# PCI-2515

**16-Bit, 1 MS/s Analog** Input Board

# User's Guide

January 2019. Rev 9

## Trademark and Copyright Information

### 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

Chapter 1

Chapter 2

Chapter 3

# 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 PCI-2515 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

Additional information about PCI-2515 hardware is available on our website at www.mccdaq.com. You can also contact Measurement Computing Corporation with specific questions.

- Knowledgebase: kb.mccdaq.com
- Tech support form: www.mccdaq.com/support/support_form.aspx
- Email: techsupport@mccdaq.com
- Phone: 508-946-5100 and follow the instructions for reaching Tech Support

For international customers, contact your local distributor. Refer to the International Distributors section on our web site at www.mccdaq.com/International.

# Introducing the PCI-2515

## Overview: PCI-2515 features

The PCI-2515 board is a multifunction measurement and control board that is supported under popular Microsoft® Windows® operating systems.

The PCI-2515 provides the following features:

- Eight differential or 16 single-ended analog inputs with 16-bit resolution
  - Software-selectable analog input ranges: ±10 V, ±5 V, ±2 V, ±1 V, ±0.5 V, ±0.2 V, ±0.1V
- Two 16-bit, 1 MHz analog output channels with an output range of –10V to +10V
  - Software-selectable analog input ranges: ±10 V, ±5 V, ±2 V, ±1 V, ±0.5 V, ±0.2 V, ±0.1V
- 24 high-speed digital I/O lines
  - Up to 4 MHz scanning on all digital input lines[1]
- Two timer outputs
- Four 32-bit counters
- Synchronous analog I/O, digital I/O, and counter/timer I/O operations

---

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

# Installing the PCI-2515

## Unpacking

As with any electronic device, you should take care while handling to avoid damage from static electricity. Before removing the device 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.

Contact us immediately if any components are missing or damaged.

## Installing the software

Refer to the *MCC DAQ Quick Start* and the PCI-2515 product page on our website for information about the software that supports the device.

**Install the software before you install your device**

The driver needed to run the PCI-2515 is installed with the software. Therefore, you need to install the software package you plan to use before you install the hardware.

## Installing the PCI-2515

The PCI-2515 board is completely plug-and-play. There are no switches or jumpers to set on the board. Configuration is controlled by your system's BIOS.

**Before you install the PCI-2515**

**Enable Bus Mastering DMA**: For a PCI-2515 to operate properly, you must enable Bus Mastering DMA on the PCI slot where you will install the board. Make sure that your computer can perform Bus Mastering DMA for the applicable PCI slot. Some computers have BIOS settings that enable and disable Bus Mastering DMA. If your computer has this BIOS option, make sure you enable Bus Mastering DMA on the appropriate PCI slot.

Refer to your PC Owner's Manual for additional information regarding your PC and enabling Bus Mastering DMA for PCI slots.

**Install the MCC DAQ software**: The driver needed to run your PCI-2515 is installed with the MCC DAQ software. Therefore, you need to install the MCC DAQ software before you install your board. Refer to the *MCC DAQ Quick Start* for instructions on installing the software.

Perform the following procedure to install the PCI-2515:

1. Turn the computer off, open it up, and insert the board into an available PCI slot that has Bus Mastering DMA enabled.
2. Close the computer and turn it on.

   When connected for the first time, a **Found New Hardware** dialog opens when the operating system detects the device. When the dialog closes, the installation is complete. If you have not installed the MCC DAQ software, cancel the dialog and install it now.
3. To test the installation and configure your board, run the InstaCal utility installed in the previous section. Refer to the *MCC DAQ Quick Start* for information on how to initially set up and load InstaCal.

If the board has been powered-off for more than 10 minutes, allow the computer to warm up for at least 30 minutes before acquiring data. This warm-up period is required for the board to achieve its rated accuracy. The high-speed components used on the board generate heat, and it takes this amount of time for a board to reach steady state if it has been powered off for a significant amount of time.

# Configuring the hardware

All hardware configuration options on the PCI-2515 are software-controlled. You can select some of the configuration options using *Insta*Cal, such as the analog input configuration (16 single-ended or eight differential channels), and the edge used for pacing when using an external clock. Once selected, any program that uses the Universal Library initializes the hardware according to these selections.

**Information on signal connections**

For general information regarding signal connection and configuration, refer to the *Guide to Signal Connections* (available on our website at www.mccdaq.com/support/DAQ-Signal-Connections.aspx).

# Signal connections

The table below lists board connectors, applicable cables, and compatible accessory products for the PCI-2515.

Board connectors, cables, and compatible hardware

| | |
|---|---|
| Connector type | ■ 68-pin standard "SCSI TYPE III" female connector<br>■ HDMI connector (targeted for future expansion) |
| Compatible cables (for the 68-pin SCSI connector) | CA-68-3R — 68-pin ribbon cable; 3 feet.<br>CA-68-3S — 68-pin shielded round cable; 3 feet.<br>CA-68-6S — 68-pin shielded round cable; 6 feet. |
| Compatible accessory products | TB-100 terminal connector<br>RM-TB-100 |

## Connector pinout

16-channel single-ended mode pinout (8-channel differential signals in parentheses)

| Signal name | Pin | | Pin | Signal name |
|---|---|---|---|---|
| ACH0 (ACH0 HI) | 68 | • • | 34 | ACH8 (ACH0 LO) |
| AGND | 67 | • • | 33 | ACH1 (ACH1 HI) |
| ACH9 (ACH1 LO) | 66 | • • | 32 | AGND |
| ACH2 (ACH2 HI) | 65 | • • | 31 | ACH10 (ACH2 LO) |
| AGND | 64 | • • | 30 | ACH3 (ACH3 HI) |
| ACH11 (ACH3 LO) | 63 | • • | 29 | AGND |
| SGND | 62 | • • | 28 | ACH4 (ACH4 HI) |
| ACH12 (ACH4 LO) | 61 | • • | 27 | AGND |
| ACH5 (ACH5 HI) | 60 | • • | 26 | ACH13 (ACH5 LO) |
| AGND | 59 | • • | 25 | ACH6 (ACH6 HI) |
| ACH14 (ACH6 LO) | 58 | • • | 24 | AGND |
| ACH7 (ACH7 HI) | 57 | • • | 23 | ACH15 (ACH7 LO) |
| NC | 56 | • • | 22 | XDAC0 |
| NC | 55 | • • | 21 | XDAC1 |
| NEGREF (reserved for self-calibration) | 54 | • • | 20 | POSREF (reserved for self-calibration) |
| GND | 53 | • • | 19 | +5V |
| A1 | 52 | • • | 18 | A0 |
| A3 | 51 | • • | 17 | A2 |
| A5 | 50 | • • | 16 | A4 |
| A7 | 49 | • • | 15 | A6 |
| B1 | 48 | • • | 14 | B0 |
| B3 | 47 | • • | 13 | B2 |
| B5 | 46 | • • | 12 | B4 |
| B7 | 45 | • • | 11 | B6 |
| C1 | 44 | • • | 10 | C0 |
| C3 | 43 | • • | 9 | C2 |
| C5 | 42 | • • | 8 | C4 |
| C7 | 41 | • • | 7 | C6 |
| GND | 40 | • • | 6 | TTL TRG |
| CNT1 | 39 | • • | 5 | CNT0 |
| CNT3 | 38 | • • | 4 | CNT2 |
| TMR1 | 37 | • • | 3 | TMR0 |
| GND | 36 | • • | 2 | XAPCR |
| GND | 35 | • • | 1 | XDPCR |

PCI slot ↓

# Cabling

Use a CA-68-3R 68-pin ribbon expansion cable (Figure 1), or a CA-68-3S (3-foot) or CA-68-6S (6-foot) 68-pin shielded expansion cable (Figure 2) to connect signals to the PCI-2515 board.)

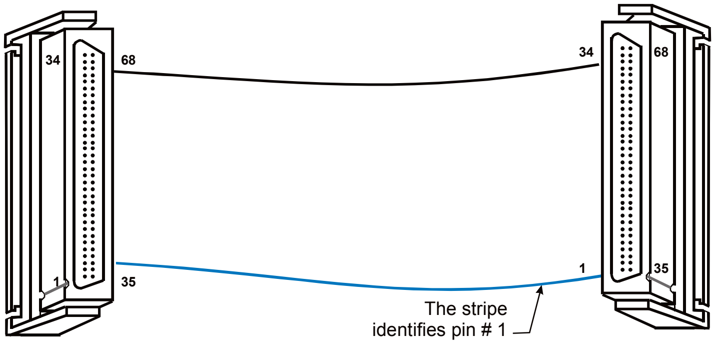


Figure 1. CA-68-3R cable

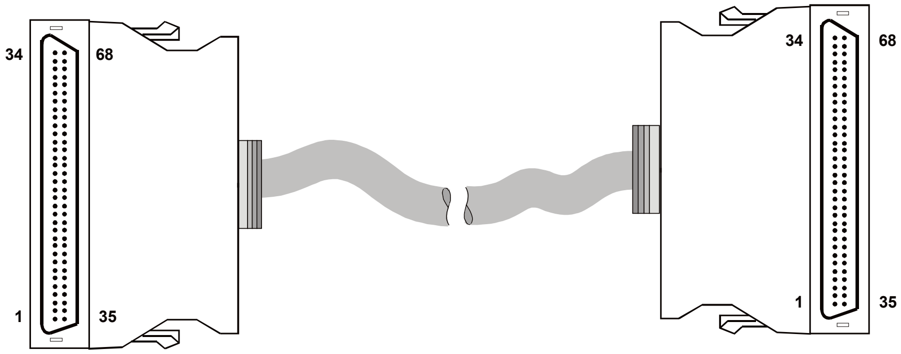


Figure 2. CA-68-3S and CA-68-6S cable

## Field wiring and signal termination

You can use the following MCC screw terminal boards to terminate field signals and route them into the PCI-2515 board using the CA-68-3R, CA-68-3S, or CA-68-6S cable:

- **TB-100**: Termination board with screw terminals.
- **RM-TB-100**: 19-inch rack mount kit for the TB-100 termination board.

Details on these products are available on our website.

# Using multiple PCI-2515s per PC

PCI-2515 features can be replicated up to four times, as up to four boards can be installed in a single host PC. The serial number on each PCI-2515 distinguishes one from another. You can operate multiple PCI-2515 boards synchronously. To do this, set up one PCI-2515 with the pacer pin you want to use (XAPCR or XDPCR) configured for output. Set up the PCI-2515 boards you want to synchronize to this board with the pacer pin you want to use (XAPCR or XDPCR) configured for input. Wire the pacer pin configured for output to each of the pacer input pins that you want to synchronize.

# Functional Details

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

- a block diagram of board functions
- information on how to use the signals generated by the board
- diagrams of signals using default or conventional board settings

## PCI-2515 block diagram

Figure 3 is a simplified block diagram of the PCI-2515. This board provides all of the functional elements shown in the figure.

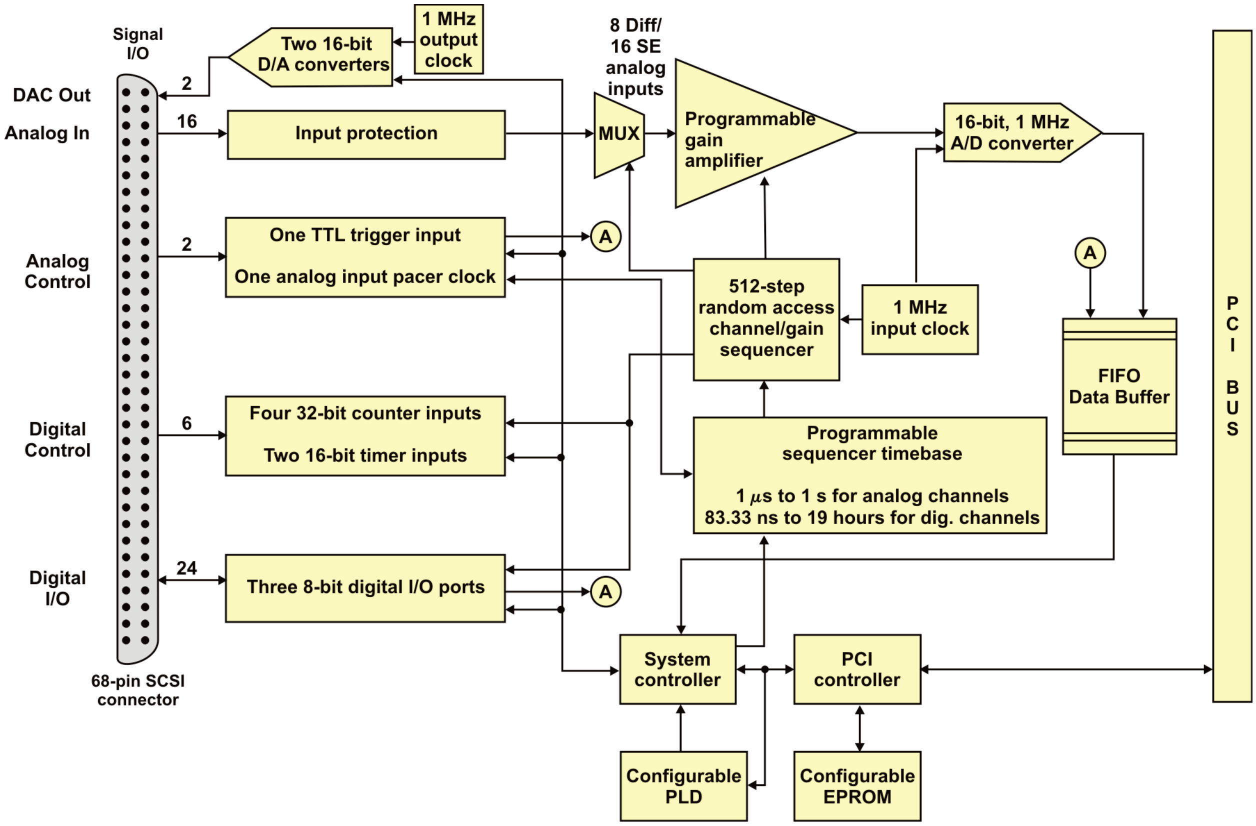


Figure 3. PCI-2515 functional block diagram

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

The PCI-2515 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.

The ability to scan digital and counter channels along with analog channels provides for a more deterministic collection of data.

# Bus mastering DMA

The PCI-2515 supports bus mastering DMA. With multiple DMA channels, analog, digital, and counter input data, as well as digital output data, can flow between the PC and the PCI-2515 without consuming valuable CPU time. The driver supplied with the PCI-2515 automatically uses bus mastering DMA to efficiently conduct I/O from the PC to the PCI-2515.

# Analog input

The PCI-2515 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 PCI-2515 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 1 µs settling time for the acquisition of each channel, 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 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 4 shows a simple acquisition. The scan is programmed pre-acquisition and is made up of six analog channels (Ch0, Ch1, Ch3, Ch4, Ch6, Ch7). Each of these analog channels can have a different gain. The acquisition is triggered and the samples stream to the PC via DMA. Using the default settling time, 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 second. The maximum scan frequency is 1 divided by 6 µs, or 166,666 Hz.

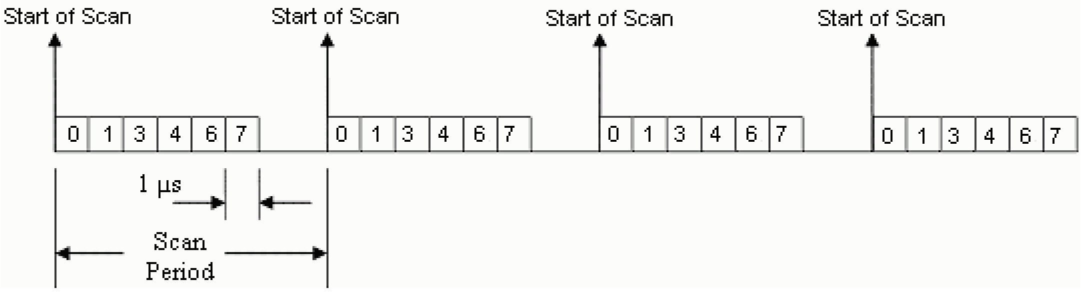


Figure 4. Analog channel scan of voltage inputs example

# Analog output

The PCI-2515 has two 16-bit, 1 MHz analog output channels.

The channels have an output range of -10V to +10V. Using Bus Mastering DMA, each D/A output can continuously output a waveform at up to 1 MHz. In addition, a program can asynchronously output a value to any of the D/A channels for non-waveform applications, if 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 on-board programmable clock can generate updates ranging from 1 Hz to 1 MHz.
- **External output scan clock (XDPCR)**: A user-supplied external clock.
- **Internal input scan pacer clock**: The internal ADC pacer clock can pace both the D/A and the analog input.
- **External input scan pacer clock (XAPCR)**: The external ADC pacer clock can pace both the D/A and the analog input.

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

The example shown in Figure 5 adds two DACs a 16-bit digital pattern output to the example presented in Figure 4.

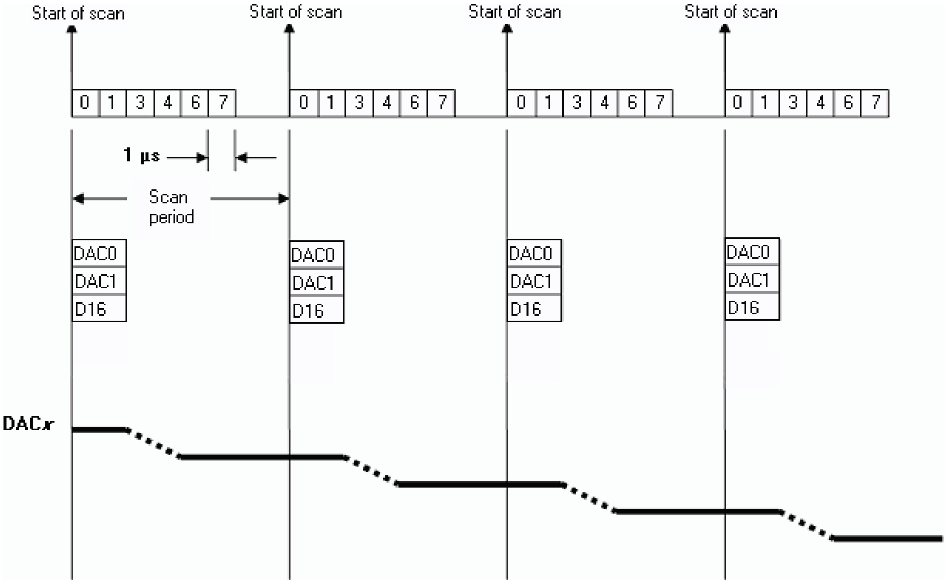


Figure 5. Analog channel scan of voltage inputs and streaming analog outputs example

This example updates both DACs and the 16-bits of digital I/O. These updates happen at the same time as the acquisition pacer clock—also called the input scan clock. Both 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 PCI bus to the PCI-2515 by the DMA.

In this example, the outputs are updated by the input scan clock, but you can also 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.

# Digital I/O

Twenty-four TTL-level digital I/O lines are included in each PCI-2515. 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. In both modes, adding digital input scans has no effect 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 any time 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 PCI-2515 supports digital pattern generation with bus mastering DMA. 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 on-board 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 PCI-2515 supports the following trigger modes to accommodate certain measurement situations.

## Hardware analog triggering

The PCI-2515 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, 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, 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, you would 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 (TTL TRG), allowing TTL-level triggering with latencies guaranteed to be less than 1 µs. You can program both 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 to detect the trigger event.

### Analog triggering

You can select any analog channel in the scan as the trigger channel. You can program the trigger level, the rising or falling edge, 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 PCI-2515 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 PCI-2515 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.

**Set pre-trigger > 0 when using counter as trigger source**

When using a counter for a trigger source, you should use a pre-trigger with a value of at least 1. Since all counters start at zero with the first scan, there is no valid reference regarding 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 will be legitimate.

## 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 PCI-2515 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 PCI-2515. Each counter accepts frequency inputs up to 20 MHz.

PCI-2515 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 has been reached. See counter mode descriptions below.

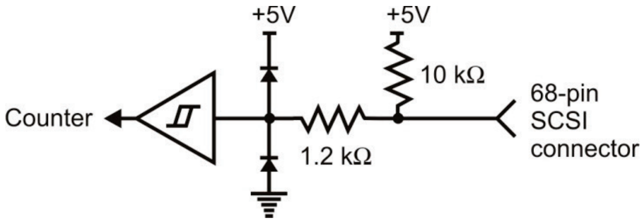


Figure 6. Typical PCI-2515 counter channel

## 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 to 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 channel—known as the *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 PCI-2515.

### 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 PCI-2515 pulses once every scan period to indicate the start of a scan group—latches the count, so 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 7. 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.

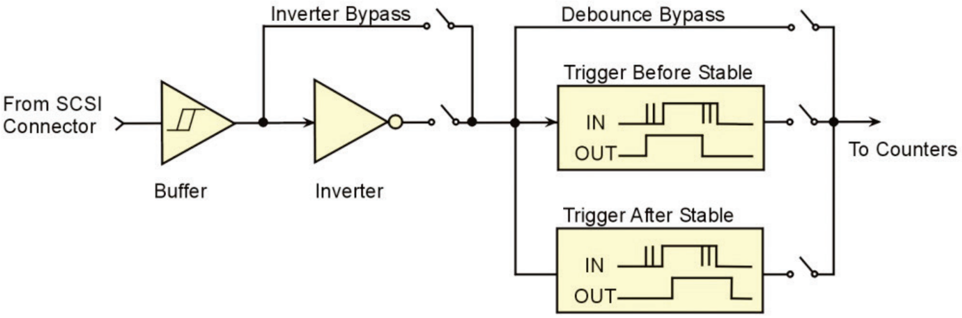


Figure 7. Debounce model block diagram

### 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.

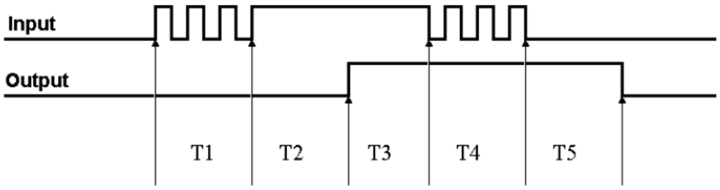


Figure 8. Debounce module – trigger after stable mode

The following time periods (T1 through T5) pertain to Figure 8. 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.

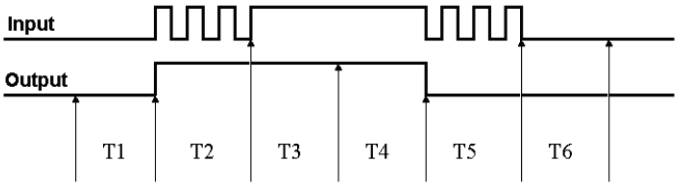


Figure 9. Debounce module – Trigger before stable mode

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 10 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.

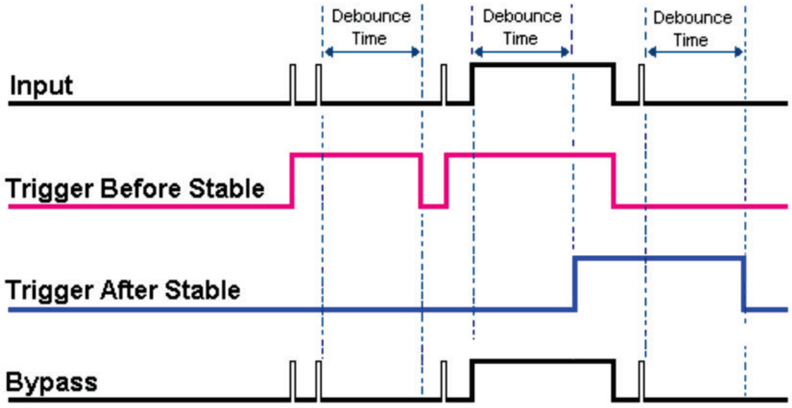


Figure 10. Example of two debounce modes interpreting the same signal

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.

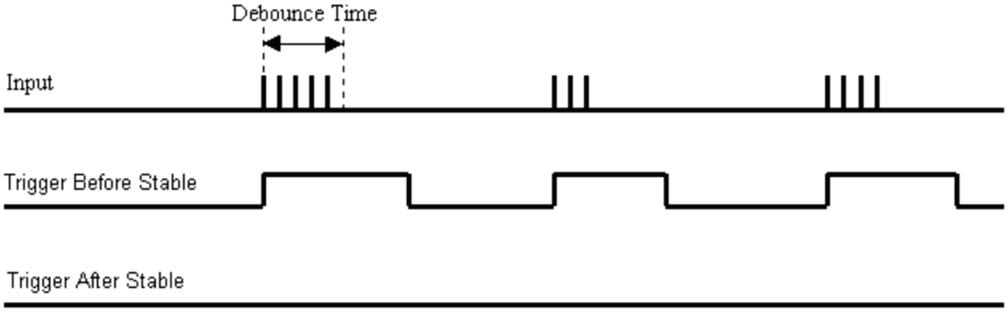


Figure 11.Optimal debounce time for trigger before stable mode

*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 12.

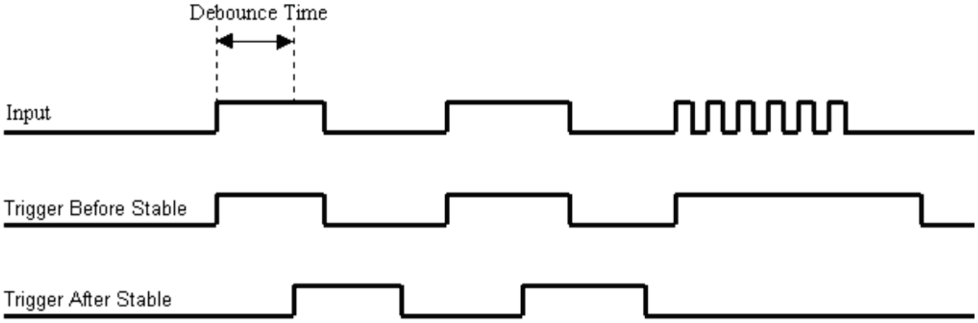


Figure 12. Optimal debounce time for trigger after stable mode

## 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.

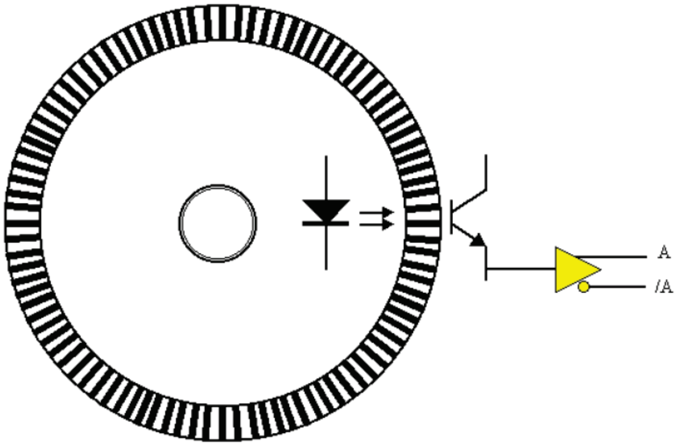
The *encoder mode* allows the PCI-2515 to make use of data from optical incremental quadrature encoders. In encoder mode, the PCI-2515 accepts *single-ended* inputs. When reading phase A, phase B, and index Z signals, the PCI-2515 provides positioning, direction, and velocity data.

The PCI-2515 can receive input from up to two encoders.

The PCI-2515 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 -5 V to +10 V 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.



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.

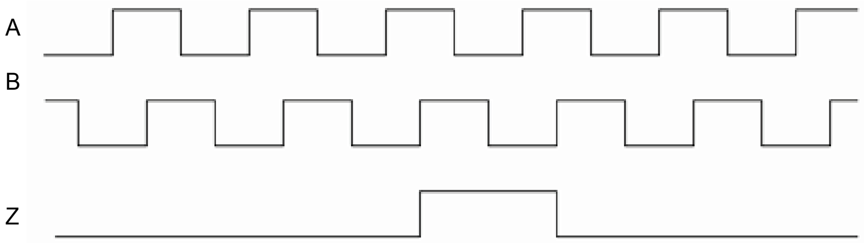


Figure 13. Representation of quadrature encoder outputs: A, B, and Z

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 gives 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 leads B and when the encoder is rotating counterclockwise, A lags behind 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 gives 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 PCI-2515 to an encoder

You can use up to two encoders with each PCI-2515 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 PCI-2515.
- Connect each encoder ground to GND.

You can also connect external pull-up resistors to the PCI-2515 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 PCI-2515. Lower values of pull-up resistors cause less distortion, but also cause the encoder's output driver to pull down with more current.

### Connecting external pull-up resistors to the PCI-2515

For open-collector outputs, you can connect external pull-up resistors to the PCI-2515's counter input terminal blocks. You can place a pull-up resistor between any input channel and the provided +5 V power supply.

Choose a pull-up resistor value based on the encoder's output drive capability and the input impedance of the PCI-2515. 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 14 shows the connections for one encoder to a module.

The following figure illustrates connections for one encoder to a 68-pin SCSI connector on a PCI-2515.

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 CTR0, "B" would be connected to CTR1.

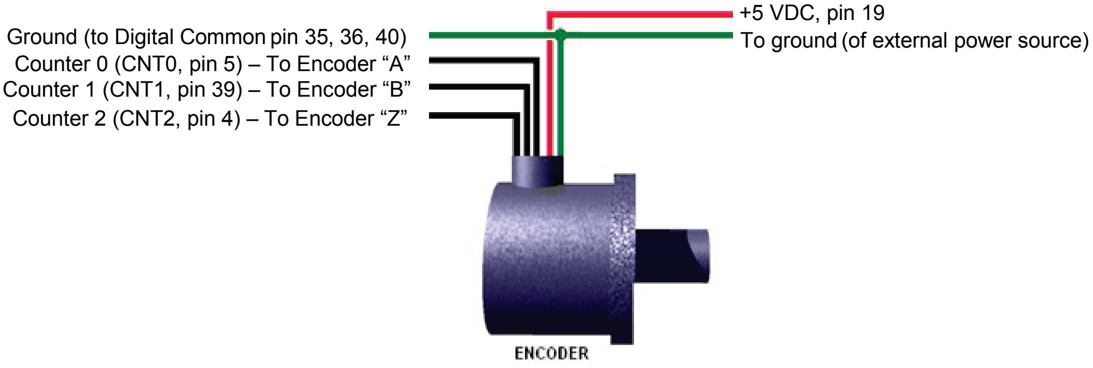


Figure 14. Encoder connections to pins on the SCSI connector*

* Connections can instead be made to the associated screw-terminals of a connected TB-100 terminal connector option.

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.

If the encoder stops rotating, but is vibrating (due to it being mounted to a machine), you can use the debounce feature to eliminate false edges. Choose an appropriate debounce time and apply it to each encoder channel. Refer to the *Debounce modes* section in the *Functional Details* chapter in this manual for additional information regarding debounce times.

You can get the relative position and velocity from the encoder. However, during an acquisition, you cannot get data that is relative to the Z-position until the encoder locates the Z-reference.

Note that the number of Z-reference crossings can be tabulated. If the encoder was turning in only one direction, then the Z-reference crossings equal the number of complete revolutions. This means that the data streaming to the PC *is relative position*, *period = 1/velocity*, and *revolutions*.

A typical acquisition might take six readings from the PCI-2515 as illustrated below. The user determines the scan rate and the number of scans to take.

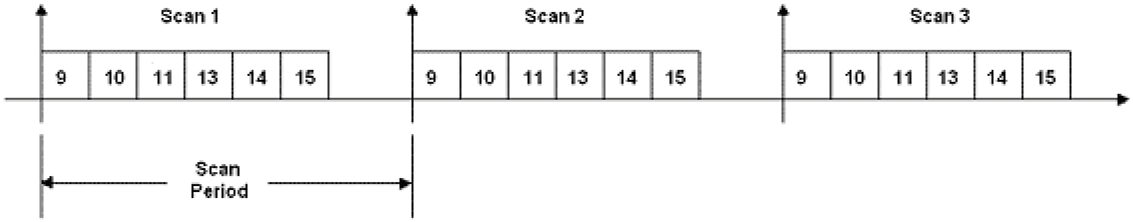


Figure 15. PCI-2515 acquisition of six readings per scan

**Note**: Digital channels do not take up analog channel scan time.

In general, the output of each channel's counter is latched at the beginning of each scan period (called the *start-of-scan*.) Every time the PCI-2515 receives a *start-of-scan* signal, the counter values are latched and are available to the PCI-2515.

The PCI-2515 clears all counter channels at the beginning of the acquisition. This means that the values returned during scan period 1 are always zero. The values returned during scan period 2 reflect what happened during scan period 1.

The scan period defines the timing resolution for the PCI-2515. If you need a higher timing resolution, shorten the scan period.

**Wiring for two encoders:** Figure 16 shows the single-ended connections for two encoders. Differential connections do not apply.

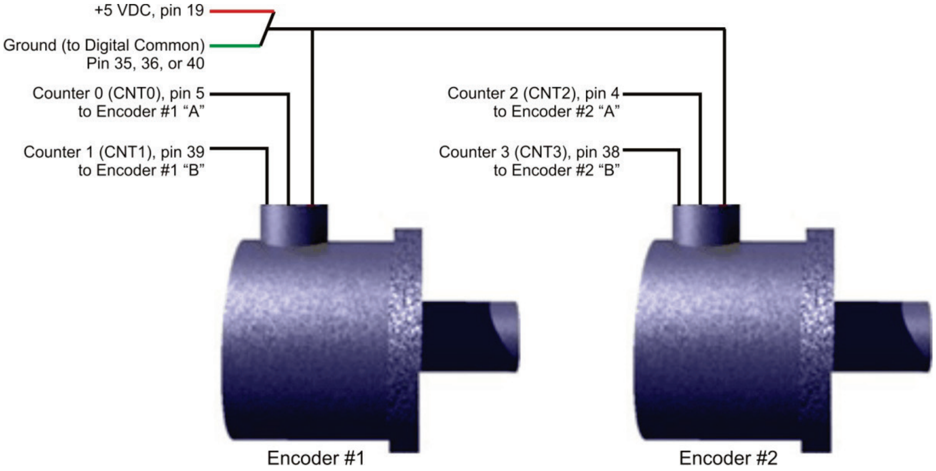


Figure 16. Two encoders connected to pins on the SCSI connector*

* Connections can instead be made to the associated screw-terminals of a connected TB-100 terminal connector option.

Each signal (A, B) can be connected as a single-ended connection with respect to the common digital ground (GND). Both encoders can draw their power from the +5 V power output (pin 19) on the 68-pin SCSI connector.

Connect each encoder's power input to +5 V power. Connect the return to digital common (GND) on the same connector. Make sure that the current output spec is not violated.

With the encoders connected in this manner, there is no relative positioning information available on encoder #1 or #2 since there is no Z signal connection for either. Therefore only distance traveled and velocity can be measured for each encoder.

# Timer outputs

Two 16-bit timer outputs are built into the PCI-2515. Each timer can generate a different square wave with a programmable frequency in the range of 16 Hz to 1 MHz.

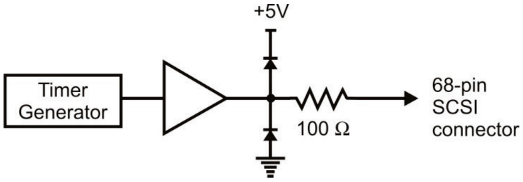


Figure 17. Typical PCI-2515 timer channel

## 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. The table below lists some examples.

Timer output frequency examples

| **Divisor** | **Timer output frequency** |
|---|---|
| 1 | 1 MHz |
| 100 | 10 kHz |
| 1000 | 1 kHz |
| 10000 | 100 Hz |
| 65535 | 15.259 Hz |

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

# Using detection setpoints for output control

## What are detection setpoints?

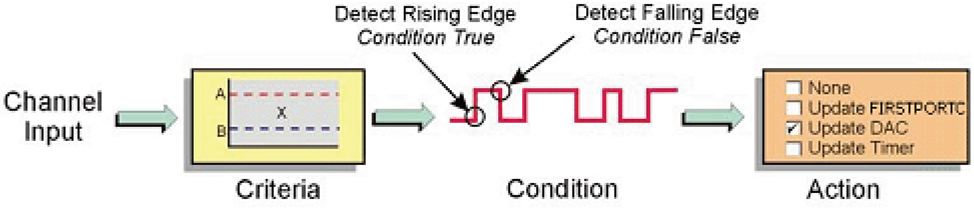
With the PCI-2515's setpoint configuration feature, you can configure up to 16 detection setpoints associated with channels in a scan group. Each setpoint can update the following, allowing for real-time control based on acquisition data:

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

## Setpoint configuration overview

You can program each detection setpoint 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).



A digital detect signal is used to indicate when a signal condition is *True* or *False*—for example, whether 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 26.

| **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)<br>■ Below A (X < A)<br>■ Above B (X > B) | *True* only:<br>■ If *True*, then output value 1<br>■ If *False*, then perform no action<br>*True* and *False*:<br>■ If *True*, then output value 1<br>■ If *False*, then output value 2 |
| Window* (non-hysteresis mode) | ■ Inside (B < X < A)<br>■ Outside: B > X; or, X > A | *True* only<br>■ If *True*, then output value 1<br>■ If *False*, then perform no action<br>*True* and *False*<br>■ If *True*, then output value 1<br>■ If *False*, then output value 2 |
| Window* (hysteresis mode) | ■ Above A (X > A)<br>■ Below (B X < B) *(Both conditions are checked when in hysteresis mode* | Hysteresis mode (forced update)<br>■ If X > A is *True*, then output value 2 until X < B is *True*, then output value 1.<br>■ If X < B is *True*, then output value 1 until X > A is *True*, then output value 2.<br>This is saying:<br>(a) If the input signal is outside the window *high*, then output value 2 until the signal goes outside the window *low,* and<br>(b) if the signal is outside the window *low*, then 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.)

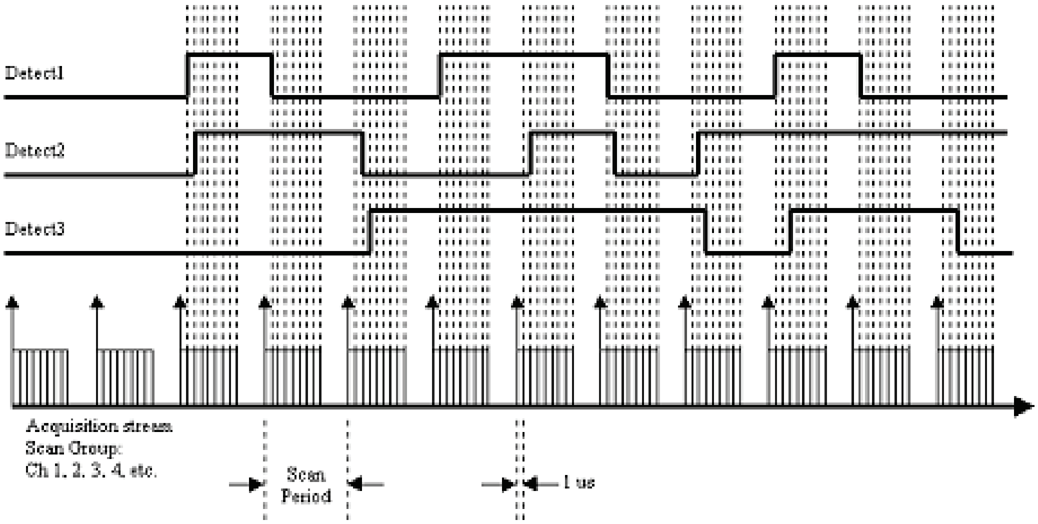


Figure 18. Example diagram of detection signals for channels 1, 2, and 3

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 PCI-2515 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 PCI-2515 through software.

### Set low limit

You can set the 16-bit low limit (*limit B*) when configuring the PCI-2515 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 is not met—in other words, the condition is *False*.
- A value of 1 indicates that the criteria has 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`)<br>*False* (`0`) | `0` | `0` | `0` | `0` | `0` | `0` | `0` | `0` | `0` | `0` | `0` | `1` | `0` | `0` | `1` | `1` |
| | <<< Most significant bit | | | | | | | Least significant bit >>> | | | | | | | | |

From the above table 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<br>(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 *minus* 1.0 V. |
| 4 | Within window (between limit A and limit B)<br>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. |

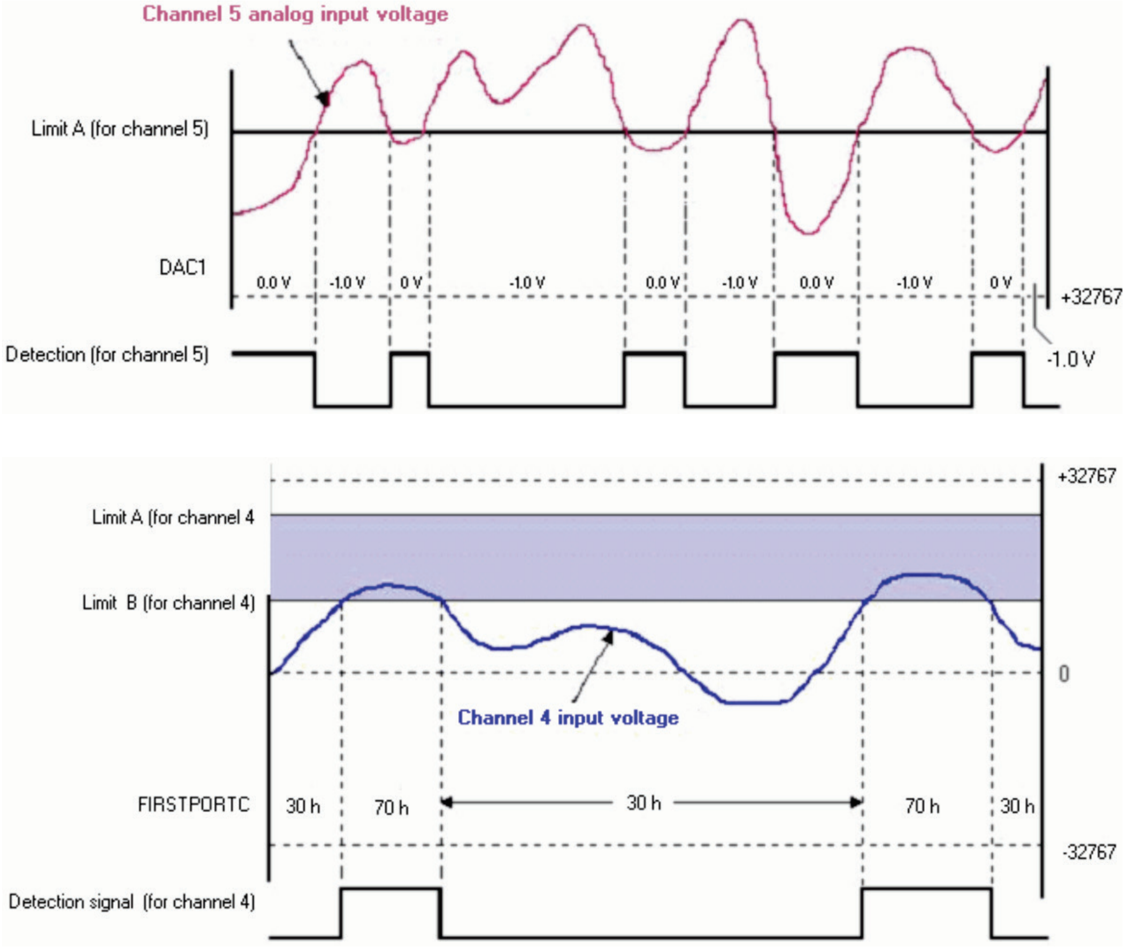


Figure 19. Analog inputs with setpoints update on *True* and *False*

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 *minus*1.0V.

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

The figure below 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.

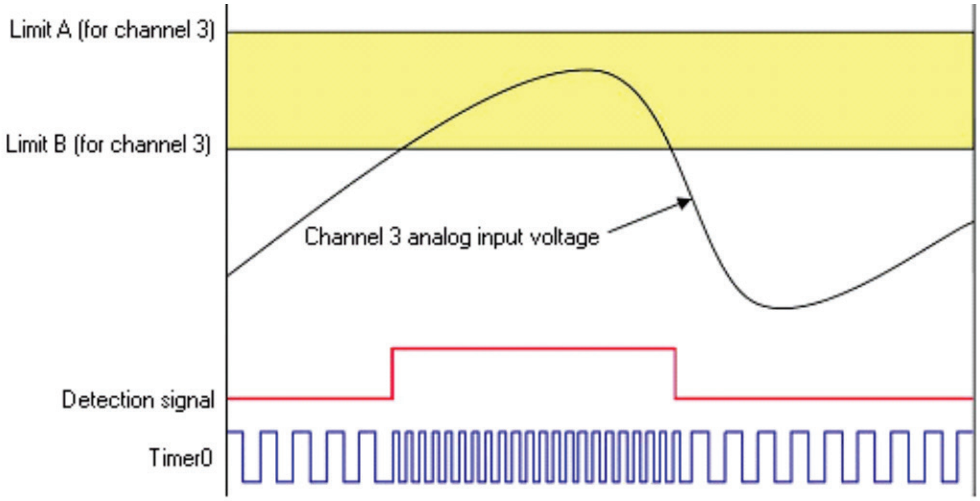


Figure 20. Timer output update on *True* and *False*

### 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

The figure below 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.

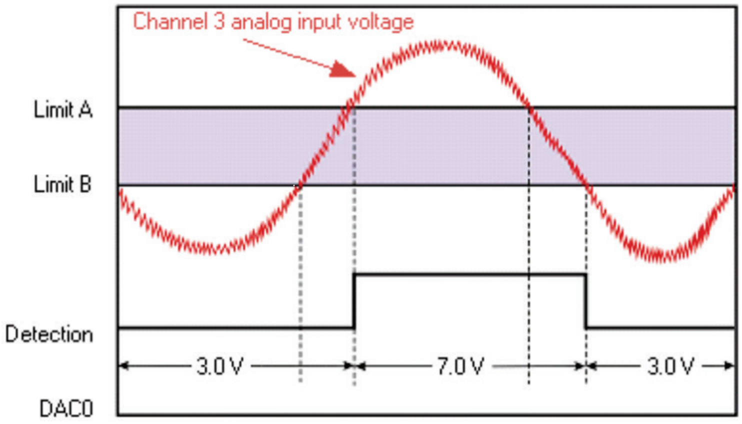


Figure 21. Channel 3 in hysteresis mode

### 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 DAC1 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, DAC1 will be updated with 3.0V.

Analog input Channel 7 also has an inside-the-window setpoint applied. Whenever channel 7's input goes inside the programmed window, DAC1 is updated with - 7.0V.

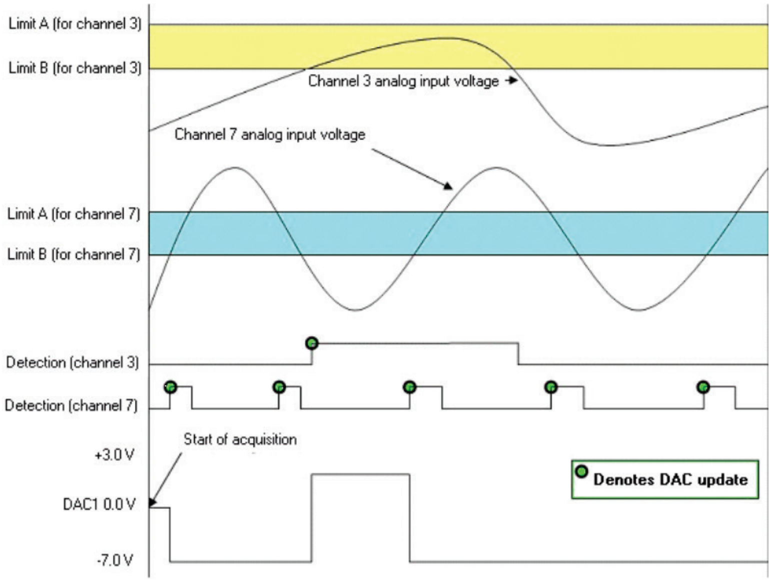


Figure 22. Using two criteria to control an output*

The update on *True* only mode was selected, and therefore the updates for DAC1 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 DAC1 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). Alternatively, one of the DAC output channels, or timer outputs, could be updated with a value.

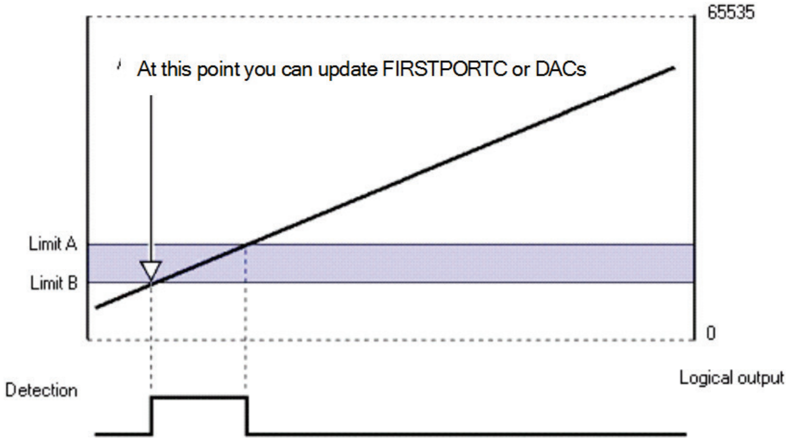


Figure 23. Channel 1 in totalizing counter mode, inside the window setpoint

## 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. You can also program each setpoint with:

- a 16-bit DAC update value, and any one of the two DAC outputs can be updated in real time
- 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 DAC pacer 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 outputs, 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.

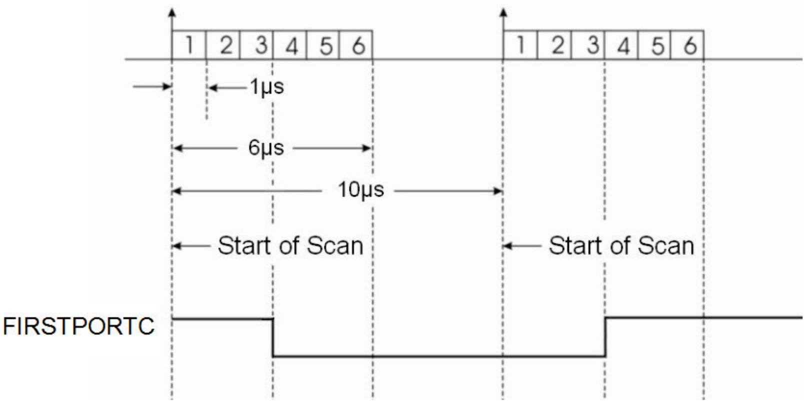


Figure 24. Example of FIRSTPORTC or DAC latency

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 10th, 100th, 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 PCI-2515

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.

## Chapter 5

# Specifications

**All specifications are subject to change without notice.**
**Typical for 25 °C unless otherwise specified.**
**Specifications in *italic text* are guaranteed by design.**

## Analog input

Table 1. Analog input specifications

| **Parameter** | **Specification** |
|---|---|
| A/D converter type | Successive approximation |
| Resolution | 16 bits |
| Number of channels | 16 single-ended/8 differential, software selectable |
| Input ranges (SW 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 max |
| Nonlinearity (differential) | ±1 LSB max |
| A/D pacing | Onboard input scan clock, external source (XAPCR) |
| Trigger sources and modes | Refer to Table 7 |
| Data transfer | DMA |
| Configuration memory | Programmable I/O |
| Maximum usable input voltage + common mode voltage (CMV + $V_{in}$) | Range: ±10 V, ±5 V. 10.5 V max<br>Range: ±2 V, ±1 V, ±0.5 V, ±0.2 V, ±0.1 V. 6.0 V max |
| *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)* |
| *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* | 1.5 | Note 2 |
| –5 V to 5 V | | *0.031% + 0.009%* | *14 + 9* | 2.0 | |
| –2 V to 2 V | | *0.031% + 0.010%* | *14 +10* | 1.6 | |
| –1 V to 1 V | | *0.031% + 0.02%* | *14 + 12* | 2.5 | |
| –500 mV to 500 mV | | *0.031% + 0.04%* | *14 +18* | 4.0 | |
| –200 mV to 200 mV | | *0.036% + 0.075%* | *14 +12* | 5.0 | |
| –100 mV to 100 mV | | *0.042% + 0.15%* | *14 +18* | 9.0 | |

**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.

**Note 2:** Noise reflects 10,000 samples at 1 MHz, typical, differential short, using CA-68-3S cable.

# 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 3. Analog output specifications

| **Parameter** | **Specification** |
|---|---|
| Channels | 2 |
| Resolution | 16 bits |
| Data buffer | PC-based memory |
| Output voltage range | ±10 V |
| Output current | ±10 mA |
| Offset error | ±0.0045 V max |
| 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 max, resolution 20.83 ns |
| Settling time | 2 µs to rated accuracy |
| Pacer sources | Four programmable sources:<br>■ Onboard output scan clock, independent of scanning input clock<br>■ Onboard input scan clock<br>■ External output scan clock (XDPCR), independent of external input scan clock (XAPCR)<br>■ External input scan clock (XAPCR) |
| Trigger sources | Start of input scan |

# Digital input/output

Table 4. Digital input/output specifications

| **Parameter** | **Specification** |
|---|---|
| Number of I/O | 24 |
| Ports | Three banks of eight.<br>Each port is programmable as input or output |
| *Input scanning mode* | *Asynchronous, under program control at any time relative to input scanning* |
| Configuration | 10 kΩ pull-up to +5 V, 20 pf to analog common |
| Input protection | ±15 kV ESD clamp diodes |
| *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 12 mA per pin, 200 mA total continuous |
| Digital input pacing | Onboard clock, external input scan clock (XAPCR) |
| Digital output pacing | Four programmable sources:<br>■ Onboard output scan clock, independent of input scan clock<br>■ Onboard input scan clock<br>■ External output scan clock (XDPCR), independent of external input scan clock (XAPCR)<br>■ External input scan clock (XAPCR) |
| Digital input trigger sources and modes | Refer to Table 7 |
| Digital output trigger sources | Start of input scan |
| Data transfer | DMA |

| **Parameter** | **Specification** |
|---|---|
| Sampling/update rate | 4 MHz max (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 5. Counter specifications

| **Parameter** | **Specification** |
|---|---|
| Channels | 4 independent |
| Resolution | 32 bits |
| Input frequency | 20 MHz max |
| Input signal range | –5 V to 10 V |
| Input characteristics | 10 kΩ pull-up, ±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 | 30 ppm (0 °C to 50 °C) |
| Counter read pacer | Onboard input scan clock, external input scan clock (XAPCR) |
| Trigger sources and modes | Refer to Table 7 |
| Programmable mode | Counter |
| Counter mode options | Totalize, clear on read, rollover, stop at all Fs, 16-bit 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 6. Input sequencer specifications

| **Parameter** | **Specification** |
|---|---|
| Input scan clock sources<br>(Note 3) | Internal:<br>▪ Analog channels from 1 µs to 1 sec in 20.83 ns steps<br>▪ Digital channels and counters from 250 ns to 1 sec in 20.83 ns steps<br>External. TTL-level input (XAPCR):<br>▪ Analog channels down to 1 µs min<br>▪ Digital channels and counters down to 250 ns min |
| Programmable parameters per scan | Programmable channels (random order), programmable gain |
| Depth | 512 locations |
| Onboard channel-to-channel scan rate | ▪ Analog: 1 MHz max<br>▪ Digital: 4 MHz if no analog channels are enabled, 1 MHz with analog channels enabled |
| External input scan clock (XAPCR) maximum rate | ▪ Analog: 1 MHz<br>▪ 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<br>▪ Logical one: 2.4 V to 5.0 V |
| Minimum pulse width | 50 ns high, 50 ns low |

**Note 3:** 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 scan rates up to 83.33 ns for digital-only scans.

# Trigger sources and modes

Table 7. Trigger sources and modes

| | |
|---|---|
| Input scan trigger sources | ▪ Single channel analog hardware trigger<br>▪ Single channel analog software trigger<br>▪ External-single channel digital trigger (TTL TRG input)<br>▪ Digital pattern trigger<br>▪ Counter/totalizer trigger |
| Input scan triggering modes | Single channel analog hardware trigger: The first analog input channel in the scan is the analog trigger channel.<br>▪ Input signal range: –10 V to +10 V max<br>▪ Trigger level: Programmable (12-bit resolution)<br>▪ Latency: 350 ns typical<br>▪ Accuracy: ±0.5% of reading, ±2 mV offset max<br>▪ Noise: 2 mV RMS typical |
| | Single channel analog software trigger: The first analog input channel in the scan is the analog trigger channel.<br>▪ Input signal range: Anywhere within range of the trigger channel<br>▪ Trigger level: Programmable (16-bit resolution)<br>▪ Latency: One scan period max |
| | External-single channel digital trigger (TTL trigger input):<br>▪ Input signal range: –15 V to +15 V max<br>▪ Trigger level: TTL-level sensitive<br>▪ Minimum pulse width: 50 ns high, 50 ns low<br>▪ Latency: One scan period max |
| | ▪ 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.<br>▪ Latency: One scan period max |
| | ▪ 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.<br>▪ Latency: One scan period max |

# Frequency/pulse generators

Table 8. Frequency/pulse generator specifications

| **Parameter** | **Specification** |
|---|---|
| Channels | 2 x 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 min @ –1.0 mA, 2.9 V min @ –400 µA |
| Low-level output voltage | 0.4 V max @ 400 µA |

# Power consumption

Table 9. Power consumption specifications

| **Parameter** | **Specification** |
|---|---|
| Power consumption (per board) | 3 W |

# PCI compatibility

Table 10. PCI compatibility specifications

| **Parameter** | **Specification** |
|---|---|
| PCI bus | PCI r2.2 compliant, universal 3.3 V/5 V signaling support, compatible with PCI-X |

# Environmental

Table 11. Environmental specifications

| **Parameter** | **Specification** |
|---|---|
| Operating temperature range | 0 °C to +60 °C |
| Storage temperature range | –40 °C to +80 °C |
| Relative humidity | 0% to 95% non-condensing |

# Mechanical

Table 12. Mechanical specifications

| **Parameter** | **Specification** |
|---|---|
| Vibration | MIL STD 810E cat 1 and 10 |
| Dimensions | 165 mm (W) x 15 mm (D) x 108 mm (H) (6.5" x 0.6" x 4.2") |
| Weight | 160 g (0.35 lbs) |

# Main connector and pinout

Table 13. Main connector specifications

| **Parameter** | **Specification** |
|---|---|
| Connector type | 68-pin standard "SCSI TYPE III" female connector<br>HDMI connector (targeted for future expansion) |
| Compatible cables (for the 68-pin SCSI connector) | CA-68-3R — 68-pin ribbon cable; 3 feet.<br>CA-68-3S — 68-pin shielded round cable; 3 feet.<br>CA-68-6S — 68-pin shielded round cable; 6 feet. |
| Compatible accessory products | TB-100 termination board with screw terminals |
|  | RM-TB-100, 19-inch rack mount kit for TB-100 |

Table 14. 16-channel single-ended pinout

| Pin | Function | Pin | Function |
|---|---|---|---|
| 68 | ACH0 | 34 | ACH8 |
| 67 | AGND | 33 | ACH1 |
| 66 | ACH9 | 32 | AGND |
| 65 | ACH2 | 31 | ACH10 |
| 64 | AGND | 30 | ACH3 |
| 63 | ACH11 | 29 | AGND |
| 62 | SGND (low level sense – not for general use) | 28 | ACH4 |
| 61 | ACH12 | 27 | AGND |
| 60 | ACH5 | 26 | ACH13 |
| 59 | AGND | 25 | ACH6 |
| 58 | ACH14 | 24 | AGND |
| 57 | ACH7 | 23 | ACH15 |
| 56 | NC | 22 | XDAC0 |
| 55 | NC | 21 | XDAC1 |
| 54 | NEGREF (reserved for self-calibration) | 20 | POSREF (reserved for self-calibration) |
| 53 | GND | 19 | +5 V (refer to Note 4) |
| 52 | A1 | 18 | A0 |
| 51 | A3 | 17 | A2 |
| 50 | A5 | 16 | A4 |
| 49 | A7 | 15 | A6 |
| 48 | B1 | 14 | B0 |
| 47 | B3 | 13 | B2 |
| 46 | B5 | 12 | B4 |
| 45 | B7 | 11 | B6 |
| 44 | C1 | 10 | C0 |
| 43 | C3 | 9 | C2 |
| 42 | C5 | 8 | C4 |
| 41 | C7 | 7 | C6 |
| 40 | GND | 6 | TTL TRG |
| 39 | CNT1 | 5 | CNT0 |
| 38 | CNT3 | 4 | CNT2 |
| 37 | TMR1 | 3 | TMR0 |
| 36 | GND | 2 | XAPCR (input scan clock) |
| 35 | GND | 1 | XDPCR (output scan clock) |

Table 15. 8-channel differential pinout

| Pin | Function | Pin | Function |
|---|---|---|---|
| 68 | ACH0 HI | 34 | ACH0 LO |
| 67 | AGND | 33 | ACH1 HI |
| 66 | ACH1 LO | 32 | AGND |
| 65 | ACH2 HI | 31 | ACH2 LO |
| 64 | AGND | 30 | ACH3 HI |
| 63 | ACH3 LO | 29 | AGND |
| 62 | SGND (low level sense – not for general use) | 28 | ACH4 HI |
| 61 | ACH4 LO | 27 | AGND |
| 60 | ACH5 HI | 26 | ACH5 LO |
| 59 | AGND | 25 | ACH6 HI |
| 58 | ACH6 LO | 24 | AGND |
| 57 | ACH7 HI | 23 | ACH7 LO |
| 56 | NC | 22 | XDAC0 |
| 55 | NC | 21 | XDAC1 |
| 54 | NEGREF (reserved for self-calibration) | 20 | POSREF (reserved for self-calibration) |
| 53 | GND | 19 | +5 V (refer to Note 4) |
| 52 | A1 | 18 | A0 |
| 51 | A3 | 17 | A2 |
| 50 | A5 | 16 | A4 |
| 49 | A7 | 15 | A6 |
| 48 | B1 | 14 | B0 |
| 47 | B3 | 13 | B2 |
| 46 | B5 | 12 | B4 |
| 45 | B7 | 11 | B6 |
| 44 | C1 | 10 | C0 |
| 43 | C3 | 9 | C2 |
| 42 | C5 | 8 | C4 |
| 41 | C7 | 7 | C6 |
| 40 | GND | 6 | TTL TRG |
| 39 | CNT1 | 5 | CNT0 |
| 38 | CNT3 | 4 | CNT2 |
| 37 | TMR1 | 3 | TMR0 |
| 36 | GND | 2 | XAPCR (input scan clock) |
| 35 | GND | 1 | XDPCR (output scan clock) |

**Note 4:** 5 V output, up to 500 mA.

CE

# Declaration of Conformity

Manufacturer: Measurement Computing Corporation
Address: 10 Commerce Way
Norton, MA 02766
USA
Category: Information technology equipment.

Measurement Computing Corporation declares under sole responsibility that the product

**PCI-2515**

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 (1990)/CISPR 22: Radiated and Conducted emissions.

Immunity: EN61326-1:2006, (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): 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.

- Part CA-68-3S or CA-68-6S must be properly installed.
- The host computer, peripheral equipment, power sources, and expansion hardware must be CE compliant.
- All I/O cables must be shielded, with the shields connected to CHASSIS ground stud.
- I/O cables must be less than 3 meters (9.75 feet) in length.
- The host computer must be properly grounded.
- Equipment must be operated in a controlled electromagnetic environment as defined by Standards EN 61326-1:2006, or IEC 61326-1:2005.

**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 “Daqboard 3000 with PDQ30 Expansion Module”. Further testing was conducted by Chomerics Test Services, Woburn, MA. 01801, USA in January, 2009. Test records are outlined in Chomerics Test report #EMI5250.09.

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

Carl Haapaoja

Carl Haapaoja, Director of Quality Assurance

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

**NI Hungary Kft**
**H-4031 Debrecen, Hátar út 1/A, Hungary**
**Phone: +36 (52) 515400**
**Fax: +36 (52) 515414**
**http://hungary.ni.com/debrecen**