USB-1616HS-2

High-Speed Analog Input and Digital I/O

User's Guide

Document Revision 3
December 2012

MEASUREMENT COMPUTING™

Your new Measurement Computing product comes with a fantastic extra —

Management committed to your satisfaction!

Thank you for choosing a Measurement Computing product—and congratulations! You own the finest, and you can now enjoy the protection of the most comprehensive warranties and unmatched phone tech support. It's the embodiment of our mission:

- To provide data acquisition hardware and software that will save time and save money.

Simple installations minimize the time between setting up your system and actually making measurements. We offer quick and simple access to outstanding live FREE technical support to help integrate MCC products into a DAQ system.

Limited Lifetime Warranty: Most MCC products are covered by a limited lifetime warranty against defects in materials or workmanship for the life of the product, to the original purchaser, unless otherwise noted. Any products found to be defective in material or workmanship will be repaired, replaced with same or similar device, or refunded at MCC's discretion. For specific information, please refer to the terms and conditions of sale.

Harsh Environment Program: Any Measurement Computing product that is damaged due to misuse, or any reason, may be eligible for replacement with the same or similar device for 50% of the current list price. I/O boards face some harsh environments, some harsher than the boards are designed to withstand. Contact MCC to determine your product's eligibility for this program.

30 Day Money-Back Guarantee: Any Measurement Computing Corporation product may be returned within 30 days of purchase for a full refund of the price paid for the product being returned. If you are not satisfied, or chose the wrong product by mistake, you do not have to keep it.

These warranties are in lieu of all other warranties, expressed or implied, including any implied warranty of merchantability or fitness for a particular application. The remedies provided herein are the buyer's sole and exclusive remedies. Neither Measurement Computing Corporation, nor its employees shall be liable for any direct or indirect, special, incidental or consequential damage arising from the use of its products, even if Measurement Computing Corporation has been notified in advance of the possibility of such damages.

Notice

Measurement Computing Corporation does not authorize any Measurement Computing Corporation product for use in life support systems and/or devices without prior written consent from Measurement Computing Corporation. Life support devices/systems are devices or systems that, a) are intended for surgical implantation into the body, or b) support or sustain life and whose failure to perform can be reasonably expected to result in injury. Measurement Computing Corporation products are not designed with the components required, and are not subject to the testing required to ensure a level of reliability suitable for the treatment and diagnosis of people.

Table of Contents

Preface
About this User's Guide 5
What you will learn from this user's guide 5
Conventions in this user's guide 5
Where to find more information 5

Chapter 1
Introducing the USB-1616HS-2 6
Overview: USB-1616HS-2 features 6

Chapter 2
Installing the USB-1616HS-2 7
What comes with your USB-1616HS-2 shipment? 7
Hardware 7
Optional components 7
Additional documentation 8
Unpacking the USB-1616HS-2 8
Installing the software 8
Installing the hardware 9
Configuring the hardware 10
Connecting the board for I/O operations 10
Connectors, cables – main I/O connector 10
Screw terminal pin outs 11
DSUB25F expansion connector 12
Cabling 13

Chapter 3
Functional Details 14
USB-1616HS-2 components 14
USB-1616HS-2 block diagram 16
Synchronous I/O – mixing analog, digital, and counter scanning 16
Analog input 17
Analog input scanning 17
Thermocouple input 19
Tips for making accurate temperature measurements 20
Analog output 20
Example: Analog channel scanning of voltage inputs and streaming analog outputs 21
Digital I/O 21
Digital input scanning 21
Digital outputs and pattern generation 22
Triggering 22
Hardware analog triggering 22
Digital triggering 22
Software-based triggering 22
Stop trigger modes 23
Pre-triggering and post-triggering modes 23
Counter inputs 23
Tips for making high-speed counter measurements (> 1 MHz) 24
Mapped channels 24
Counter modes 24
Debounce modes 25
Encoder mode 28

Timer outputs....30
Example: Timer outputs....31
Using multiple USB-1616HS-2s per PC....31
Using detection setpoints for output control....31
What are detection setpoints?....31
Setpoint configuration overview....32
Setpoint configuration....33
Using the setpoint status register....34
Examples of control outputs....34
Detection setpoint details....38
FIRSTPORTC, DAC, or timer update latency....38
Chapter 4
Calibrating the USB-1616HS-2....40
Chapter 5
Specifications....41
Analog input....41
Accuracy....41
Thermocouples....42
Analog outputs....42
Digital input/output....43
Counters....43
Input sequencer....44
Triggering....45
Frequency/pulse generators....45
Power consumption....45
External power....46
USB specifications....46
Environmental....46
Mechanical....46
Signal I/O connectors and pin out....46
Declaration of Conformity....49

About this User's Guide

What you will learn from this user's guide

This user's guide explains how to install, configure, and use the USB-1616HS-2 so that you get the most out of its analog input, digital I/O, and counter/timer I/O features.

This user's guide also refers you to related documents available on our web site and to technical support resources.

Conventions in this user's guide

For more information

Text presented in a box signifies additional information and helpful hints related to the subject matter you are reading.

Caution! Shaded caution statements present information to help you avoid injuring yourself and others, damaging your hardware, or losing your data.

bold text **Bold** text is used for the names of objects on the screen, such as buttons, text boxes, and check boxes.

italic text *Italic* text is used for the names of manuals and help topic titles, and to emphasize a word or phrase.

Where to find more information

For additional information relevant to the operation of your hardware, refer to the *Documents* subdirectory where you installed the MCC DAQ software (C:\Program Files\Measurement Computing\DAQ by default), or search for your device on our website at www.mccdaq.com.

Introducing the USB-1616HS-2

Overview: USB-1616HS-2 features

The USB-1616HS-2 is supported under popular Microsoft® Windows® operating systems. The USB-1616HS-2 board is a multifunction measurement and control board designed for the USB bus.

The USB-1616HS-2 provides either eight differential or 16 single-ended analog inputs with 16-bit resolution. It offers seven software-selectable analog input ranges of ±10 V, ±5 V, ±2 V, ±1 V, ±0.5 V, ±0.2 V, and ±0.1V. You can configure up to eight of the analog inputs as differential thermocouple (TC) inputs.

The USB-1616HS-2 has two 16-bit, 1 MHz analog output channels with an output range of -10 V to +10 V.

The board has 24 high-speed lines of digital I/O, two timer outputs, and four 32-bit counters. It provides up to 4 MHz scanning on all digital input lines[1].

Six banks of removable screw-terminal blocks provide connectivity to the analog input channels, digital I/O lines, counter/timer channels, and analog outputs.

You can operate all analog I/O, digital I/O, and counter/timer I/O synchronously.

[1] Higher rates—up to 12 MHz—are possible depending on the platform and the amount of data being transferred.

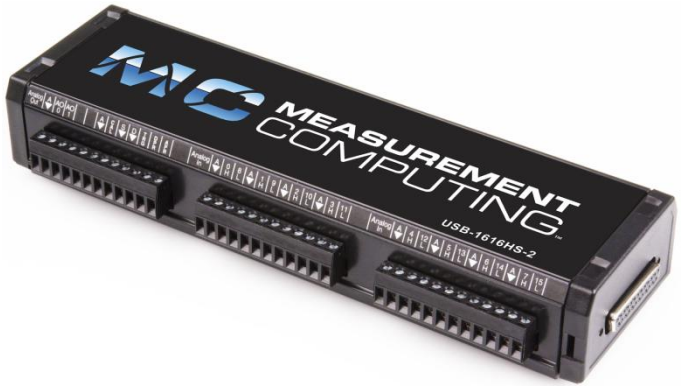
Installing the USB-1616HS-2

What comes with your USB-1616HS-2 shipment?

As you unpack your USB-1616HS-2, verify that the following components are included.

Hardware

- USB-1616HS-2



- USB cable (2-meter length)

- TR-2U power supply and CA-1* line cord
  AC-to-DC power supply cord plugs into the external power connector of the USB-1616HS-2.

* **European customers**: Contact Measurement Computing to order the CA-261 line cord for your region.

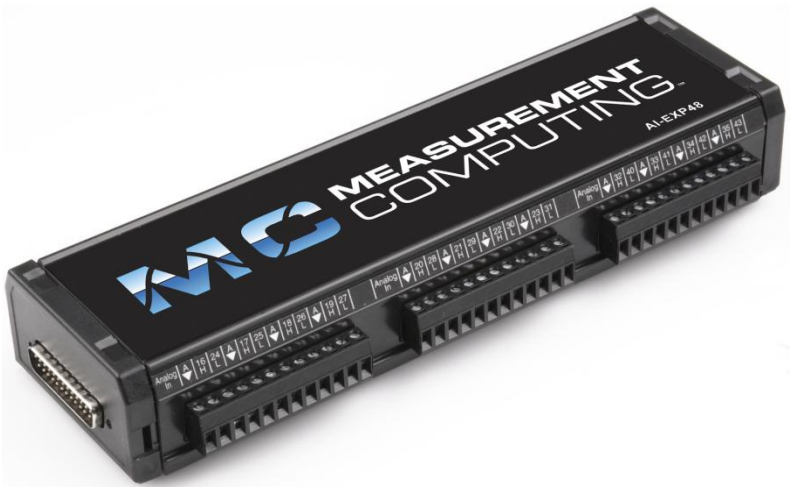
Optional components

Expansion devices and cables and that are compatible with the USB-1616HS-2 and must be ordered separately.

If you ordered any of the following products with your device, they should be included with your shipment.

AI-EXP48

Analog input expansion module adds up to 24 differential or 48 single-ended inputs to the USB-1616HS-2.



CA-96A expansion cable

Expansion cable for connecting to the AI-EXP48 expansion board.

Additional documentation

In addition to this hardware user's guide, you should also receive the *Quick Start Guide* (available in PDF at www.mccdaq.com/PDFmanuals/DAQ-Software-Quick-Start.pdf). This booklet supplies a brief description of the software you received with your USB-1616HS-2 and information regarding installation of that software. Please read this booklet completely before installing any software or hardware.

Unpacking the USB-1616HS-2

As with any electronic device, you should take care while handling to avoid damage from static electricity. Before removing the USB-1616HS-2 from its packaging, ground yourself using a wrist strap or by simply touching the computer chassis or other grounded object to eliminate any stored static charge.

If any components are missing or damaged, notify Measurement Computing Corporation immediately by phone, fax, or e-mail:

- Phone: 508-946-5100 and follow the instructions for reaching Tech Support
- Fax: 508-946-9500 to the attention of Tech Support
- Email: techsupport@mccdaq.com

Installing the software

Refer to the *Quick Start Guide* for instructions on installing the software on the *Measurement Computing Data Acquisition Software CD*. This booklet is available in PDF at www.mccdaq.com/PDFmanuals/DAQ-Software-Quick-Start.pdf.

We recommend that you download the latest Windows Update onto your computer before installing and operating the USB-1616HS-2.

Installing the hardware

To connect the USB-1616HS-2 to your system, turn your computer on, and then do the following:

1. Connect signal lines to the USB-1616HS-2's removable screw terminal blocks.
   - Connect voltage signals as single-ended or differential connections (see Figure 1).
   - Connect thermocouple signals as differential connections (see Figure 1). The negative (typically, the red) thermocouple wire connects to the channel's LO connector, and the other color wire connects to the channel's HI connector.

     Always use differential input mode for thermocouple connections.

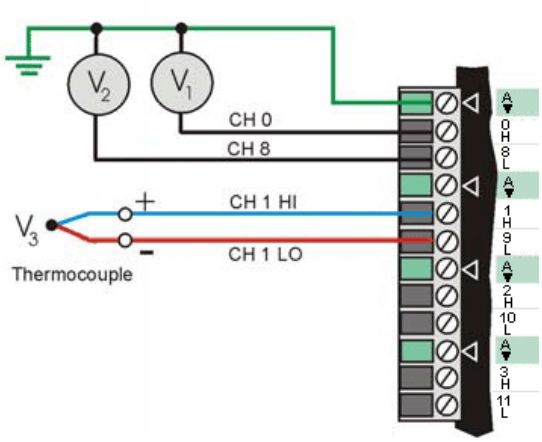


Figure 1. Single-ended voltage connections (V_1 and V_2) and differential thermocouple connections (V_3)

2. If you are using an AI-EXP48 expansion device, connect it to the USB-1616HS-2. Make sure you do not connect the AI-EXP48 to a live USB-1616HS-2. If the USB cable is connected to the computer, unplug it before you connect the AI-EXP48.
3. If you are using the TR-2U external supply (sold separately), connect the power supply to the USB-1616HS-2's external power connector, and plug the other end into a power outlet.

   The TR-2U is optional, but can be used in any scenario. You may need a TR-2U power supply if the USB port does not provide enough power for your USB-1616HS-2 application.

   The USB-1616HS-2 requires 3000 mW by itself, and 3400 mW when connected to the AI-EXP48.

   By USB2 standards, USB 2.0 ports are required to provide at least 2500 mW.
4. Connect the USB cable to the USB-1616HS-2 USB connector and to a USB port on your computer. A USB2.0 port is recommended—connecting to a USB1.1 port results in lower performance.

   When you connect the USB-1616HS-2 for the first time, a **Found New Hardware** message opens as the USB-1616HS-2 is detected. When the message closes, the installation is complete.

The **power LED** (bottom LED) blinks during device detection and initialization, and then remains solid if properly detected. If not, check if the USB-1616HS-2 has sufficient power. When the board is first powered on, there is usually a momentary delay before the power LED begins to blink, or come on solid.

Caution! Do not disconnect **any** device from the USB bus while the computer is communicating with the USB-1616HS-2, or you may lose data and/or your ability to communicate with the USB-1616HS-2.

Configuring the hardware

All hardware configuration options on the USB-1616HS-2 are software-controlled. You can select some of the configuration options using *Insta*Cal, such as the analog input configuration (16 single-ended or 8 differential channels), and the edge used for pacing when using an external clock. When measuring from thermocouples, make sure you configure the channels for differential mode.

Once selected, any program that uses the Universal Library initializes the hardware according to these selections.

Caution! Turn off power to all devices connected to the system before making connections. Electrical shock or damage to equipment can result even under low-voltage conditions.

Information on signal connections

General information regarding signal connection and configuration is available in the *Guide to Signal Connections*. This document is available on our web site at www.mccdaq.com/signals/signals.pdf.

Caution! Always handle components carefully, and never touch connector terminals or circuit components unless you are following ESD guidelines in an appropriate ESD-controlled area. These guidelines include using properly-grounded mats and wrist straps, ESD bags and cartons, and related procedures.

Avoid touching board surfaces and onboard components. Only handle boards by their edges. Make sure the USB-1616HS-2 does not come into contact with foreign elements such as oils, water, and industrial particulate.

The discharge of static electricity can damage some electronic components. Semiconductor devices are especially susceptible to ESD damage.

Connecting the board for I/O operations

Connectors, cables – main I/O connector

The following table lists the board connectors, applicable cables, and compatible accessory products for the USB-1616HS-2.

Main connector specifications

Main connectors	Six banks of removable screw-terminal blocks
Expansion connector	25-pin DSUB, female (DSUB25F)
Compatible cable for the 25-pin expansion connector	CA-96A
Compatible accessory product for the 25-pin expansion connector	AI-EXP48 expansion board with screw terminals (can connect to the USB-1616HS-2 directly, or with the CA-96A cable)

Screw terminal pinout

USB-1616HS-2 screw terminal pinout – single-ended connections

Analog Out	Analog common (A▼)		Port A	Digital common (D▼)	**DIG-Tmr I/O**
	Analog output 0 (AO0)			FIRSTPORTA Bit 0 (A0)	
	Analog output 1 (AO1)			FIRSTPORTA Bit 1 (A1)	
	NC			FIRSTPORTA Bit 2 (A2)	
	NC			FIRSTPORTA Bit 3 (A3)	
	Analog common (A▼)			FIRSTPORTA Bit 4 (A4)	
	CAL (Reserved for self-calibration)			FIRSTPORTA Bit 5 (A5)	
	Signal ground (S▼)			FIRSTPORTA Bit 6 (A6)	
	Digital common (D▼)			FIRSTPORTA Bit 7 (A7)	
	TTL trigger (TRG)			Digital common (D▼)	
	Output scan clock I/O (DPR)			Timer 0 (T0)	
	Input scan clock I/O (APR)			Timer 1 (T1)	
Analog In	Analog common (A▼)		Port B	Digital common (D▼)	**Dig-Ctr I/O**
	CH 0 (0H)			FIRSTPORTB Bit 0 (B0)	
	CH 8 (8L)			FIRSTPORTB Bit 1 (B1)	
	Analog common (A▼)			FIRSTPORTB Bit 2 (B2)	
	CH 1 (1H)			FIRSTPORTB Bit 3 (B3)	
	CH 9 (9L)			FIRSTPORTB Bit 4 (B4)	
	Analog common (A▼)			FIRSTPORTB Bit 5 (B5)	
	CH 2 (2H)			FIRSTPORTB Bit 6 (B6)	
	CH 10 (10L)			FIRSTPORTB Bit 7 (B7)	
	Analog common (A▼)			Digital common (D▼)	
	CH 3 (3H)			Counter 0 (CT0)	
	CH 11 (11L)			Counter 1 (CT1)	
Analog In	Analog common (A▼)		Port C	Digital common (D▼)	**Dig-Ctr I/O**
	CH 4 (4H)			FIRSTPORTC Bit 0 (C0)	
	CH 12 (12L)			FIRSTPORTC Bit 1 (C1)	
	Analog common (A▼)			FIRSTPORTC Bit 2 (C2)	
	CH 5 (5H)			FIRSTPORTC Bit 3 (C3)	
	CH 13 (13L)			FIRSTPORTC Bit 4 (C4)	
	Analog common (A▼)			FIRSTPORTC Bit 5 (C5)	
	CH 6 (6H)			FIRSTPORTC Bit 6 (C6)	
	CH 14 (14L)			FIRSTPORTC Bit 7 (C7)	
	Analog common (A▼)			Digital common (D▼)	
	CH 7 (7H)			Counter 2 (CT2)	
	CH 15 (15L)			Counter 3 (CT3)	

USB-1616HS-2 screw terminal pinout – differential connections

Analog Out	Analog common (A▾)		Port A	Digital common (D▾)	**DIG-Tmr I/O**
	Analog output 0 (AO0)			FIRSTPORTA Bit 0 (A0)	
	Analog output 1 (AO1)			FIRSTPORTA Bit 1 (A1)	
	NC			FIRSTPORTA Bit 2 (A2)	
	NC			FIRSTPORTA Bit 3 (A3)	
	Analog common (A▾)			FIRSTPORTA Bit 4 (A4)	
	CAL (Reserved for self-calibration)			FIRSTPORTA Bit 5 (A5)	
	Signal ground (S▾)			FIRSTPORTA Bit 6 (A6)	
	Digital common (D▾)			FIRSTPORTA Bit 7 (A7)	
	TTL trigger (TRG)			Digital common (D▾)	
	Output scan clock I/O (DPR)			Timer 0 (T0)	
	Input scan clock I/O (APR)			Timer 1 (T1)	
Analog In	Analog common (A▾)		Port B	Digital common (D▾)	**Dig-Ctr I/O**
	CH 0 HI (0H)			FIRSTPORTB Bit 0 (B0)	
	CH 0 LO (8L)			FIRSTPORTB Bit 1 (B1)	
	Analog common (A▾)			FIRSTPORTB Bit 2 (B2)	
	CH 1 HI (1H)			FIRSTPORTB Bit 3 (B3)	
	CH 1 LO (9L)			FIRSTPORTB Bit 4 (B4)	
	Analog common (A▾)			FIRSTPORTB Bit 5 (B5)	
	CH 2 HI (2H)			FIRSTPORTB Bit 6 (B6)	
	CH 2 LO (10L)			FIRSTPORTB Bit 7 (B7)	
	Analog common (A▾)			Digital common (D▾)	
	CH 3 HI (3H)			Counter 0 (CT0)	
	CH 3 LO (11L)			Counter 1 (CT1)	
Analog In	Analog common (A▾)		Port C	Digital common (D▾)	**Dig-Ctr I/O**
	CH 4 HI (4H)			FIRSTPORTC Bit 0 (C0)	
	CH 4 LO (12L)			FIRSTPORTC Bit 1 (C1)	
	Analog common (A▾)			FIRSTPORTC Bit 2 (C2)	
	CH 5 HI (5H)			FIRSTPORTC Bit 3 (C3)	
	CH 5 LO (13L)			FIRSTPORTC Bit 4 (C4)	
	Analog common (A▾)			FIRSTPORTC Bit 5 (C5)	
	CH 6 HI (6H)			FIRSTPORTC Bit 6 (C6)	
	CH 6 LO (14L)			FIRSTPORTC Bit 7 (C7)	
	Analog common (A▾)			Digital common (D▾)	
	CH 7 HI (7H)			Counter 2 (CT2)	
	CH 7 LO (15L)			Counter 3 (CT3)	

DSUB25F expansion connector

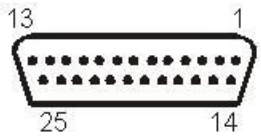


Figure 2. DSUB25 expansion connector pinout

Cabling

Use a CA-96A 25-pin expansion cable (CA-96A expansion cable) to connect to the USB-1616HS-2's 25-pin expansion connector.

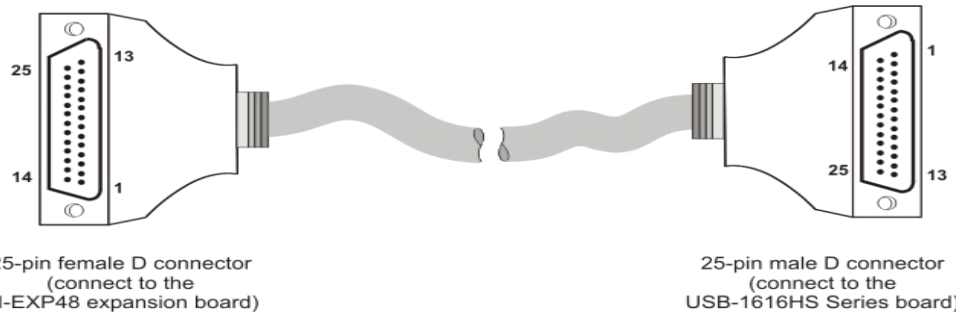


Figure 3. CA-96A expansion cable

Functional Details

This chapter contains detailed information on all of the features available from the board, including:

- a diagram and explanations of physical board components
- a functional block diagram
- information on how to use the signals generated by the board
- diagrams of signals using default or conventional board settings

USB-1616HS-2 components

These USB-1616HS-2 components are shown in Figure 4.

- Six removable screw terminal blocks
- One USB port
- One external power connector
- One 25-pin expansion connector
- Two LED indicators ("Active" and "Power")

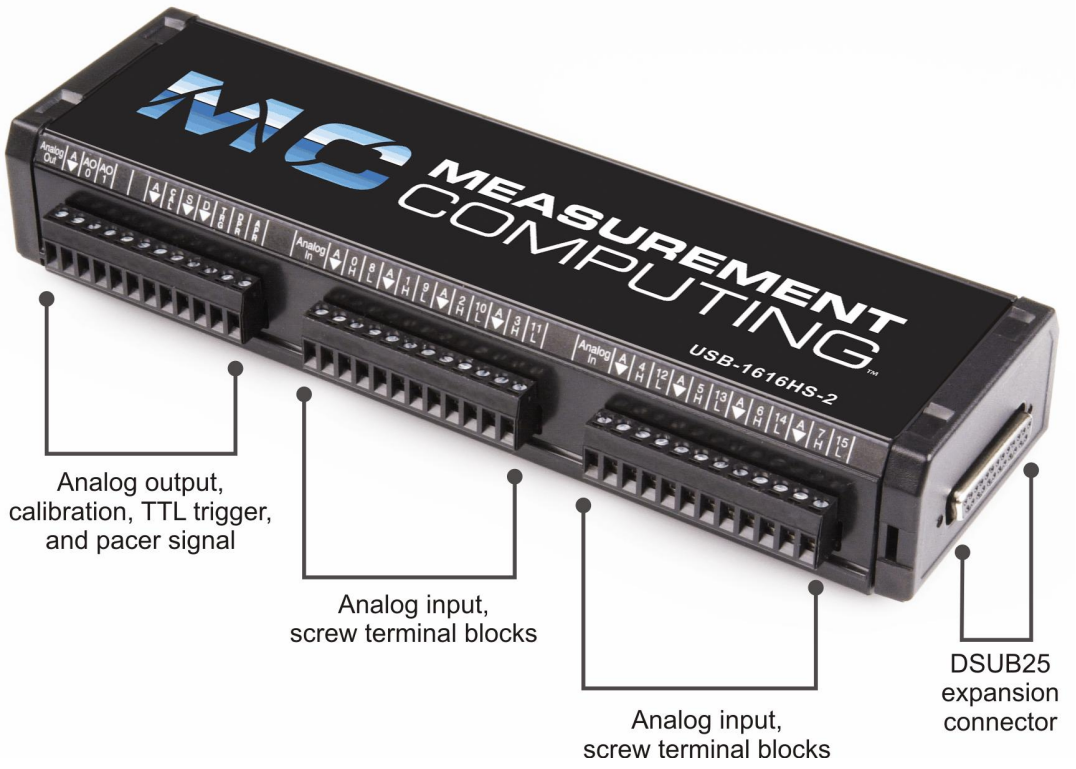


Figure 4. USB-1616HS-2 components – front view

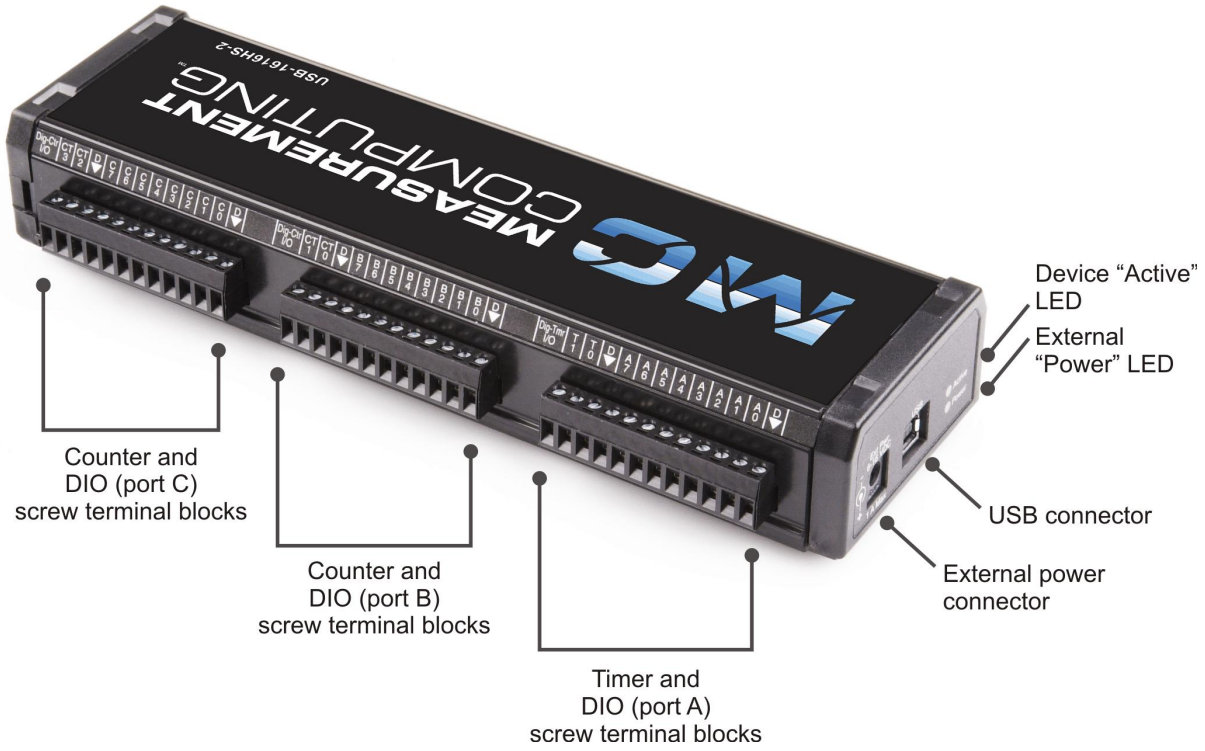


Figure 5. USB-1616HS-2 components – rear view

External power connector

Although the USB-1616HS-2 is powered by a USB port on a host PC, an external power connector may also be required to provide sufficient power for the USB-1616HS-2.

Connect the optional TR-2U power supply to the external power supply connector. This power supply provides 9 VDC, 1 A power to the USB-1616HS-2.

USB-1616HS-2 block diagram

Figure 6 shows a simplified block diagram of the USB-1616HS-2. This board provides all of the functional elements shown in the figure.

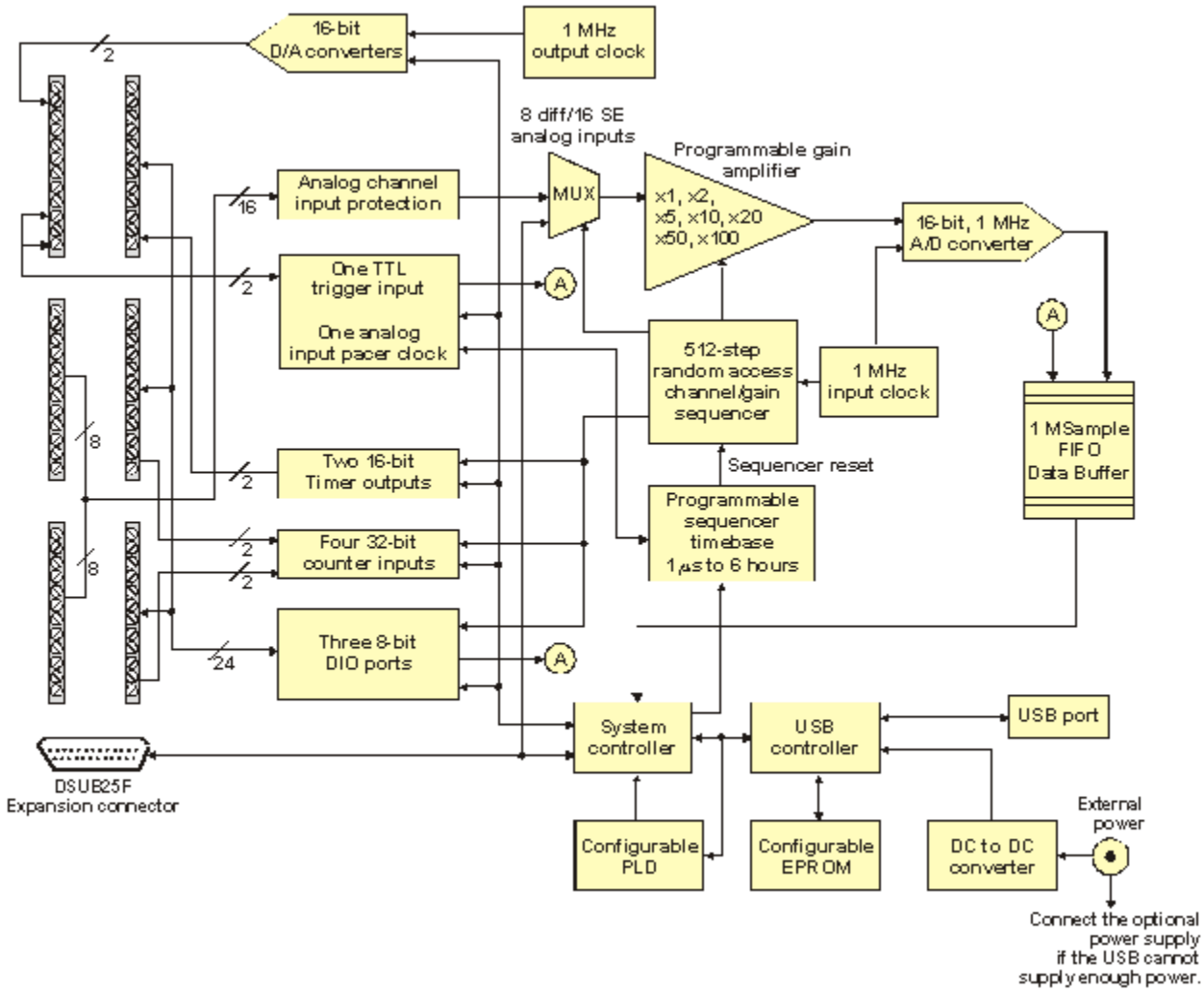


Figure 6. USB-1616HS-2 functional block diagram

Synchronous I/O – mixing analog, digital, and counter scanning

The USB-1616HS-2 can read analog, digital, and counter inputs, while generating up to two analog outputs and digital pattern outputs at the same time. Digital and counter inputs do not affect the overall A/D rate because these inputs use no time slot in the scanning sequencer.

For example, one analog input channel can be scanned at the full 1 MHz A/D rate along with digital and counter input channels. Each analog channel can have a different gain, and counter and digital channels do not need additional scanning bandwidth as long as there is at least one analog channel in the scan group.

Digital input channel sampling is not done during the "dead time" of the scan period where no analog sampling is being done either.

Analog input

The USB-1616HS-2 has a 16-bit, 1-MHz A/D coupled with 16 single-ended, or eight differential analog inputs. Seven software programmable ranges provide inputs from ±10 V to ±100 mV full scale.

Analog input scanning

The USB-1616HS-2 has several scanning modes to address various applications. You can load the 512-location scan buffer with any combination of analog input channels. All analog input channels in the scan buffer are measured sequentially at 1 µs per channel by default.

For example, in the fastest mode, with ADC settling time set to 1 µs, a single analog channel can be scanned continuously at 1 MS/s; two analog channels can be scanned at 500 kS/s each; 16 analog input channels can be scanned at 62.5 kS/s.

Settling time

For most applications, leave the settling time at its default of 1 µs.

However, if you are scanning multiple channels, and one or more channels are connected to a high-impedance source, you may get better results by increasing the settling time. Remember that increasing the settling time reduces the maximum acquisition rate.

You can set the settling time to 1 µs, 5 µs, 10 µs, or 1 ms.

Example: Analog channel scanning of voltage inputs

Figure 7 shows a simple acquisition. The scan is programmed pre-acquisition and is made up of six analog channels (Ch0, Ch1, Ch3, Ch4, Ch6, and Ch7). Each of these analog channels can have a different gain. The acquisition is triggered and the samples stream to the PC. Each analog channel requires one microsecond of scan time—therefore the scan period can be no shorter than 6 µs for this example. The scan period can be made much longer than 6 µs—up to 1 s. The maximum scan frequency is one divided by 6 µs, or 166,666 Hz.

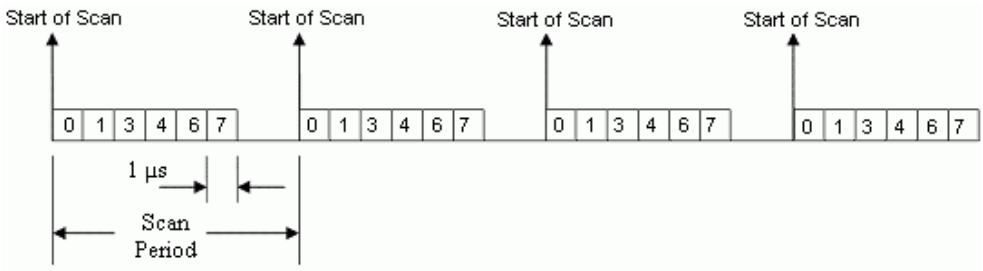


Figure 7. Analog channel scan of voltage inputs example

Example: Analog channel scanning of voltage and temperature inputs

Figure 8 shows a programmed pre-acquisition scan made up of six analog channels (Ch0, Ch1, Ch5, Ch11, Ch12, Ch13). Each of these analog channels can have a different gain. You can program channels 0 and 1 to directly measure TCs.

In this mode, oversampling is programmable up to 16384 oversamples per channel in the scan group. When oversampling is applied, it is applied to all analog channels in the scan group, including temperature and voltage channels. Digital channels are not oversampled.

If you want 256 oversamples, then each analog channel in the scan group will take 256 µs, and the returned 16-bit value represents an average of 256 consecutive 1 µs samples of that channel. The acquisition is triggered and 16-bit values—each representing an average of 256—stream to the PC via the USB cable. Since two of the channels in the scan group are temperature channels, you need the acquisition engine to read a cold-junction-compensation (CJC) temperature every scan.

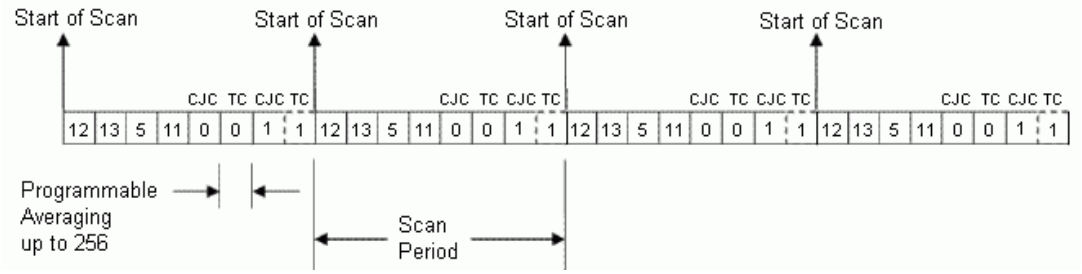


Figure 8. Analog channel scanning of voltage and temperature inputs example

Since the targeted number of oversamples is 256 in this example, each analog channel in the scan group requires 256 microseconds to return one 16-bit value. The oversampling is also done for CJC temperature measurement channels, making the minimum scan period for this example 7 x 256 µs, or 1792 µs. The maximum scan frequency is the inverse of this number, 558 Hz.

For accurate measurements, you must associate TC and CJC channels properly

The TC channels must immediately follow their associated CJC channels in the channel array. For accurate TC readings, associate CJC0 with TC0, CJC1 with TC1 and TC2, CJC2 with TC3, CJC3 with TC4, CJC4 with TC5 and TC6, and CJC5 with TC7.

When the AI-EXP48 module is connected to the USB-1616HS-2, associate CJC6 with TC8 through TC11, CJC7 with TC12 through TC15, CJC8 with TC16 through TC19, CJC9 with TC20 through TC23, CJC10 with TC24 through TC27, and CJC11 with TC28 through TC31.

Example: Analog and digital scanning, once per scan mode

The scan is programmed pre-acquisition and is made up of six analog channels (Ch0, Ch2, Ch5, Ch11, Ch13, Ch15) and four digital channels (16-bits of digital IO, three counter inputs.) Each of the analog channels can have a different gain.

The acquisition is triggered and the samples stream to the PC via the USB cable. Each analog channel requires one microsecond of scan time. Therefore, the scan period can be no shorter than 6 µs for this example. All of the digital channels are sampled at the start of scan and do not require additional scanning bandwidth as long as there is at least one analog channel in the scan group. The scan period can be made much longer than 6 µs, up to 1 second. The maximum scan frequency is one divided by 6 µs or 166,666 Hz.

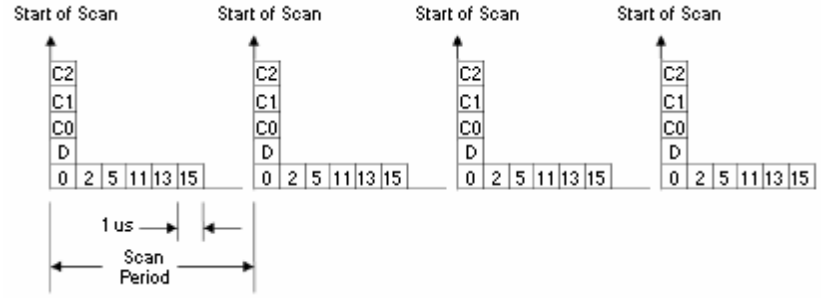


Figure 9. Analog and digital scanning, once per scan mode example

The counter channels may return only the lower 16-bits of count value if that is sufficient for the application. They could also return the full 32-bit result if necessary. Similarly, the digital input channel could be the full 24 bits if desired or only eight bits if that is sufficient. If the three counter channels are all returning 32-bit values and the digital input channel is returning a 16-bit value, then 13 samples are being returned to the PC every scan period, with each sample being 16-bits. The 32-bit counter channels are divided into two 16-bit samples—one for the low word, and the other for the high word. If the maximum scan frequency is 166,666 Hz, then the data bandwidth streaming into the PC is 2.167 MS/s. Some slower PCs may have a problem with data bandwidths greater than 6 MS/s.

The USB-1616HS-2 has an onboard 1 MS buffer for acquired data.

Example: Sampling digital inputs for every analog sample in a scan group

The scan is programmed pre-acquisition and is made up of six analog channels (Ch0, Ch2, Ch5, Ch11, Ch13, Ch15) and four digital channels (16-bits of digital input, three counter inputs.) Each of the analog channels can have a different gain.

The acquisition is triggered and the samples stream to the PC via the USB cable. Each analog channel requires one microsecond of scan time therefore the scan period can be no shorter than 6 µs for this example. All of the digital channels are sampled at the start of scan and do not require additional scanning bandwidth as long as there is at least one analog channel in the scan group. The 16-bits of digital input are sampled for every analog sample in the scan group. This allows up to 1 MHz digital input sampling while the 1 MHz analog sampling bandwidth is aggregated across many analog input channels.

The scan period can be made much longer than 6 µs—up to 1 second. The maximum scan frequency is one divided by 6 µs, or 166,666 Hz. Note that digital input channel sampling is not done during the "dead time" of the scan period where no analog sampling is being done either.

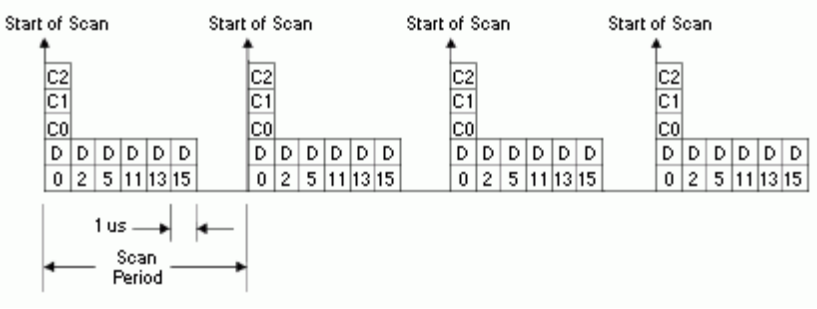


Figure 10. Analog and digital scanning, once per scan mode example

If the three counter channels are all returning 32-bit values and the digital input channel is returning a 1-bit value, then 18 samples are returned to the PC every scan period, with each sample being 16-bits. Each 32-bit counter channel is divided into two 16-bit samples—one for the low word and the other for the high word. If the maximum scan frequency is 166,666 Hz, then the data bandwidth streaming into the PC is 3 MS/s. Some slower PCs may have a problem with data bandwidths greater than 6 MS/s.

The USB-1616HS-2 has an onboard 1 MS buffer for acquired data.

Thermocouple input

You can configure up to eight analog inputs in differential mode on the USB-1616HS-2 to accept a thermocouple (TC) input. Built-in cold-junction sensors are provided for each of the screw-terminal connectors, and any TC type can be attached to any of the eight thermocouple channels.

When measuring TCs, the USB-1616HS-2 can operate in an averaging mode, taking multiple readings on each channel, applying digital filtering and cold-junction compensation, and then converting the readings to temperature.

As a result, the USB-1616HS-2 measures channels with TCs attached at a rate from 50 Hz to 10 kHz, depending on how much over-sampling is selected.

Additionally, a rejection frequency can be specified in which over sampling occurs during one cycle of either 50 Hz or 60 Hz, providing a high level of 50 Hz or 60 Hz rejection.

The USB-1616HS-2 does not have open thermocouple detection.

Tips for making accurate temperature measurements

- Set the rejection frequency to equal the line frequency.
- Warm up the USB-1616HS-2 for 60 minutes—including TC wires—so that it is thermally stabilized. This warm-up time enables the CJC thermistors to more accurately measure the junction at the terminal block.
- Make sure the surrounding environment is thermally stabilized and ideally around 20 °C to 30 °C. If the device's ambient temperature is changing due to a local heating or cooling source, then the TC junction temperature may be changing and the CJC thermistor will have a larger error.
- Use small-diameter, *instrument-grade* TC wire. Small diameter TC wire has less effect on the TC junction at the terminal block because less heat is transferred from the ambient environment to the junction.
- Use shielded TC wire (see "Shielding" below) with the shield connected to analog common to reduce noise. The USB-1616HS-2 has several analog commons on the screw terminals.

  You can also minimize the effect of noise by averaging readings (see "Averaging" below), or combining both shielding and averaging.

  Refer to "Screw terminal pinout" section starting on page 11 for the locations of these analog common screw terminals.
- Make sure the USB-1616HS-2 is mounted on a flat surface.
- Be careful to avoid loading down the digital outputs too heavily (>1 mA). Heavy load down causes significant heat generation inside the unit and increase the CJC thermistor error.

Shielding

Use shielded TC wire with the shield connected to analog common to reduce noise.

The USB-1616HS-2 several analog common screw terminals (see "Connecting the board for I/O operations" starting on page 10). You can connect the shield of a shielded thermocouple to one of the analog commons. When this connection is made, leave the shield at the other end of the thermocouple unconnected.

Caution! Connecting the shield to common at both ends results in a ground loop.

Averaging

Certain acquisition programs apply *averaging* after several samples have been collected. Depending on the nature of the noise, averaging can reduce noise by the square root of the number of averaged samples.

Although averaging can be effective, it suffers from several drawbacks:

- Noise in measurements only decreases as the square root of the number of measurements—reducing RMS noise significantly may require many samples. Thus, averaging is suited to low-speed applications that can provide many samples.
- Only random noise is reduced or eliminated by averaging. Averaging does not reduce or eliminate periodic signals.

Analog output

The USB-1616HS-2 has two 16-bit, 1 MHz analog output channels. Analog outputs can be updated at a maximum rate of 1 MHz.

The channels have an output range of -10 V to +10 V. Each D/A can continuously output a waveform. In addition, a program can asynchronously output a value to any of the D/A channels for non-waveform applications, assuming that the D/A is not already being used in the waveform output mode.

When used to generate waveforms, you can clock the D/As in several different modes.

- **Internal output scan clock**: The onboard programmable clock can generate updates ranging from 1 Hz to 1 MHz.
- **External output scan clock**: A user-supplied external output scan clock at the DPR screw terminal.
- **External input scan clock**: A user-supplied external input scan clock at APR can pace both the D/A and the analog input.
- **Internal input scan clock**: The internal ADC scan clock.

Example: Analog channel scanning of voltage inputs and streaming analog outputs

The example shown in Figure 11 adds two DACs and a 16-bit digital pattern output to the example presented in Figure 7 on page 17.

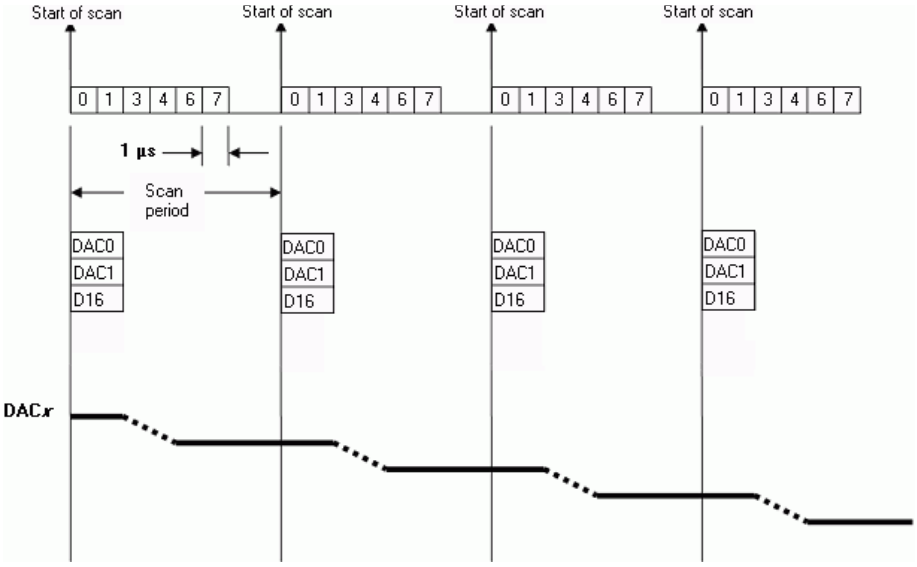


Figure 11. Analog channel scan of voltage inputs and streaming analog outputs example

This example updates all DACs and the 16-bits of digital I/O. These updates happen at the same time as the output scan clock. All DACs and the 16-bits of pattern digital output are updated at the beginning of each scan.

Due to the time it takes to shift the digital data out to the DACs, plus the actual settling time of the digital-to-analog conversion, the DACs actually take up to 4 µs after the start of scan to settle on the updated value.

The data for the DACs and pattern digital output comes from a PC-based buffer. The data is streamed across the USB2 bus to the USB-1616HS-2.

You can update the DACs and pattern digital output with the output scan clock—either internally-generated or externally-applied. In this scenario, the acquisition input scans are not synchronized to the analog outputs or pattern digital outputs.

You can also synchronize everything—input scans, DACs, pattern digital outputs—to one clock, which is either internally-generated or externally-applied.

Digital I/O

Twenty-four TTL-level digital I/O lines are included in each USB-1616HS-2. You can program digital I/O in 8-bit groups as either inputs or outputs and scan them in several modes (see "Digital input scanning" below). You can access input ports asynchronously from the PC at any time, including when a scanned acquisition is occurring.

Digital input scanning

Digital input ports can be read asynchronously before, during, or after an analog input scan.

Digital input ports can be part of the scan group and *scanned along with analog input channels*. Two synchronous modes are supported when digital inputs are scanned along with analog inputs. Refer to "Example 4: Sampling digital inputs for every analog sample in a scan group" on page 11 for more information.

In both modes, adding digital input scans has no affect on the analog scan rate limitations.

If no analog inputs are being scanned, the digital inputs can sustain rates up to 4 MHz.

Higher rates—up to 12 MHz—are possible depending on the platform and the amount of data being transferred.

Digital outputs and pattern generation

Digital outputs can be updated asynchronously at anytime before, during, or after an acquisition. You can use two of the 8-bit ports to generate a digital pattern at up to 4 MHz. The USB-1616HS-2 supports digital pattern generation. The digital pattern can be read from PC RAM.

Higher rates—up to 12 MHz—are possible depending on the platform and the amount of data being transferred.

Digital pattern generation is clocked using an internal clock. The onboard programmable clock generates updates ranging from once every 1 second to 1 MHz, independent of any acquisition rate.

Triggering

Triggering can be the most critical aspect of a data acquisition application. The USB-1616HS-2 supports the following trigger modes to accommodate certain measurement situations.

Hardware analog triggering

The USB-1616HS-2 uses true analog triggering in which the trigger level you program sets an analog DAC, which is then compared in hardware to the analog input level on the selected channel. This guarantees an analog trigger latency that is less than 1 µs.

You can select any analog channel as the trigger channel, but the selected channel must be the first channel in the scan. You can program the trigger level, the rising or falling edge to trigger on, and hysteresis.

A note on the hardware analog level trigger and comparator change state

When analog input voltage starts near the trigger level, and you are performing a rising or falling hardware analog level trigger, the analog level comparator may have already tripped before the sweep was enabled. If this is the case, the circuit waits for the comparator to change state. However, since the comparator has already changed state, the circuit does not see the transition.

To resolve this problem, do the following:

1. Set the analog level trigger to the threshold you want.
2. Apply an analog input signal that is *more than* 2.5% of the full-scale range *away from the desired threshold*. This ensures that the comparator is in the proper state at the beginning of the acquisition.
3. Bring the analog input signal toward the desired threshold. When the input signal is at the threshold (± some tolerance) the sweep will be triggered.
4. Before re-arming the trigger, again move the analog input signal to a level that is more than 2.5% of the full-scale range *away from* the desired threshold.

For example, if you are using the ±2 V full-scale range (gain = 5), and you want to trigger at +1 V on the rising edge, set the analog input voltage to a start value that is less than +0.9 V (1 V – (2 V * 2 * 2.5%)).

Digital triggering

A separate digital trigger input line is provided (TRG), allowing TTL-level triggering with latencies guaranteed to be less than 1 µs. You can program both of the logic levels (1 or 0) and the rising or falling edge for the discrete digital trigger input.

Software-based triggering

The three software-based trigger modes differ from hardware analog triggering and digital triggering because the readings—analog, digital, or counter—are checked by the PC in order to detect the trigger event.

Analog triggering

You can select any analog channel as the trigger channel. You can program the trigger level, the rising or falling edge to trigger on, and hysteresis.

Pattern triggering

You can select any scanned digital input channel pattern to trigger an acquisition, including the ability to mask or ignore specific bits.

Counter triggering

You can program triggering to occur when one of the counters meets or exceeds a set value, or is within a range of values. You can program any of the included counter channels as the trigger source.

Software-based triggering usually results in a long period of inactivity between the trigger condition being detected and the data being acquired. However, the USB-1616HS-2 avoids this situation by using pre-trigger data. When software-based-triggering is used, and the PC detects the trigger condition—which may be thousands of readings after the actual occurrence of the signal—the USB-1616HS-2 driver automatically looks back to the location in memory where the actual trigger-causing measurement occurred, and presents the acquired data that begins at the point where the trigger-causing measurement occurs. The maximum inactive period in this mode equals one scan period.

Stop trigger modes

You can use any of the software trigger modes explained previously to stop an acquisition.

For example, you can program an acquisition to begin on one event—such as a voltage level—and then stop on another event—such as a digital pattern.

Pre-triggering and post-triggering modes

The USB-1616HS-2 supports four modes of pre-triggering and post-triggering, providing a wide-variety of options to accommodate any measurement requirement.

When using pre-trigger, you must use software-based triggering to initiate an acquisition.

No pre-trigger, post-trigger stop event

In this simple mode, data acquisition starts when the trigger is received, and the acquisition stops when the stop-trigger event is received.

Fixed pre-trigger with post-trigger stop event

In this mode, you set the number of pre-trigger readings to acquire. The acquisition continues until a stop-trigger event occurs.

No pre-trigger, infinite post-trigger

In this mode, no pre-trigger data is acquired. Instead, data is acquired beginning with the trigger event, and is terminated when you issue a command to halt the acquisition.

Fixed pre-trigger with infinite post-trigger

You set the amount of pre-trigger data to acquire. Then, the system continues to acquire data until the program issues a command to halt acquisition.

Counter inputs

Four 32-bit counters are built into the USB-1616HS-2. Each counter accepts frequency inputs up to 20 MHz.

USB-1616HS-2 counter channels can be configured as standard counters or as multi-axis quadrature encoders.

The counters can concurrently monitor time periods, frequencies, pulses, and other event driven incremental occurrences directly from pulse-generators, limit switches, proximity switches, and magnetic pick-ups.

Counter inputs can be read asynchronously under program control, or synchronously as part of an analog or digital scan group.

When reading synchronously, all counters are set to zero at the start of an acquisition. When reading asynchronously, counters may be cleared on each read, count up continually, or count until the 16-bit or 32-bit limit is reached.

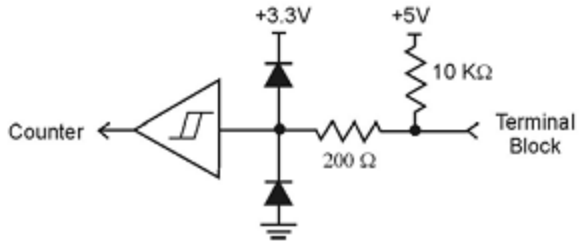


Figure 12. Typical USB-1616HS-2 counter channel

Tips for making high-speed counter measurements (> 1 MHz)

- Use coax or twisted-pair wire. Connect one side to Digital Common.
- If the frequency source is tolerant, parallel-terminate the coax or twisted-pair with a 50 Ω or 100 Ω resistor at the terminal block.
- The amplitude of the driving waveform should be as high as possible without violating the over-voltage specification.
- To ensure adequate switching, waveforms should swing at least 0 V to 5 V and have a high slew rate.

Mapped channels

A *mapped channel* is one of four counter input signals that can get multiplexed into a counter module. The mapped channel can participate with the counter's input signal by gating the counter, latching the counter, and so on. The four possible choices for the mapped channel are the four counter input signals (post-debounce).

A mapped channel can be used to:

- gate the counter
- decrement the counter
- latch the current count to the count register

Usually, all counter outputs are latched at the beginning of each scan within the acquisition. However, you can use a second mapped channel to latch the counter output.

Counter modes

A counter can be asynchronously read with or without *clear on read*. The asynchronous read-signals strobe when the lower 16-bits of the counter are read by software. The software can read the counter's high 16-bits some time later after reading the lower 16-bits. The full 32-bit result reflects the timing of the first asynchronous read strobe.

Totalize mode

The *Totalize mode* allows basic use of a 32-bit counter. While in this mode, the channel's input can only increment the counter upward. When used as a 16-bit counter (counter low), one channel can be scanned at the 12 MHz rate. When used as a 32-bit counter (*counter high*), two sample times are used to return the full 32-bit result. Therefore a 32-bit counter can only be sampled at a 6 MHz maximum rate. If you only want the upper 16 bits of a 32-bit counter, then you can acquire that upper word at the 12 MHz rate.

The counter counts up and does not clear on every new sample. However, it does clear at the start of a new scan command.

The counter rolls over on the 16-bit (*counter low*) boundary, or on the 32-bit (*counter high*) boundary.

Clear on read mode

The counter counts up and is cleared after each read. By default, the counter counts up and only clears the counter at the start of a new scan command. The final value of the counter—the value just before it was cleared—is latched and returned to the USB-1616HS-2.

Clear on read mode is only available if the counter is in asynchronous mode the. The counter's lower 16-bit value should be read first. This will latch the full 32-bit result and clear the counter. The upper 16-bit value can be read after the lower 16-bit value.

Stop at the top mode

The counter stops at the top of its count. The top of the count is FFFF hex (65,535) for the 16-bit mode, and FFFFFFFF hex (4,294,967,295) for the 32-bit mode.

32-bit or 16-bit

Sets the counter type to either **16-bits** or **32-bits**. The type of counter only matters if the counter is using the stop at the top mode—otherwise, this option is ignored.

Latch on map

Sets the signal on the mapped counter input to latch the count.

By default, the *start of scan* signal—a signal internal to the USB-1616HS-2 that pulses once every scan period to indicate the start of a scan group—latches the count so that the count is updated each time a scan is started.

Gating "on" mode

Sets the gating option to "on" for the mapped channel, enabling the mapped channel to gate the counter.

Any counter can be *gated* by the mapped channel. When the mapped channel is *high*, the counter is enabled. When the mapped channel is *low*, the counter is disabled (but holds the count value). The mapped channel can be any counter input channel other than the counter being gated.

Decrement "on" mode

Sets the counter decrement option to "on" for the mapped channel. The input channel for the counter increments the counter, and you can use the mapped channel to decrement the counter.

Debounce modes

Each channel's output can be debounced with 16 programmable debounce times from 500 ns to 25.5 ms. The debounce circuitry eliminates switch-induced transients typically associated with electro-mechanical devices including relays, proximity switches, and encoders.

There are two debounce modes, as well as a debounce bypass, as shown in Figure 13. In addition, the signal from the buffer can be inverted before it enters the debounce circuitry. The inverter is used to make the input rising-edge or falling-edge sensitive.

Edge selection is available with or without debounce. In this case the debounce time setting is ignored and the input signal goes straight from the inverter or inverter bypass to the counter module.

There are 16 different debounce times. In either debounce mode, the debounce time selected determines how fast the signal can change and still be recognized.

The two debounce modes are *trigger after stable* and *trigger before stable*. A discussion of the two modes follows.

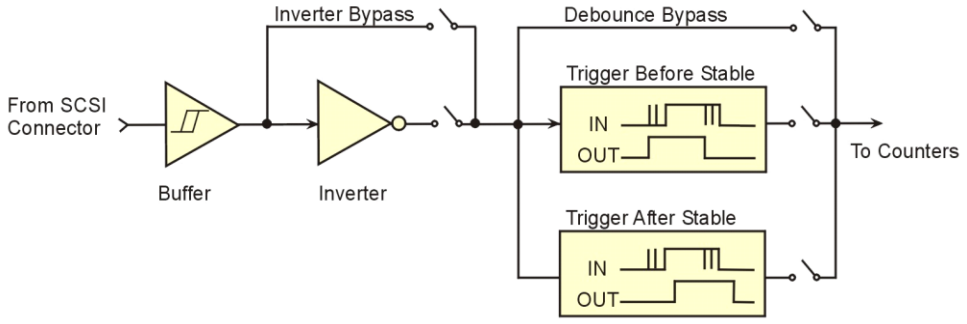


Figure 13. Debounce model block diagram

Trigger after stable mode

In the *trigger after stable* mode, the output of the debounce module does not change state until a period of stability has been achieved. This means that the input has an edge, and then must be stable for a period of time equal to the debounce time.

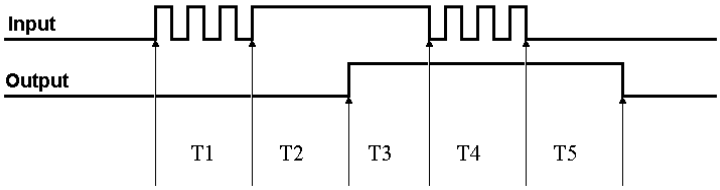


Figure 14. Debounce module – trigger after stable mode

The following time periods (T1 through T5) pertain to Figure 14. In *trigger after stable* mode, the input signal to the debounce module is required to have a period of stability after an incoming edge, in order for that edge to be accepted (passed through to the counter module.) The debounce time for this example is equal to T2 and T5.

- T1 – In the example above, the input signal goes high at the beginning of time period T1, but never stays high for a period of time equal to the debounce time setting (equal to T2 for this example.)
- T2 – At the end of time period T2, the input signal has transitioned high and stayed there for the required amount of time—therefore the output transitions high. If the input signal does not stabilize in the high state long enough, no transition would have appeared on the output and the entire disturbance on the input would have been rejected.
- T3 – During time period T3, the input signal remained steady. No change in output is seen.
- T4 – During time period T4, the input signal has more disturbances and does not stabilize in any state long enough. No change in the output is seen.
- T5 – At the end of time period T5, the input signal has transitioned low and stayed there for the required amount of time—therefore the output goes low.

Trigger before stable mode

In the *trigger before stable* mode, the output of the debounce module immediately changes state, but will not change state again until a period of stability has passed. For this reason the mode can be used to detect glitches.

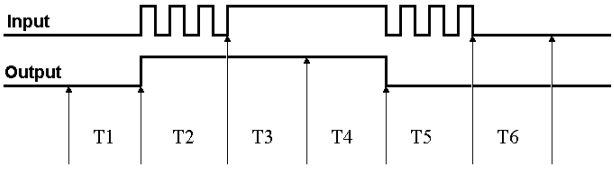


Figure 15. Debounce module – Trigger before stable mode

The following time periods (T1 through T6) pertain to the above drawing.

- T1 – In the illustrated example, the input signal is low for the debounce time (equal to T1); therefore when the input edge arrives at the end of time period T1, it is accepted and the output (of the debounce module) goes high. Note that a period of stability must precede the edge in order for the edge to be accepted.
- T2 – During time period T2, the input signal is not stable for a length of time equal to T1 (the debounce time setting for this example.) Therefore, the output stays "high" and does not change state during time period T2.
- T3 – During time period T3, the input signal is stable for a time period equal to T1, meeting the debounce requirement. The output is held at the high state. This is the same state as the input.
- T4 – At anytime during time period T4, the input can change state. When this happens, the output will also change state. At the end of time period T4, the input changes state, going low, and the output follows this action [by going low].
- T5 – During time period T5, the input signal again has disturbances that cause the input to not meet the debounce time requirement. The output does not change state.
- T6 – After time period T6, the input signal has been stable for the debounce time and therefore any edge on the input after time period T6 is immediately reflected in the output of the debounce module.

Debounce mode comparisons

Figure 16 shows how the two modes interpret the same input signal, which exhibits glitches. Notice that the *trigger before stable* mode recognizes more glitches than the *trigger after stable* mode. Use the *bypass* option to achieve maximum glitch recognition.

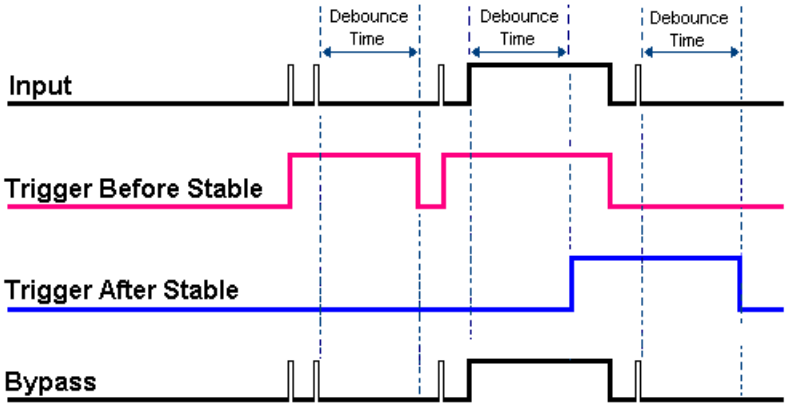


Figure 16. Example of two debounce modes interpreting the same signal

Debounce times should be set according to the amount of instability expected in the input signal. Setting a debounce time that is too short may result in unwanted glitches clocking the counter. Setting a debounce time too long may result in an input signal being rejected entirely. Some experimentation may be required to find the appropriate debounce time for a particular application.

To see the effects of different debounce time settings, simply view the analog waveform along with the counter output. This can be done by connecting the source to an analog input.

Use *trigger before stable* mode when the input signal has groups of glitches and each group is to be counted as one. The trigger before stable mode recognizes and counts the first glitch within a group but rejects the subsequent glitches within the group if the debounce time is set accordingly. The debounce time should be set to encompass one entire group of glitches as shown in the following diagram.

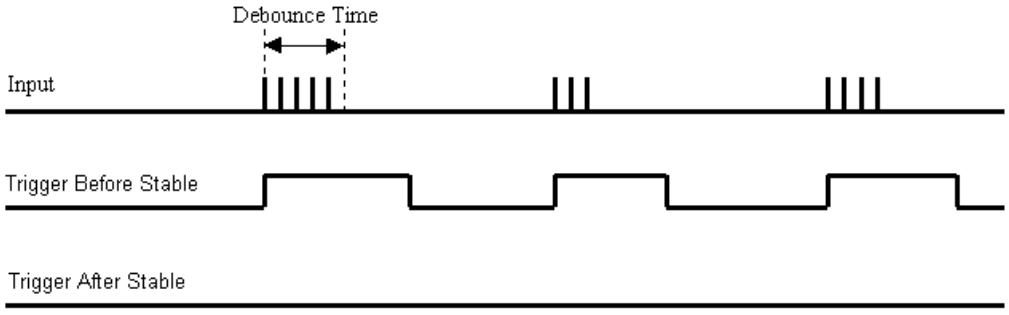


Figure 17. Optimal debounce time for trigger before stable mode

Trigger after stable mode behaves more like a traditional debounce function: rejecting glitches and only passing state transitions after a required period of stability. *Trigger after stable* mode is used with electro-mechanical devices like encoders and mechanical switches to reject switch bounce and disturbances due to a vibrating encoder that is not otherwise moving. The debounce time should be set short enough to accept the desired input pulse but longer than the period of the undesired disturbance as shown in Figure 18.

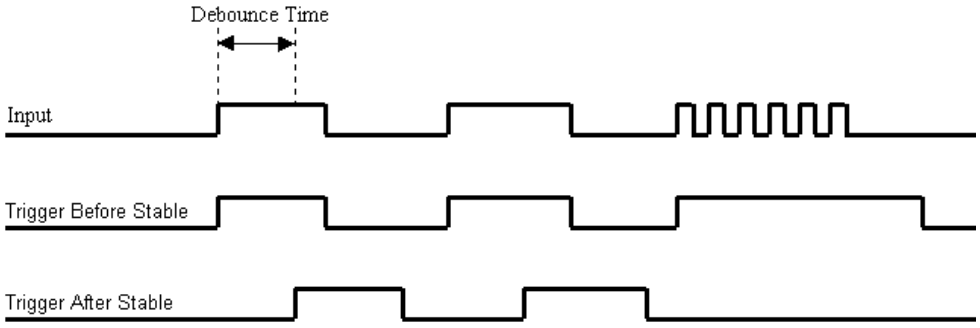


Figure 18. Optimal debounce time for trigger after stable mode

Encoder mode

Rotary shaft encoders are frequently used with CNC equipment, metal-working machines, packaging equipment, elevators, valve control systems, and in a multitude of other applications in which rotary shafts are involved.

The USB-1616HS-2 supports quadrature encoders with up to 2 billion pulses per revolution, 20 MHz input frequencies, and x1, x2, x4 count modes.

The *encoder mode* allows the USB-1616HS-2 to make use of data from optical incremental quadrature encoders. In encoder mode, the USB-1616HS-2 accepts *single-ended* inputs. When reading phase A, phase B, and index Z signals, the USB-1616HS-2 provides positioning, direction, and velocity data.

The USB-1616HS-2 can receive input from up to two encoders.

The USB-1616HS-2 supports quadrature encoders with a 16-bit (counter low) or a 32-bit (counter high) counter, 20 MHz frequency, and X1, X2, and X4 count modes. With only phase A and phase B signals, two channels are supported; with phase A, phase B, and index Z signals, 1 channel is supported. Each input can be debounced from 500 ns to 25.5 ms (total of 16 selections) to eliminate extraneous noise or switch induced transients. Encoder input signals must be within -5V to +10V and the switching threshold is TTL (1.3V).

Quadrature encoders generally have three outputs: A, B, and Z. The A and B signals are pulse trains driven by an optical sensor inside the encoder. As the encoder shaft rotates, a laminated optical shield rotates inside the encoder. The shield has three concentric circular patterns of alternating opaque and transparent windows through which an LED shines. There is one LED and one phototransistor for each of the concentric circular patterns. One phototransistor produces the A signal, another phototransistor produces the B signal and the last phototransistor produces the Z signal. The concentric pattern for A has 512 window pairs (or 1024, 4096, etc.)

When using a counter for a trigger source, use a pre-trigger with a value of at least 1. Since all counters start at zero with the initial scan, there is no valid reference in regard to rising or falling edge. Setting a pre-trigger to 1 or more ensures that a valid reference value is present, and that the first trigger is legitimate.

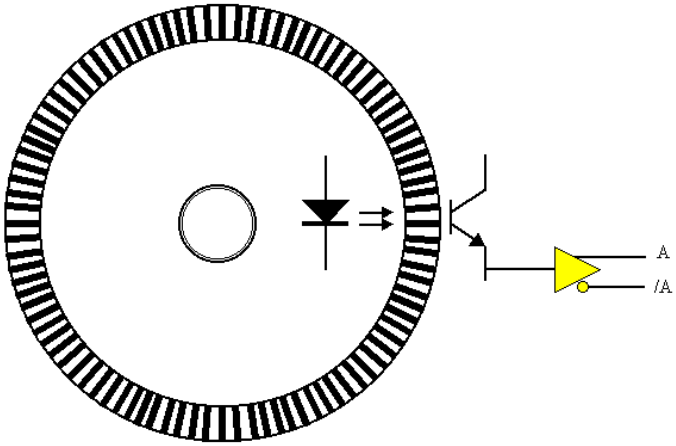


Figure 19. Representation of rotary shaft quadrature encoder

The concentric pattern for B has the same number of window pairs as A—except that the entire pattern is rotated by 1/4 of a window-pair. Thus the B signal is always 90 degrees out of phase from the A signal. The A and B signals pulse 512 times (or 1024, 4096, etc.) per complete rotation of the encoder.

The concentric pattern for the Z signal has only one transparent window and therefore pulses only once per complete rotation. Representative signals are shown in the following figure.

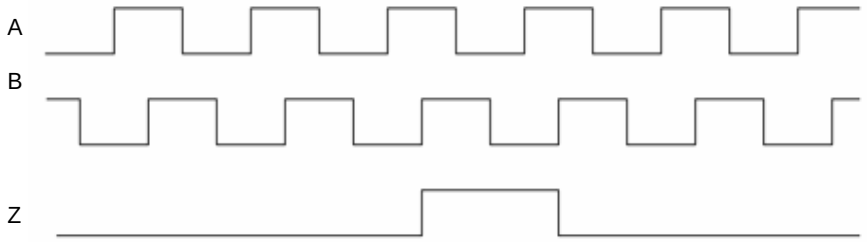


Figure 20. Representation of quadrature encoder outputs: A, B, and Z

As the encoder rotates, the A (or B) signal indicates the distance the encoder has traveled. The frequency of A (or B) indicates the velocity of rotation of the encoder. If the Z signal is used to zero a counter (that is clocked by A) then that counter will give the number of pulses the encoder has rotated from its reference. The Z signal is a reference marker for the encoder. It should be noted that when the encoder is rotating clockwise (as viewed from the back), A will lead B and when the encoder is rotating counterclockwise, A will lag B. If the counter direction control logic is such that the counter counts upward when A leads B and counts downward when A lags B, then the counter will give direction control as well as distance from the reference.

Maximizing encoder accuracy

If there are 512 pulses on A, then the encoder position is accurate to within 360°/512.

You can get even greater accuracy by counting not only rising edges on A but also falling edges on A, giving position accuracy to 360 degrees/1024.

You get maximum accuracy counting rising and falling edges on A and on B (since B also has 512 pulses.) This gives a position accuracy of 360°/2048. These different modes are known as X1, X2, and X4.

Connecting the USB-1616HS-2 to an encoder

You can use up to two encoders with each USB-1616HS-2 in your acquisition system. Each A and B signal can be made as a single-ended connection with respect to common ground.

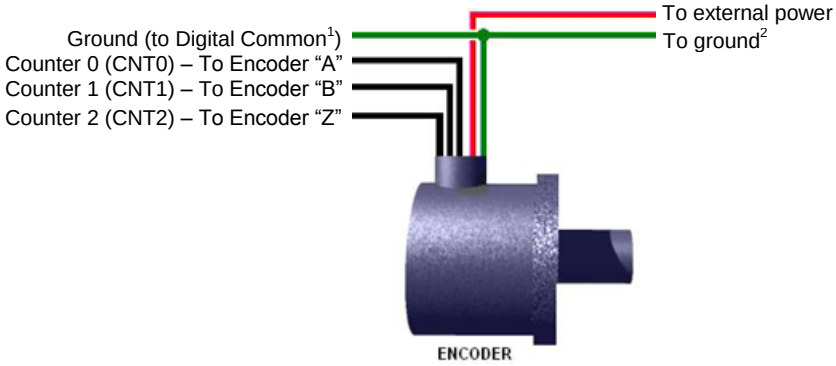
Differential applications are not supported.

For single-ended applications:

- Connect signals A, B, and Z to the counter inputs on the USB-1616HS-2.
- Connect each encoder ground to GND.

You can also connect external pull-up resistors to the USB-1616HS-2 counter input terminal blocks by placing a pull-up resistor between any input channel and the encoder power supply. Choose a pull-up resistor value based on the encoder's output drive capability and the input impedance of the USB-1616HS-2. Lower values of pull-up resistors cause less distortion, but also cause the encoder's output driver to pull down with more current.

Wiring to one encoder: Figure 21 shows the connections for one encoder to a USB-1616HS-2 module.



[1] The ground depicted at the left is associated with Digital Common on the USB-1616HS-2.
[2] The ground depicted at the right is associated with the external power source.

Figure 21. Connections from single encoder to screw terminals on the USB-1616HS-2

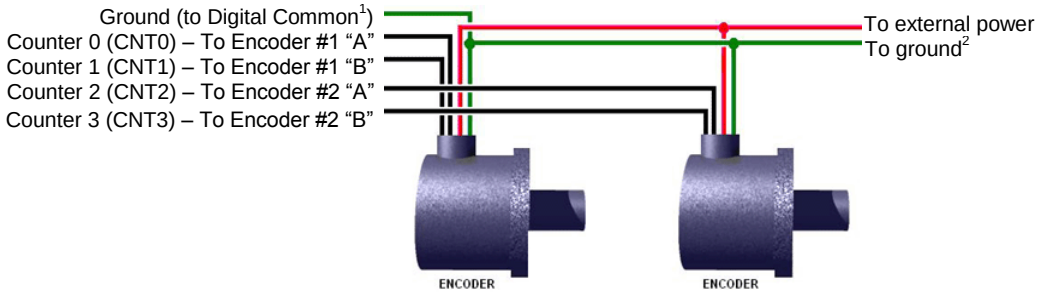
The "A" signal must be connected to an even-numbered channel and the associated "B" signal must be connected to the next higher odd-numbered channel. For example, if "A" were connected to counter 0, then "B" would be connected to counter 1.

Connect each signal (A, B, Z) as a single-ended connection with respect to the common ground. The encoder needs power from an external power output (typically +5 VDC). Connect the encoder's power input to the power source and connect the return to the digital common of that source.

Wiring for two encoders: The following figure illustrates single-ended connections for two encoders. Differential connections are not applicable.

Each signal (A, B) can be connected as a single-ended connection with respect to the common digital ground (GND). Both encoders need power from an external power source (typically +5 VDC).

Connect each encoder's power input to the external power source. Connect the return to digital common (GND) on the same source.



[1] The ground depicted at the left is associated with Digital Common on the USB-1616HS-2.
[2] The ground depicted at the right is associated with the external power source.

Figure 22. Connections from two encoders to screw terminals on the USB-1616HS-2

Timer outputs

Two 16-bit timer outputs are built into every USB-1616HS-2. Each timer output can generate a different square wave with a programmable frequency in the range of 16 Hz to 1 MHz.

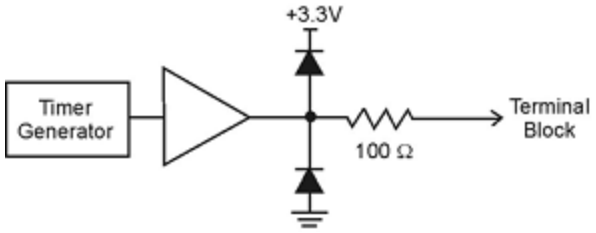


Figure 23. Typical USB-1616HS-2 timer channel

Example: Timer outputs

Timer outputs are programmable square waves. The period of the square wave can be as short as 1 µs or as long as 65535 µs. Refer to the table below for examples of timer output frequencies.

Timer output frequency examples

Divisor	Timer output frequency
1	1 MHz
100	10 kHz
1000	1 kHz
10000	100 Hz
65535	▪ 15.259 Hz (in asynchronous write) ▪ Turns timer off (for setpoint operation).

The two timer outputs can generate different square waves. The timer outputs can be updated asynchronously at any time.

Both timer outputs can also be updated during an acquisition as the result of setpoints applied to analog or digital inputs.

Using multiple USB-1616HS-2s per PC

USB-1616HS-2 features can be replicated up to four times, as up to four devices can be connected to a single host PC. The serial number on each USB-1616HS-2 distinguishes one from another. You can operate multiple USB-1616HS-2 boards synchronously. To do this, set up one USB-1616HS-2 with the pacer terminal you want to use (APR or DPR) configured for output. Set up the USB-1616HS-2 boards you want to synchronize to this board with the pacer screw terminal you want to use (APR or DPR) configured for input. Wire the pacer terminal configured for output to each of the pacer input terminals that you want to synchronize.

To operate two or more USB-1616HS-2s synchronously:

1. Use coax (or twisted-pair wire) to connect the output signal to the input(s).
2. Connect Digital Common of each USB-1616HS-2 to one of the twisted pairs or to the shield of the coax.

Using detection setpoints for output control

What are detection setpoints?

With the USB-1616HS-2's setpoint configuration feature, you can configure up to 16 detection setpoints associated with channels in a scan group. Each setpoint can update the following, allowing for real-time control based on acquisition data:

- FIRSTPORTC digital output port with a data byte and mask byte
- analog outputs (DACs)
- timers

Setpoint configuration overview

You can program each as one of the following:

- Single point referenced – Above, below, or equal to the defined setpoint.
- Window (dual point) referenced – Inside or outside the window.
- Window (dual point) referenced, hysteresis mode – Outside the window high forces one output (designated Output 2; outside the window low-forces another output, designated as Output 1).

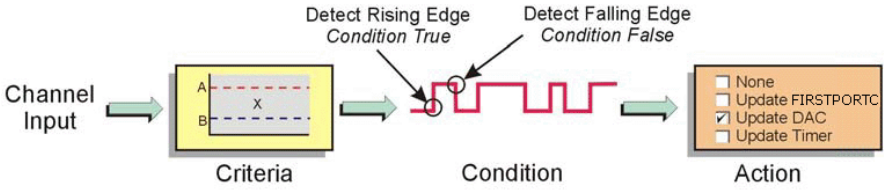


Figure 24. Diagram of detection setpoints

A digital detect signal is used to indicate when a signal condition is *True* or *False*—for example, whether or not the signal has met the defined criteria. The detect signals can be part of the scan group and can be measured as any other input channel, thus allowing real time data analysis during an acquisition.

The detection module looks at the 16-bit data being returned on a channel and generates another signal for each channel with a setpoint applied (*Detect1* for Channel 1, *Detect2* for Channel 2, and so on). These signals serve as data markers for each channel's data. It does not matter whether that data is volts, counts, or timing.

A channel's detect signal shows a rising edge and is *True* (1) when the channel's data meets the setpoint criteria. The detect signal shows a falling edge and is *False* (0) when the channel's data does not meet the setpoint criteria. The *True* and *False* states for each setpoint criteria are explained in the "Using the setpoint status register" section on page 34.

Criteria – input signal is equal to X		**Action - driven by condition**
Compare X to:	Setpoint definition (choose one)	Update conditions:
Limit A or Limit B	▪ Equal to A (X = A) ▪ Below A (X < A) ▪ Above B (X > B)	*True* only: ▪ If *True*, then output value 1 ▪ If *False*, then perform no action *True* and *False*: ▪ If *True*, then output value 1 ▪ If *False*, then output value 2
Window* (non-hysteresis mode)	▪ Inside (B < X < A) ▪ Outside (B > X; or, X > A)	*True* only ▪ If *True*, then output value 1 ▪ If *False*, then perform no action *True* and *False* ▪ If *True*, then output value 1 ▪ If *False*, then output value 2
Window* (hysteresis mode)	▪ Above A (X > A) ▪ Below (A < X < B) *(Both conditions are checked when in hysteresis mode*	Hysteresis mode (forced update) ▪ If X > A is *True*, then output value 2 until X < B is *True*, then output value 1. ▪ If X < B is *True*, then output value 1 until X > A is *True*, then output value 2. This is saying: (a) If the input signal is outside the window *high*, output value 2 until the signal goes outside the window *low*, and (b) if the signal is outside the window *low*, output value 1 until the signal goes outside the window *high*. There is no change to the detect signal while within the window.

The detect signal has the timing resolution of the scan period as seen in the diagram below. The detect signal can change no faster than the scan frequency (1/scan period.)

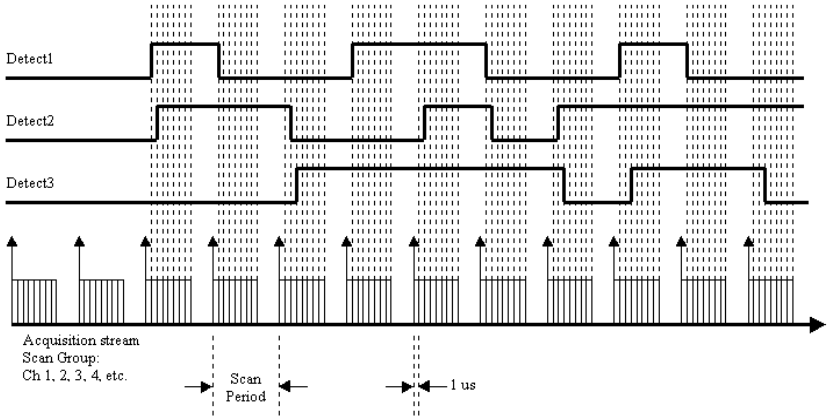


Figure 25. Example diagram of detection signals for channels 1, 2, and 3

Each channel in the scan group can have one detection setpoint. There can be no more than 16 total setpoints total applied to channels within a scan group.

Detection setpoints act on 16-bit data only. Since the USB-1616HS-2 has 32-bit counters, data is returned 16-bits at a time. The lower word, the higher word, or both lower and higher words can be part of the scan group. Each counter input channel can have one detection setpoint for the counter's lower 16-bit value and one detection setpoint for the counter's higher 16-bit value.

Setpoint configuration

You program all setpoints as part of the pre-acquisition setup, similar to setting up an external trigger. Since each setpoint acts on 16-bit data, each has two 16-bit compare values: a high limit (*limit A*) and a low limit (*limit B*). These limits define the setpoint window.

There are several possible conditions (criteria) and effectively three update modes, as explained in the following configuration summary.

Set high limit

You can set the 16-bit high limit (*limit A*) when configuring the USB-1616HS-2 through software.

Set low limit

You can set the 16-bit low limit (*limit B*) when configuring the USB-1616HS-2 through software.

Set criteria

- **Inside window**: Signal is below 16-bit high limit and above 16-bit low limit.
- **Outside window**: Signal is above 16-bit high limit, or below 16-bit low limit.
- **Greater than value**: Signal is above 16-bit low limit, so 16-bit high limit is not used.
- **Less than value**: Signal is below 16-bit high limit, so 16-bit low limit is not used.
- **Equal to value**: Signal is equal to 16-bit high limit, and limit B is not used.

  The *equal to* mode is intended for use when the counter or digital input channels are the source channel.

  You should only use the *equal to*16-bit high limit *(limit A)* mode with counter or digital input channels as the channel source. If you want similar functionality for analog channels, then use the *inside window* mode
- **Hysteresis mode**: Outside the window, high forces output 2 until an outside the window low condition exists, then output 1 is forced. Output 1 continues until an outside the window high condition exists. The cycle repeats as long as the acquisition is running in hysteresis mode.

Set output channel

- None
- Update FIRSTPORTC
- Update DAC
- Update timer*x*

Update modes

- Update on *True* only
- Update on *True* and *False*

Set values for output

- 16-bit DAC value, FIRSTPORTC* value, or timer value when input meets criteria.
- 16-bit DAC value, FIRSTPORTC* value, or timer value when does not meet criteria.

  * By default, FIRSTPORTC comes up as a digital input. You may want to initialize FIRSTPORTC to a known state before running the input scan to detect the setpoints.

When using setpoints with triggers other than immediate, hardware analog, or TLL, the setpoint criteria evaluation begins immediately upon arming the acquisition.

Using the setpoint status register

You can use the setpoint status register to check the current state of the 16 possible setpoints. In the register, Setpoint 0 is the least-significant bit and Setpoint 15 is the most-significant bit. Each setpoint is assigned a value of 0 or 1.

- A value of 0 indicates that the setpoint criteria are not met—in other words, the condition is *False*.
- A value of 1 indicates that the criteria have been met—in other words, the condition is *True*.

In the following example, the criteria for setpoints 0, 1, and 4 is satisfied (*True*), but the criteria for the other 13 setpoints has not been met.

Setpoint #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
True (1)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
False (0)	<<< Most significant bit							Least significant bit >>>								

From the table above we have `10011` binary, or 19 decimal, derived as follows:

- Setpoint 0, having a *True* state, shows 1, giving us decimal 1.
- Setpoint 1, having a *True* state, shows 1, giving us decimal 2.
- Setpoint 4, having a *True* state, shows 1, giving us decimal 16.

For proper operation, the setpoint status register must be the last channel in the scan list.

Examples of control outputs

Detecting on analog input, DAC, and FIRSTPORTC updates

Update mode: Update on *True* and *False*

Criteria: Channel 5 example: *below limit*; channel 4 example: *inside window*

In this example, channel 5 is programmed with reference to one setpoint (*limit A*), defining a low limit.

Channel 4 is programmed with reference to two setpoints (limit A and limit B) which define a window for that channel.

Channel	Condition	State of detect signal	Action
5	Below limit A (for channel 5)	*True*	When channel 5 analog input voltage is below the limit A, update DAC1 with output value 0.0 V.
		False	When the above stated condition is false, update DAC1 with the Output Value of - 1.0 V.
4	Within window (between limit A and limit B) for channel 4	*True*	When Channel 4's analog input voltage is within the window, update FIRSTPORTC with 70h.
		False	When the above stated condition is *False* (channel 4 analog input voltage is outside the window), update FIRSTPORTC with 30h.

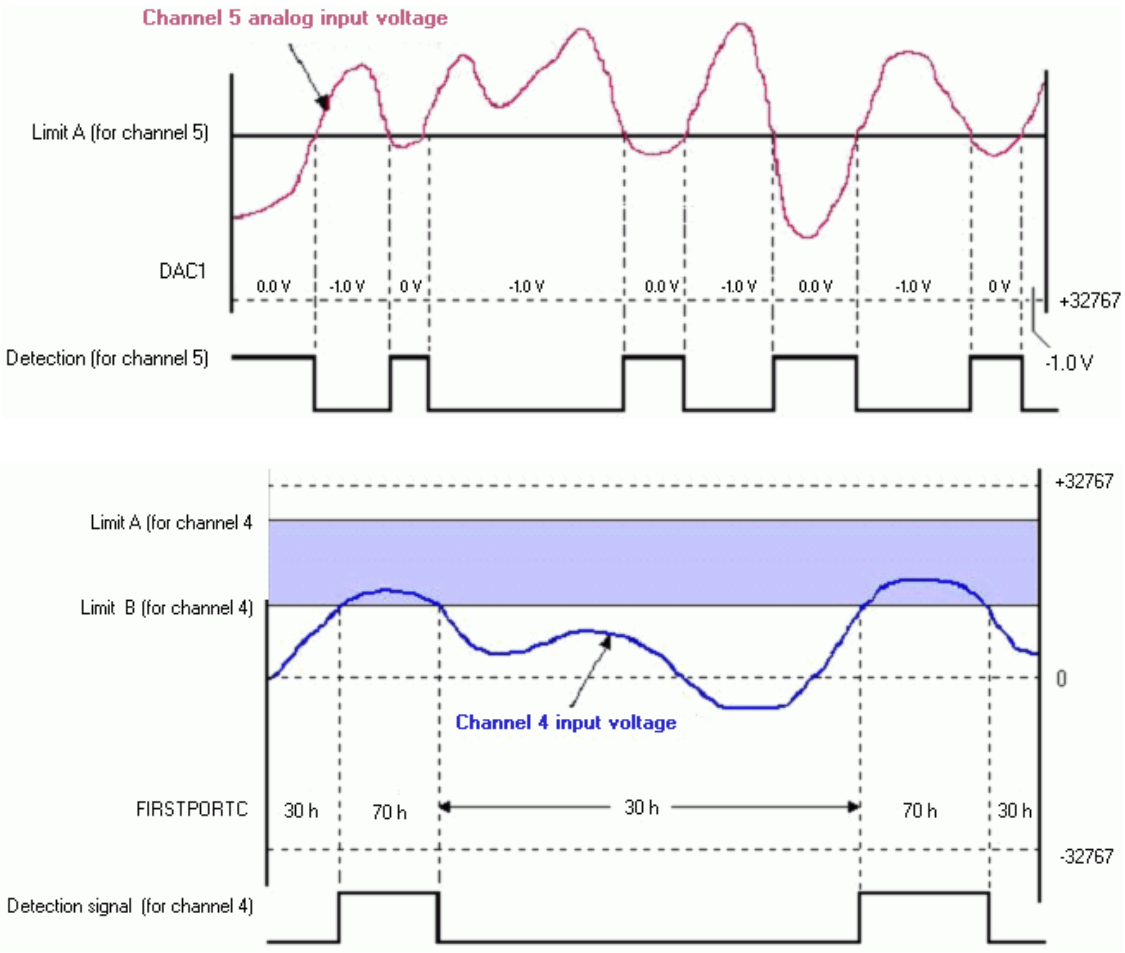


Figure 26. Analog inputs with setpoints update on *True* and *False*

In the channel 5 example, the setpoint placed on analog Channel 5 updated DAC1 with 0.0 V. The update occurred when channel 5's input was less than the setpoint (limit A). When the value of channel 5's input was above setpoint limit A, the condition of <A was false and DAC1 was then updated with -1.0 V.

You can program control outputs programmed on each setpoint, and use the detection for channel 4 to update the FIRSTPORTC digital output port with one value (70 h in the example) when the analog input voltage is within the shaded region and a different value when the analog input voltage is outside the shaded region (30 h in the example).

Detection on an analog input, timer output updates

Update Mode: Update on *True* and *False*

Criteria Used: Inside window

Figure 27 shows how a setpoint can be used to update a timer output. Channel 3 is an analog input channel. A setpoint is applied using *update on True and False*, with a criteria of *inside-the-window*, where the signal value is inside the window when simultaneously less than Limit A but greater than Limit B.

Whenever the channel 3 analog input voltage is inside the setpoint window (condition *True*), Timer0 is updated with one value; and whenever the channel 3 analog input voltage is outside the setpoint window (condition False) timer0 will be updated with a second output value.

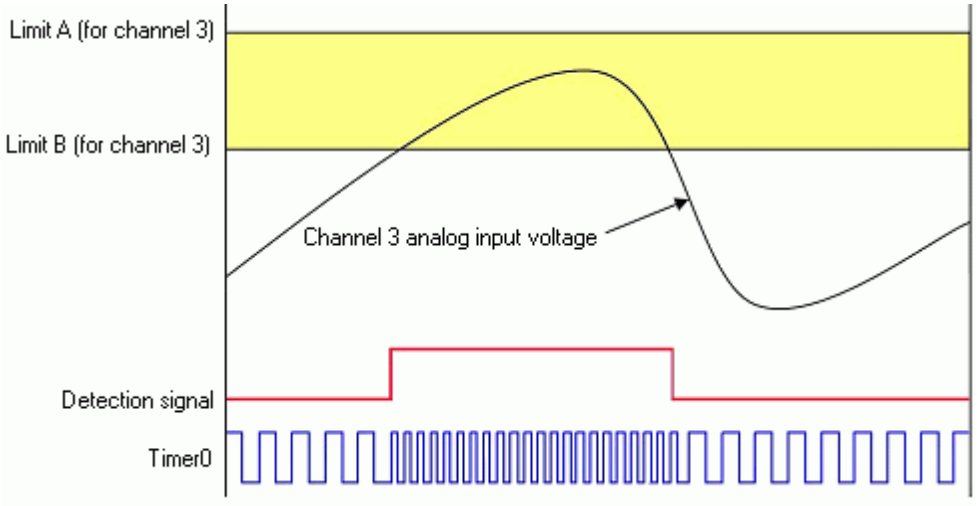


Figure 27. Timer output update on *True* and *False*

Using the hysteresis function

Update mode: N/A, the hysteresis option has a forced update built into the function

Criteria used: Window criteria for above and below the set limits

Figure 28 shows analog input Channel 3 with a setpoint which defines two 16-bit limits, Limit A (High) and Limit B (Low). These are being applied in the hysteresis mode and DAC channel 0 is updated accordingly.

In this example, Channel 3's analog input voltage is being used to update DAC0 as follows:

- When outside the window, low (below limit B) DAC0 is updated with 3.0 V. This update remains in effect until the analog input voltage goes above Limit A.
- When outside the window, high (above limit A), DAC0 is updated with 7.0 V. This update remains in effect until the analog input signal falls below limit B. At that time we are again outside the limit "low" and the update process repeats itself.

Hysteresis mode can also be done with FIRSTPORTC digital output port, or a timer output, instead of a DAC.

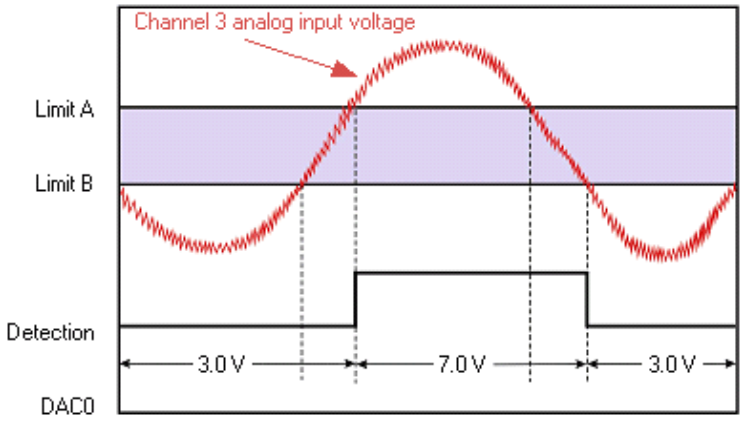


Figure 28. Channel 3 in hysteresis mode

Using multiple inputs to control one DAC output

Update mode: Rising edge, for each of two channels

Criteria used: Inside window, for each of two channels

The figure below shows how multiple inputs can update one output. In the following figure the DAC2 analog output is being updated. Analog input Channel 3 has an inside-the-window setpoint applied. Whenever Channel 3's input goes inside the programmed window, DAC2 will be updated with 3.0 V.

Analog input Channel 7 also has an inside-the-window setpoint applied. Whenever channel 7's input goes inside the programmed window, DAC2 is updated with - 7.0 V.

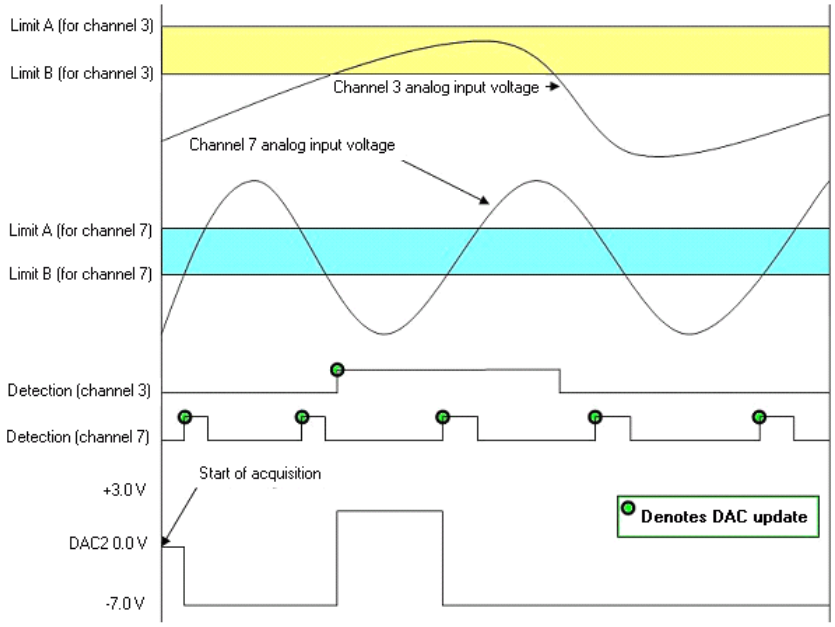


Figure 29. Using two criteria to control an output*

The update on *True*-only mode was selected, and therefore the updates for DAC2 only occur when the criteria is met. However, in the above figure we see that there are two setpoints acting on one DAC. We can also see that the two criteria can be met simultaneously. When both criteria are *True* at the same time, the DAC2 voltage is associated with the criteria that has been most recently met.

Detecting setpoints on a totalizing counter

In the following figure, Channel 1 is a counter in totalize mode. Two setpoints define a point of change for Detect 1 as the counter counts upward. The detect output is high when inside the window (greater than Limit B (the low limit) but less than Limit A (the high limit).

In this case, the Channel 1 setpoint is defined for the 16 lower bits of channel 1's 32-bit value. The FIRSTPORTC digital output port could be updated on a *True* condition (the rising edge of the detection signal). Alternately, one of the DAC output channels, or timer outputs, could be updated with a value.

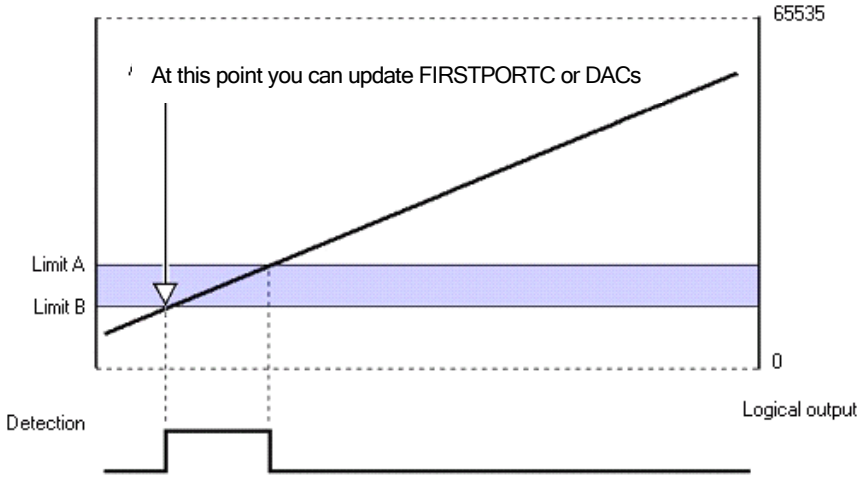


Figure 30. Channel 1 in totalizing counter mode, inside the window setpoint

Detection setpoint details

Controlling analog, digital, and timer outputs

You can program each setpoint with an 8-bit digital output byte and corresponding 8-bit mask byte. When the setpoint criteria is met, the FIRSTPORTC digital output port can be updated with the given byte and mask. Alternately, you can program each setpoint with a 16-bit DAC update value, and any one of the two DAC outputs can be updated in real time. Any setpoint can also be programmed with a timer update value.

In *hysteresis mode*, each setpoint has two forced update values. Each update value can drive one DAC, one timer, or the FIRSTPORTC digital output port. *In hysteresis mode, the outputs do not change when the input values are inside the window*. There is one update value that gets applied when the input values are less than the window and a different update value that gets applied when the input values are greater than the window.

Update on *True* and *False* uses two update values. The update values can drive DACs, FIRSTPORTC, or timer outputs.

FIRSTPORTC digital outputs can be updated immediately upon setpoint detection. This is not the case for analog outputs, as these incur another 3 µs delay. This is due to the shifting of the digital data out to the D/A converter which takes 1µs, plus the actual conversion time of the D/A converter, i.e., another 2µs (worst case). Going back to the above example, if the setpoint for analog input Channel 2 required a DAC update it would occur 5µs after the ADC conversion for Channel 2, or 6µs after the start of the scan.

When using setpoints to control any of the DAC outputs, increased latencies may occur if attempting to stream data to DACs or pattern digital output at the same time. The increased latency can be as long as the period of the output scan clock. For these reasons, avoid streaming outputs on any DAC or pattern digital output when using setpoints to control DACs.

FIRSTPORTC, DAC, or timer update latency

Setpoints allow analog output DACs, timers, or FIRSTPORTC digital outputs to update very quickly. Exactly how fast an output can update is determined by these factors:

- scan rate
- synchronous sampling mode
- type of output to be updated

For example, you set an acquisition to have a scan rate of 100 kHz, which means each scan period is 10 µs. Within the scan period you sample six analog input channels. These are shown in the following figure as channels 1 through 6. The ADC conversion occurs at the beginning of each channel's 1 µs time block.

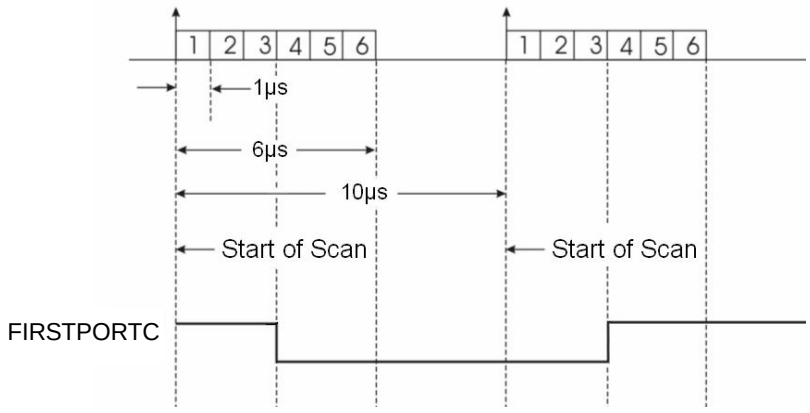


Figure 31. Example of FIRSTPORTC or DAC latency

By applying a setpoint on analog input channel 2, that setpoint gets evaluated every 10 µs with respect to the sampled data for channel 2.

Due to the pipelined architecture of the analog-to-digital converter system, the setpoint cannot be evaluated until 2 µs after the ADC conversion. In the example above, the FIRSTPORTC digital output port can be updated no sooner than 2 µs after channel 2 has been sampled, or 3 µs after the start of the scan. This 2 µs delay

is due to the pipelined ADC architecture. The setpoint is evaluated 2 µs after the ADC conversion and then FIRSTPORTC can be updated immediately.

The detection circuit works on data that is put into the acquisition stream at the scan rate. This data is acquired according to the pre-acquisition setup (scan group, scan period, etc.) and returned to the PC. Counters are latched into the acquisition stream at the beginning of every scan. The actual counters may be counting much faster than the scan rate, and therefore only every 10^{th}, 100^{th}, or n^{th} count shows up in the acquisition data.

As a result, you can set a small detection window on a totalizing counter channel and have the detection setpoint "stepped over" since the scan period was too long. Even though the counter value stepped into and out of the detection window, the actual values going back to the PC may not. This is true no matter what mode the counter channel is in.

When setting a detection window, keep a scan period in mind. This applies to analog inputs and counter inputs. Quickly changing analog input voltages can step over a setpoint window if not sampled often enough.

There are three possible solutions for overcoming this problem:

- Shorten the scan period to give more timing resolution on the counter values or analog values.
- Widen the setpoint window by increasing limit A and/or lowering limit B.
- A combination of both solutions (1 and 2) could be made.

Calibrating the USB-1616HS-2

Board ranges are calibrated at the factory using a digital NIST traceable calibration method in which a correction factor for each range is stored on the unit at the time of calibration.

Two calibration tables are stored on the board in EPROM — one table contains the factory calibration, and the other is available for field calibration. You can adjust the AI calibration while the board is installed in the acquisition system without destroying the factory calibration supplied with the board.

You can perform field calibration automatically in seconds with InstaCal. No external hardware or instruments are required. Field calibration derives its traceability through an on-board reference which has a stability of 0.005% per year.

Calibrate the board after it has fully warmed up; the recommended warm-up time is 30 minutes. For best results, calibrate the board immediately before making critical measurements. The high resolution analog components on the board are somewhat sensitive to temperature. Pre-measurement calibration ensures that your board is operating at optimum calibration values.

The recommended calibration interval is one year.

Specifications

Typical for 25 °C unless otherwise specified.
Specifications in *italic text* are guaranteed by design.

Analog input

Table 1. Analog input specifications

A/D converter type	Successive approximation	
Resolution	16 bits	
Number of channels	16 single-ended/8 differential, software-selectable. Up to 48 additional analog inputs per module are available with the optional AI-EXP48 module. Expansion channel features are the same as those of the main channels.	
Input ranges (software and sequencer programmable)	Bipolar: ±10 V, ±5 V, ±2 V, ±1 V , ±0.5 V, ±0.2 V, ±0.1 V	
Maximum sample rate	1 MHz	
Nonlinearity (integral)	±2 LSB maximum	
Nonlinearity (differential)	±1 LSB maximum	
A/D pacing	Onboard input scan clock, external source (APR)	
Trigger sources and modes	See Table 8	
Acquisition data buffer	1 MSample	
Data transfer	DMA	
Configuration memory	Programmable I/O	
Maximum usable input voltage + common mode voltage (CMV + V_{in})	Range: ±10 V, ±5 V, ±2 V, ±1 V, ±0.5 V	10.5 V maximum
	Range: ±0.2 V, ±0.1 V	2.1 V maximum
Signal to noise and distortion	*72 dB typical for ±10 V range, 1 kHz fundamental*	
Total harmonic distortion	*-80 dB typical for ±10 V range, 1 kHz fundamental*	
Calibration	Auto-calibration, calibration factors for each range stored onboard in non-volatile RAM.	
CMRR @ 60 Hz	-70 dB typical DC to 1 kHz	
Bias current	*40 pA typical (0 °C to 35°C)*	
Crosstalk	*-75 dB typical DC to 60 Hz; -65 dB typical @ 10 kHz*	
Input impedance	*10 MΩ single-ended, 20 MΩ differential*	
Absolute maximum input voltage	*±30 V*	

Accuracy

Table 2. Analog input accuracy specifications

Voltage range		**Accuracy ±(% of reading + % range) 23°C ±10 °C, 1 year**	**Temperature coefficient ±(ppm of reading + ppm range)/°C**	**Noise (cts RMS)**	
-10 V to 10 V	Note 1	*0.031% + 0.008%*	*14 + 8*	2.0	Note 2
-5 V to 5 V		*0.031% + 0.009%*	*14 + 9*	3.0	
-2 V to 2 V		*0.031% + 0.010%*	*14 +10*	2.0	
-1 V to 1 V		*0.031% + 0.02%*	*14 + 12*	3.5	
-500 mV to 500 mV		*0.031% + 0.04%*	*14 +18*	5.5	
-200 mV to 200 mV		*0.036% + 0.075%*	*14 +12*	8.0	
-100 mV to 100 mV		*0.042% + 0.15%*	*14 +18*	14.0	

Note 1: Specifications assume differential input single-channel scan, 1 MHz scan rate, unfiltered, CMV=0.0 V, 30 minute warm-up, exclusive of noise, range is +FS to -FS.

Note 2: Noise reflects 10,000 samples at 1 MHz, typical, differential short.

Thermocouples

Table 3. Thermocouple (TC) types and accuracy (Note 3)

TC type	Temperature range (°C)	Accuracy (±°C)	Noise typical (±°C)
J	-200 to + 760	1.7	0.2
K	-200 to + 1200	1.8	0.2
T	-200 to + 400	1.8	0.2
E	-270 to + 650	1.7	0.2
R	-50 to + 1768	4.8	1.5
S	-50 to + 1768	4.7	1.5
N	-270 to + 1300	2.7	0.3
B	+300 to + 1400	3.0	1.0

Note 3: Assumes 16384 oversampling applied, CMV = 0.0V, 60 minute warm-up, still environment, and 25 °C ambient temperature; excludes thermocouple error; TC_{in} = 0° C for all types except B (1000 °C), TR-2U power supply for external power.

Analog outputs

Analog output channels can be updated synchronously relative to scanned inputs, and clocked from either an internal onboard clock, or an external clock source. Analog outputs can also be updated asynchronously, independent of any other scanning system.

Table 4. Analog output specifications

Channels	2
Resolution	16-bits
Data buffer	PC-based memory
Output voltage range	±10 V
Output current	±1 mA Sourcing more current (1 to 10 mA) may require a TR-2U power supply.
Offset error	±0.0045 V maximum
Digital feed-through	<10 mV when updated
DAC analog glitch	<12 mV typical at major carry
Gain error	±0.01%
Coupling	DC
Update rate	1 MHz maximum, resolution 20.83 ns
Settling time	2 µs to rated accuracy
Pacer sources	Four programmable sources: ▪ Onboard output scan clock, independent of input scan clock ▪ Onboard input scan clock ▪ External output scan clock (DPR), independent of external input scan clock (APR) ▪ External input scan clock (APR)
Trigger sources	Start of input scan

Digital input/output

Table 5. Digital input/output specifications

Number of I/O	24
Ports	Three banks of eight. Each port is programmable as input or output
Input scanning modes	Two programmable ▪ Asynchronous, under program control at any time relative to input scanning ▪ Synchronous with input scanning
Input characteristics	220 Ω series resistors, 20 pF to common
Logic keeper circuit	Holds the logic value to 0 or 1 when there is no external driver
Input protection	±15 kV ESD clamp diodes parallel
Input high	*+2.0 V to +5.0 V*
Input low	*0 to 0.8 V*
Output high	*>2.0 V*
Output low	*<0.8 V*
Output current	Output 1.0 mA per pin Sourcing more current may require a TR-2U power supply.
Digital input pacing	Onboard input scan clock, external input scan clock (APR)
Digital output pacing	Four programmable sources: ▪ Onboard output scan clock, independent of input scan clock ▪ Onboard input scan clock ▪ External output scan clock (DPR), independent of external input scan clock (APR) ▪ External input scan clock (APR)
Digital input trigger sources and modes	See Table 8
Digital output trigger sources	Start of input scan
Data transfer	DMA
Sampling/update rate	4 MHz maximum (rates up to 12 MHz are sustainable on some platforms)
Pattern generation output	Two of the 8-bit ports can be configured for 16-bit pattern generation. The pattern can also be updated synchronously with an acquisition at up to 4 MHz.

Counters

Counter inputs can be scanned based on an internal programmable timer or an external clock source.

Table 6. Counter specifications

Channels	4 independent
Resolution	32-bit
Input frequency	20 MHz maximum
Input signal range	-5 V to 10 V
Input characteristics	10 kΩ pull-up, 200 Ω series resistor, ±15 kV ESD protection
Trigger level	TTL
Minimum pulse width	25 ns high, 25 ns low
Debounce times	16 selections from 500 ns to 25.5 ms, positive or negative edge sensitive, glitch detect mode or debounce mode
Time base accuracy	50 ppm (0 ° to 50 °C)
Counter read pacer	Onboard input scan clock, external input scan clock (APR)
Trigger sources and modes	See Table 8
Programmable mode	Counter
Counter mode options	Totalize, clear on read, rollover, stop at all Fs, 16- or 32-bit, any other channel can gate the counter

Input sequencer

Analog, digital, and counter inputs can be scanned based on either an internal programmable timer or an external clock source.

Table 7. Input sequencer specifications

Input scan clock sources: two (see Note 4)	Internal, programmable: ▪ Analog channels from 1 µs to 1 s in 20.83 ns steps. ▪ Digital channels and counters from 250 ns to 1 s in 20.83 ns steps. External. TTL level input (APR): ▪ Analog channels down to 1 µs minimum ▪ Digital channels and counters down to 250 ns minimum
Programmable parameters per scan:	Programmable channels (random order), programmable gain
Depth	512 locations
Onboard channel-to-channel scan rate	Analog: 1 MHz maximum Digital: 4 MHz if no analog channels are enabled, 1 MHz with analog channels enabled
External input scan clock (APR) maximum rate	Analog: 1.0 MHz Digital: 4 MHz if no analog channels are enabled, 1 MHz with analog channels enabled
Clock signal range:	Logical zero: 0 V to 0.8 V Logical one: 2.4 V to 5.0 V
Minimum pulse width	50 ns high, 50 ns low

Note 4: The maximum scan clock rate is the inverse of the minimum scan period. The minimum scan period is equal to 1 µs times the number of analog channels. If a scan contains only digital channels, then the minimum scan period is 250 ns.

Some platforms can sustain clock rates up to 83.33 ns.

Triggering

Table 8. Trigger sources and modes

Trigger source	Explanation
Single channel analog hardware trigger	Any analog input channel can be software programmed as the analog trigger channel, including any of the analog expansion channels. ▪ Input signal range: -10 V to +10 V maximum ▪ Trigger level: Programmable (12-bit resolution) ▪ Latency: 350 ns typical, 1.3 µs max ▪ Accuracy: ±0.5% of reading, ±2 mV offset maximum ▪ Noise: 2 mV RMS typical
Single channel analog software trigger	Any analog input channel—including any of the analog expansion channels, can be selected as the software trigger channel. If the trigger channel involves a calculation, such as temperature, then the driver automatically compensates for the delay required to obtain the reading, resulting in a maximum latency of one scan period. ▪ Input signal range: Anywhere within range of the trigger channel ▪ Trigger level: Programmable (16-bit resolution) ▪ Latency: One scan period (maximum)
External-single channel digital trigger	A separate digital input is provided for digital triggering. ▪ Input signal range: -15 V to +15 V maximum ▪ Trigger level: TTL level sensitive ▪ Minimum pulse width: 50 ns high, 50 ns low ▪ Latency: One scan period maximum
Digital pattern triggering	8-bit or 16-bit pattern triggering on any of the digital ports. Programmable for trigger on equal, not equal, above, or below a value. ▪ Individual bits can be masked for "don't care" condition. ▪ Latency: One scan period, maximum
Counter/totalizer triggering	Counter/totalizer inputs can trigger an acquisition. User can select to trigger on a frequency or on total counts that are equal, not equal, above, or below a value, or within/outside of a window rising/falling edge. ▪ Latency: One scan period, maximum

Frequency/pulse generators

Table 9. Frequency/pulse generator specifications

Channels	2 × 16-bit
Output waveform	Square wave
Output rate	1 MHz base rate divided by 1 to 65535 (programmable)
High-level output voltage	2.0 V minimum @ -1.0 mA, 2.9 V minimum @ -400 µA
Low-level output voltage	0.4 V maximum @ 400 µA

Power consumption

Power consumption specification is for a USB-1616HS-2. Add 400mW for a USB-1616HS-2 connected to an AI-EXP48 expansion module.

Table 10. Power consumption specifications (Note 5)

Power consumption (per board)	3000 mW

External power

Table 11. External power specifications (Note 5)

Connector	Switchcraft # RAPC-712
Power range	6 to 16 VDC (used when USB port supplies insufficient power, or when an independent power supply is desired)
Over-voltage	20 V for 10 seconds, maximum

Note 5: The power supply (MCC p/n TR-2U) and line cord (MCC p/n CA-1) are required if the USB port cannot supply adequate power. By USB 2.0 standards, USB 2.0 ports must supply 2500 mW (nominal at 5 V, 500 mA)

USB specifications

Table 12. USB specifications

USB-device type	USB 2.0 high-speed mode (480 Mbps) if available (recommended), otherwise, USB1.1 full-speed mode (12 Mbps)
Device compatibility	USB 2.0 (recommended) or USB 1.1

Environmental

Table 13. Environmental specifications

Operating temperature range	-30 °C to +70 °C
Storage temperature range	-40 °C to +80 °C
Relative humidity	0 to 95% non-condensing

Mechanical

Table 14. Mechanical specifications

Vibration	MIL STD 810E category 1 and 10
Dimensions	269 mm (W) x 92 mm (D) x 45 mm (H)(10.6” x 3.6” x 1.6”)
Weight	431 g (0.95 lbs)

Signal I/O connectors and pin out

Table 15. Screw connector specifications

Connector type	Screw terminal
Wire gauge range	14 AWG to 30 AWG
Expansion connector type	25-pin DSUB, female
Compatible expansion products	AI-EXP48 expansion module with screw terminals

Table 16. USB-1616HS-2 screw terminal pin out – single-ended connections

Analog Out	Analog common (A⏷)		Port A	Digital common (D⏷)	**DIG-Tmr I/O**
	Analog output 0 (AO0)			FIRSTPORTA Bit 0 (A0)	
	Analog output 1 (AO1)			FIRSTPORTA Bit 1 (A1)	
	NC			FIRSTPORTA Bit 2 (A2)	
	NC			FIRSTPORTA Bit 3 (A3)	
	Analog common (A⏷)			FIRSTPORTA Bit 4 (A4)	
	CAL (Reserved for self-calibration)			FIRSTPORTA Bit 5 (A5)	
	Signal ground (S⏷)			FIRSTPORTA Bit 6 (A6)	
	Digital common (D⏷)			FIRSTPORTA Bit 7 (A7)	
	TTL trigger (TRG)			Digital common (D⏷)	
	Output scan clock I/O (DPR)			Timer 0 (T0)	
	Input scan clock I/O (APR)			Timer 1 (T1)	
Analog In	Analog common (A⏷)		Port B	Digital common (D⏷)	**Dig-Ctr I/O**
	CH 0 (0H)			FIRSTPORTB Bit 0 (B0)	
	CH 8 (8L)			FIRSTPORTB Bit 1 (B1)	
	Analog common (A⏷)			FIRSTPORTB Bit 2 (B2)	
	CH 1 (1H)			FIRSTPORTB Bit 3 (B3)	
	CH 9 (9L)			FIRSTPORTB Bit 4 (B4)	
	Analog common (A⏷)			FIRSTPORTB Bit 5 (B5)	
	CH 2 (2H)			FIRSTPORTB Bit 6 (B6)	
	CH 10 (10L)			FIRSTPORTB Bit 7 (B7)	
	Analog common (A⏷)			Digital common (D⏷)	
	CH 3 (3H)			Counter 0 (CT0)	
	CH 11 (11L)			Counter 1 (CT1)	
Analog In	Analog common (A⏷)		Port C	Digital common (D⏷)	**Dig-Ctr I/O**
	CH 4 (4H)			FIRSTPORTC Bit 0 (C0)	
	CH 12 (12L)			FIRSTPORTC Bit 1 (C1)	
	Analog common (A⏷)			FIRSTPORTC Bit 2 (C2)	
	CH 5 (5H)			FIRSTPORTC Bit 3 (C3)	
	CH 13 (13L)			FIRSTPORTC Bit 4 (C4)	
	Analog common (A⏷)			FIRSTPORTC Bit 5 (C5)	
	CH 6 (6H)			FIRSTPORTC Bit 6 (C6)	
	CH 14 (14L)			FIRSTPORTC Bit 7 (C7)	
	Analog common (A⏷)			Digital common (D⏷)	
	CH 7 (7H)			Counter 2 (CT2)	
	CH 15 (15L)			Counter 3 (CT3)	

Table 17. USB-1616HS-2 screw terminal pin out – differential connections

Analog Out	Analog common (A⏷)		Port A	Digital common (D⏷)	**DIG-Tmr I/O**
	Analog output 0 (AO0)			FIRSTPORTA Bit 0 (A0)	
	Analog output 1 (AO1)			FIRSTPORTA Bit 1 (A1)	
	NC			FIRSTPORTA Bit 2 (A2)	
	NC			FIRSTPORTA Bit 3 (A3)	
	Analog common (A⏷)			FIRSTPORTA Bit 4 (A4)	
	CAL (Reserved for self-calibration)			FIRSTPORTA Bit 5 (A5)	
	Signal ground (S⏷)			FIRSTPORTA Bit 6 (A6)	
	Digital common (D⏷)			FIRSTPORTA Bit 7 (A7)	
	TTL trigger (TRG)			Digital common (D⏷)	
	Output scan clock I/O (DPR)			Timer 0 (T0)	
	Input scan clock I/O (APR)			Timer 1 (T1)	
Analog In	Analog common (A⏷)		Port B	Digital common (D⏷)	**Dig-Ctr I/O**
	CH 0 HI (0H)			FIRSTPORTB Bit 0 (B0)	
	CH 0 LO (8L)			FIRSTPORTB Bit 1 (B1)	
	Analog common (A⏷)			FIRSTPORTB Bit 2 (B2)	
	CH 1 HI (1H)			FIRSTPORTB Bit 3 (B3)	
	CH 1 LO (9L)			FIRSTPORTB Bit 4 (B4)	
	Analog common (A⏷)			FIRSTPORTB Bit 5 (B5)	
	CH 2 HI (2H)			FIRSTPORTB Bit 6 (B6)	
	CH 2 LO (10L)			FIRSTPORTB Bit 7 (B7)	
	Analog common (A⏷)			Digital common (D⏷)	
	CH 3 HI (3H)			Counter 0 (CT0)	
	CH 3 LO (11L)			Counter 1 (CT1)	
Analog In	Analog common (A⏷)		Port C	Digital common (D⏷)	**Dig-Ctr I/O**
	CH 4 HI (4H)			FIRSTPORTC Bit 0 (C0)	
	CH 4 LO (12L)			FIRSTPORTC Bit 1 (C1)	
	Analog common (A⏷)			FIRSTPORTC Bit 2 (C2)	
	CH 5 HI (5H)			FIRSTPORTC Bit 3 (C3)	
	CH 5 LO (13L)			FIRSTPORTC Bit 4 (C4)	
	Analog common (A⏷)			FIRSTPORTC Bit 5 (C5)	
	CH 6 HI (6H)			FIRSTPORTC Bit 6 (C6)	
	CH 6 LO (14L)			FIRSTPORTC Bit 7 (C7)	
	Analog common (A⏷)			Digital common (D⏷)	
	CH 7 HI (7H)			Counter 2 (CT2)	
	CH 7 LO (15L)			Counter 3 (CT3)	

CE Declaration of Conformity

Manufacturer:	IOTech, Incorporated
Address:	25971 Cannon Road Cleveland, OH 44146 USA
Category:	Information technology equipment.

IOTech, Incorporated declares under sole responsibility that the product

USB-1616HS-2

to which this declaration relates is in conformity with the relevant provisions of the following standards or other documents:

EC EMC Directive 2004/108/EC: Electromagnetic Compatibility, EN 61326-1:2006, (IEC 61326-1:2005)

Emissions: Group 1, Class A

- EN 55022 (1993)/CISPR 22: Radiated and Conducted emissions.

Immunity: EN 61326-1:2006, (IEC 61326-1:2005)

- IEC 61000-4-2 (2001): Electrostatic Discharge immunity.
- IEC 61000-4-3 (2002): Radiated Electromagnetic Field immunity.
- IEC 61000-4-4 (2004): Electric Fast Transient Burst immunity.
- IEC 61000-4-5 (2001): Fast Surge immunity.
- IEC 61000-4-6 (2003): Radio Frequency Common Mode immunity.

To maintain the safety, emission, and immunity standards of this declaration, the following conditions must be met.

- The host computer, peripheral equipment, power sources, and expansion hardware must be CE compliant.
- Equipment must be operated in a controlled electromagnetic environment as defined by Standards EN 61326-1:2006, or IEC 61326-1:2005.
- Shielded wires must be used for all I/Os and must be less than 3 meters (9.75 feet) in length. Clips must be used with the AI-EXP48.
- The host computer must be properly grounded.
- The host computer must be USB2.0 compliant and IOtech USB cables (CA-179-x) must be used.
- A protective ESD wrist strap should be used when connecting or disconnecting leads from screw terminal blocks. Alternatively, unplug the unit from the host computer when making connections. Protective housings (IOtech p/n CN-241-12) can be placed over the removable terminal blocks to protect signals from ESD during operation.
- If external DC power is needed, a TR-2U power supply must be used.

Note: Data acquisition equipment may exhibit noise or increased offsets when exposed to high RF fields (>1V/m) or transients.

Declaration of Conformity based on tests conducted by Smith Electronics, Inc., Cleveland, OH 44141, USA in December, 2005. Test records are outlined in Smith Electronics Test Report “Personal Daq/3000 Series with PDQ30 Expansion Module” and “PDAQ3000-PDQ30 Addenda”. Further testing was conducted by Chomerics Test Services, Woburn, MA. 01801, USA in January, 2009. Test records are outlined in Chomerics Test report #EMI5245.09.

We hereby declare that the equipment specified conforms to the above Directives and Standards.

Carl Haapaoja, Director of Quality Assurance

Measurement Computing Corporation
10 Commerce Way
Suite 1008
Norton, Massachusetts 02766
(508) 946-5100
Fax: (508) 946-9500
E-mail: info@mccdaq.com
www.mccdaq.com