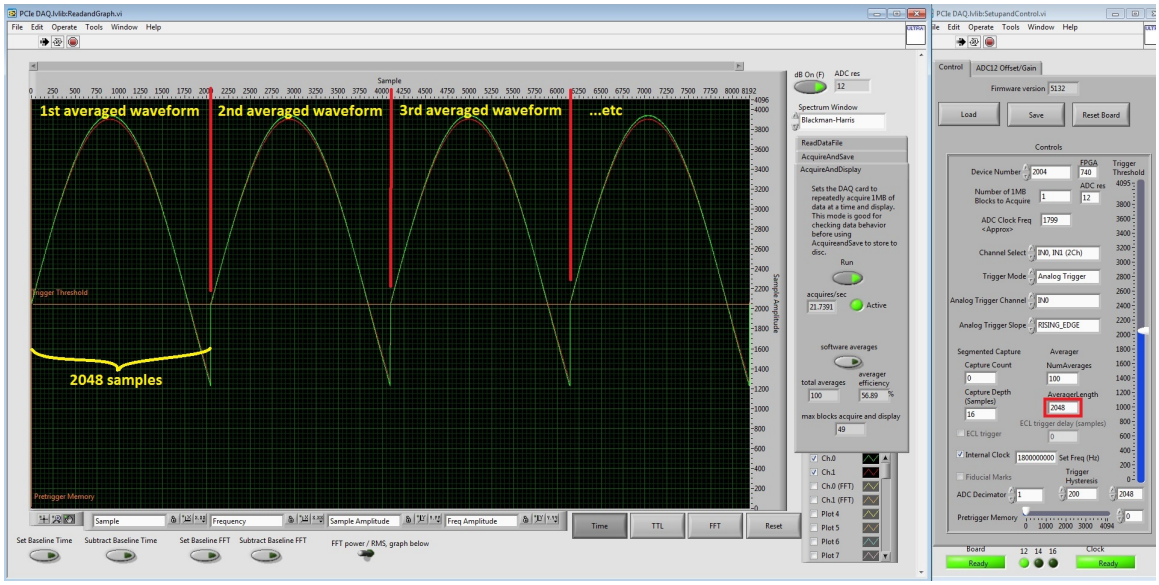


Figure 13.1. Display and/or storage of sequential hardware-averaged waveforms, each one 2048 samples in length, and hardware-averaged 100 times.

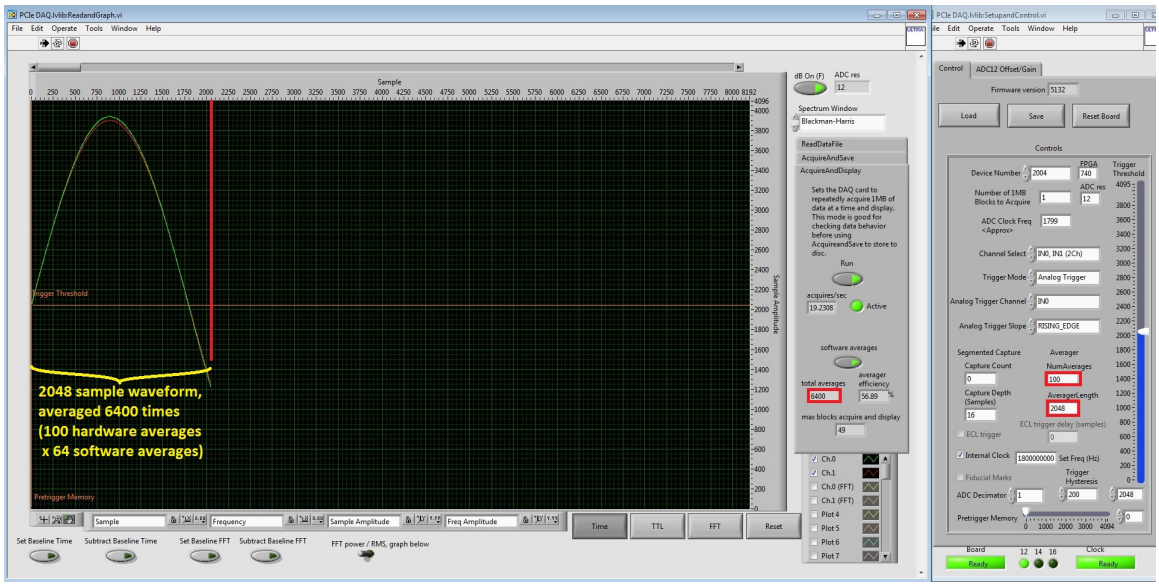


Figure 13.2. Display of waveform averaged 6400 times. The board hardware averages the waveform 100 times, and the host software then averages together 64 of these waveforms. By using hardware and then software averaging, display latency is reduced (improved).

14. APPENDIX – Installing Clock/Trigger Splitter Boards

14.1 Installing the ADSPLTB4 Clock/Trigger Splitter

The ADSPLTB4 is an optional clock/trigger splitter boards that can, from a single clock and optional trigger input, generate as many as four matched clock and trigger outputs for concurrently triggering and running up to four boards. Powered by a “wall-cube” power supply (included), the ADSPLTB4 boards are connected via short SMA-to-SMA cables to the clock and trigger inputs on boards that are to be run concurrently.

To install the ADSPLITB4 board, make the following cable connections:

1) Connect the TRIG0 (and TRIG1, 2 and 3, if used) outputs of the ADSPLTB4 to the TRIGGER inputs on the boards you wish to use. Be sure that the cables used are as short as possible (less than 1 foot recommended) and matched to within ¼”. Also, check that all SMA cable connectors are firmly hand-tightened onto their respective jacks.

2) Connect the CLK0 (and CLK1, 2 and 3, if used) outputs of the ADSPLTB4 to the CLK inputs on the boards you wish to use. Be sure the cables used are short and matched to within ¼”. Check that all SMA cable connectors are firmly hand-tightened.

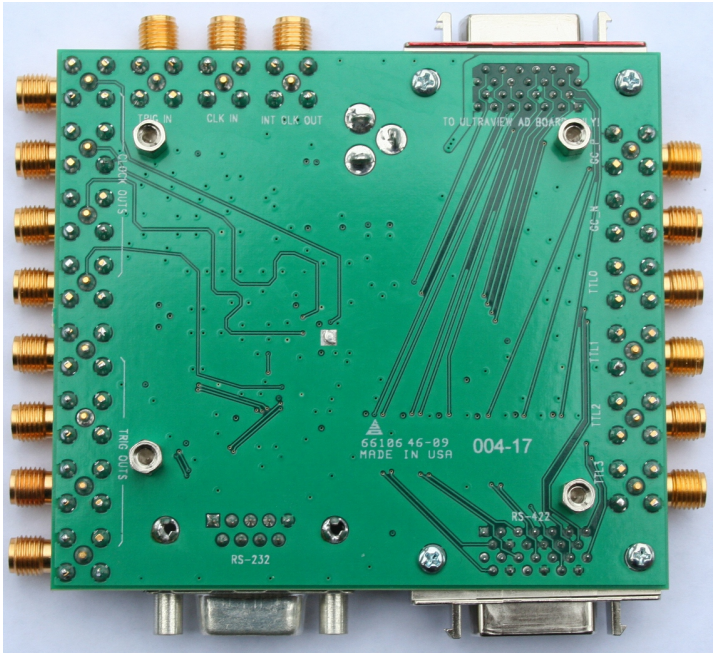
3) Connect the input clock source (0dBm amplitude) to the ADSPLTB4’s “CLOCK IN” jack.

4) Plug the +5V adapter’s cord into the ADSPLTB4’s “DC5V IN” jack. Only the included 5V regulated DC (center + terminal) adapter should be used with the ADSPLTB4.

5) Plug the +5V AC adapter into a standard 120VAC outlet.

After installing the ADSPLTB4 board, the acquire disable LED on boards connected to the ADSPLTB4 should change in response to the single external trigger input, to verify this try connecting a very low frequency (a few tens of hertz is OK) square wave to the ADSPLTB4's trigger input.

14.2 Installing The TTL I/O, RS232-RS422, Clock/Trigger Splitter



The TTL I/O, RS232-RS422, Clock/Trigger Splitter.

This board connects to an Ultraview PCIe main board via cable using the 26 pin connector marked: TO ULTRAVIEW A/D BOARD ONLY!". **Do not use the connector market RS-422 for this purpose**! The board allows 4 TTL inputs to be captured along with 14 bit data, on AD14-400x2 and AD14-500x2 models, by connecting the desired external TTL signals to the TTL3, TTL2, TTL1 and TTL0 SMA jacks. TTL signals TTL1 and TTL0 are sampled as bits 15 and bit 14, occupying the most significant two bits of the 16-bit word containing the data from analog channel IN1. Similarly, TTL signals TTL3 and TTL2 are sampled as bits 15 and bit 14, occupying the most significant two bits of the 16-bit word containing the data from analog channel IN0, as shown below:

See Figures 14.2 and 14.3 for TTL output format for AD14-400x2 and AD14-500x2.

There are two additional unused TTL SMA jacks, GC_N and GC_P which are connected to global clock input/outputs on the PCIe main board's FPGA. These TTL I/O are shown on the right hand side (RHS) of the figure above.

The 26 pin digital connector which connects the PCIe main board and optionally powers this board is seen on the top RHS of this board. **This connector is shown with a red stripe as the 26 pin connector on the PCIe main board must only be connected to this connector to prevent a power short.**

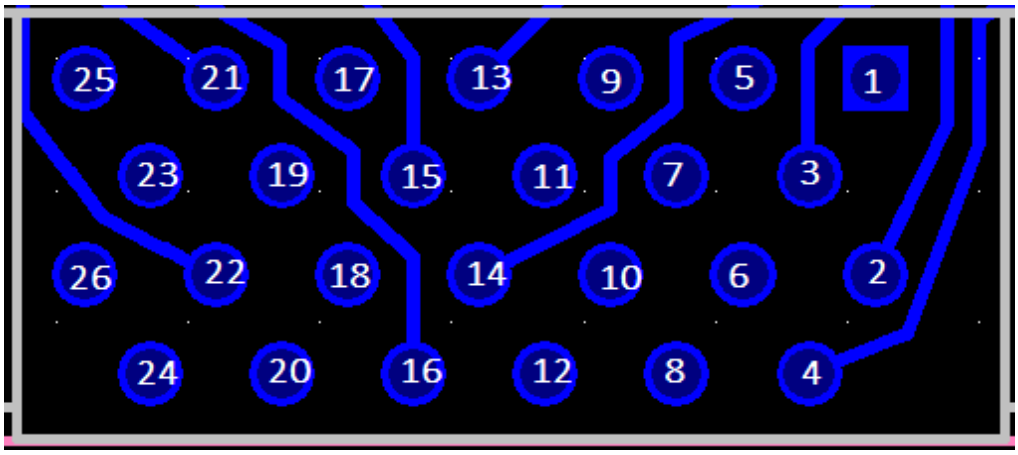
The bottom RHS of this board has the same type of 26pin connector, but must not be connected to the PCIe main board. This connector is labeled RS-422 and contains differential RS422 transceivers. The bottom LHS of the board has a DB9 connector with RS232 serial transceivers.
The top LHS of the board has the SMA connectors for the TTL trigger in, clock in, and clock out. The LHS of the board has 4 synchronized trigger outputs and 4 co-phase clock outputs that are buffered versions of the clock in. The clock in SMA must always be driven, if the internal clock is to be used then the user must connect the clock out SMA to the clock in SMA with a short cable.

TTL bits are captured only during non-averaging modes for AD14-400x2, AD14-500x2, and AD12-2000x2. If averaging, leave the TTL bits unconnected.

14.3 TTL input/output lines for custom firmware

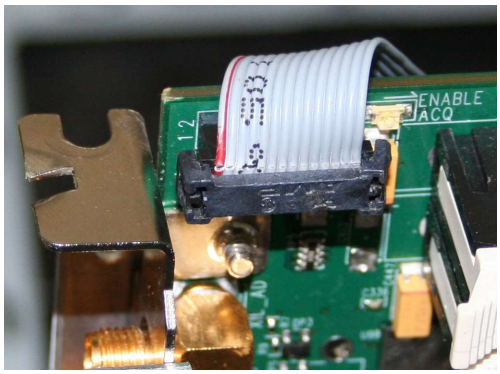
For those wishing to develop custom firmware, listed below are the TTL boardlet pins with their corresponding Xilinx pin.

Connector Pin	Function	Xilinx Pin	SMA Jack Connector
1	Ground		
2	VTrans		
3	Mezz P3	G15	TTL 4
4	Mezz N3	G14	TTL 5
5	Mezz P5	K16	
6	Mezz N5	J16	
7	Mezz N2	J14	
8	Mezz P2	J15	
9	Mezz P1	K18	
10	Mezz N1	K17	
11	Ground		
12	Misc Xil L19	A23	
13	Mezz P7	J19	TTL 2
14	Mezz N7	J18	TTL 3
15	Mezz P6	H17	TTL 1
16	Mezz N6	H16	TTL 0
17	Mezz P4	L19	
18	Mezz N4	L18	
19	Misc Xil L15	NA	
20	Misc Xil H14	NA	
21	Mezz N0	J10	GC_N
22	Mezz P0	K10	Sync Select Record / TTL_TRIG / GC_P
23	Ground		
24	Misc HDR Bank2	AD8	
25	VTrans		
26	VTrans		

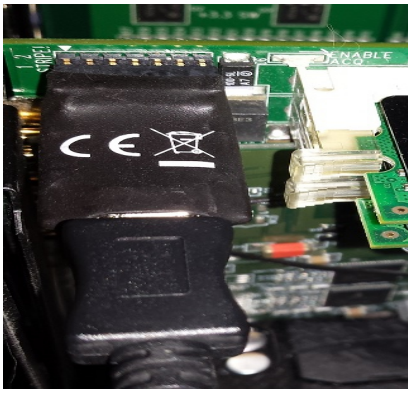


15. APPENDIX – Firmware update using programming cable

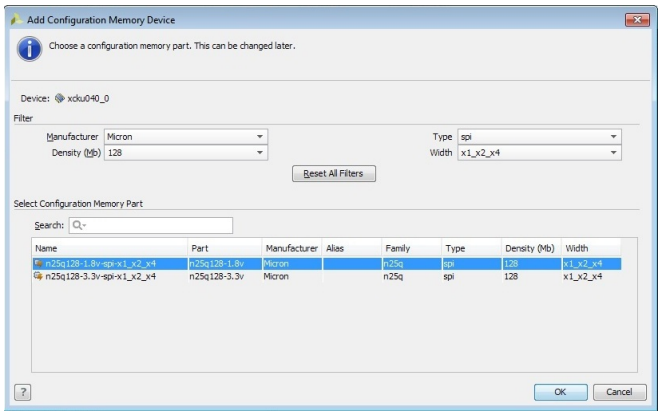
1. With the programming cable connected to the system that can run Vivado hardware manager connect the programming cable's ribbon cable to the programming header on the board. **Ensure that the red stripe on the ribbon cable is connected to pin 1 on the programming header, as shown below. For the digilent cable, ensure the alignment key is facing downwards. Improper connection could damage the board.**
2. Power up the system containing the board to be programmed. Booting the system into the BIOS is the safest thing to do.
3. In Vivado hardware manager, click “open target”, then “auto connect”.
4. If auto connect is successful the part should now show up and display the results of the DDR calibration. To program the prom, it must first be selected. Select “tools → add configuration memory device → XCKU040”
5. Select the n25q128-1.8V-spi-x1_x2_x4 part
6. Select the .bin file that contains the programming information for the prom/FPGA and click OK.

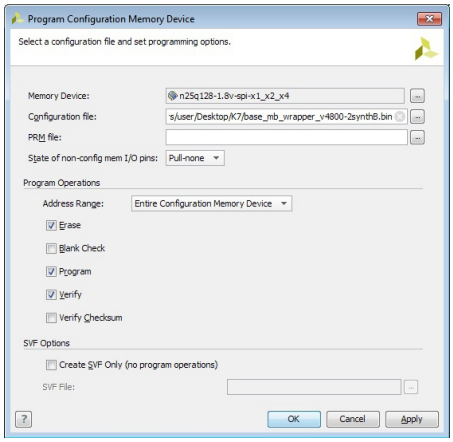


Programming cable orientation (original)



Programming cable orientation (Digilent)





Prom settings on Kintex 7 ultrascale FPGA based boards

16. APPENDIX – ADC Gain/Offset/Bias/Microsynth Calibration

Normally, users should not need to recalibrate unless application specific precise calibration is required. The board specific calibration parameters are now contained in the on board prom on Kintex 7 Ultrascale FPGA based boards. To pull the factory values out of the prom, run the program “promtofile” and this will create a file “config.dat” that contains your board specific calibration. After using Labview to change these values, if appropriate, enter the new values into config.dat” and then run the program “filetoprom” and this will flash the prom with the new values so they will be loaded every time the board is run. In order to wipe the user settings from the prom and restore it to factory calibration, run the program “wipeuserpromsettings”. The program source for “promtofilefactory” is included, but no executable for it is included and it should not be run by users as this will wipe the factory calibration. User calibration will always supercede factory calibration.

--(Serial Number)
ADC_RES= (2 digits, base 10) (required always) (ADC bit resolution)
ADC_CHAN= (2 digits, base 10) (required always) (Number of ADC channels)
DAC_RES= (2 digits, base 10) (required always) (DAC bit resolution)
DAC_CHAN= (2 digits, base 10) (required always) (Number of DAC channels)

ADC3OFF= (2-3 digits, base 16) (00-> no offset, FF-> maximum offset)
ADC3OFFNEG= (1 digit, base 2) (1-> offset is negative, 0-> offset is positive)
ADC3FSR= (3-4 digits, base 16) (Gain adjustment, 000 to 1FF)

ADC3OFF_Q= (2-3 digits, base 16) (00-> no offset, FF-> maximum offset)
ADC2OFFNEG_Q= (1 digit, base 2) (1-> offset is negative, 0-> offset is positive)
ADC3OFFNEG_Q= (1 digit, base 2) (1-> offset is negative, 0-> offset is positive)
ADC3FSR_Q= (3-4 digits, base 16) (Gain adjustment, 000 to 1FF)

ADCGAIN0= (3-4 digits, base 16) (only required for 14-bit boards) (Gain adjustment, 400 to 700)
ADCGAIN1= (3-4 digits, base 16) (only required for 14-bit boards) (Gain adjustment, 400 to 700)
ADCOFFS0= (3 digits, base 16) (only required for 14-bit boards) (Offset adjustment, ? to ?)
ADCOFFS1= (3 digits, base 16) (only required for 14-bit boards) (Offset adjustment, ? to ?)
ADCOFFS2= (3 digits, base 16)
ADCOFFS3= (3 digits, base 16)
ADCBIAS0= (3 digits, base 16) (only required for 14-bit boards) (Bias current adjustment, ? to ?)
ADCBIAS1= (3 digits, base 16) (only required for 14-bit boards) (Bias current adjustment, ? to ?)
ADCBIAS2= (3 digits, base 16)
ADCBIAS3= (3 digits, base 16)

ADC12D2000= (2 digits, base 2)
ADC12D2000_DESI_Q_OFFSET= (3 digits, base 16)(Q offset adjustments to match when in single channel I)
ADC12D2000_DESQ_I_OFFSET= (3 digits, base 16)(I offset adjustments to match when in single channel Q)
ADC12D2000_DESIQ_I_OFFSET= (3 digits, base 16)(I offset adjust when IQ are electrically tied in SCM)
ADC12D2000_DESIQ_Q_OFFSET= (3 digits, base 16)(Q offset adjust when IQ are electrically tied in SCM)

ISLA_ADCX_GAIN_COARSE0= (4 digits, base 16) (only used for AD16-250 boards)
ISLA_ADCX_GAIN_MEDIUM0= (2 digits, base 16) (only used for AD16-250 boards)
ISLA_ADCX_GAIN_FINE0= (2 digits, base 16) (only used for AD16-250 boards)
ISLA_ADCX_GAIN_COARSE1= (4 digits, base 16) (only used for AD16-250 boards)
ISLA_ADCX_GAIN_MEDIUM1= (2 digits, base 16) (only used for AD16-250 boards)
ISLA_ADCX_GAIN_FINE1= (2 digits, base 16) (only used for AD16-250 boards)
(Refer to Figure 18.1 for proper input values)
Where X = 0,1,2,3 the channel numbers on the board

TABLE 6. COARSE GAIN ADJUSTMENT

0x22[3:0] core 0 0x26[3:0] core 1	NOMINAL COARSE GAIN ADJUST (%)
Bit3	+2.8
Bit2	+1.4
Bit1	-2.8
Bit0	-1.4

TABLE 7. MEDIUM AND FINE GAIN ADJUSTMENTS

PARAMETER	0x23[7:0] MEDIUM GAIN	0x24[7:0] FINE GAIN
Steps	256	256
-Full Scale (0x00)	-2%	-0.20%
Mid-Scale (0x80)	0.00%	0.00%
+Full Scale (0xFF)	+2%	+0.2%
Nominal Step Size	0.016%	0.0016%

Figure 18.1 ISLA216P25 Coarse, Medium, and Fine Gain Adjustment
http://www.intersil.com/content/dam/Intersil/documents/fn75/fn7574.pdf

For AD12-2000 boards:

Set to dual channel mode
Adjust ADC3OFF/ADC3OFF_Q so that an open input gives a 50% 2047/2048 distribution.
Adjust ADC3FSR/ADC3FSR_Q so that a 350mVpp / 634mVpp (Rev B / Rev C boards) 10.1MHz input is just short of clipping when the internal 2000MHz clock is used.

Set to single channel mode, channel 0
Adjust ADC12D2000_DESQ_I_OFFSET so that an open input gives a 50% 2047/2048 distribution. Check FFT.

Set to single channel mode, channel 1
Adjust ADC12D2000_DESI_Q_OFFSET so that an open input gives a 50% 2047/2048 distribution. Check FFT.

Set to DESIQ
Adjust DESIQ_I_OFFSET and DESIQ_Q_OFFSET so that an open input gives a 50% 2047/2048 distribution. Check FFT.

For AD14-400 / AD14-500 boards:
Adjust ADCXOFF so that a shorted input gives a midscale value. Adjust the gains and note the range of gains where the midscale value doesn't change.
Then adjust ADCXBIAS so that an open input gives a midscale value.
Then adjust ADCXFSR so that a 750mVpp 1.1MHz input is just short of clipping, FSR should be in range 0x400-0x700 where the midscale value doesn't change when the gain is adjusted slightly.

For AD16-250 boards:
Adjust ADCXOFF so that a shorted input gives a midscale value.
Then adjust ADCXBIAS so that an open input gives a midscale value.
Then adjust ISLA_ADCX_GAIN_COARSEY, ISLA_ADCX_GAIN_MEDIUMY, and ISLA_ADCX_GAIN_FINEY so that all channels are aligned (where X = 0,1,2,3 the channel

numbers on the board, and Y = 0,1 the two cores in each ADC).

.

In LabVIEW, there are controls for adjusting bias, offset, and gains. User must manually input these values into “config.dat” and then run the program “filetoprom” so that it can be used next time the ADCs are calibrated.

17. APPENDIX – How to Modify Your LabVIEW™ Project

18.1 – Modifying VI's

This section is for users who wish to modify their LabVIEW™ VI's from their original configuration.

1. Open the LabVIEW project Ultraview PCIe DAQ.lvproj.
2. Open and modify the VI file of interest.
3. Build the executable which is based on the modified VI. Some permissions may need to be altered depending on operating system and environment. If you are not running as administrator and the LabVIEW project explorer says you do not have access to the required folder, create the folder, go to properties, and change access on everyone group from read only to full control.
4. Copy the executable from its built location to its original location, overwriting the current file.
5. If updating to a new LabVIEW version, the data file lvanlys.dll in the /data directory may need to be copied over to the release as well.

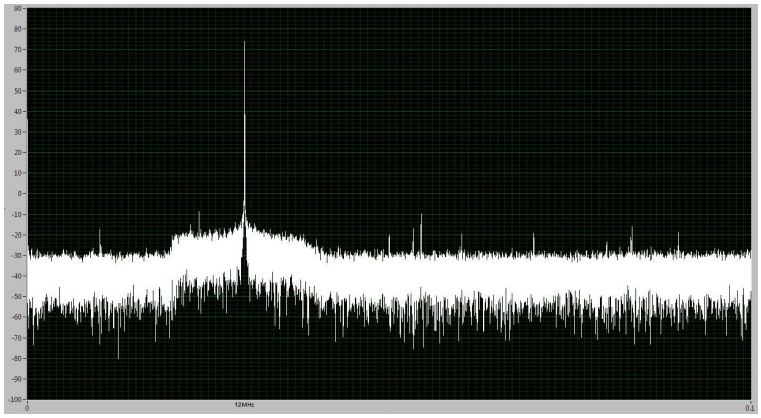
18. APPENDIX – Measurement Data

18.1 AD12-2000 FFT SNR, SFDR

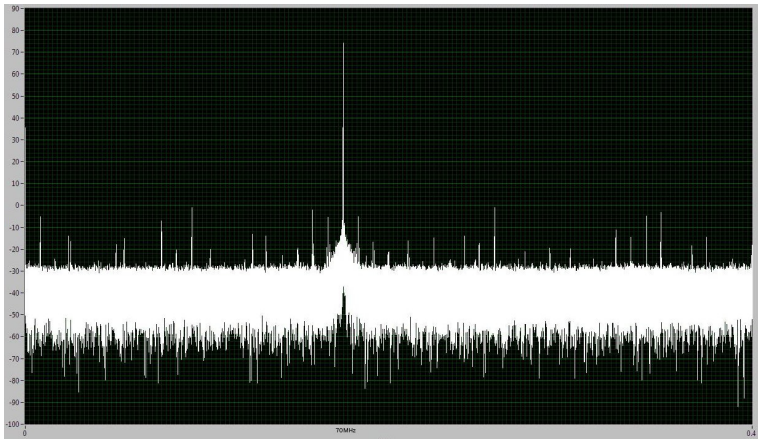
AD12-2000 1MB FFT with a 100MHz Input, 9-15MHz BPF, 500MSPS TBD

AD12-2000 1MB FFT with a 70MHz Input, 70MHz BPF, 500MSPS TBD

18.2 AD14-400 / AD14-500 FFT SNR, SFDR



AD14-400 1MB FFT with a 12MHz Input, 9-15MHz BPF, 400MSPS



AD14-400 1MB FFT with a 70MHz Input, 70MHz BPF, 400MSPS

19. APPENDIX – Data Output Format (uvdma.dat)

Data is in little-endian format.

12/14/16-bit Single Channel | [X][M][N] | where X = channel #, M = sample (2 bytes), N = byte #, 0 is LSB.

[0][0][0]	[0][0][1]	[0][1][0]	[0][1][1]	[0][2][0]	[0][2][1]	…

12/14/16-bit Single Channel (Averaging) | [X][M][N] | where X = channel #, M = sample (4 bytes), N = byte #, 0 is LSB.

[0][0][0]	[0][0][1]	[0][0][2]	[0][0][3]	[0][1][0]	[0][1][1]	[0][1][2]
[0][1][3]	[0][2][0]	[0][2][1]	[0][2][2]	[0][2][3]	…	…

12/14/16-bit Dual Channel | [X][M][N] | where X = channel #, M = sample (2 bytes), N = byte #, 0 is LSB.

[0][0][0]	[0][0][1]	[1][0][0]	[1][0][1]	[0][1][0]	[0][1][1]	[1][1][0]
[1][1][1]	[0][2][0]	[0][2][1]	[1][2][0]	[1][2][1]	…	…

12/14/16-bit Dual Channel (Averaging) | [X][M][N] | where X = channel #, M = sample (4 bytes long), N = byte #, 0 is LSB.

[0][0][0]	[0][0][1]	[0][0][2]	[0][0][3]	[1][0][0]	[1][0][1]	[1][0][2]
[1][0][3]	[0][1][0]	[0][1][1]	[0][1][2]	[0][1][3]	[1][1][0]	[1][1][1]
[1][1][2]	[1][1][3]	…	…	…	…	…

16-bit Quad Channel | [X][M][N] | where X = channel #, M = sample (2 bytes long), N = byte #, 0 is LSB.

[0][0][0]	[0][0][1]	[1][0][0]	[1][0][1]	[2][0][0]	[2][0][1]	[3][0][0]
[3][0][1]	[0][1][0]	[0][1][1]	[1][1][0]	[1][1][1]	[2][1][0]	[2][1][1]
[3][1][0]	[3][1][1]	…	…	…	…	…

16-bit Quad Channel (Averaging) | [X][M][N] | where X = channel #,
M = sample (4 bytes long),
N = byte #, 0 is LSB.

[0][0][0]	[0][0][1]	[0][0][2]	[0][0][3]	[1][0][0]	[1][0][1]	[1][0][2]
[1][0][3]	[2][0][0]	[2][0][1]	[2][0][2]	[2][0][3]	[3][0][0]	[3][0][1]
[3][0][2]	[3][0][3]	…	…	…	…	…

20. CERTIFICATE OF VOLATILITY

AD12-2000x2, AD14-400x2, AD14-500x2, AD16-250x2/x4 SERIES DATA ACQUISITION BOARDS

1) All Ultraview PCIe bus data acquisition boards contain three types of memory storage – Non-Volatile Flash memory for the COTS FPGA configuration data, Volatile SRAM inside the FPGA for elasticity buffering of incoming data, and volatile DRAM, for on-board storage of acquired data until it is transferred to the host system.

2) Only the above referenced SRAM and DRAM ever receive acquired data, and therefore only these components could contain classified data. While the most reliable way to erase this data is to run one of the active purging programs described in 3 below, the vast majority (>99.9%) of the acquired data will alternatively be passively lost from both of these memories within one hour after power is removed from the board, such as by powering off the system. The remainder of the data in the board will disappear well within 24-hours after the board is powered down.

3) The board may alternatively be actively purged of all acquired data by executing the following process:

> Disconnect all analog input cables from the data acquisition board, and run "acquire 8192 –ic". Wait until the board completes the operation and returns with a shell prompt (completion should occur within 16 seconds). All data in both the Volatile SRAM and the volatile DRAM in the board is now overwritten with baseline noise.

4) Do not remove the DIMM memory modules from the board, as the sockets are not designed for repeated insertions, and removal of the DIMM modules may void the warranty. The above procedures will reliably remove data.

21. Known Issues

The following issues have been noted and will be addressed at a later time:

1. There is a LabVIEW™ bug where the checkbox for the first channel does not work. Instead, left-click on the plot icon and deselect “Plot Visible”.

2. There is a LabVIEW™ bug where the plot legend retains all plots in the list, even if there is no data in those channels. The superfluous plots should be ignored.

3. Errors 63 and 66 are TCP/IP read and write failures in LabVIEW™. Ensure that no firewall or network limitations are in effect.

4. Our A/D boards will freeze if the PC sleeps. All power saving features should be disabled.

