



PCIe/PXIe-5500 Series

Family of Multi-functional Data Acquisition Boards User Manual



User Manual Version: V2.5.0

Revision Date: June 28, 2022

Table of Contents

1.Introduction	1
1.1 Overview	1
1.2 Main Features	1
1.3 Abbreviations	2
1.4 Learn by Example	2
2.Hardware	5
2.1 Analog Input.....	5
2.1.1 AI Hardware Specifications	5
2.1.2 Basic DC AI Accuracy, DAQ Mode	6
2.1.3 Additional DC AI Accuracy Adjustment, DAQ Mode.....	7
2.1.4 DS Mode.....	10
2.1.5 AI Bandwidth.....	13
2.1.6 CMRR.....	14
2.2 Analog Output.....	14
2.2.1 Hardware Specifications	14
2.2.2 Basic AO Accuracy	15
2.2.3 Additional AO Accuracy Adjustment.....	15
2.3 Digital IO Specifications	17
2.4 Counter/Timer Specifications	17
2.5 PFI Specifications	17
2.6 Power Specification.....	18
2.7 Physical and Environment.....	18
2.8 Front Panel and Pin Definition	19
3.Software	24
3.1 System Requirements	24
3.2 System Software	24
3.3 C# Programming Language	25
3.4 PCIe/PXIe-5500 Series Hardware Driver	25
3.5 Install the SeeSharpTools from JYTEK.....	25
3.6 Running C# Programs in Linux	26
4.Operating PCIe/PXIe-5500	27
4.1 Quick Start.....	27
4.2 Data Acquisition Methods	27
4.2.1 Continuous Acquisition	30
4.2.2 Finite Acquisition	30
4.2.3 Single Point Acquisition	30
4.2.4 Record Acquisition	30
4.3 Analog Input Terminal Type.....	30
4.3.1 DIFF Mode	31
4.3.2 RSE Mode	33

4.3.3 NRSE Mode	33
4.4 Trigger Source	36
4.4.1 Immediate trigger	36
4.4.2 Software Trigger.....	37
4.4.3 External Analog Trigger.....	38
4.4.4 External Digital Trigger.....	43
4.5 Trigger Mode.....	45
4.5.1 Start Trigger	45
4.5.2 Reference Trigger.....	46
4.5.3 ReTrigger	47
Learn by Example 4.5	47
4.6 AO Operations.....	50
4.6.1 Finite Output	50
4.6.2 Continuous NoWrapping Output.....	53
4.6.3 Continuous Wrapping Output.....	57
4.7 Digital I/O Operations	60
4.7.1 Static DI/DO	60
4.7.2 Dynamic DI/DO	61
4.8 Counter Input Operations	64
4.8.1 Edge Counting	65
4.8.2 Pulse Measurement	70
4.8.3 Frequency Measurement.....	74
4.8.4 Period Measurement	77
4.8.5 Two-Edge Separation	79
4.8.6 Quadrature Encoder	84
4.8.7 Two-Pulse Encoder.....	88
4.9 Counter Output Operations	93
4.9.1 Single Pulse Output.....	93
4.9.2 Finite Pulse Output	95
4.9.3 Continuous Pulse Output.....	97
4.10 System Synchronization Interface (SSI) for PCIe Modules.....	99
4.11 DIP Switch in PCIe-5500	99
5.Calibration	101
6.Using PCIe/PXIe-5500 in Other Software	102
6.1 Python	102
6.2 C++	102
6.3 LabVIEW	102
7.About JYTEK.....	103
7.1 JYTEK China	103
7.2 JYTEK Korea and JYTEK In Other Countries.....	103
7.3 JYTEK Hardware Products	103

7.4 JYTEK Software Platform	104
7.5 JYTEK Warranty and Support Services	104
8.Appendix 1 Common Analog Measurement Issues.....	105
8.1 Floating Signals and Ground Referenced Signals.....	105
8.2 Differential, NRSE, RSE Modes.....	105
8.3 Reducing the Common Mode Voltage Effect	106
8.4 DC, AC and DSA Mode.....	107
8.4.1 DC Mode	107
8.4.2 AC Mode.....	108
8.4.3 DSA Mode	108
9.Statement.....	109

Figure 1 JYPEDIA Information	2
Figure 2 PCIe/PXIe-5500 experiment.....	3
Figure 3 TB-68 Terminal Block	3
Figure 4 Averaging in DC Measurements.....	6
Figure 5 Common Mode Rejection Ratio.....	14
Figure 6 PXIe/PCIe 5510 Front Panel	19
Figure 7 Sample Rate and Internal AD Conversion	28
Figure 8 Continuous MultiChannel Paraments.....	29
Figure 9 MultiChannel Continuous Acquisition	29
Figure 10 Differential Mode for Grounding Signals.....	31
Figure 11 Choose Differential In AI Terminal.....	32
Figure 12 RSE Mode for Floating Signals	33
Figure 13 NRSE Mode for Grounding Signals.....	34
Figure 14 Choose NRSE In AI Terminal.....	35
Figure 15 Immediate trigger Paraments.....	36
Figure 16 Software trigger Paraments.....	37
Figure 17 Software trigger Acquisition	38
Figure 18 Rising Slope Trigger	39
Figure 19 Falling Slope Trigger	39
Figure 20 Hysteresis with Rising Slope Trigger	40
Figure 21 Hysteresis with Falling Slope Trigger	40
Figure 22 Entering Window Trigger	41
Figure 23 Leaving Window Trigger	41
Figure 24 Analog Trigger Paraments.....	42
Figure 25 Waiting For Trigger	42
Figure 26 Analog Trigger Acquisition	43
Figure 27 External Digital Trigger.....	43

Figure 28 Digital Trigger Paraments.....	44
Figure 29 Digital Trigger Acquisition	45
Figure 30 Start Trigger	46
Figure 31 Reference Trigger	47
Figure 32 ReTrigger	47
Figure 33 Retrigger Paraments	48
Figure 34 Retrigger In Start Trigger Mode	49
Figure 35 Retrigger In Reference Trigger Mode	49
Figure 36 Complete Retrigger Count	50
Figure 37 AI Continuous Paraments	51
Figure 38 AO Finite Output Paraments.....	52
Figure 39 AO Finite Signal	52
Figure 40 AI Acquisition Signal.....	53
Figure 41 AI Continuous Paraments	54
Figure 42 AO ContinuousNoWrapping Output Paraments.....	55
Figure 43 AO ContinuousNoWrapping Signal	55
Figure 44 AI Acquisition AO Sin Signal	56
Figure 45 Update AO Square Signal	56
Figure 46 AI Acquisition AO Square Signal.....	57
Figure 47 AI Continuous Paraments	58
Figure 48 AO Continuous Wrapping Paraments.....	58
Figure 49 AO Continuous Wrapping Signal.....	59
Figure 50 AI Acquisition AO Signal	59
Figure 51 Single Digital Output	60
Figure 52 Single Digital Input	61
Figure 53 DI Continuous Paraments	62
Figure 54 DO ContinuousNoWrapping Output.....	62
Figure 55 DI Continuous Acquisition.....	63
Figure 56 Counter Terminal	64
Figure 57 Counter Signal Wiring Instruction.....	65
Figure 58 Simple Edge Counting in Single Mode	65
Figure 59 Buffered Edge Counting with Internal Sample Clock.....	66
Figure 60 Simple Edge Counting with Implicit SampleClk	66
Figure 61 Count Direction	67
Figure 62 EdgeCounting For Single Mode.....	67
Figure 63 EdgeCounting For Finite Mode	68
Figure 64 Counter Values For Internal Clock	69
Figure 65 Counter Values For Implicit Clock	69
Figure 66 Pulse Measurement in Single Mode	70
Figure 67 Pulse Measurement with Internal SampleClk	70
Figure 68 Pulse Measurement with Implicit SampleClk	71

Figure 69 Pulse Measure For Single Mode	72
Figure 70 Pulse Measure Value For Single Mode	72
Figure 71 Pulse Measure For Finite Mode	73
Figure 72 Pulse Measure Values For Finite Mode	73
Figure 73 Frequency Measurement with Internal Sample Clock	75
Figure 74 Frequency Measure For Single Mode	76
Figure 75 Frequency Measure For Continuous Mode	76
Figure 76 Frequency Measure Values.....	77
Figure 77 Peroid Measure For Single Mode	78
Figure 78 Peroid Measure For Continuous Mode	79
Figure 79 Two-Edge Separation in Single Mode	80
Figure 80 Two-Edge Separation with Internal Sample Clock.....	80
Figure 81 Two-Edge Separation with Implicit Sample Clock	81
Figure 82 Two-EdgeSeparation Measure For Single Mode	82
Figure 83 Two-EdgeSeparation Measure For Finite Mode.....	83
Figure 84 Quadrature Encoder x1 Mode	84
Figure 85 Quadrature Encoder x2 Mode	84
Figure 86 Quadrature Encoder x4 mode	85
Figure 87 Quadrature Encoder x1 with Sample Clock	85
Figure 88 Quadrature Encoder x4 with Implicit Sample Clock	86
Figure 89 QuadEncoder For Single Mode	87
Figure 90 QuadEncoder For Continuous Mode	87
Figure 91 Two-Pulse Encoder in Single Mode.....	88
Figure 92 Two-Pulse Encoder with Internal Sample Clock	89
Figure 93 Two-Pulse Encoder with Implicit Sample Clock	89
Figure 94 Two-PulseEncoder For Single Mode	90
Figure 95 Two-PulseEncoder For Finite Mode.....	91
Figure 96 Two-PulseEncoder For Continuous Mode	92
Figure 97 Single Pulse Output	93
Figure 98 Single Pulse Generation	94
Figure 99 AI Acquisition Single Pulse	94
Figure 100 Finite Pulse Output	95
Figure 101 Finite Pulses Generation	95
Figure 102 AI Acquisition Finite Pulse.....	96
Figure 103 Continuous Pulse Output	97
Figure 104 Continuous Pulse Generation	97
Figure 105 AI Acquisition Continuous Pulse	98
Figure 106 SSI Connector in PCIe-5500.....	99
Figure 107 DIP Switch in PCIe-5500	100
Figure 108 Six Measurement Scenarios.....	105
Figure 109 Using Resister to Reduce Common Mode Voltage Effect	106

Figure 110 DC, AC and DSA Mode.....	107
-------------------------------------	-----

Table 1	PCIe/PXle-5500 Family and Main Features.....	1
Table 2	PCIe/PXle-5500 on Different Bus	1
Table 3	Analog Input Specifications.....	5
Table 4	Basic Accuracy in DAQ Mode	7
Table 5	Additional Accuracy Adjustment due to Temperature, DAQ Mode	8
Table 6	Additional Accuracy Adjustment due to Multi-Channel Mode.....	10
Table 7	Basic AI Accuracy in DSMode	11
Table 8	AC Accuracy for One Channel	12
Table 9	AC Accuracy for Multiple Channels.....	12
Table 10	AI Bandwidth	13
Table 11	Analog Output Specifications.....	14
Table 12	Basic AO Accuracy	15
Table 13	Additional AO Accuracy.....	16
Table 14	Digital IO Specifications.....	17
Table 15	Counter/Timer Specifications	17
Table 16	PFI Specifications.....	17
Table 17	Power Specification.....	18
Table 18	Physical and Environment.....	18
Table 19	5510 / 5511 Pin Definition	20
Table 20	5510 / 5511 Counter Pin Define.....	21
Table 21	5515 / 5516 Pin Definition	22
Table 22	5515 / 5516 Counter Pin Define.....	23
Table 23	Notes to Legend	23
Table 24	Supported Linux Versions	24
Table 25	SSI Connector Pin Assignment for PCIe-5500	99
Table 26	Relationship between switch position and slot number	100
Table 27	DC, AC, DSA Measurements.....	107

1. Introduction

This chapter presents the information how to use this manual and how to quick start if you are already familiar with Microsoft Visual Studio and C# programming language.

1.1 Overview

The PCIe/PXIe-5500 Series is a family of multifunction data acquisition boards, which can run on PCIe, PXIe, TXI(Thunderbolt) and USB buses (coming soon). Depending on the model number, a PCIe/PXIe-5500 series provide different AI channels, AO channels, sampling rate as shown in Table 1.

55xx Model	AI Channels	Sample Rate (MS/s)	AI Resolution	AO Channels	AO Update Rate (MS/s)	DIO+PFI
5510	32	2	18	4	2.86	48
5511	32	1.25	18	4	2.86	48
5515	16	2	18	2	2.86	24
5516	16	1.25	18	2	2.86	24

Table 1 PCIe/PXIe-5500 Family and Main Features

55xx Model	PCIe	PXIe	TXI	USB
5510	√	√	√	√
5511	√	√	√	√
5515	√	√	√	√
5516	√	√	√	√

Table 2 PCIe/PXIe-5500 on Different Bus

Please check with JYTEK for the latest PCIe/PXIe-5500 series offering.

1.2 Main Features

- 32 single-ended or 16 differential 18-bit analog input channels
- 18 bits ADC
- 7 voltage ranges: $\pm 10V/\pm 5V/\pm 2V/\pm 1V/\pm 0.5V/\pm 0.2V/\pm 0.1V$
- 64M samples FIFO buffer for analog input
- 4 simultaneous 16-bit analog output channels
- 32M sample FIFO buffer for analog output
- 6 ports digital IO, 8 channels per port
- 4 general 32-bit timer/counter
- DMA for analog input and output
- Analog/Digital/Software Trigger

1.3 Abbreviations

AI: Analog Input
AO: Analog Output
DI: Digital Input
DO: Digital Output
CI: Counter Input
CO: Counter Output
DAQ: Data Acquisition
ADC: Analog-to-Digital Conversion
DAC: Digital-to-Analog Conversion
PFI: Programmable Function Interface
SE: Single-Ended
RSE: Referenced Single-Ended
NRSE: Non-Referenced Single-Ended
DIFF: Differential
PPM: Parts Per Million

1.4 Learn by Example

JYTEK has added **Learn by Example** in this manual. We provide many sample programs for this device. Please download the sample programs for this device. You can download a [JYPEDIA](#) excel file from our web www.jytek.com. Open JYPEDIA and search for JY5500 in the driver sheet, select **JY5500 Examples.zip**. In addition to the download information, JYPEDIA also has a lot of other valuable information, JYTEK highly recommend you use this file to obtain information from JYTEK.



 	Drivers are often updated, please register and receive the update information.		
Drivers	Update Date		
JY5500 V3.0.1 Linux.tar	2020/12/4		
JY5500 V3.0.1 Win.zip	2020/12/4		
JY5500 V1.0.0 LV.zip	2019/10/24		
JY5500 V3.0.1 Examples.zip	2020/12/4		
JY5500 V1.0.0 Python.zip	2019/10/24		
JY5500 V1.0.0 PythonExamples.zip	2019/10/24		

Figure 1 JYPEDIA Information

In a **Learn by Example** section, the sample program is in bold style such as **Analog Input-->Winform AI Continuous MultiChannel Soft Trigger**; the property name in the sample program is also in bold style such as **SamplesToAcquire**; the technical names used in the manual is in italic style such as *SampleRate*. You can easily relate the property names in the example program with the manual documentation.

In an **Learn by Example** section, the experiment is set up as follow. A PCIe/PXIe-5500 card is plugged in a desktop computer. The PCIe/PXIe-5500 is connected to a TB-68 terminal block. A signal source is also connected to the same terminal block as shown Figure 2.

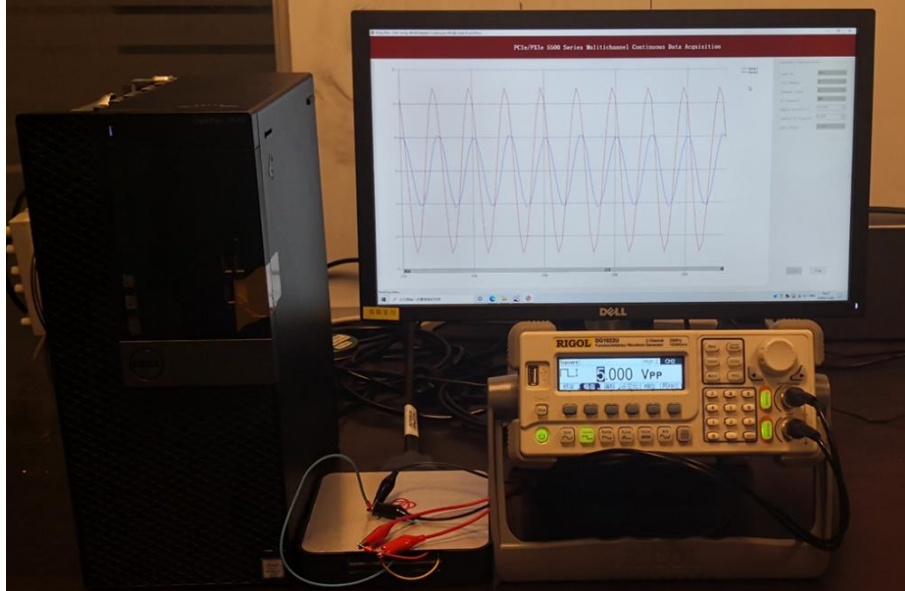


Figure 2 PCIe/PXIe-5500 experiment

The TB-68 has 4 terminal columns, J1 – J4 and is shown below as Figure 3. In the rest of this manual, the wire connection in each **Learn by Example** section will be given by the pin numbers only.

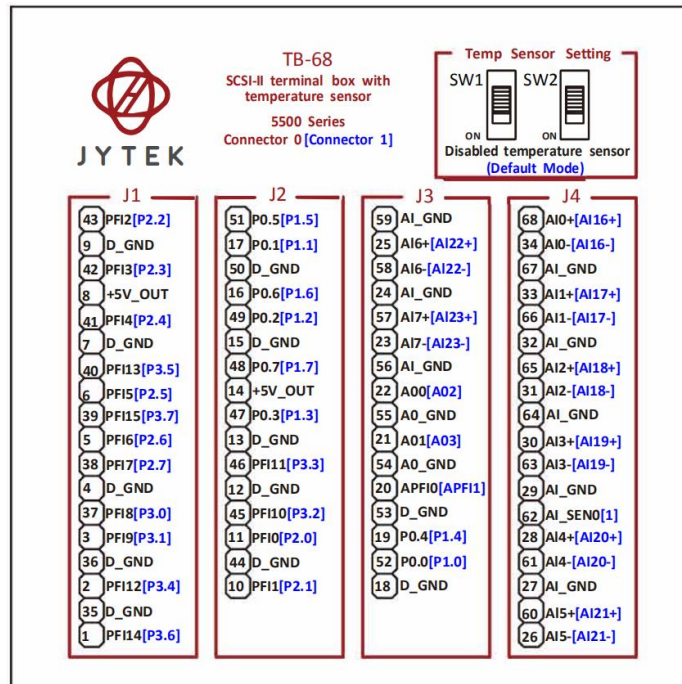


Figure 3 TB-68 Terminal Block

*Tip: PCIe/PXIe-5500 also has the analog output capability. If you do not have a signal source, you can use the outputs of PCIe/PXIe-5500 as the signal source. In this case you need first run example program **Analog Output-->Winform AO Continuous Wrapping Multichannel** to generate the output.*

2. Hardware

2.1 Analog Input

2.1.1 AI Hardware Specifications

Analog Input	5510	5511	5515	5516
Number of channels	32 SE / 16 DIFF		16 SE / 8 DIFF	
ADC resolution (Bits)	18			
Single channel maximum sample rate	2M Sample/s	1.25M Sample/s	2M Sample/s	1.25M Sample/s
Multichannel maximum sample rate (aggregate)	1M Sample/s	625K Sample/s	1M Sample/s	625K Sample/s
Clock	100 MHz			
Input range(V)	$\pm 10/\pm 5/\pm 2/\pm 1/\pm 0.5/\pm 0.2/\pm 0.1$			
Maximum Working Voltage(V)	± 11 V (ref. AIGND)			
Input mode	RSE / NRSE / Differential			
Input impedance	>1 G Ω 100 pF			
Input coupling	DC			
Overvoltage protection	± 25 V			
CMRR	85 dB			
Crosstalk	Adjacent Channel : -80 dB			
	Non-adjacent Channel : -95 dB			
	Differential: -65 dB			
DNL	No Missing Code			
INL	70 ppm of Range Typical			
Input FIFO	64M Samples			
Trigger type	Digital, Analog, Software			
Trigger mode	StartTrigger, ReferenceTrigger, ReTrigger			
External Sampling Clock	≤ 1 MHz			
Analog trigger voltage range	± 10 V Software Programmable			
Overvoltage Protection	Continuous : 20m A, ± 25 V			
	Instantaneous : 40 mA, ± 25 V			

Table 3 Analog Input Specifications

2.1.2 Basic DC AI Accuracy, DAQ Mode

The DC measurement refers to measuring a signal, which either is a DC source or has extremely slow frequency such that the signal voltage changes very little in the measurement window. Please note, the averaging can only be useful if the signal does not change in the averaging window as shown on the left of Figure 4. The noise is greatly reduced and an accurate measurement is made after averaging.

But if the signal changes in the averaging windows as shown on the right of Figure 4, the result will be the average of both the signal and the noise. While the noise is reduced as before, the averaging of the signal introduces the measurement error. **You cannot use the averaging method if your signal changes in the averaging window!**

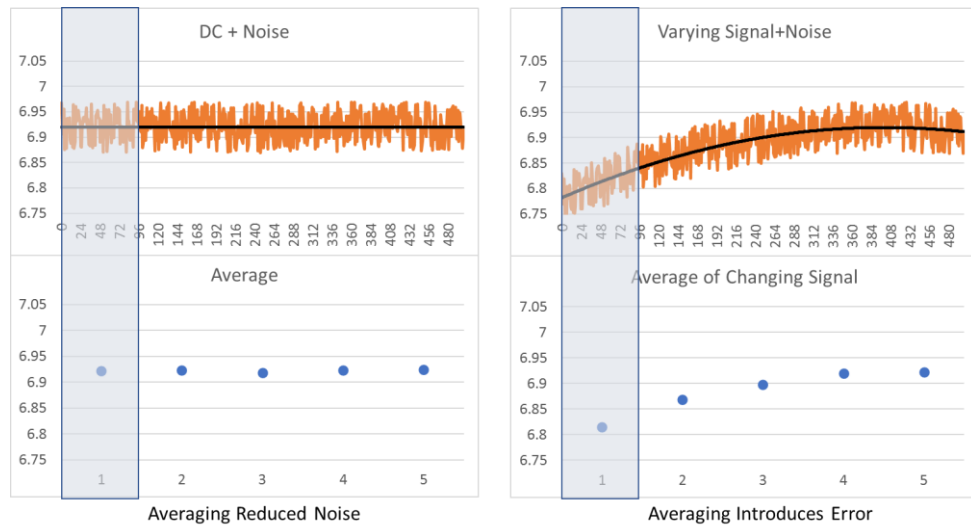


Figure 4 Averaging in DC Measurements

Table 4 also listed the maximum source signal frequency using the 24 Hr. Column accuracy at the maximum 2MHz sample rate. You can use this number for other calibration columns. But if you have other sample rate or the temperature or multiple-channel adjustment, you should use

$$f = \frac{\text{Total Accuracy}}{10 * \text{Range}} f_s$$

to calculate the maximum source frequency. The formulas are same for the DAQ and DS Mode.

The DAQ mode is the normal data acquisition mode commonly found in commercial DAQ hardware. The basic AI DC accuracy Table 4 of the DAQ mode provides accuracy entries when PCIe/PXIe-5500 operates in the single channel mode and within the indicated calibration temperature range. Please note that this accuracy is valid for every single point regardless how many sample points you acquire.

Each entry in the basic accuracy table is a pair of gain and offset coefficients. Using these gain and offset coefficients, your measurement accuracy can be calculated by following fomular:

$$Accuracy = \pm(\% \text{ of Reading} + \% \text{ of Range})$$

For example, at the 0.1V range and 24 Hours column, if your measurement or reading is 0.02V, the accuracy of this measurement is:

$$\pm(0.0024\% * 0.02 + 0.0437\% * 0.1) = \pm 0.000044V = \pm 44\mu V$$

The basic accuracy table also provides full-scale accuracy entries for a quick and convenient look-up. For example, the full-scale accuracy for the 0.1V range and the 24-Hour calibration column is 46uV.

JY5500 Basic Accuracy = $\pm(\% \text{ Reading} + \% \text{ Range})$, DAQ Mode								
Nominal Range (V)	Resolution (18-bits) (μV)	24 Hour Tcal $\pm 1C^\circ$		90 Days Tcal $\pm 5^\circ C$		24 Hr Full Scale Accuracy	90 Days Full Scale Accuracy	Max Input Frequency @2MHz Fs (Hz)
0.1	0.8	0.0024	+ 0.0437	0.0052	+ 0.0498	46 μV	55 μV	92
0.2	1.5	0.0013	+ 0.0230	0.0035	+ 0.0260	49 μV	59 μV	49
0.5	3.8	0.0013	+ 0.0100	0.0033	+ 0.0113	56 μV	73 μV	23
1	7.6	0.0008	+ 0.0056	0.0029	+ 0.0064	65 μV	93 μV	13
2	15.3	0.0007	+ 0.0039	0.0029	+ 0.0045	92 μV	148 μV	9
5	38.1	0.0008	+ 0.0040	0.0031	+ 0.0045	240 μV	378 μV	10
10	76.3	0.0007	+ 0.0034	0.0031	+ 0.0039	410 μV	702 μV	8

Valid for one channel only. 95% of Confidence Interval
 Max sampling rates for 5510,5511,5515,5516: 2M, 1.25M, 2M, 1.25M
 Add 20% to Gain and Offset Errors From 91 Days to 1 Year. Preliminary
 10 V range: valid for $\pm 9.5V$
 Source impedance $\leq 100\Omega$
 Add accuracy adjustment for temperature and multiple channels
 Max input frequency = (total accuracy/range)*Sample Rate/10, only 24 Hr. data provided
 All accuracy data in this table are tested with 1m shield cable.
 Specs subject to minor changes

Table 4 Basic Accuracy in DAQ Mode

2.1.3 Additional DC AI Accuracy Adjustment, DAQ Mode

The AI accuracy will also be affected by the operating temperatures and the multi-channel operating mode. If PCIe/PXIe-5500 operates outside the basic operating range

as defined by the basic accuracy table, the additional accuracy entries must be added to the basic accuracy table to obtain the final accuracy.

Temperature Adjustment

The temperature adjustment is needed when the operating temperature is outside the calibration temperature range, such as the $T_{cal} \pm 1^\circ \text{C}$ range, or the $T_{cal} \pm 5^\circ \text{C}$ range etc. The temperature coefficients, both the gain and offset coefficients, are provided. These coefficients must be added to the gain and offset coefficients in the basic accuracy table to calculate the accuracy. Table 5 is the additional AI accuracy adjustment table due to the operating temperatures.

JY5500 Temperature Accuracy Adjustment = $\pm(\% \text{ Reading} + \% \text{ Range})$			
Nominal Range (V)	Temperature Coefficients (/ $^\circ\text{C}$)		Full-Scale Temp Adjustment ($\mu\text{V}/^\circ\text{C}$)
0.1	0.0007	+ 0.0015	2 μV
0.2	0.0006	+ 0.0007	3 μV
0.5	0.0005	+ 0.0003	4 μV
1	0.0005	+ 0.0002	7 μV
2	0.0006	+ 0.0002	14 μV
5	0.0006	+ 0.0001	35 μV
10	0.0006	+ 0.0001	73 μV
For all sample rates			
All accuracy data in this table are tested with 1m shield cable.			
Specs subject to minor changes when more tests become available			

Table 5 Additional Accuracy Adjustment due to Temperature, DAQ Mode

For example, at the 0.1V range and 2°C outside the 24 Hour $T_{cal} \pm 1^\circ \text{C}$ range, the basic accuracy entry (0.0024%, 0.0437%) must be adjusted to

$$(0.0024\% + 0.0007\% * 2, 0.0437\% + 0.0015\% * 2) = (0.0038\%, 0.0467\%).$$

The accuracy of the same 0.02V reading value is

$$\pm(0.0038\% * 0.02 + 0.0467\% * 0.1) = \pm 0.000047\text{V} = \pm 47\mu\text{V}$$

The full-scale accuracy numbers are also provided in the additional accuracy table. For example, at the 0.1V range, and 2°C outside 24 Hour $T_{cal} \pm 1^\circ \text{C}$ range, there is a $2\mu\text{V}/^\circ\text{C}$ adjustment. You will need to add $2 * 2\mu\text{V} = 4\mu\text{V}$ to the basic accuracy, and the total accuracy is $46 + 2 * 2 = 50\mu\text{V}$.

Multi-Channel Adjustment

When PCIe/PXIe-5500 operates at the multi-channel mode, the channel switching can incur additional errors because it takes time for the voltage to discharge from one channel before the next channel starts acquisition. Because PCIe/PXIe-5500's 32 channels share one ADC, the channel sample rate of each channel is determined by

$$\text{ChannelSample Rate} = \text{Sample Rate}/N$$

where

SampleRate \leq 1M is selected by the driver software,
N is the total number of channels. N=2,3 ..., 32.

Table 6. lists the additional DC accuracy adjustment. No adjustments are needed for the sample rates lower than or equal to 200K/N. For the sample rate \geq 500K/N, the additional error increases. In most cases, these higher sample rates are not needed and not recommended for DC measurements. Please refer to 8.4 for details.

The source impedance, cable length and terminal block also affect the accuracy adjustment. Table 6. uses a 25 Ω impedance source, a JYTEK 2-meter cable and TB68 terminal block. The accuracy adjustment for the longer cable length and larger source impedance have not been tested.

In the above temperature adjustment example, at the 0.1V range, 2°C outside 24 Hour Tcal \pm 1°C range, N=2 channels, and the 500K/N sample rate, the total accuracy is 46 + 2*2 +45 uV = 95uV.

JY5500 Multi-Channel Accuracy Adjustment (μV)		
Nominal Range (V)	Sample Rate (Hz) per Channel	Full-Scale Multi-Channel Adjustment (μV)
0.1	≤ 200 K/N	0
0.1	500 K/N	45 μV
0.1	1 M/N	98 μV
0.2	≤ 200 K/N	0
0.2	500 K/N	23 μV
0.2	1 M/N	136 μV
0.5	≤ 200 K/N	0
0.5	500 K/N	32 μV
0.5	1 M/N	314 μV
1	≤ 200 K/N	0
1	500 K/N	70 μV
1	1 M/N	683 μV
2	≤ 200 K/N	0
2	500 K/N	109 μV
2	1 M/N	904 μV
5	≤ 200 K/N	0
5	500 K/N	693 μV
5	1 M/N	2,904 μV
10	≤ 200 K/N	0
10	500 K/N	2,375 μV
10	1 M/N	5,713 μV
N: Number of channels from 2 to 32 Use the next higher sample rate for the not listed sample rates 1 M/N not recommended for a DC measurement. TB68 Terminal Block + 2m ACL-2006868-2 Cable Source Impedance: $\leq 25 \Omega$ All accuracy data in this table are tested with 1m shield cable. Preliminary, subject to changes.		

Table 6 Additional Accuracy Adjustment due to Multi-Channel Mode

2.1.4 Basic DC AI Accuracy, DS Mode

PCIe/PXIe-5510 and 5515 is capable to make very accurate DC measurement by using the DS Mode operation. Please note the DS mode is currently available for one channel measurement only. The reason is that the channel switching induces errors that affect the measurement accuracy. The DS mode is not intended for high frequency measurement at all.

In the DS Mode, these boards use the onboard FPGA processing to reduce the noise effect and to improve the AI accuracy. The FPGA processing uses more sample points and will reduce the overall sample rates. For each range, the processing can be

different, so the sample rates can also be different. Typically, a slower sample rate delivers better accuracy.

PCIe/PXIe-5510 and 5515 only provides the accuracy entries for several predetermined sample rates. If you choose a different sample rate, you need to use the accuracy entries in the next higher sample rate. For instance, if you choose a 5K sample rate, you need to use the accuracy of the 10K sample rate.

Table 7 is the basic accuracy table in the DS Mode. The additional DC accuracy adjustment due to temperature in the DS Mode is identical to that in the normal DAQ mode as in Table 5.

JY5510, 5515 Basic Accuracy = ±(% Reading+% Range),DSMode									
Nominal Range (V)	Max Sample Rate for 1 Channel	Resolution (uV)	24 Hour Tcal ±1C*		90 Days Tcal ± 5°C		24 Hr Full Scale Accuracy	90 Days Full Scale Accuracy	Max Input Frequency Fs (Hz)
0.1	1K	0.8	0.0024	+ 0.0106	0.0052	+ 0.0166	13 uV	22 uV	0.013
0.1	10K	0.8	0.0024	+ 0.0087	0.0052	+ 0.0148	11 uV	20 uV	0.111
1	1K	7.6	0.0008	+ 0.0015	0.0029	+ 0.0023	23 uV	52 uV	0.002
1	10K	7.6	0.0008	+ 0.0013	0.0029	+ 0.0021	22 uV	50 uV	0.022
10	1K	76.3	0.0007	+ 0.0010	0.0031	+ 0.0015	171 uV	463 uV	0.002
10	10K	76.3	0.0007	+ 0.0009	0.0031	+ 0.0014	161 uV	453 uV	0.016

Valid for one channel only. 95% of Confidence Interval
DS Mode for 5510, 5515 only
Use the next higher sample rate for not listed sample rates
Use DAQ Mode when the sample rate>10KHz
Add 20% to Gain and Offset Errors From 91 Days to 1 Year. Preliminary
10 V range: valid for ±9.5V
Source impedance <=100Ω
Add accuracy adjustment for temperature and multiple channels
Max input frequency = (total accuracy/range)*Sample Rate/10, only 24 Hr. data provided
All accuracy data in this table are tested with 1m shield cable.
Specs subject to minor changes when more tests become available

Table 7 Basic AI Accuracy in DSMode

2.1.5 AC Accuracy for One Channel

The AC accuracy is defined as the measurement accuracy of a single tone sinusoidal input signal. If the input the signal is not sinusoidal, the accuracy is not applicable.

Table 8 gives the total AC accuracy specification for PCIe/PXIe-5510. It is for one channel only. There is no AC accuracy specification when the sinusoidal input is greater than 200KHz. The reason is we need at least 5 sample points to describe a time domain sinusoidal waveform.

JY5500 Total Absolute AC Accuracy for One Channel							
	0.1V	0.2V	0.5V	1V	2V	5V	10V
[10Hz ~ 50Hz)	0.28%	0.24%	0.21%	0.18%	0.16%	0.14%	0.14%
[50Hz ~ 20K)	0.13%	0.11%	0.11%	0.09%	0.09%	0.08%	0.09%
[20K ~ 50K)	0.27%	0.24%	0.23%	0.22%	0.17%	0.16%	0.15%
[50K ~ 100K)	0.79%	0.66%	0.57%	0.74%	0.48%	0.39%	0.38%
[100K ~ 200K]	2.42%	2.09%	1.90%	2.12%	1.28%	1.00%	1.00%

Valid for one channel measurement of sinusoidal input
90 days, Tcal±5°C. For one year AC accuracy, add 20%. Preliminary
Sample Rate ≥ 1.25M. No specs for Sample Rate < 1.25MHz.
DC Coupling
10 V range: valid for ±9.5V
Source impedance ≤ 100Ω
All accuracy data in this table are tested with 1m shield cable.
95% confidence level

Table 8 AC Accuracy for One Channel

2.1.6 AC Accuracy for Multiple Channels

Table 9 shows the AC accuracy for multiple channels. Please note you can only use the 200K/N sample rate. No specifications are provided for other sample rates. At this sample rate, there is no channel switching error. min(20K, 40K/N) indicates the smaller frequency number of 20K and 40K/N Hz. The reason is we need at least 5 sample points to describe a time domain sinusoidal waveform. 200K/N is the channel sample rate, the sinusoidal waveform frequency should ≤ 40K/N.

JY5500 Total Absolute AC Accuracy for Multiple Channels							
	0.1V	0.2V	0.5V	1V	2V	5V	10V
[10Hz ~ 50Hz)	0.28%	0.24%	0.21%	0.18%	0.16%	0.14%	0.14%
[50Hz ~ min(20K, 40K/N))	0.13%	0.11%	0.11%	0.09%	0.09%	0.08%	0.09%

Valid for N channels of sinusoidal inputs. 2 ≤ N ≤ 32.
90 days, Tcal±5°C. For one year AC accuracy, add 20%. Preliminary
Sample Rate = 200K/N. No specs for other sample rates.
DC Coupling
10 V range: valid for ±9.5V
Source impedance ≤ 100Ω
All accuracy data in this table are tested with 1m shield cable.
95% confidence level

Table 9 AC Accuracy for Multiple Channels

For example, if we use all 32 channels, ie N=32, PCIe/PXIe-5510 can still provide AC accuracy from 10Hz to 1.25KHz.

2.1.7 Temperature Adjustment for AC Accuracy

At this time, JYTEK only provides 90 days AC accuracy within $T_{cal} \pm 5^{\circ}C$ range. At this time, no other temperature adjustment specifications are provided.

2.1.8 AI Bandwidth

Analog Input Bandwidth	
Nominal Range Full Scale (V)	-3dB Bandwidth (MHz)
± 10	2.06
± 5	2.14
± 2	1.72
± 1	1.21
± 0.5	1.21
± 0.2	1.21
± 0.1	1.21

Table 10 AI Bandwidth

2.1.9 CMRR

A typical CMRR performance is shown below.

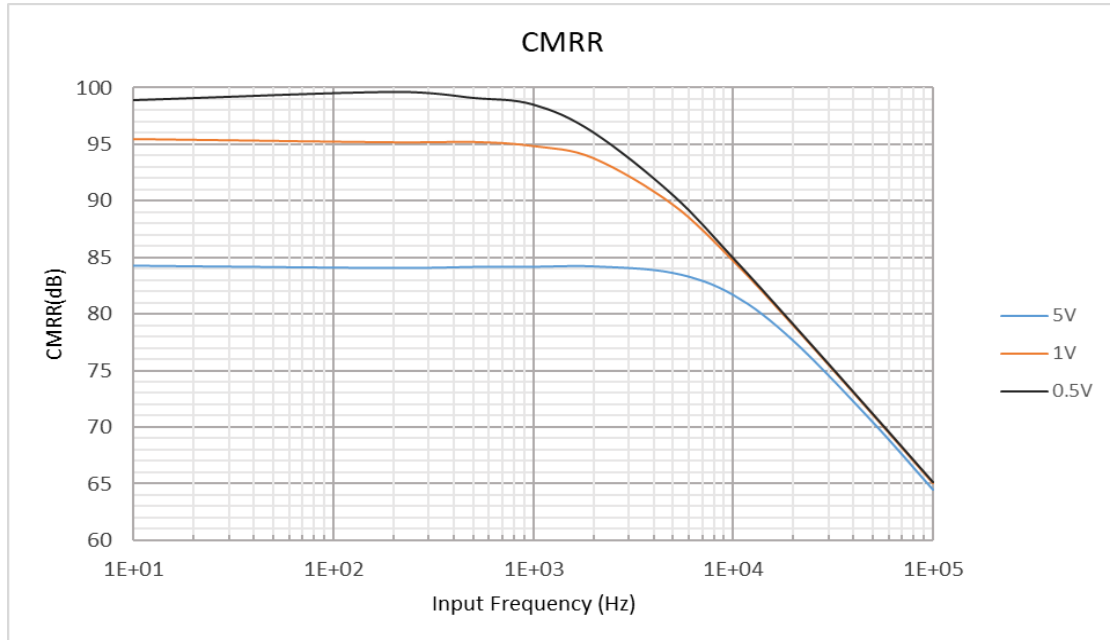


Figure 5 Common Mode Rejection Ratio

2.2 Analog Output

2.2.1 Hardware Specifications

Analog Output	5510	5511	5515	5516
Number of channels	4		2	
DAC resolution	16 bits			
Maximum update rate(simultaneous)	1 channel	2.86 M Sample/s		
	2 channels	2 M Sample/s		
	3 channels	1.54 M Sample/s		
	4 channels	1.25 M Sample/s		
Clock	100 MHz			
Clock accuracy	Jitter <20 ps			
Output range(V)	±10, ±5			
Output mode	RSE			
Output impedance	2 ohm			
Output coupling	DC			
Output current drive	±10 mA			
Output FIFO	32M Samples			
Trigger type	Digital, Software			
Trigger mode	StartTrigger			

Table 11 Analog Output Specifications

2.2.2 Basic AO Accuracy

The AO output accuracy of PCIe/PXIe-5500 Series when using the analog output function can be calculated according to the corresponding parameters in the following table

Each entry in the basic accuracy table is a pair of gain and offset coefficients. Using these gain and offset coefficients, your AO output basic accuracy is calculated by following fomular:

$$Accuracy = \pm(\% \text{ of Output} + \% \text{ of Range})$$

For example, at the 5V range and 24 Hours column, if your output is 2V, the accuracy of this measurement is:

$$\pm(0.0015\% * 2 + 0.0056\% * 5) = \pm 0.00031V = \pm 310uV$$

The basic accuracy table also provides full-scale accuracy entries for a quick and convenient look-up. For example, the full-scale accuracy for the 5V range and the 24-Hour calibration column is 355uV.

JY5500 Basic AO Accuracy = $\pm(\% \text{ of Output} + \% \text{ of Range})$						
Nominal Range (V)	Resolution (16-bits) (μV)	24 Hour Tcal $\pm 1\text{C}^\circ$	90 Days Tcal $\pm 5^\circ$	24 Hr Full-Scale Accuracy	90 Days Full-Scale Accuracy	Max Update Rate (S/s)
5	153	0.0015 + 0.0056	0.0043 + 0.0064	355 μV	535 μV	2.86 M
10	305	0.0020 + 0.0070	0.0060 + 0.0078	900 μV	1380 μV	2.86 M

Valid for all update rates.
 Add accuracy adjustment if temperature is outside calibration temperature range.
 Add 20% to Gain and Offset Errors From 91 Days to 1 Year. Preliminary.
 Maximum update rates(simultaneous)
 1 Ch: 2.86 M; 2 Ch: 2 M; 3 Ch: 1.54 M; 4 Ch: 1.25 M
 All accuracy data in this table are tested with 1m shield cable.
 Specs subject to minor changes when more tests become available.

Table 12 Basic AO Accuracy

2.2.3 Additional AO Accuracy Adjustment

Table 13 is the additional AO accuracy adjustment table. It provides additional accuracy adjustments: temperature adjustment.

JY5500 AO Additional Accuracy Adjustment			
Nominal Range (V)	Resolution (18-bits) (μV)	Temperature Coefficients (/ $^{\circ}\text{C}$)	Full-Scale Temp Adjustment ($\mu\text{V}/^{\circ}\text{C}$)
5	153	0.0007 + 0.0002	45 μV
10	305	0.0010 + 0.0002	120 μV

Valid for all update rates.
All accuracy data in this table are tested with 1m shield cable.
Specs subject to minor changes when more tests become available.

Table 13 Additional AO Accuracy

The temperature adjustment is needed when the operating temperature is outside the calibration temperature range, such as the $T_{cal} \pm 1^{\circ}\text{C}$ range, or the $T_{cal} \pm 5^{\circ}\text{C}$ range etc. The temperature coefficients, both the gain and offset coefficients, are provided. These coefficients must be added to the gain and offset coefficients in the basic accuracy table to calculate the accuracy. For example, at the 5V output and 2°C outside the 24 Hour $T_{cal} \pm 1^{\circ}\text{C}$ range, the basic accuracy entry (0.0015%, 0.0056%) must be adjusted to

$$(0.0015\% + 0.0007\% * 2, 0.0056\% + 0.0002\% * 2) = (0.0029\%, 0.0060\%).$$

The accuracy of the same 2V output value is

$$\pm(0.0029\% * 2 + 0.0060\% * 5) = \pm 0.000358\text{V} = \pm 358\mu\text{V}$$

The full-scale accuracy numbers are also provided in the additional accuracy table. For example, at the 5V range, and 2°C outside 24 Hour $T_{cal} \pm 1^{\circ}\text{C}$ range, there is a $45\mu\text{V}/^{\circ}\text{C}$ adjustment. You will need to add $2 * 45\mu\text{V} = 90\mu\text{V}$ to the basic accuracy, and the total accuracy is $358 + 90 = 448\mu\text{V}$.

2.3 Digital IO Specifications

DIO	5510/5511	5515/5516
Number of channels	Port (0,1,2,3,4,5)	Port (0,1,2)
Ground reference	D GND	
Directional control	Independent control of each port	
Clock	10 MHz	
DI FIFO	16M Samples	
DO FIFO	16M Samples	
Initial state	Input	
Digital Input	Logic Low: V_{IL} Min : 0 / Max : 1.0 V	
	Logic High: V_{IH} Min : 2V / Max : 5.3V	
Digital Output	Logic Low : 0 V, I_{OL} Max: 24 mA	
	Logic High : 2.6 V~5 V, I_{OH} : -24 mA~0 mA	
Overvoltage Protection	Continuous 30 mA, -3.9 V~8.9 V Instantaneous 200 mA, ± 25 V	
	Duty cycle of instantaneous current pulse does not exceed 15%	

Table 14 Digital IO Specifications

2.4 Counter/Timer Specifications

CI/CO	5510	5511	5515	5516
Number of channels	4		2	
Resolution	32			
CI	edge count, period measurement, frequency measurement, pulse width measurement, two-edge interval measurement, orthogonal coding, etc.			
CO	Single, finite and continuous pulse			
Clock	200 MHz			
FIFO	4M Samples			
Input	Gate, Source, Aux			
Output	OUT			

Table 15 Counter/Timer Specifications

2.5 PFI Specifications

PFI	5510	5511	5515	5516
Number of channels	16			
External digital trigger interface	Trigger voltage 3.3 V TTL; trigger edge: Rising/Falling			
Initial state	Input			

Table 16 PFI Specifications

2.6 Power Specification

Power	5510	5511	5515	5516
3.3V	2.09 A		2.12 A	
12V	0.28 A		0.25 A	

Table 17 Power Specification

2.7 Physical and Environment

Operating Environment

Ambient temperature range	0 °C to 50 °C
Relative humidity range	20% to 80%, noncondensing

Storage Environment

Ambient temperature range	-20 °C to 80 °C
Relative humidity range	10% to 90%, noncondensing

Table 18 Physical and Environment

2.8 Front Panel and Pin Definition

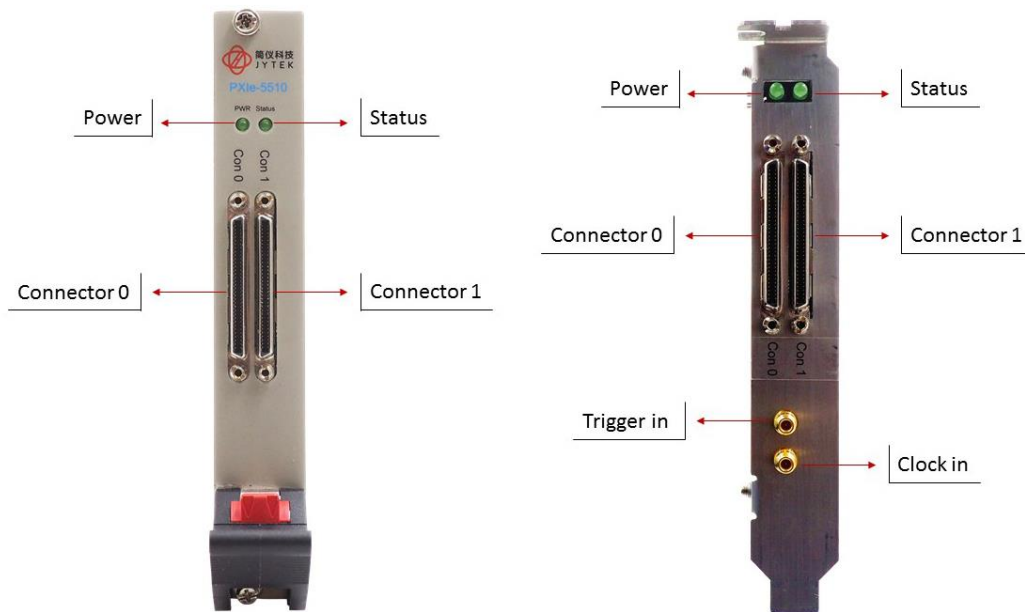


Figure 6 PXIe/PCIe 5510 Front Panel

PCIe/PXIe-5500 series boards are connected to outside signals by either two 68-pin cables for the 32 channel configurations or one 68-pin cable for 16 channel configurations. Table 19, Table 20, Table 21, Table 22 show the pin definitions for 32 channels and 16 channels of PCIe/PXIe-5500 series boards respectively. Please note that pin definition of connector 0 and 1 is different!

Please also note, for 32 channel device in DIFF mode, the 16 analog input channels are 0~7 and 16~23.

5510/5511 Connector Pin Definition

Connector 0				Connector 1			
Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
1	PFI 14 /P5.6/DO_ECLK	35	D_GND	1	P3.6	35	D_GND
2	PFI 12/P5.4	36	D_GND	2	P3.4	36	D_GND
3	PFI 9/P5.1	37	PFI 8/P5.0	3	P3.1	37	P3.0
4	D_GND	38	PFI 7/P4.7	4	D_GND	38	P2.7
5	PFI 6 /P4.6/ AO_ECLK	39	PFI 15/P5.7	5	P2.6	39	P3.7
6	PFI 5/P4.5	40	PFI 13/P5.5	6	P2.5	40	P3.5
7	D_GND	41	PFI 4/P4.4	7	D_GND	41	P2.4
8	+5V_OUT	42	PFI 3/P4.3	8	+5V_OUT	42	P2.3
9	D_GND	43	PFI 2 /P4.2/AI_ECLK	9	D_GND	43	P2.2
10	PFI 1/P4.1	44	D_GND	10	P2.1	44	D_GND
11	PFI 0/P4.0	45	PFI10 /P5.2/DI_ECLK	11	P2.0	45	P3.2
12	D_GND	46	PFI 11/P5.3	12	D_GND	46	P3.3
13	D_GND	47	P0.3	13	D_GND	47	P1.3
14	+5V_OUT	48	P0.7	14	+5V_OUT	48	P1.7
15	D_GND	49	P0.2	15	D_GND	49	P1.2
16	P0.6	50	D_GND	16	P1.6	50	D_GND
17	P0.1	51	P0.5	17	P1.1	51	P1.5
18	D_GND	52	P0.0	18	D_GND	52	P1.0
19	P0.4	53	D_GND	19	P1.4	53	D_GND
20	APFI 0	54	AO_GND	20	APFI 1	54	AO_GND
21	AO 1	55	AO_GND	21	AO 3	55	AO_GND
22	AO 0	56	AI_GND	22	AO 2	56	AI_GND
23	AI 15 (AI 7-)	57	AI 7 (AI 7+)	23	AI 31 (AI 23-)	57	AI 23 (AI 23+)
24	AI_GND	58	AI 14 (AI 6-)	24	AI_GND	58	AI 30 (AI 22-)
25	AI 6 (AI 6+)	59	AI_GND	25	AI 22 (AI 22+)	59	AI_GND
26	AI 13 (AI 5-)	60	AI 5 (AI 5+)	26	AI 29 (AI 21-)	60	AI 21 (AI 21+)
27	AI_GND	61	AI 12 (AI 4-)	27	AI_GND	61	AI 28 (AI 20-)
28	AI 4 (AI 4+)	62	AI_SENSE 0	28	AI 20 (AI 20+)	62	AI_SENSE 1
29	AI_GND	63	AI 11 (AI 3-)	29	AI_GND	63	AI 27 (AI 19-)
30	AI 3 (AI 3+)	64	AI_GND	30	AI 19 (AI 19+)	64	AI_GND
31	AI10 (AI 2-)	65	AI 2 (AI 2+)	31	AI26 (AI 18-)	65	AI 18 (AI 18+)
32	AI_GND	66	AI 9 (AI 1-)	32	AI_GND	66	AI 25 (AI 17-)
33	AI 1 (AI 1+)	67	AI_GND	33	AI 17 (AI 17+)	67	AI_GND
34	AI 8 (AI 0-)	68	AI 0 (AI 0+)	34	AI 24 (AI 16-)	68	AI 16 (AI 16+)

Table 19 5510 / 5511 Pin Definition

5510/5511 Counter Pin Definition (Connector 0)

Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
2	CTR0_OUT	40	CTR1_OUT
5	CTR2_Source/A	3	CTR3_Source/A
38	CTR2_Gate/Z	45	CTR3_Gate/Z
37	CTR2_AUX/B	46	CTR3_AUX/B
1	CTR2_OUT	39	CTR3_OUT

Table 20 5510 / 5511 Counter Pin Define

5515/5516 Connector Pin Definition

Connector 0			
Pin	Signal Name	Pin	Signal Name
1	PFI14/P2.6/DO_ELCK	35	D_GND
2	P2.4/PFI12	36	D_GND
3	P2.1/PFI9	37	P2.0/PFI8
4	D_GND	38	P1.7/PFI7
5	P1.6/PFI6/AO_ELCK	39	P2.7/PFI15
6	P1.5/PFI5	40	P2.5/PFI13
7	D_GND	41	P1.4/PFI4
8	+5V_OUT	42	P1.3/PFI3
9	D_GND	43	P1.2/PFI2/AI_ECLK
10	P1.1/PFI1	44	DGND
11	P1.0/PFI0	45	P2.2/PFI10/DI_ECLK
12	D_GND	46	P2.3/PFI11
13	D_GND	47	P0.3
14	+5V_OUT	48	P0.7
15	D_GND	49	P0.2
16	P0.6	50	D_GND
17	P0.1	51	P0.5
18	D_GND	52	P0.0
19	P0.4	53	D_GND
20	APFI	54	AO_GND
21	AO 1	55	AO_GND
22	AO 0	56	AI_GND
23	AI 15 (AI 7-)	57	AI 7 (AI 7+)
24	AI_GND	58	AI 14 (AI 6-)
25	AI 6 (AI 6+)	59	AI_GND
26	AI 13 (AI 5-)	60	AI 5 (AI 5+)
27	AI_GND	61	AI 12 (AI 4-)
28	AI 4 (AI 4+)	62	AI_SENSE
29	AI_GND	63	AI 11 (AI 3-)
30	AI 3 (AI 3+)	64	AI_GND
31	AI10 (AI 2-)	65	AI 2 (AI 2+)
32	AI_GND	66	AI 9 (AI 1-)
33	AI 1 (AI 1+)	67	AI_GND
34	AI 8 (AI 0-)	68	AI 0 (AI 0+)

Table 21 5515 / 5516 Pin Definition

5515/5516 Counter Pin Definition

Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
37	CTR0_OUT	3	CTR1_OUT

Table 22 5515 / 5516 Counter Pin Define

Notes to Legend in the Pin Definitions

AI_GND	Analog Input Reference Ground
AI<0..31>	Analog Input Channel
AI SENSE	Analog Input Signal, Suitable for NRSE mode
AO_GND	Analog Output Reference Ground
AO<0..3>	Analog Output Channel
D_GND	Digital Signal Reference Ground
P<0..3>.<0..7>	Digital I/O Channel
PFI<0..15>	Programmable Function Interface
+5V_OUT	5V power supply

Table 23 Notes to Legend

3. Software

3.1 System Requirements

PCIe/PXle-5500 boards can be used in a Windows or a Linux operating system.

Microsoft Windows: Windows 7 32/64 bit, Windows 10 32/64 bit.

Linux Kernel Versions: There are many Linux versions. It is not possible JYTEK can support and test our devices under all different Linux versions. JYTEK will at the best support the following Linux versions.

Linux Version	
Ubuntu LTS	
16.04:	4.4.0-21-generic(desktop/server)
16.04.6:	4.15.0-45-generic(desktop) 4.4.0-142-generic(server)
18.04:	4.15.0-20-generic(desktop) 4.15.0-91-generic(server)
18.04.4:	5.3.0-28-generic (desktop) 4.15.0-91-generic(server)
Localized Chinese Version	
中标麒麟桌面操作系统软件（兆芯版）V7.0（Build61）: 3.10.0-862.9.1.nd7.zx.18.x86_64	
中标麒麟高级服务器操作系统软件V7.0U6: 3.10.0-957.el7.x86_64	

Table 24 Supported Linux Versions

3.2 System Software

When using the PCIe/PXle-5500 in the Window environment, you need to install the following software from Microsoft website:

Microsoft Visual Studio Version 2015 or above,

.NET Framework version is 4.0 or above.

.NET Framework is coming with Windows 10. For Windows 7, please check .NET Framework version and upgrade to 4.0 or later version.

Given the resources limitation, JYTEK only tested PCIe/PXle-5500 be with .NET Framework 4.0 with Microsoft Visual Studio 2015. JYTEK relies on Microsoft to maintain the compatibility for the newer versions.

3.3 C# Programming Language

All JYTEK default programming language is Microsoft C#. This is Microsoft recommended programming language in Microsoft Visual Studio and is particularly suitable for the test and measurement applications. C# is also a cross platform programming language.

3.4 PCIe/PXIe-5500 Series Hardware Driver

After installing the required application development environment as described above, you need to install the PCIe/PXIe-5500 hardware driver.

JYTEK hardware driver has two parts: the shared common driver kernel software (FirmDrive) and the specific hardware driver.

Common Driver Kernel Software (FirmDrive): FirmDrive is the JYTEK's kernel software for all hardware products of JYTEK instruments. You need to install the FirmDrive software before using any other JYTEK hardware products. FirmDrive only needs to be installed once. After that, you can install the specific hardware driver.

Specific Hardware Driver: Each JYTEK hardware has a C# specific hardware driver. This driver provides rich and easy-to-use C# interfaces for users to operate various PCIe/PXIe-5500 function. JYTEK has standardized the ways which JYTEK and other vendor's DAQ boards are used by providing a consistent user interface, using the methods, properties and enumerations in the object-oriented programming environment. Once you get yourself familiar with how one JYTEK DAQ card works, you should be able to know how to use all other DAQ hardware by using the same methods.

Note that this driver does not support cross-process, and if you are using more than one function, it is best to operate in one process.

3.5 Install the SeeSharpTools from JYTEK

To efficiently and effectively use PCIe/PXIe-5500 boards, you need to install a set of free C# utilities, SeeSharpTools from JYTEK. The SeeSharpTools offers rich user interface functions you will find convenient in developing your applications. They are also needed to run the examples come with PCIe/PXIe-5500 hardware. Please register and download the latest SeeSharpTools from our website, www.jytek.com.

3.6 Running C# Programs in Linux

Most C# written programs in Windows can be run by MonoDevelop development system in a Linux environment. You would develop your C# applications in Windows using Microsoft Visual Studio. Once it is done, run this application in the MonoDevelop environment. This is JYTEK recommended way to run your C# programs in a Linux environment.

If you want to use your own Linux development system other than MonoDevelop, you can do it by using our Linux driver. However, JYTEK does not have the capability to support the Linux applications. JYTEK completely relies upon Microsoft to maintain the cross-platform compatibility between Windows and Linux using MonoDevelop.

4. Operating PCIe/PXIe-5500

This chapter provides the operation guides for PCIe/PXIe-5500, including AI, AO, DI, DO, Timer and programmable I/O interface, etc.

JYTEK provides extensive examples, on-line help and documentation to assist you to operate the PCIe/PXIe-5500 board. JYTEK strongly recommends you go through these examples before writing your own application. In many cases, an example can also be a good starting point for a user application.

4.1 Quick Start

After you have installed the driver software and the SeeSharpTools, you are ready to use Microsoft Visual Studio C# to operate the PCIe/PXIe-5500 products.

If you are already familiar with Microsoft Visual Studio C#, the quickest way to use PCIe/PXIe-5500 boards is to go through our extensive examples. We provide source code of our examples. In many cases, you can modify the source code and start to write your applications.

We also provide **Learn by Example** in the following sections. These examples will help you navigate and learn how to use this PCIe/PXIe-5500.

4.2 Data Acquisition Methods

PCIe/PXIe-5500 uses a scanning method to acquire analog data, meaning there is only one ADC chip on the device and all input channels share this ADC. In the scan acquisition mode, you need to configure AI channels and set up some parameters through PCIe/PXIe-5500 driver software. The most important parameters are *Data Acquisition mode*, *Sample Rate*, *SamplesToAcquire*, *Channel Count*, *ChannelRange* and *Analog Input Terminal Type*.

AI Acquisition mode (AIMode): PCIe/PXIe-5500 provides 4 acquisition modes, **Continuous**, **Finite**, **Single Point**, **Record**, which will be described in details in Section 4.2.1-4.2.4.

SampleRate: How fast data are acquired per second per channel. For example, if the sample rate is 1000Hz, you acquire two channels of data, you will have 2000 points/second.

SamplesToAcquire: This parameter behaves differently in the different AI acquisition modes. In the continuous acquisition mode, *SamplesToAcquire* is the buffer size used in the AI acquisition task, please see Section 4.2.1; in the finite acquisition mode, it is the total number of samples to capture, please see Section 4.2.2.

Channel Count: how many channels you want to collect data. You can set up the channels in different orders, for instance 2,3,1,0. The acquired data will be arranged in the way you specify as shown in Figure 7. In this particular case, *Channel Count* is 4.

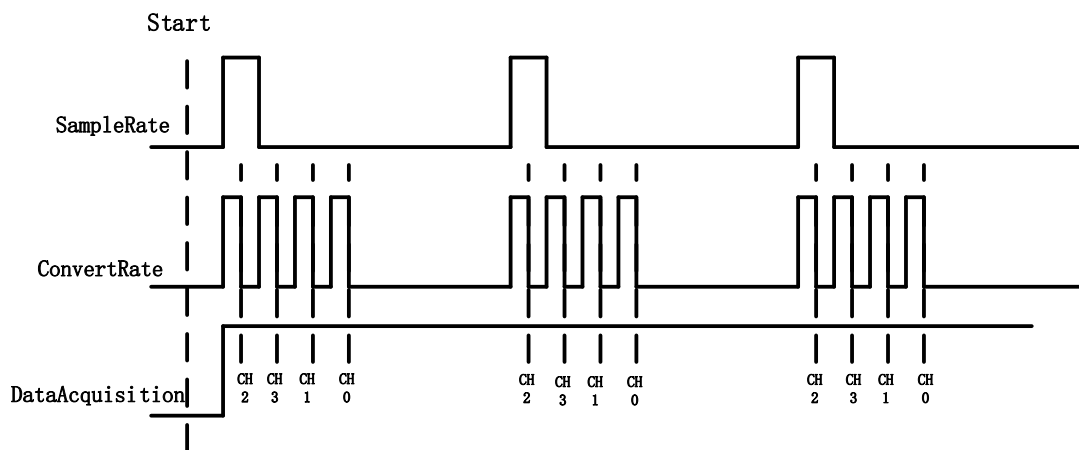


Figure 7 Sample Rate and Internal AD Conversion

ConvertRate denotes the working rate of ADC. In default: $ConvertRate = SampleRate * ChannelCount$. User can redefine the *ConvertRate* in our software. If user want to redefine *ConvertRate*, The following conditions must be met:

*Multichannel maximum sample rate (aggregate) $\geq ConvertRate \geq SampleRate * ChannelCount$.*

User can get Multichannel maximum sample rate (aggregate) from section 2.1.1

Learn by Example 4.2

- Connect the two signal source's positive outputs to PCIe/PXIe-5500 AI Ch0 (AI0+, Pin #68) and AI Ch1 (AI1+, Pin#33), two negative terminals to the ground (AI_GND, Pin#67) as shown in Figure 2 and Figure 3. (AI0+, AI_GND) and (AI1+, AI_GND) consist of two channels of RSE inputs and they share the same GND.
- Set a sinewave signal (f=4Hz, Vpp=5V) and a squarewave signal (f=4Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous MultiChannel**, set the following numbers as shown. This sample program will continuously acquire data from multiple channels.

Basic Param Configuration

Card ID	5510
Slot Number	0
Channel Count	2
AI Terminal	RSE
Sample Rate(Sa/s)	10,000
Samples to Acquire	1,000
Input Range	±10V

Figure 8 Continuous MultiChannel Parameters

- SampleRate is set by **Sample Rate**
- **Samples to Acquire** is the samples to be acquired for each channel in one block. The continuous mode will acquire blocks after blocks until **Stop** button is pressed.
- When **start** is clicked, it generates a software trigger, which starts the acquisition. The result is shown below.

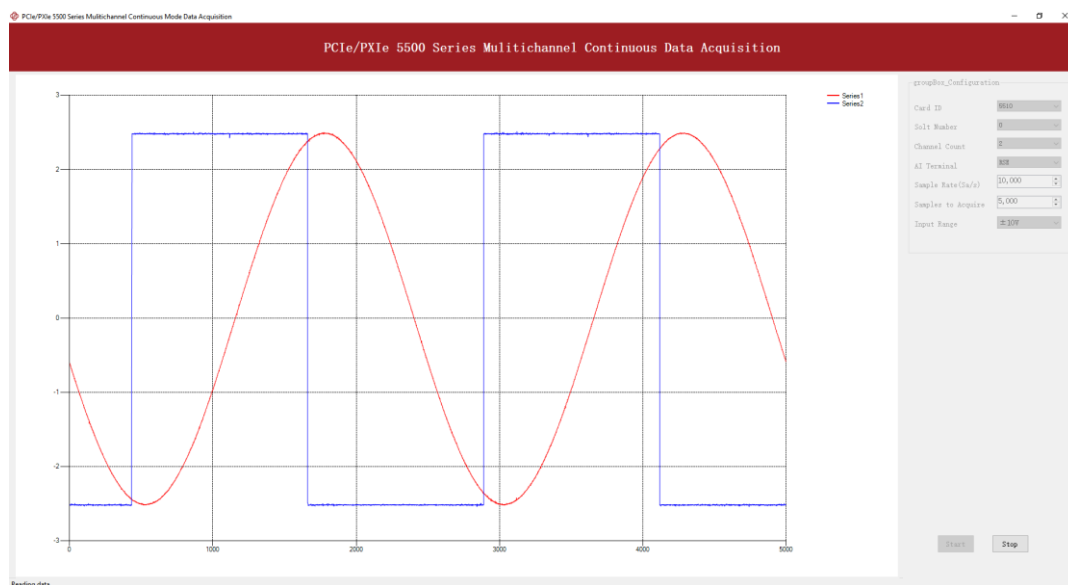


Figure 9 MultiChannel Continuous Acquisition

4.2.1 Continuous Acquisition

An AI acquisition task will acquire the data continuously until the task is stopped. The PCIe/PXIe-5500 device will continue acquiring data and save the data in a circular buffer. You specify how many samples to read back by the user buffer's length, if your program does not read the data fast enough, the circular buffer may overflow. In this case, the driver software will throw out an error message.

Tip: User buffer's length $1/10^{\text{th}}$ to $1/4^{\text{th}}$ *SampleRate* is a good start.

4.2.2 Finite Acquisition

In the Finite Acquisition mode, an AI acquisition task will capture specific total number of samples by the parameter, *SamplesToAcquire*.

You can use the sample program **Analog Input --> Winform AI Finite** to learn more about Finite Acquisition.

4.2.3 Single Point Acquisition

In the Single Acquisition mode, it is to capture a single sample for each acquisition.

You can use sample program: **Analog Input --> Console AI Single Point** to learn more about the single point Acquisition.

4.2.4 Record Acquisition

AI Task will continuously capture the data and then save them to a storage disk. During the capturing process, user can preview the captured data randomly when the capturing process is available. The mode is particularly useful for high-speed acquisition and recording applications.

4.3 Analog Input Terminal Type

The PCIe/PXIe-5500 provide 3 analog input terminal types:

- Differential (DIFF)
- Referenced Single-Ended (RSE)
- Non-Referenced Single-Ended (NRSE)

The DIFF connection is recommended for ground-referenced signal sources and it is usually better in rejecting the common-mode noise. However, to acquire one input signal, two AI channels are required to form the differential pair. The RSE and NRSE are recommended when the input signal sources are floating signals. In RSE and NRSE

modes, these floating signal sources all share the same ground reference (AI_GND). Because of it, the RSE and NRSE modes can acquire twice as many channels than the DIFF mode. Appendix 8 has more details on these 3 modes.

4.3.1 DIFF Mode

The DIFF mode connects signal's positive side to AI's positive input, signal's grounded negative side to AI's negative input as shown in Figure 10. The common noise appears on both positive and negative terminals of the differential amplifier, thus it will be cancelled out. Therefore, the DIFF mode has better signal-to-noise ratio (SNR). Please see Appendix 1 Common Analog Measurement Issues for more explanations.

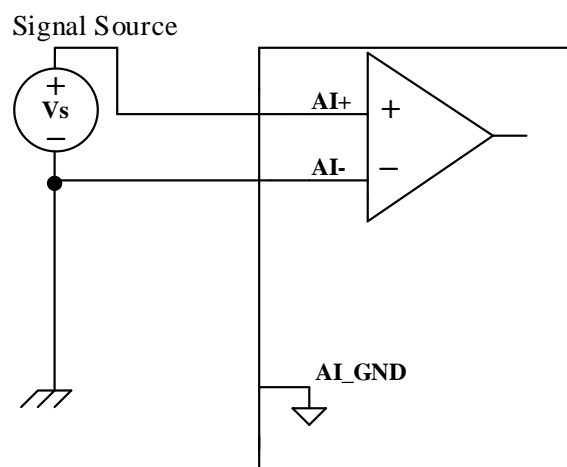


Figure 10 Differential Mode for Grounding Signals

Learn by Example 4.3.1

- Open the program **Analog Input-->Winform AI Continuous MultiChannel**
- Connect the two signal source's positive outputs to PCIe/PXIe-5500 AI Ch0 (AI0+, Pin #68) and AI Ch1 (AI1+, Pin#33), two negative terminals to AI Ch0 negative (AI0-, Pin#34) and AI Ch1 negative (AI1-, Pin#66) as shown in Figure 2 and Figure 3. (AI0+, AI0-) and (AI1+, AI1-) consist of two pairs of DIFF inputs;
- Choose Differential in **AI Terminal**;
- Set other numbers as shown and click **start**.

Basic Param Configuration

Card ID	5510
Slot Number	0
Channel Count	2
AI Terminal	Differential
Sample Rate(Sa/s)	10,000
Samples to Acquire	1,000
Input Range	$\pm 10V$

Start Stop

Figure 11 Choose Differential In AI Terminal

4.3.2 RSE Mode

In the RSE mode, all input signals' negative sides are connected to the AI ground of Instrumentation Amplifier, as shown in Figure 12. This mode works for measurements from floating sources. The RSE mode is suitable when these two conditions exist:

- The input signals are floating, meaning they are not connected to the ground
- When the common mode noise is low, meaning a clean environment.

The RSE mode offers twice as many measurement channels as the DIFF mode. Please see Appendix 1 Common Analog Measurement Issues for more explanations.

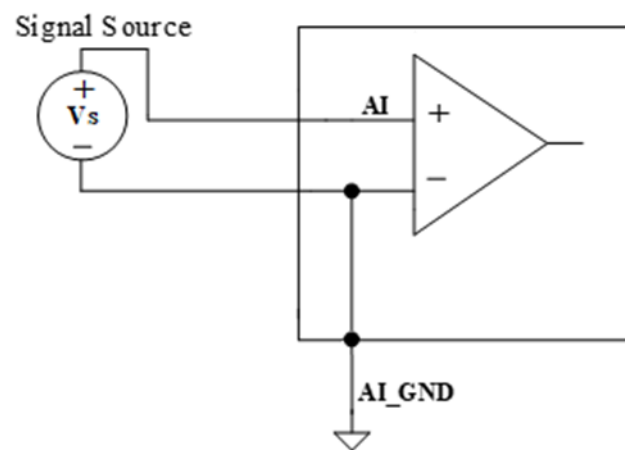


Figure 12 RSE Mode for Floating Signals

4.3.3 NRSE Mode

The NRSE mode is recommended for the measurement of ground-referenced signals, as shown in Figure 13. NRSE is also called the pseudo differential mode, because it looks very similar to a DIFF connection. In this mode, the PCIe/PXIe-5500 device offers a special reference point, AI SENSE. Instead of connecting two grounds directly, signal's ground and PXI device's ground, the input signals' ground is connected to AI SENSE to avoid the ground loop bias. The PCIe/PXIe-5500 is also designed to better reject the common mode noise than the RSE mode. Therefore the NRSE model still offers twice many channels as the DIFF mode. Please see Appendix 1 Common Analog Measurement Issues for more explanations.

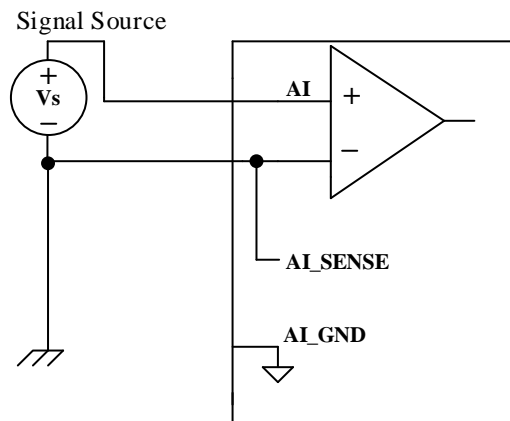


Figure 13 NRSE Mode for Grounding Signals

Learn by Example 5.3.3

- Open the program **Analog Input-->Winform AI Continuous MultiChannel**.
- This Example needs two TB-68 terminal blocks, Connector0 and Connector1 and two cables, which are connected to PCIe/PXIe-5500. Connect the two signal source's positive outputs to PCIe-5510 AI Ch0 (AI0+, Pin #68) and AI Ch1 (AI1+, Pin#33), two negative terminals to AI_SENSE 0 (Pin#62) of the first TB-68 and AI_SENSE 1 (Pin#62) of the second TB-68 as shown in Figure 2 and Figure 3. (AI0+, AI_SENSE 0) and (AI1+, AI_SENSE 1) consist of two channels of NRSE inputs.
- Choose the NRSE in **AI Terminal**
- Set other numbers as shown and click **start**.

Basic Param Configuration

Card ID	5510	▼
Slot Number	0	▼
Channel Count	2	▼
AI Terminal	NRSE	▼
Sample Rate(Sa/s)	10,000	▲▼
Samples to Acquire	1,000	▲▼
Input Range	±10V	▼

Start Stop

Figure 14 Choose NRSE In AI Terminal

4.4 Trigger Source

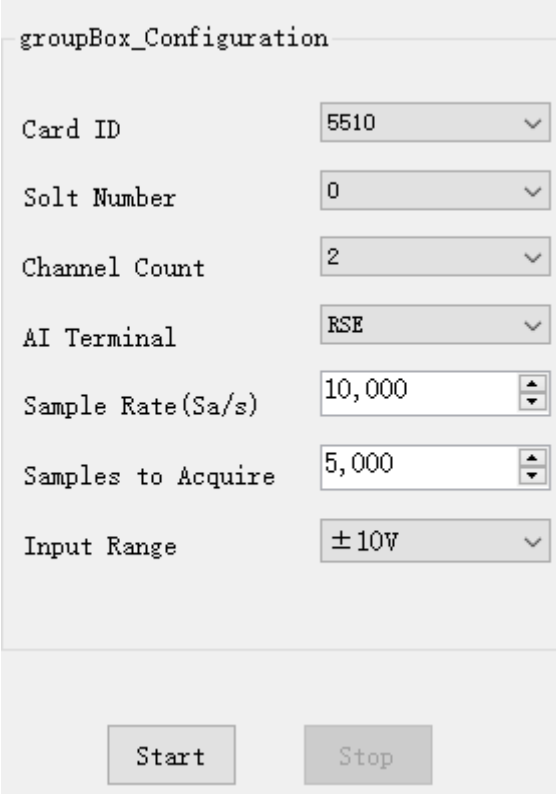
There are 4 trigger types: Immediate trigger, Software trigger, Analog trigger, and Digital trigger. The trigger type is a property and set by driver software.

4.4.1 Immediate trigger

This trigger mode does not require configuration and is triggered immediately when an operation starts. The operation can be AI, AO, DI, DO, CI, CO etc.

Learn by Example 4.4.1

- Use the same program and connection as in **Learn by Example 4.2**.



The screenshot shows a configuration window titled "groupBox_Configuration". It contains several parameters, each with a control element:

Parameter	Value	Control Type
Card ID	5510	Dropdown menu
Solt Number	0	Dropdown menu
Channel Count	2	Dropdown menu
AI Terminal	RSE	Dropdown menu
Sample Rate (Sa/s)	10,000	Spin box
Samples to Acquire	5,000	Spin box
Input Range	± 10V	Dropdown menu

At the bottom of the window, there are two buttons: "Start" and "Stop".

Figure 15 Immediate trigger Parameters

- With Immediate trigger you can click **Start** to generate the task instead of sending a trigger signal.

4.4.2 Software Trigger

A software trigger must be configured by the driver software. The trigger starts when a trigger software routine is called.

Learn by Example 4.4.2

- Connect the signal source's positive terminal to PCIe/PXIe-5500 AI Ch0 (AI0+, Pin#68), the negative terminal to the ground (AI_GND, Pin#67) as shown in Figure 2 and Figure 3. (AI0+, AI_GND) consists of a RSE input.
- Set a sinewave signal (f=4Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous Soft Trigger**, set the following numbers as shown.
- Click **Start** to run the task.

Parameter	Value
Card ID	5510
Solt Number	0
Channel ID	0
AI Terminal	RSE
Sample Rate (Sa/s)	10,000
Samples to Acquire	10,000
Input Range	±10V

Start Send Soft Trigger Stop

Figure 16 Software trigger Parameters

- Data will not be acquired until there is a positive signal from *Software Trigger* when **Send Soft Trigger** is clicked.
- After sending the trigger signal, the result will be like this:

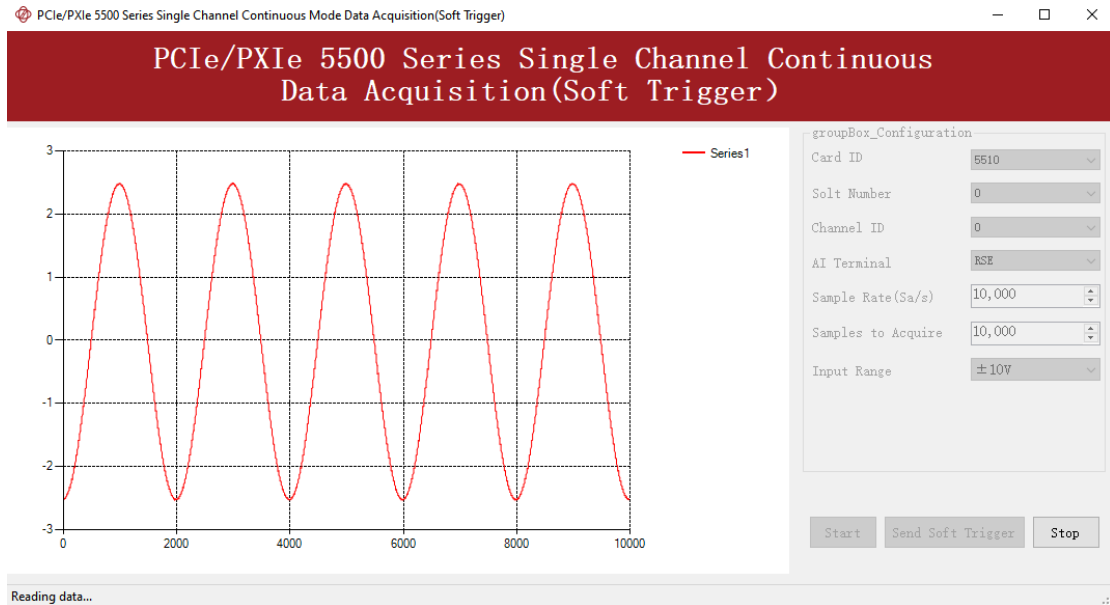


Figure 17 Software trigger Acquisition

4.4.3 External Analog Trigger

You can assign one of measurement channels as the analog trigger source. PCIe/PXIe-5500 provides three analog trigger modes:

- Edge comparator,
- Hysteresis comparator,
- Window comparator.

Analog trigger threshold range can be arbitrarily selected in the effective range of the selected channel. When setting the threshold, please pay attention to the physical unit currently in use.

Edge comparator

In the Edge comparator, there are two trigger conditions: *Rising Slope Trigger* and *Falling Slope Trigger*.

Rising Slope Trigger: The Edge comparator output is high when the signal goes above the threshold; the output is low when the signal goes below the threshold as shown in Figure 18.

Falling Slope Trigger: The Edge comparator output is high when the signal goes below the threshold; the output is low when the signal goes above the threshold as shown in Figure 19.

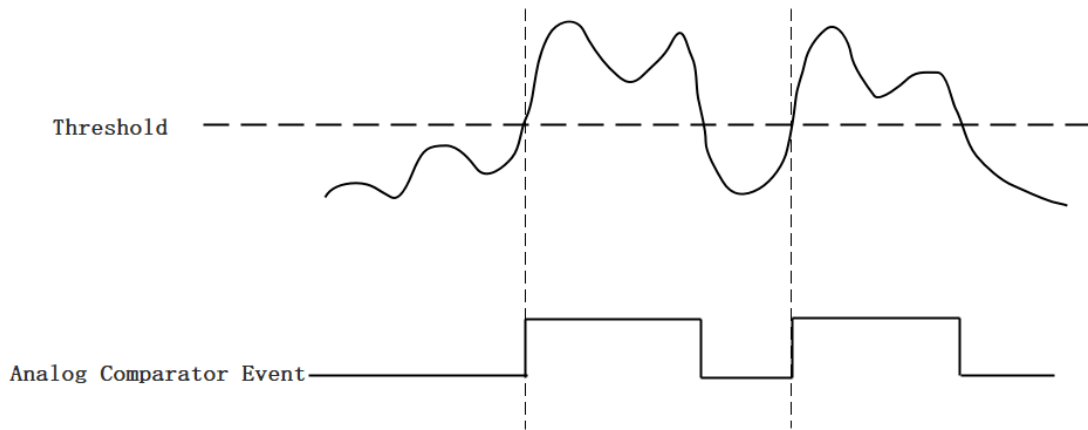


Figure 18 Rising Slope Trigger

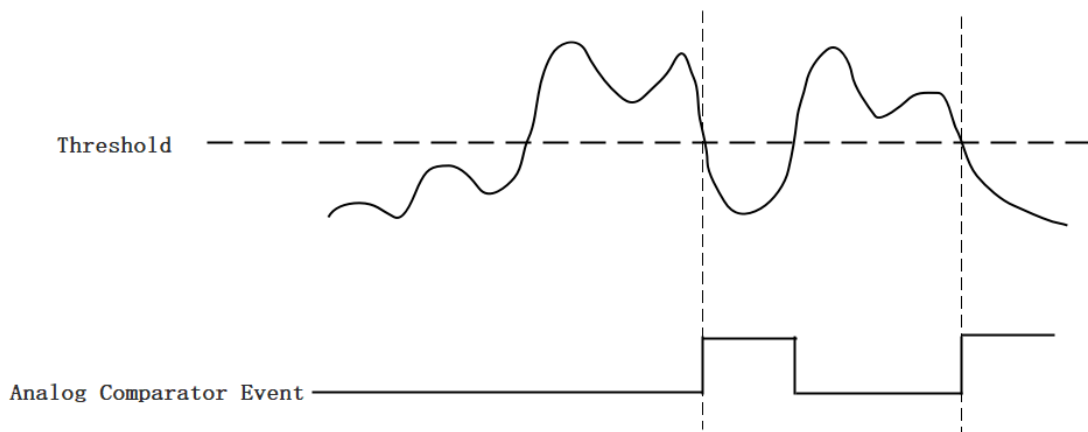


Figure 19 Falling Slope Trigger

Hysteresis Comparator

The hysteresis comparator is designed for preventing spurious triggering. You can set hysteresis region by setting high threshold and low threshold. There are two trigger conditions: *Hysteresis with Rising Slope Trigger* and *Hysteresis with Falling Slope Trigger*.

Hysteresis with Rising Slope Trigger: The Hysteresis comparator output is high when the signal must first be below the low threshold, then goes above the high threshold. The output will change to low when the signal goes below the low threshold as shown in Figure 20.

Hysteresis with Falling Slope Trigger: The Hysteresis comparator output is high when the signal must first be above the high threshold, then goes below the low threshold. The output will change to low when the signal goes above the high threshold as shown in Figure 21.

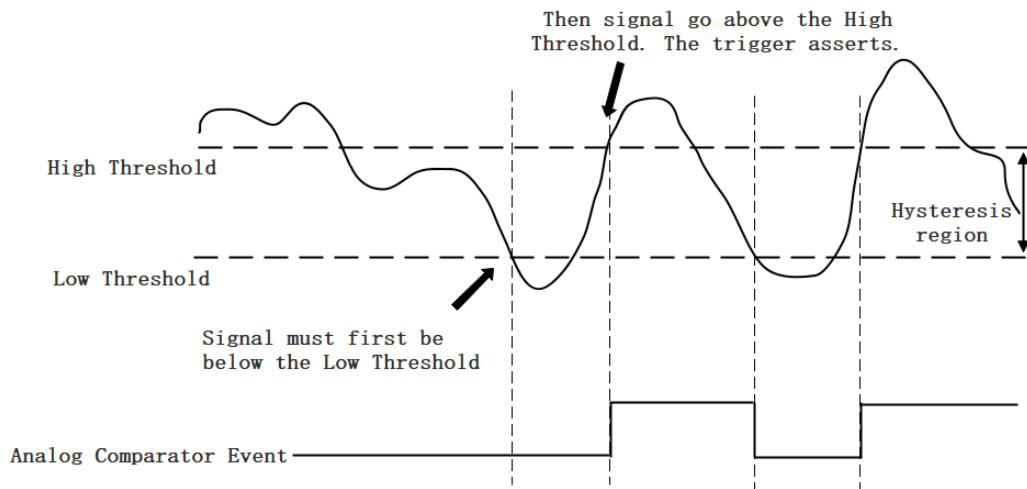


Figure 20 Hysteresis with Rising Slope Trigger

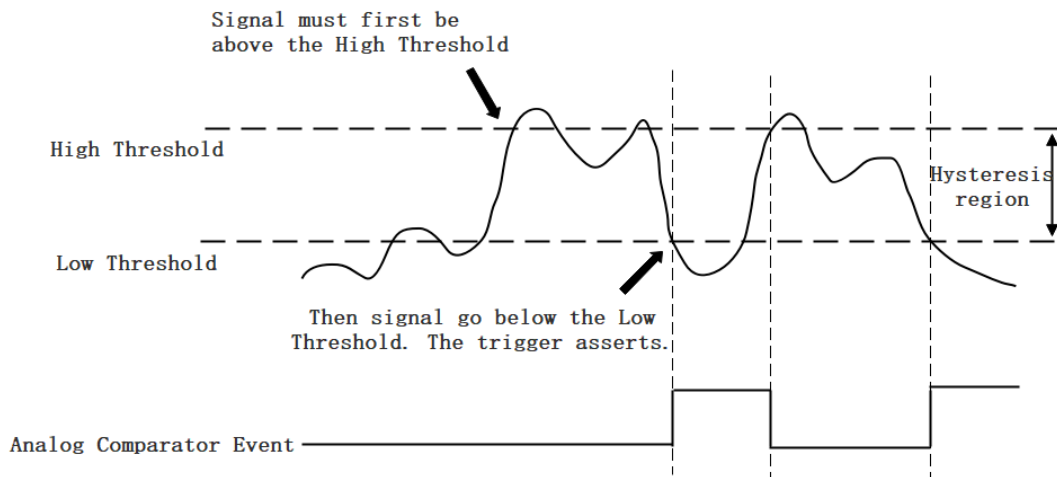


Figure 21 Hysteresis with Falling Slope Trigger

Window comparator

The window comparator is designed to acquire signal from interesting window by setting High Threshold and Low Threshold. There are two trigger conditions: *Entering Window Trigger* and *Leaving Window Trigger*.

Entering Window Trigger: The window comparator output is high when the signal enters the window defined by the *Low Threshold* and *High Threshold*. The output will change to low when the signal leaves the window as shown in Figure 22.

Leaving Window Trigger: The window comparator output is high when the signal leaves the window defined by the *Low Threshold* and *High Threshold*. The output will

change to low when the signal enters the window as shown in Figure 23 Leaving Window Trigger.

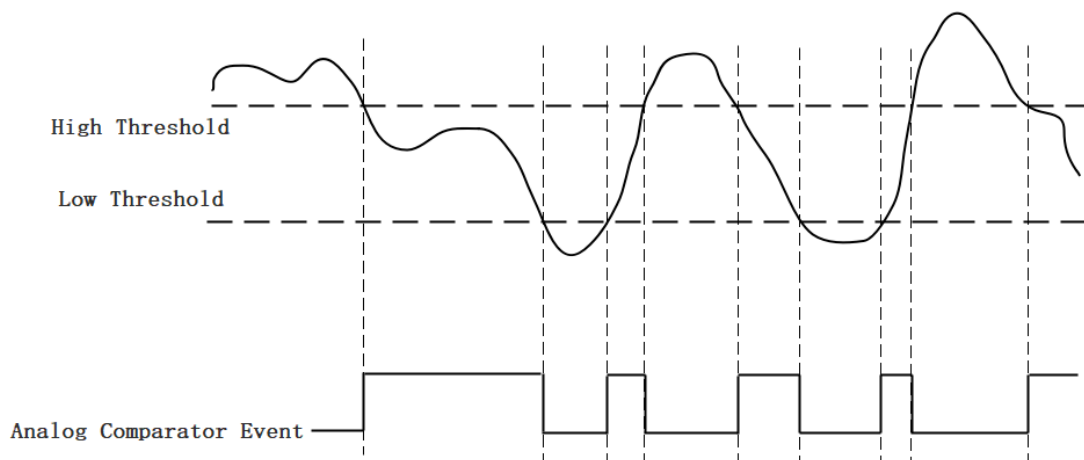


Figure 22 Entering Window Trigger

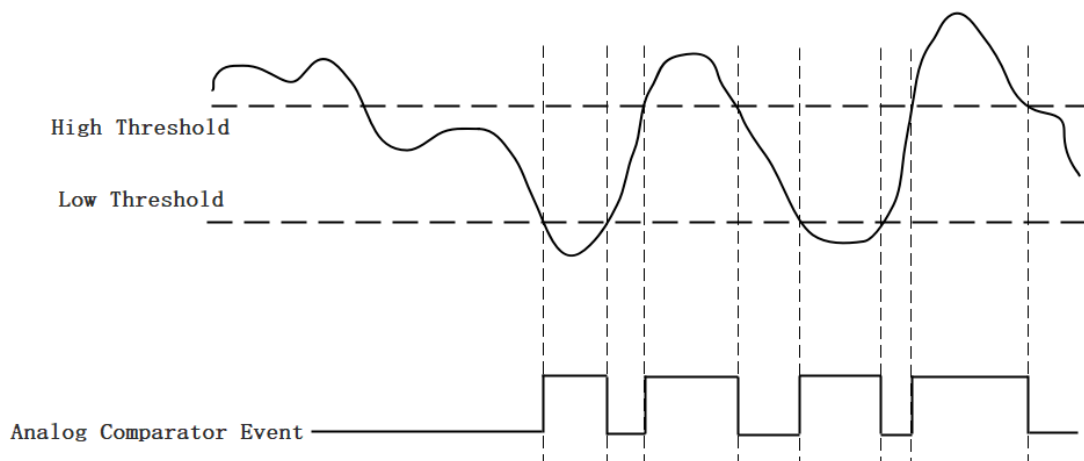


Figure 23 Leaving Window Trigger

Learn by Example 4.4.3

- Connect the signal source's positive terminal to PCIe/PXIe-5500 AI Ch0 (AI0+, Pin#68), the negative terminal to the ground (AI_GND, Pin#67) as shown in Figure 2 and Figure 3. (AI0+, AI_GND) consists of a RSE input.
- Set a sinewave signal ($f=4\text{Hz}$, $V_{pp}=5\text{V}$).
- Open **Analog Input-->Winform AI Continuous Analog Trigger**, set the following numbers as shown.

groupBox_Configuration

Card ID 5510

Slot number 0

Channel ID 0

AI Terminal RSE

Sample Rate(Sa/s) 10,000

Samples to Acquire 10,000

Input Range ±10V

groupBox_TriggerConfiguration

Trigger Source Channel_0

Trigger Comparator Edge

Trigger Edge Rising

Threshold 2.0

Start Stop

Figure 24 Analog Trigger Parameters

- Modes of the *Analog Trigger* are set by **Trigger Comparator**. Set it to **Edge**.
 - The edge of *EdgeComparator* set by **Trigger Edge**. (**Rising** and **Falling**)
 - **Trigger source** can be any channel of PCIe/PXIe-5500 analog input. Set it to **Channel_0**.
 - According to the rules of **Rising** mentioned above, the signal acquisition will not start until it raises to 2.0 V, which is set by **Threshold** above.
- Click Start, a message will appear in the lower left corner:

Waiting for the trigger signal

Figure 25 Waiting For Trigger

- This indicates the data acquisition will start only after a triggering event. In this example a trigger signal will occur when the *hysteresis comparator* meets the condition explained in 4.4.3.

■ The result is shown below:

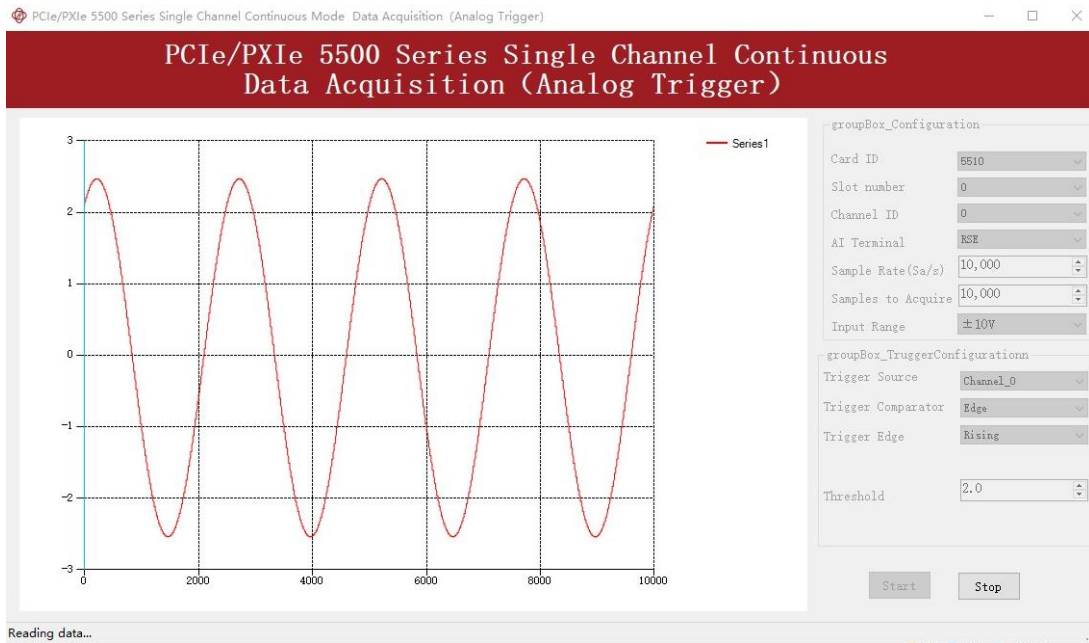


Figure 26 Analog Trigger Acquisition

- The signal starts at 2.0V, which matches the **Edge** mode set before.

4.4.4 External Digital Trigger

PCIe/PXIe-5500 supports different external digital trigger sources from PXI Trigger bus (PXI_TRIG<0..7>), PXI_STAR and connectors of front panel (PFI). The high pulse width of digital trigger signal must be longer than 20 ns for effective trigger. The module will monitor the signal on digital trigger source and wait for the rising edge or falling edge of digital signal which depending on the set trigger condition, then cause the module to acquire the data as shown in Figure 27

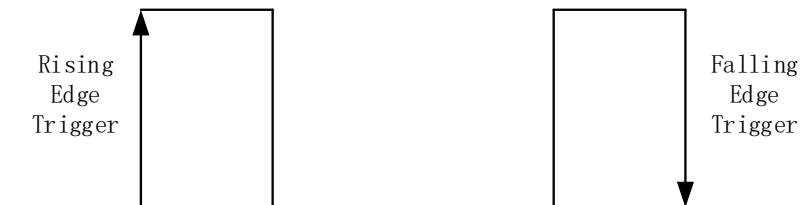


Figure 27 External Digital Trigger

Learn by Example 4.4.4

- Connect the signal source two positive terminals to PCIe/PXIe-5500 AI Ch0, (AI0+, Pin #68) and digital trigger source (PFI 0, Pin#11), two negative terminals to the ground of analog input (AI_GND, Pin#67) and the ground of digital input/output (DGND, Pin#44) as shown in Figure 2 and Figure 3. (AI0+, AI_GND) consists of a RSE input. (PFI0, DGND) provides the trigger signal.
- Set a sinewave signal (f=4Hz, Vpp=5V) and a squarewave signal (f=4Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous Digital Trigger**, set the following numbers as shown.

Card ID	5510
Slot Number	0
Channel ID	0
AI Terminal	RSE
Sample Rate (Sa/s)	10,000
Samples to Acquire	10,000
Input Range	±10V
groupBox_TriggerConfiguration	
Trigger Source	PFI0
Trigger Edge	Rising

Start Stop

Figure 28 Digital Trigger Parameters

- **Trigger Source** must match the pin on 5510.
- There are two **Trigger Edge: Rising** and **Falling**.
- Click **Start** and the result shows below:

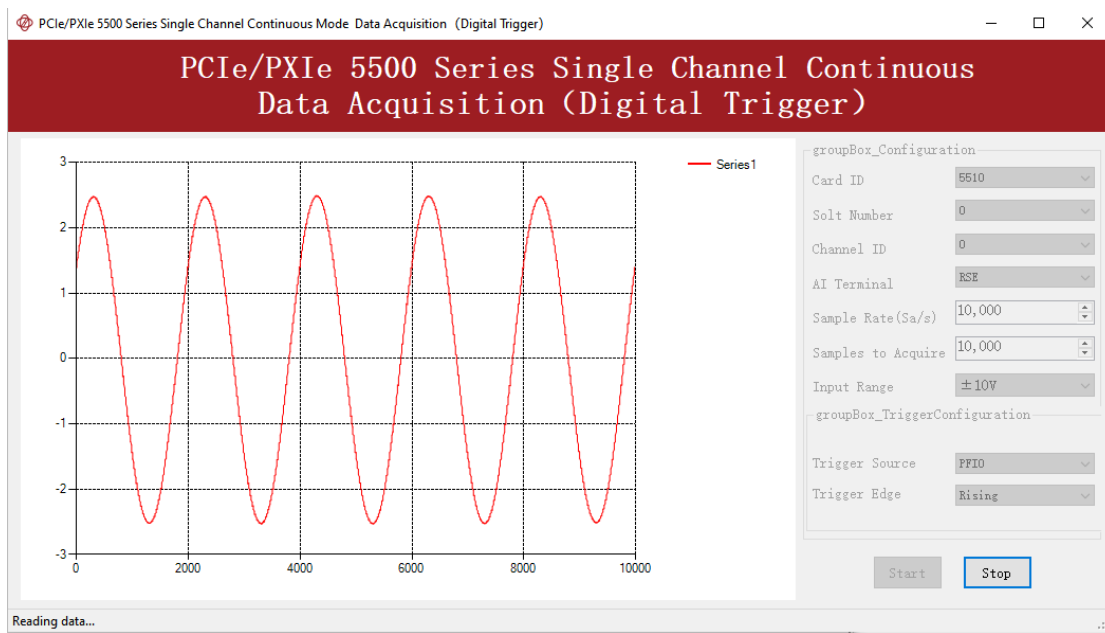


Figure 29 Digital Trigger Acquisition

- Since the squarewave is used for the digital trigger source, when a rising edge of the squarewave occurs, the digital trigger will be activated, and the data acquisition will start.

4.5 Trigger Mode

The PCIe/PXIe-5500's analog inputs support several trigger modes: start trigger, reference trigger, and re-trigger.

4.5.1 Start Trigger

In this mode, data acquisition begins immediately after the trigger. This trigger mode is suitable for continuous acquisition and finite acquisition. As shown in Figure 30.

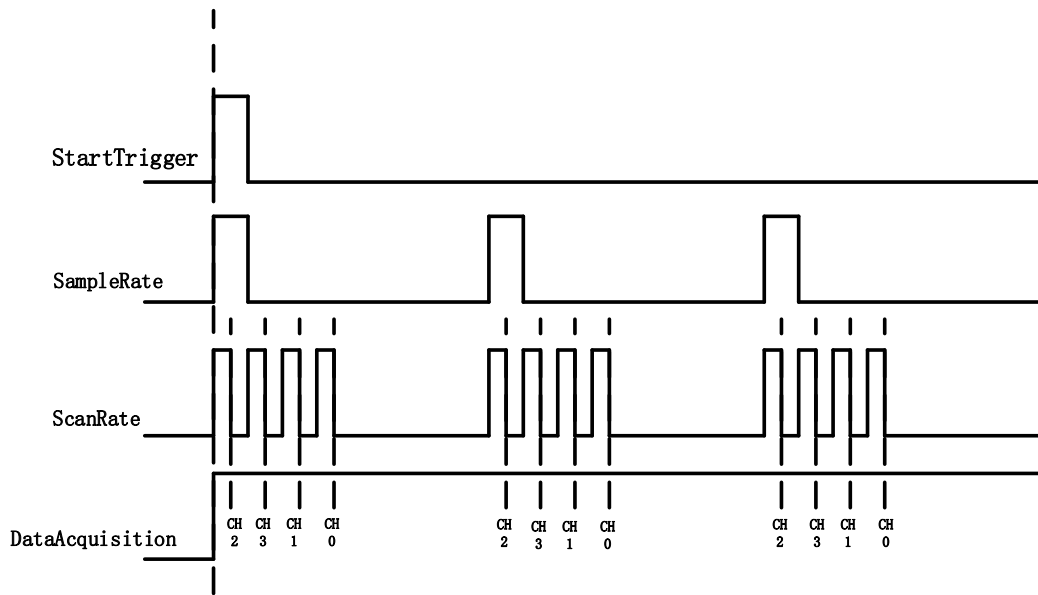


Figure 30 Start Trigger

4.5.2 Reference Trigger

This trigger mode is suitable for finite acquisition. In this mode, user can set the number of pre-trigger samples. The default number of pre-trigger points is 0. First you need to start the data acquisition. When the reference trigger condition is met, the routine will return the acquired data points. If when the points less than the pre-trigger samples, the trigger signal be ignored. An example is show below.

Example

- Total samples: 1000;
- Channel Count: 1
- Pre-trigger samples: 10;
- After triggering, it returns total 1000 samples, 10 being pre-triggered, 990 after triggering

The principle is shown in Figure 31.

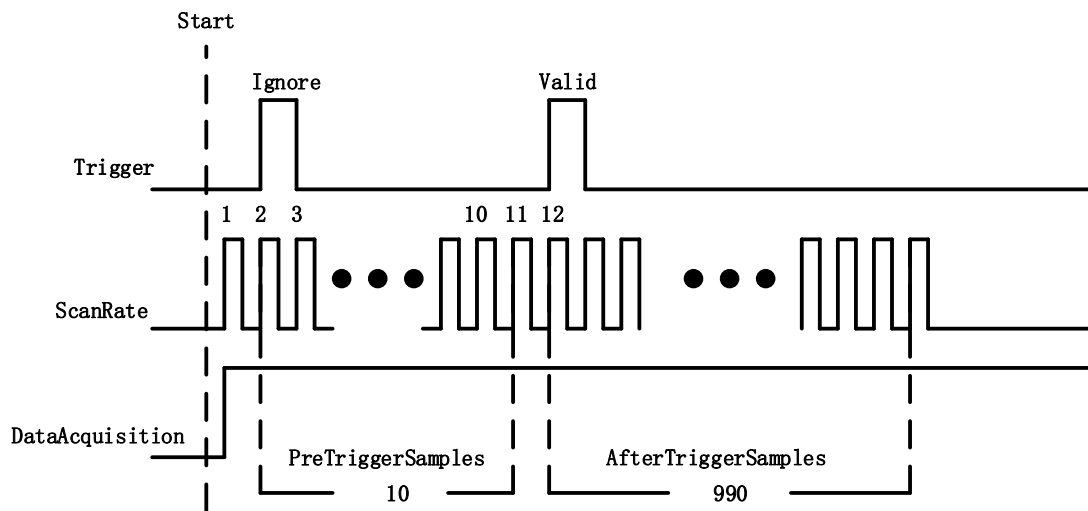


Figure 31 Reference Trigger

4.5.3 ReTrigger

PCIe/PXIe-5500 series products support retrigger mode. In the retrigger mode, you can set the number of retrigger and the length of each acquisition. Assuming that the number of re triggers is n and the length of each trigger acquisition is m , the length of all acquisition data is $n * m * \text{channelcount}$. Show in Figure 32.

When the number of retrigger is - 1, it is infinite.

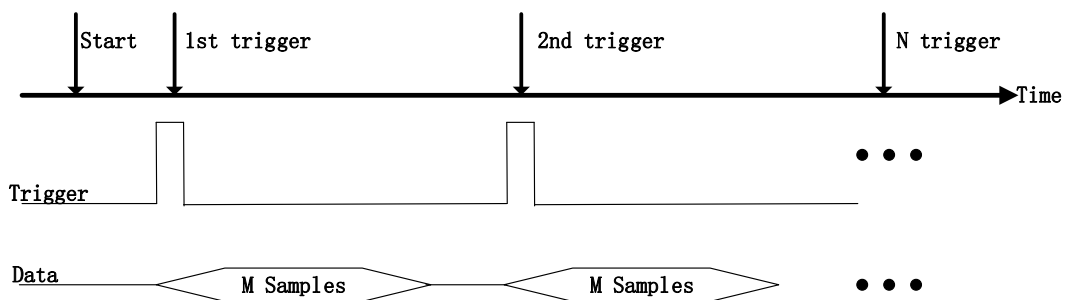


Figure 32 ReTrigger

Learn by Example 4.5

- Connect the signal source's positive terminal to PCIe-5500PCIe/PXIe-5500 AI Ch0 (AI0+, Pin#68), the negative terminal to the ground (AI_GND, Pin#67) as shown in Figure 2 and Figure 3. (AI0+, AI_GND) consists of an RSE input.
- Set a sinewave signal ($f=4\text{Hz}$, $V_{pp}=5\text{V}$).
- Open **Analog Input-->Winform AI Finite Analog Trigger**, set the following

numbers as shown.

Basic Param Configuration	
Card ID	5510
Slot Number	0
Channel ID	0
AI Terminal	RSE
Sample Rate(Sa/s)	10,000
Samples to Acquire	1,000
Input Range	±10V
Trigger Param Configuration	
Trigger Mode	Reference
Trigger Source	Channel_0
Trigger Comparator	Edge
Trigger Edge	Rising
Threshold	3.0
Retrigger Count	1
Pretrigger Samples	0

Start Stop

Figure 33 Retrigger Parameters

- You can use three different kinds of triggers in this program as mentioned in 4.5. *Start Trigger* and *Reference Trigger* can be set by **Trigger Mode**. For *ReTrigger* can be used by changing the numbers in **Retrigger Count**.
- PretriggerSamples is set by **Pretrigger Samples**.
- Now the trigger is a **Start Trigger**. Click **Start** to begin the data acquisition, the result is shown below:

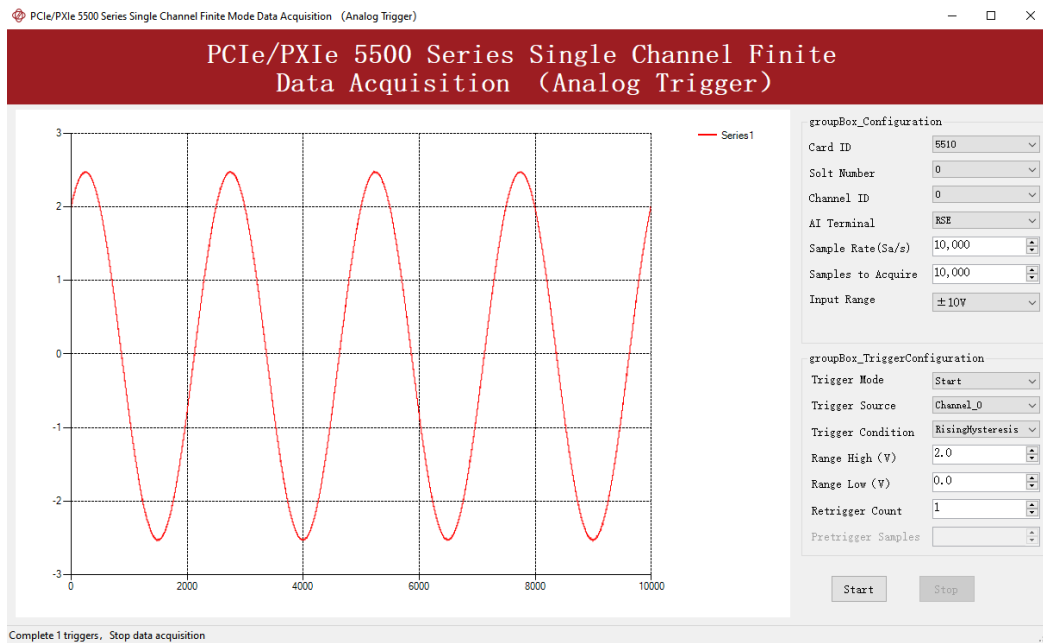


Figure 34 Retrigger In Start Trigger Mode

- Now change the **Trigger Mode** to **Reference** mode with **Pretrigger Samples 1000**. A different result shows below:

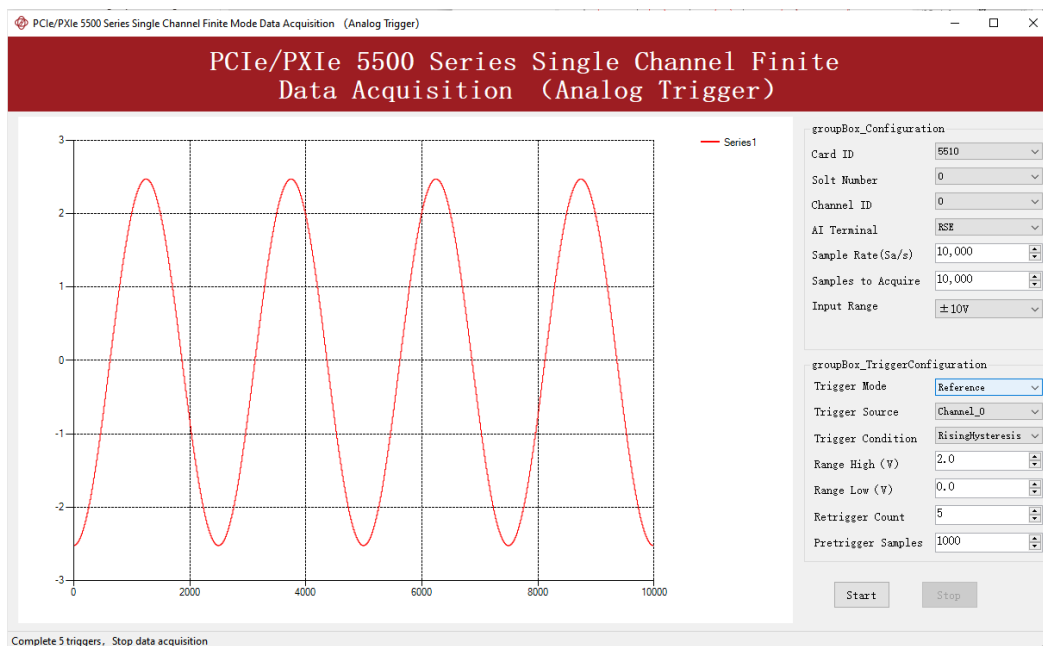
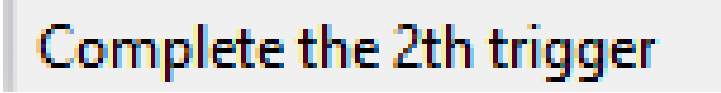


Figure 35 Retrigger In Reference Trigger Mode

- You can see the horizontal movement between two signals due to the change of **Trigger Mode**.

-
- Now change the mode of trigger to *Retrigger* through giving **Retrigger Count** a number other than 0 and click **Start**. A message will appear in the lower left corner: “Complete the n^{th} trigger”.



Complete the 2th trigger

Figure 36 Complete Retrigger Count

- It shows the acquisition process through every trigger signal.

4.6 AO Operations

The PCIe/PXIe-5500 AO provides 16-bit simultaneous outputs. The analog output has three modes of operation: Finite, ContinuousWrapping, and ContinuousNoWrapping.

4.6.1 Finite Output

The finite output requires the user to write a piece of data. After starting the AO, it starts to output the written data until the output is completed.

Learn by Example 4.6.1

- Connect PCIe/PXIe-5500 AO Ch0 (AO0, Pin #22) to AI Ch0 (AI0+, Pin#68), Ground of AO0 (AO_GND, Pin#55) to Ground of AI0 (AI_GND, Pin#67). (AI0+, AI_GND) consists of a RSE input; (AO0, AO_GND) consists of an output. PCIe/PXIe-5500 sends an analog signal through (AO0, AO_GND) and reads back the signal from (AI0+, AI_GND).
- Open **Analog Input-->Winform AI Continuous**, set the following numbers as shown.

Card ID	5510	▼
Slot Number	0	▼
Channel ID	0	▼
AI Terminal	RSE	▼
Sample Clock	Internal	▼
External Clock	PFI2	▼
Sample Rate(Sa/s)	10,000	▲▼
Samples to Acquire	3000	▲▼
Input Range	±10V	▼

Start Stop

Figure 37 AI Continuous Parameters

- Click **Start** to start the data acquisition.
- Open **Analog Output-->Winform AO Finite**, set the following numbers as shown:

Card ID	5510	Update Rate(Sa/s)	2,000,000
Solt Number	0	Samples to Update	1,000,000
Channel ID	0	Output Range	±10V

Waveform Configuration

Wave Type	Wave Amplitude	Wave Frequency
SineWave	5	10

Start

Stop

Figure 38 AO Finite Output Parameters

- Click **Start** to generate a **SineWave**. The generated signal is shown below:

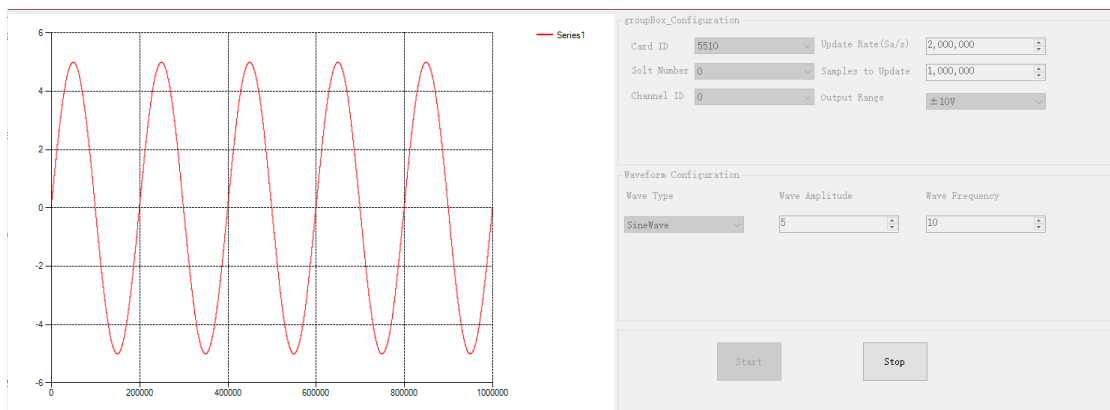


Figure 39 AO Finite Signal

- And the received signal is shown below.

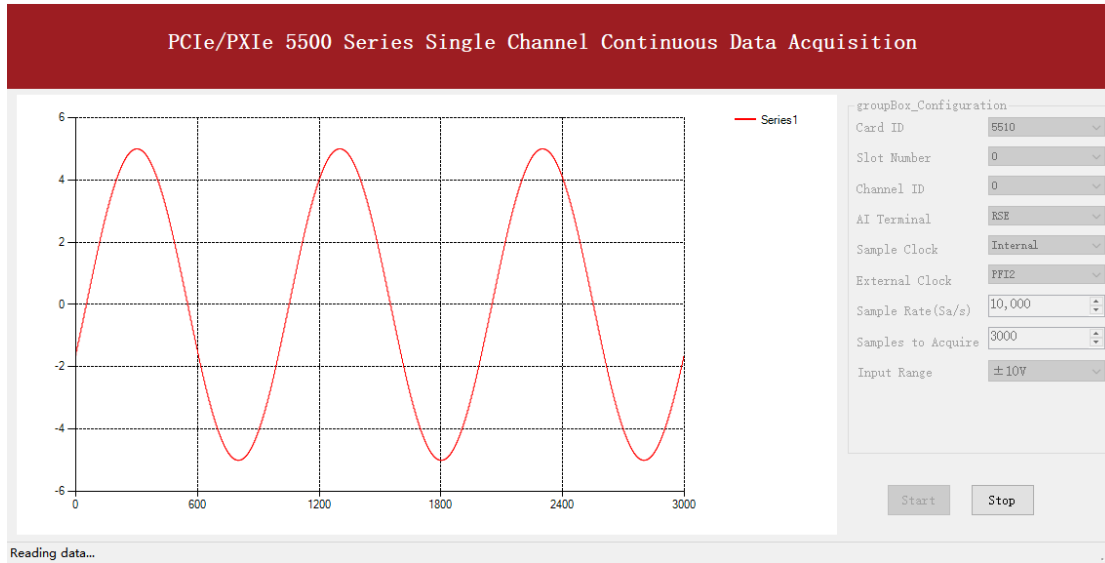


Figure 40 AI Acquisition Signal

- The analog signal is successfully generated and received by PCIe/PXIe-5500.

4.6.2 Continuous NoWrapping Output

The continuous acyclic output needs to write a piece of data before starting the AO. After the AO starts, user needs to continuously write new data to ensure the continuous output of the AO.

Learn by Example 4.6.2

- Connect PCIe/PXIe-5500 AO Ch0 (AO0, Pin #22) to AI Ch0 (AI0+, Pin#68), Ground of AO0 (AO_GND, Pin#55) to Ground of AI0 (AI_GND, Pin#67). (AI0+, AI_GND) consists of a RSE input; (AO0, AO_GND) consists of an output.
- PCIe/PXIe-5500 sends an analog signal through (AO0, AO_GND) and reads back the signal from (AI0+, AI_GND).
- Open **Analog Input-->Winform AI Continuous**, set the following numbers as shown.

Card ID	5510	▼
Slot Number	0	▼
Channel ID	0	▼
AI Terminal	RSE	▼
Sample Clock	Internal	▼
External Clock	PFI2	▼
Sample Rate(Sa/s)	10,000	▲▼
Samples to Acquire	3000	▲▼
Input Range	±10V	▼

Start Stop

Figure 41 AI Continuous Parameters

- Click **Start** to start the data acquisition.
- Open **Analog Output-->Winform AO Continuous NoWrapping**, set the following numbers as shown:

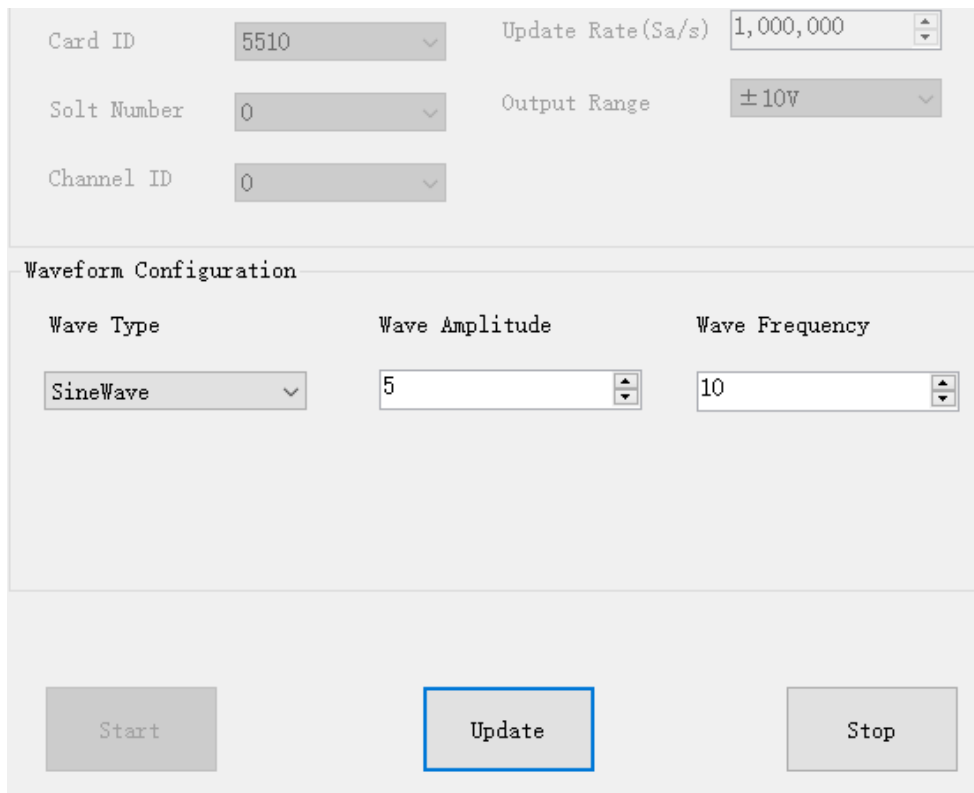


Figure 42 AO ContinuousNoWrapping Output Parameters

➤ In no wrapping analog output you can change the parameter of the signal whenever you want in **Waveform Configuration** when generating the wave. After the configuration you should click **Update** to apply the changes.

■ Click **Start** to generate a sine wave first. The result is shown below.

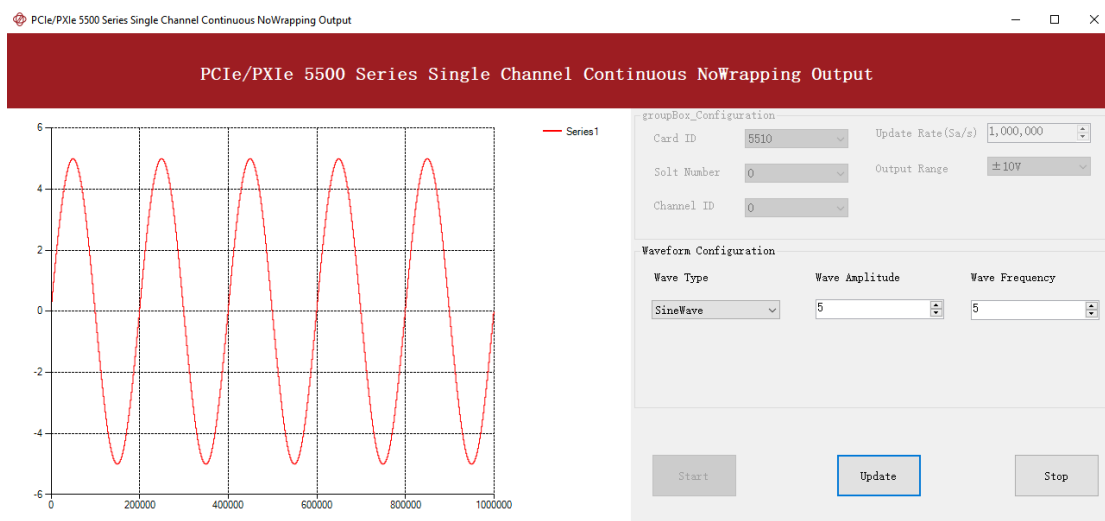


Figure 43 AO ContinuousNoWrapping Signal

- And the received signal is shown below.

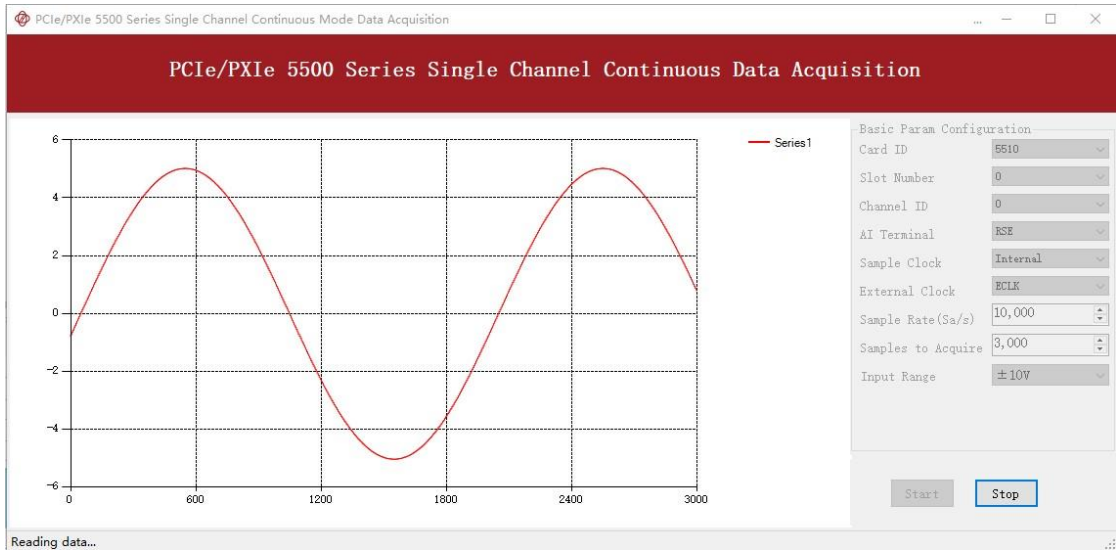


Figure 44 AI Acquisition AO Sin Signal

- Now change the **Wave Type** to **SquareWave** and click **Update** to generate it. The result is shown below.

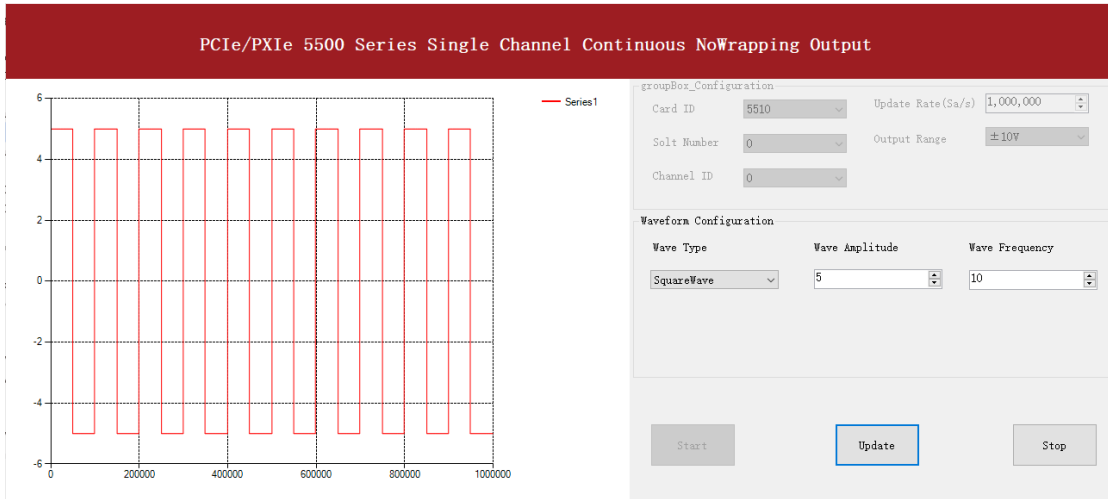


Figure 45 Update AO Square Signal

- And the received signal is shown below.

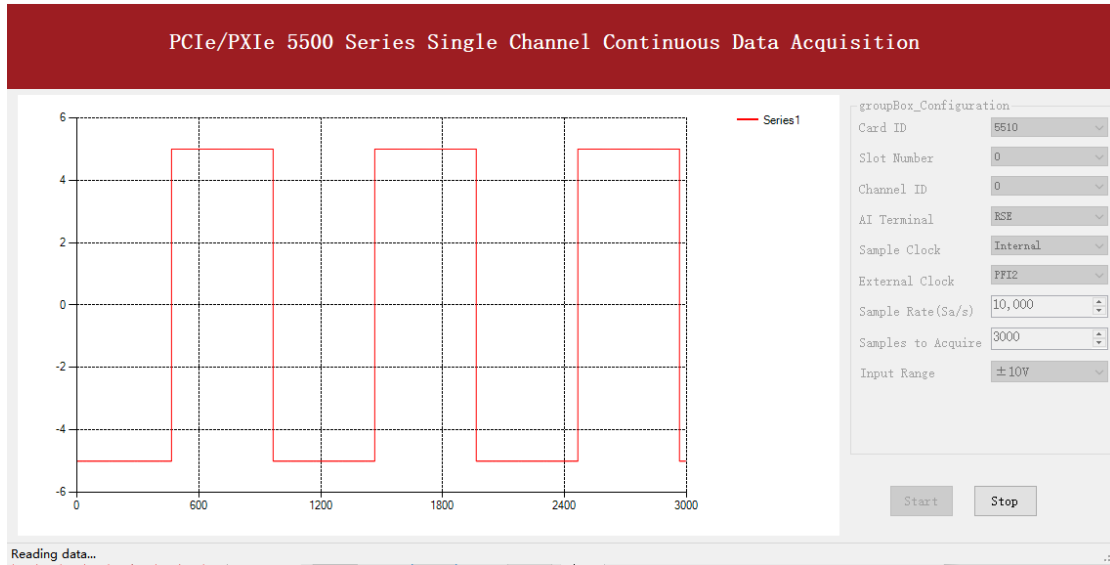


Figure 46 AI Acquisition AO Square Signal

- The analog signal is successfully generated and received by PCIe/PXIe-5500.

4.6.3 Continuous Wrapping Output

The continuous loop output first writes a piece of data before starting the AO. After the AO starts, the board will repeatedly output this data until user sends a stop command.

Learn by Example 4.6.3

- Connect PCIe/PXIe-5500 AO Ch0 (AO0, Pin #22) to AI Ch0 (AI0+, Pin#68), Ground of AO0 (AO_GND, Pin#55) to Ground of AI0 (AI_GND, Pin#67). (AI0+, AI_GND) consists of a RSE input; (AO0, AO_GND) consists of an output.
- PCIe/PXIe-5500 sends an analog signal through (AO0, AO_GND) and reads back the signal from (AI0+, AI_GND).
- Open **Analog Input-->Winform AI Continuous**, set the following numbers as shown.

Card ID	5510
Slot Number	0
Channel ID	0
AI Terminal	RSE
Sample Clock	Internal
External Clock	PFI2
Sample Rate(Sa/s)	10,000
Samples to Acquire	3000
Input Range	±10V

Figure 47 AI Continuous Parameters

- Click **Start** to start the data acquisition.
- Open **Analog Output-->Winform AO Continuous Wrapping**, set the numbers as shown.

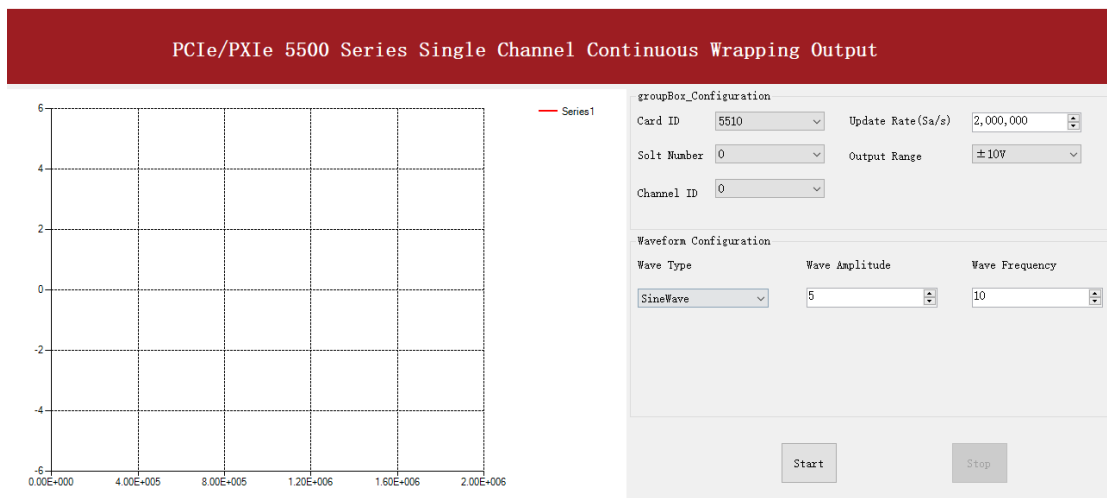


Figure 48 AO Continuous Wrapping Parameters

- Click **Start** to generate the signal. The result is shown below.



Figure 49 AO Continuous Wrapping Signal

- And the received signal is shown below.

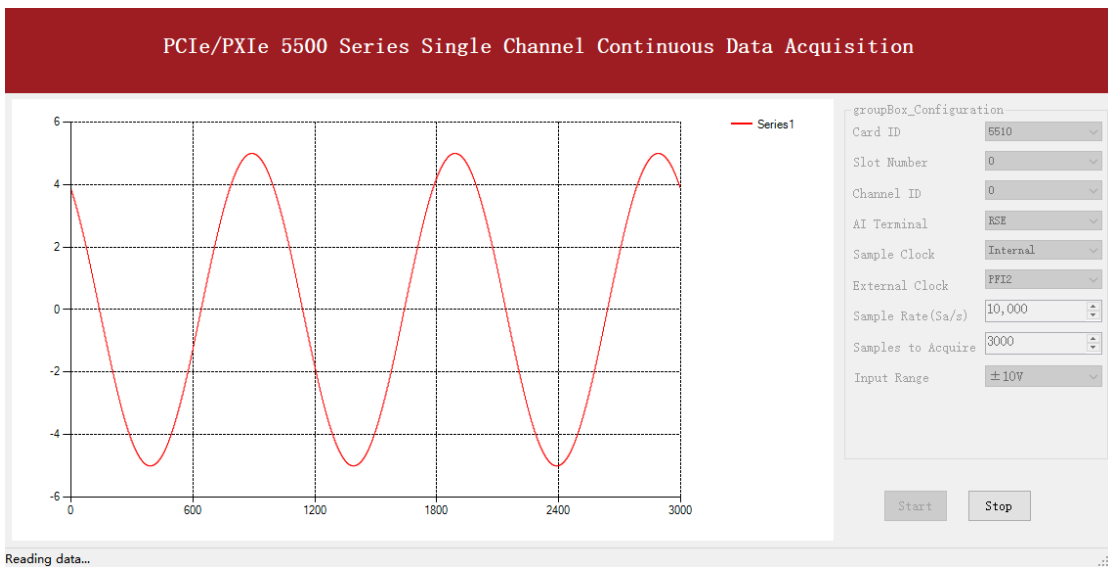


Figure 50 AI Acquisition AO Signal

- The analog signal is successfully generated and received by PCIe/PXIE-5500.

4.7 Digital I/O Operations

The PCIe/PXle-5500 provides powerful programmable digital I/O functions.

4.7.1 Static DI/DO

Programmable I/O supports static TTL, 6 ports (0,1,2,3,4,5) which are in total 48 digital I/O channels. User can access these I / O information through software polling.

Learn by Example 4.7.1

- In this example PCIe/PXle-5500 outputs a digital signal by its DO function and reads it back by its DI function.
- Connect Connector1 of PCIe/PXle-5500 to the TB-68 terminal block according to Figure 3.
- Connect Port 1/Line 0~7 (P1.0~P1.7) to Port 2/Line 0~7 (P2.0~2.7). PCIe/PXle-5500 sends a digital signal through Port 1 and reads the signal back from Port 2.
- Open the first program **Digital Output-->Winform DO SinglePoint**.
- Select **port 1** for **Digital Output**, Set Line 1,3,5,7 in High-Level positions, make sure all other lines are in Low-Level positions. Click **Start** to generate the High-Levels as shown.

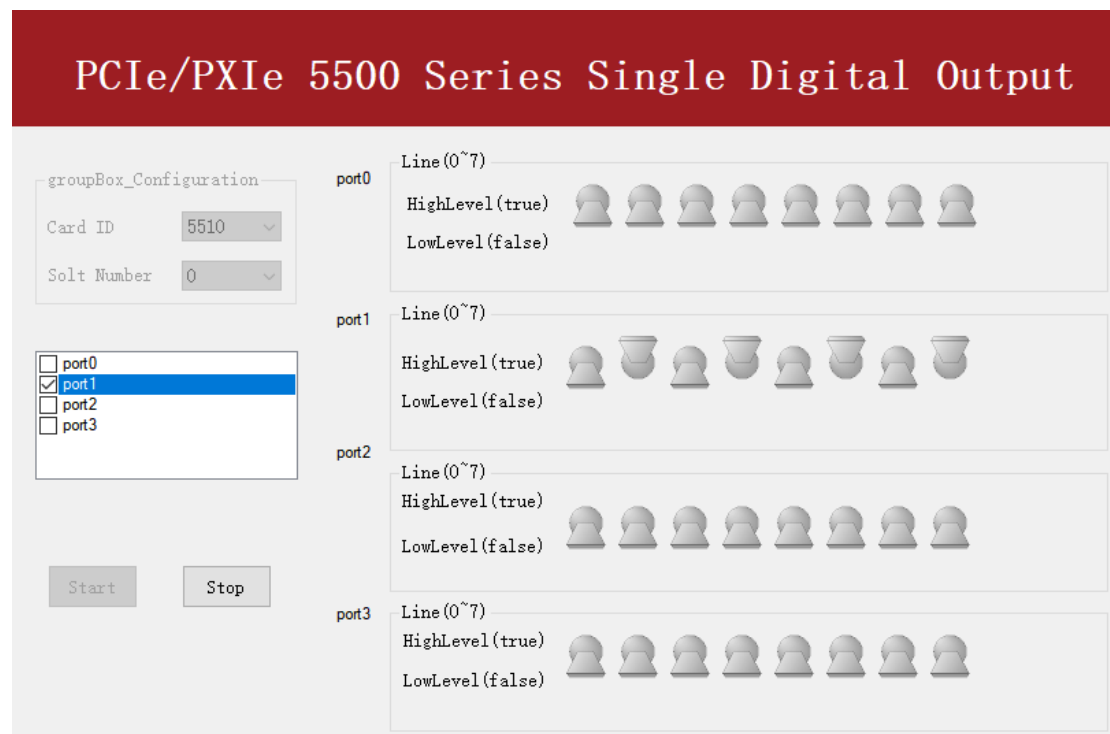


Figure 51 Single Digital Output

- Open the second program **Digital Input-->Winform DI SinglePoint**.

- Select **port 2** for **Digital Input** as shown, and click **Check DI Status**. The result is shown below.

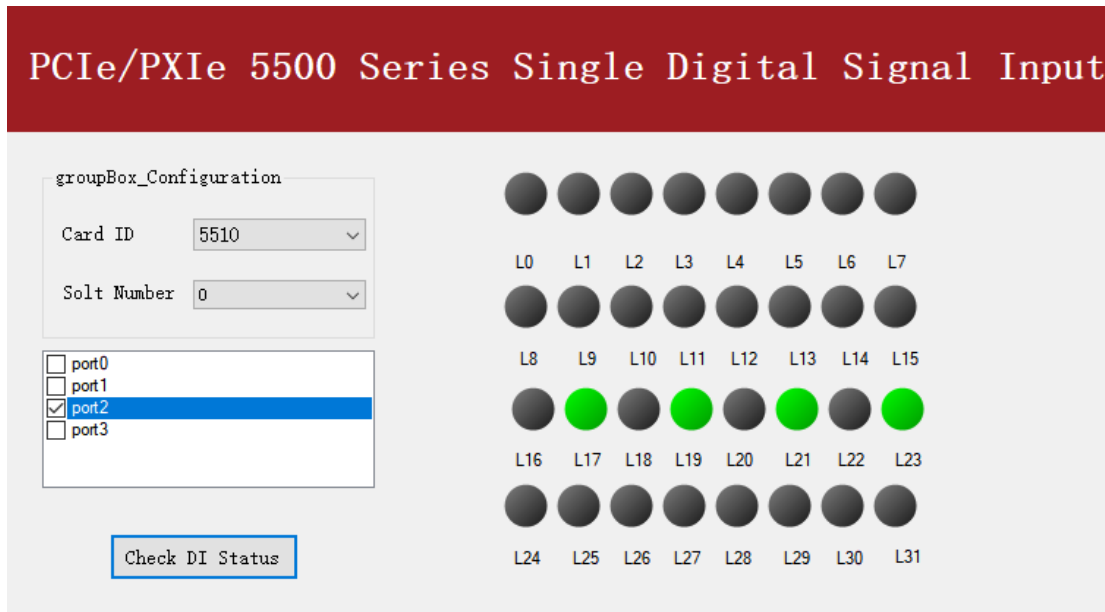


Figure 52 Single Digital Input

- The result matches the high and low levels set before.

4.7.2 Dynamic DI/DO

The PCIe/PXIE-5500 supports both dynamic DI/DO operation with a maximum sample rate (update rate) of up to 10MHz. User can acquire or output digital waveforms in this way.

Learn by Example 4.7.2

- In this example PCIe/PXIE-5500 outputs a squarewave by its DO function and reads it back by its DI function.
- Connect Connector1 of PCIe/PXIE-5510/5511 or Connector0 of PCIe/PXIE-5515/5516 to the TB-68 terminal block according to Figure 3.
- Connect PCIe/PXIE-5510/5511 Port 1/Line 0~7 (P1.0~P1.7) to Port 2/Line 0~7 (P2.0~2.7) or PCIe/PXIE-5515/5516 Port 0/Line 0~7 (P0.0~P0.7) to Port 1/Line 0~7 (P1.0~1.7). PCIe/PXIE-5500 sends digital signals through Port 1/Port 0 and reads them back from Port 2/Port 1.
- Open **Digital Input-->Winform DI Continuous** and set the numbers as shown. Select **port 2**(PCIe/PXIE-5510/5511) or **port 1**(PCIe/PXIE-5515/5516).

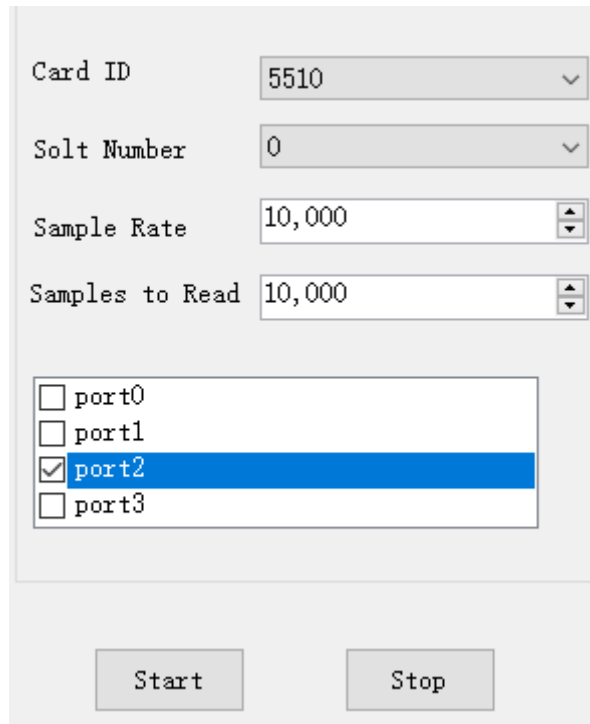


Figure 53 DI Continuous Parameters

- Click **Start** to begin the data acquisition.
- Open **Digital Output--> Winform DO Continuous NoWrapping** and set the numbers as shown.
- Click **Start** to generate the signal. The result is shown below.

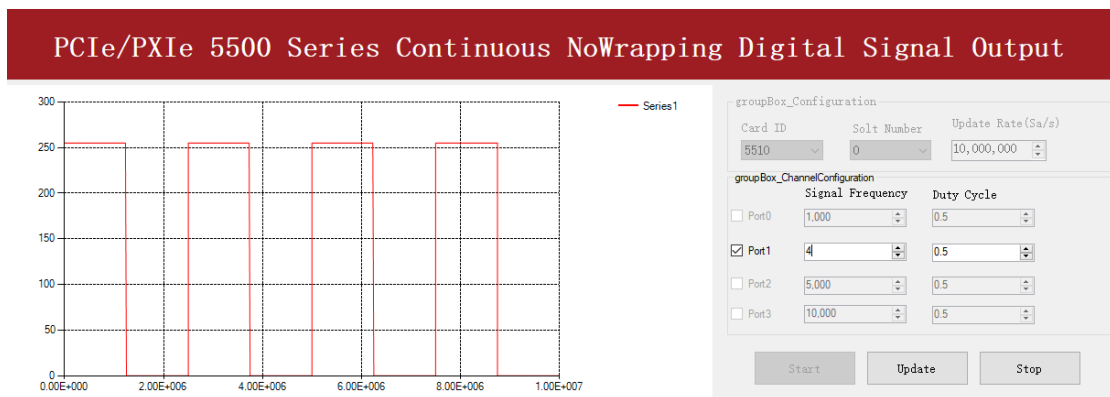


Figure 54 DO ContinuousNoWrapping Output

- In program **Winform DI Continuous**, you can see the acquired signal. Select **port 1**(PCIe/PXIe-5510/5511) or **port 0**(PCIe/PXIe-5515/5516).

PCIe/PXIe 5500 Series Continuous Digital Signal Input

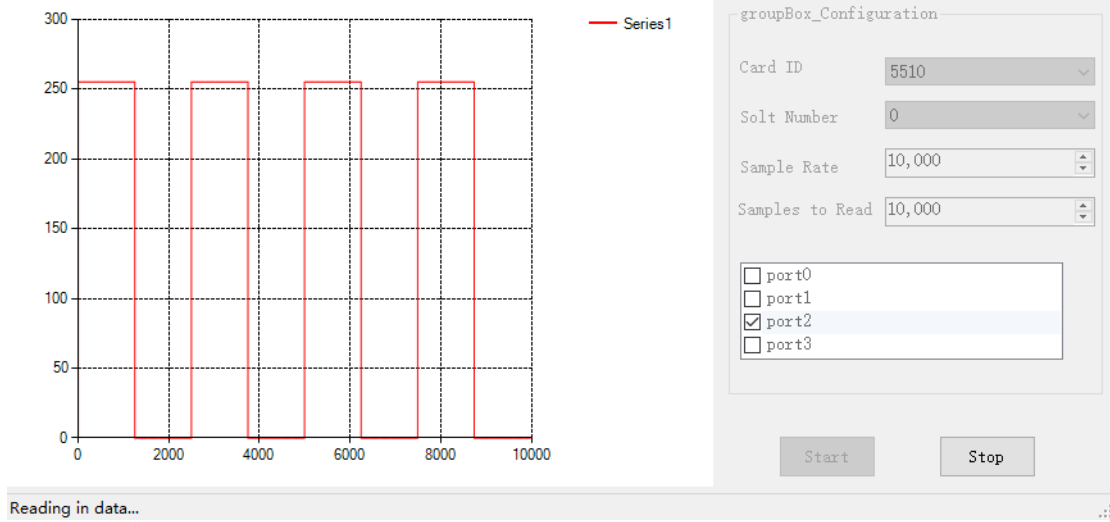


Figure 55 DI Continuous Acquisition

- The digital signal is successfully generated and acquired by PCIe/PXIe-5500.

4.8 Counter Input Operations

The PCIe/PXIe-5500 has four or two identical 32 bits timers/counters.

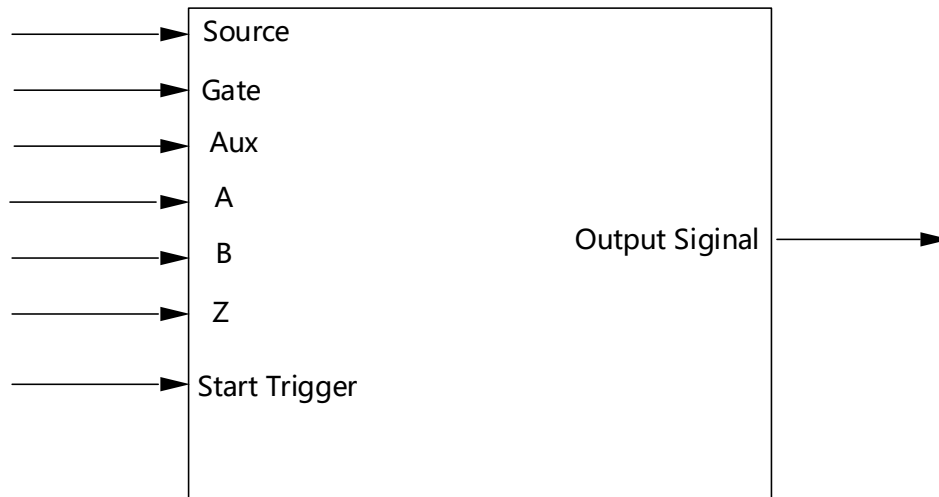


Figure 56 Counter Terminal

Each counter has seven input terminals and one output terminal, and these terminals have different functions in different counter input application types, including:

- Edge Counting
- Pulse Measurement
- Frequency Measurement
- Period Measurement
- Two-Edge Separation
- Quadrature Encoder (X1, X2, X4)
- Two-Pulse Encoder

For buffered acquisition, each counter has a separate DDR storage space and requires a sample clock.

For each counter input application type, the measured signal needs to be connected to different terminals, as shown in the following table.

Measured Signal	Terminal
Edge Counting	Source
Pulse Measurement	Gate
Frequency Measurement	Gate
Period Measurement	Gate
Two-Edge Separation	Gate、Aux
Quadrature Encoder (X1, X2, X4)	A、B、Z
Two-Pulse Encoder	A、B

Figure 57 Counter Signal Wiring Instruction

4.8.1 Edge Counting

The counter counts the number of active edges of input signal.

Timing

1) Single Mode

The count value is written to the register on each rising edge or falling edge of the signal to measure as shown in Figure 58.

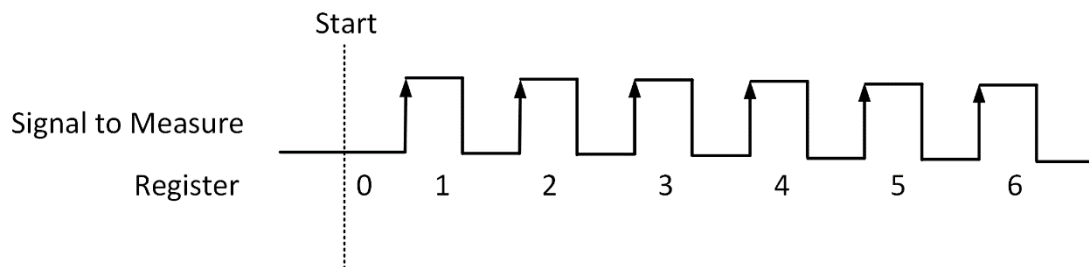


Figure 58 Simple Edge Counting in Single Mode

2) Finite/Continuous Mode with Internal Sample Clock

The count value is stored into the buffer on each rising edge or falling edge of the sample clock as shown in Figure 59.

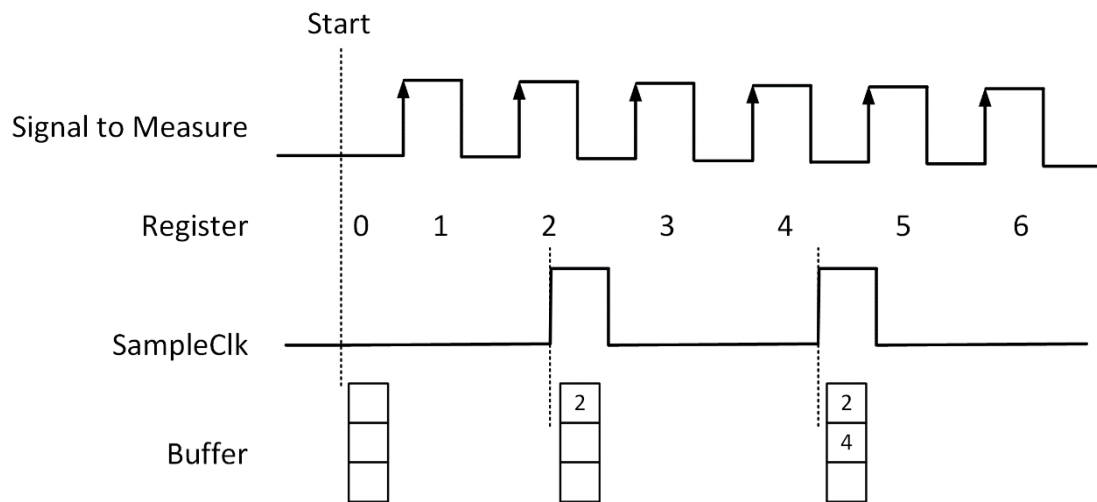


Figure 59 Buffered Edge Counting with Internal Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

The count value is stored into the buffer on each rising edge or falling edge of the signal to measure as shown in Figure 60.

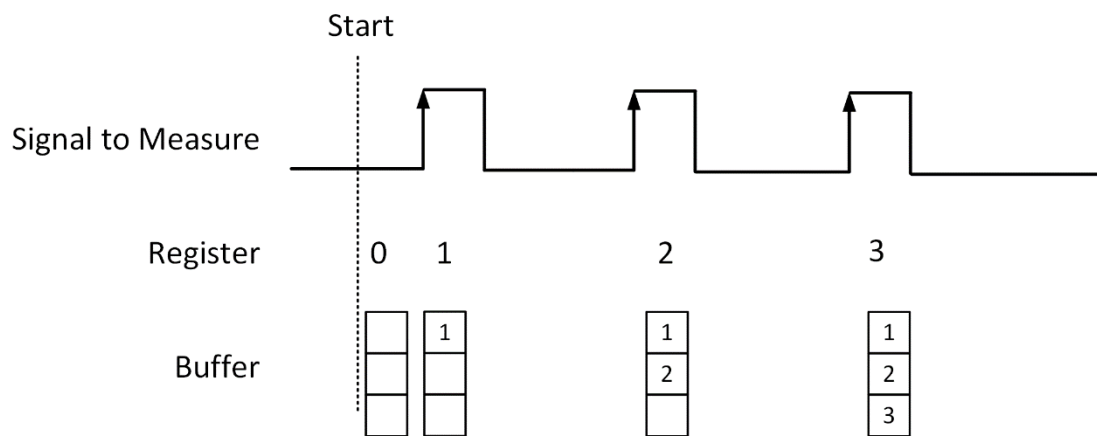


Figure 60 Simple Edge Counting with Implicit SampleClk

Counting Direction

User can control the counting direction through software configuration or by an input signal with AUX terminal. When using an input signal to control the counting direction, the counter counts up when the signal is high and counts down when the signal is low as shown in Figure 61.

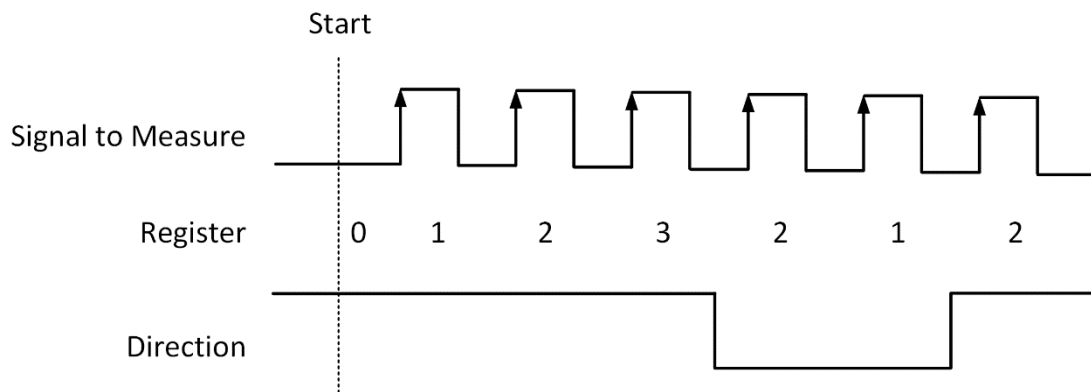


Figure 61 Count Direction

Learn by Examples 4.8.1

- Connect the signal source's positive terminal of a signal source to PCIe-5510 counter0's edge counting source (CTR0_Source/A, Pin#11), negative terminal to the ground (DGND, Pin#44) as shown in Figure 2 and Figure 3. (CTR0_Source, DGND) consists of an edge counting counter input and they share the same ground.
- Set a squarewave signal (f=1Hz, Vpp=5V).

Single Mode

- Open **Counter Input-->Winform CI Single EdgeCounting**, set the following numbers as shown:

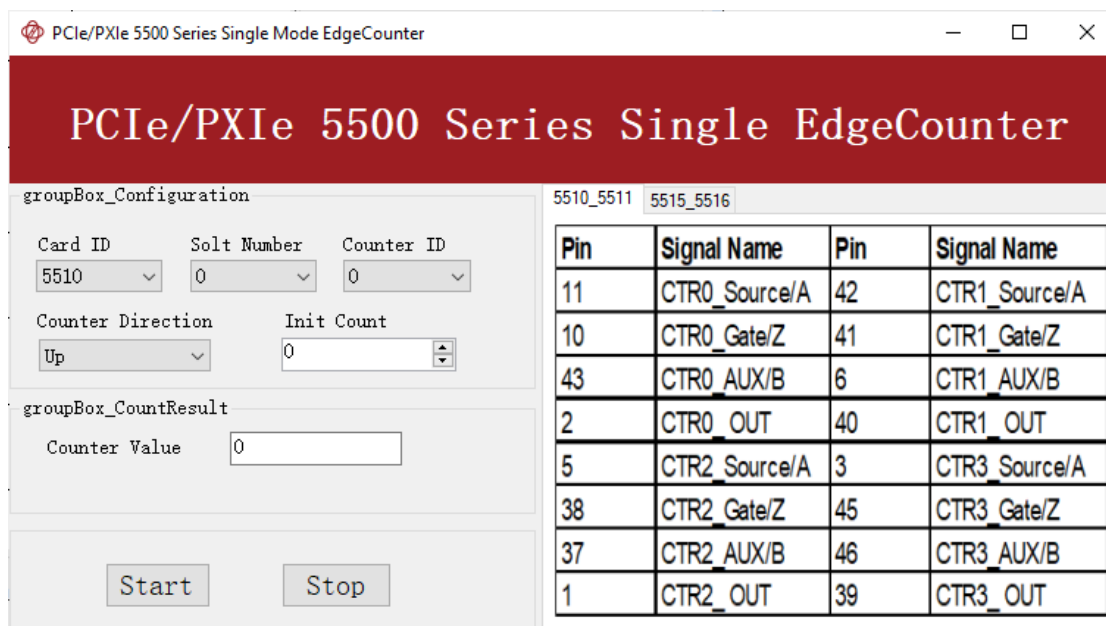


Figure 62 EdgeCounting For Single Mode

- Counter Direction is set by **Counter Direction**.

- The table in the sample program is a connection diagram for your convenience.
- The *rising edge counter* works when **Start** is clicked.
- The result is shown by **Counter Value**. In this example the **Counter Value** increases by 1 every second for a 1Hz sinewave.

Finite/Continuous Mode

- Change the squarewave frequency to 50 Hz.
- Open **Counter Input-->Winform CI Finite/Continuous EdgeCounting**, set the following numbers as shown:

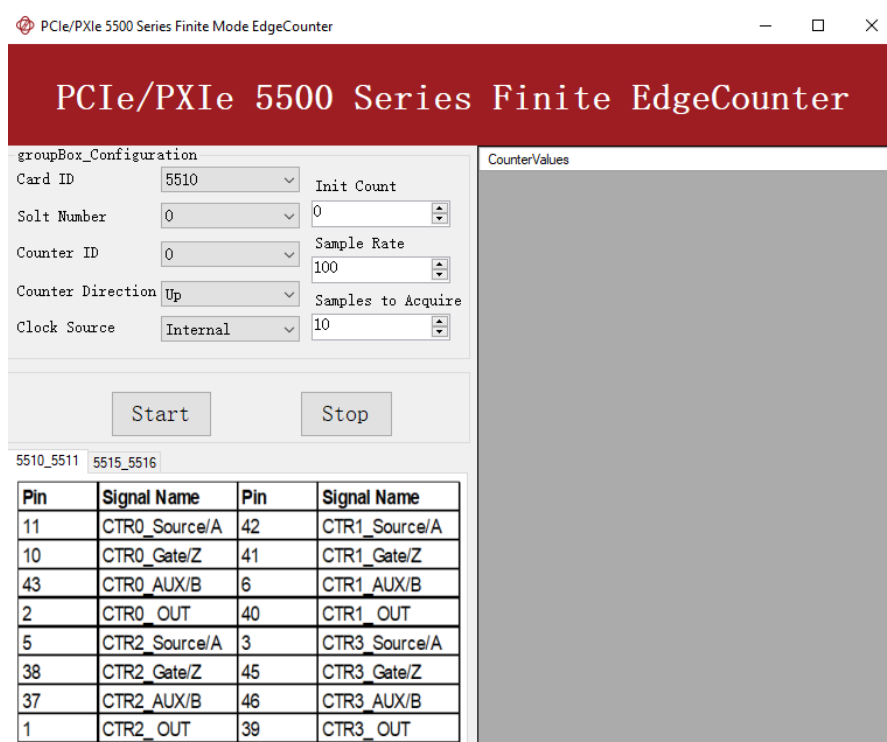


Figure 63 EdgeCounting For Finite Mode

- The table in the sample program is a connection diagram for your convenience.
- Counter Direction is set by **Counter Direction**.
- There are two clock sources in PCIe/PXle-5500 *Internal* and *Implicit*: This example uses **Internal** mode set by **Clock Source**.
- Click **Start** to start counting by rising edge. The result is shown below:

CounterValues
0
1
1
2
2
3
3
4
4
5

Figure 64 Counter Values For Internal Clock

- The numbers are stored in a buffer **CounterValues**.
- Change the **Clock Source** to **Implicit**:

CounterValues
1
2
3
4
5
6
7
8
9
10

Figure 65 Counter Values For Implicit Clock

- The numbers are stored in a buffer **CounterValues**.
- The counter values are different as before because of the change from **Clock Source**.

4.8.2 Pulse Measurement

The counter measures the high-level and low-level duration of signal.

Timing

1) Single Mode

The count value of the duration of the high-level or low-level is written to the register on each rising or falling edge of the pulse to measure, as shown in Figure 66.

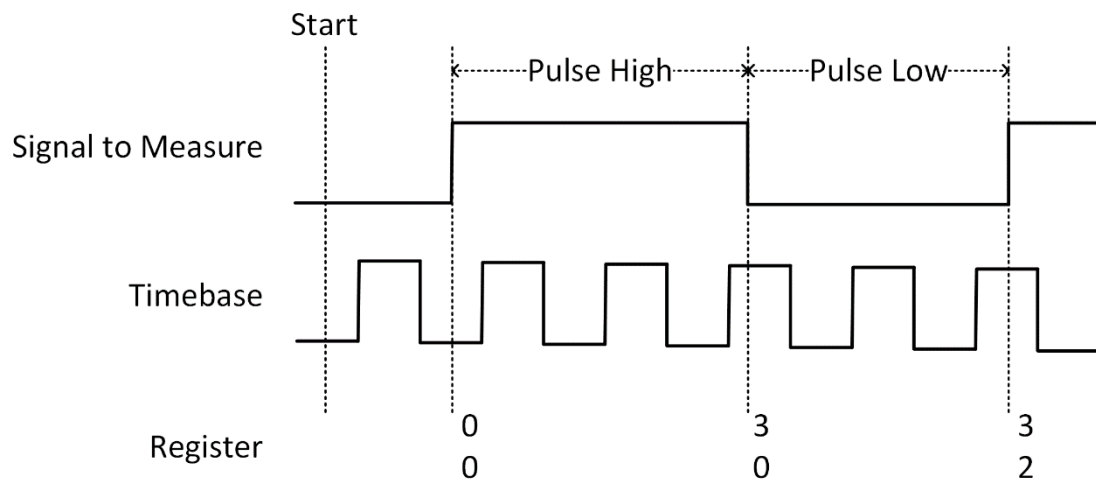


Figure 66 Pulse Measurement in Single Mode

2) Finite/Continuous Mode with Internal Sample Clock

The count value of the duration of the high or low level is stored into the buffer on each rising or falling edge of the sample clock, as shown in Figure 67.

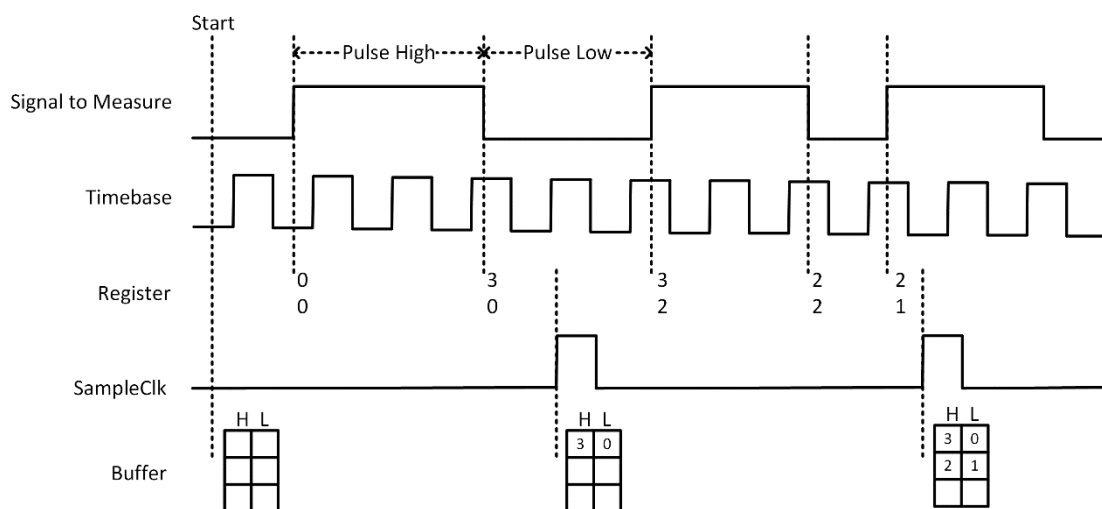


Figure 67 Pulse Measurement with Internal SampleClk

3) Finite/Continuous Mode with Implicit Sample Clock

The count value of the duration of the high-level or low-level is stored into the buffer on each rising or falling edge of the pulse to measure, as shown in Figure 68.

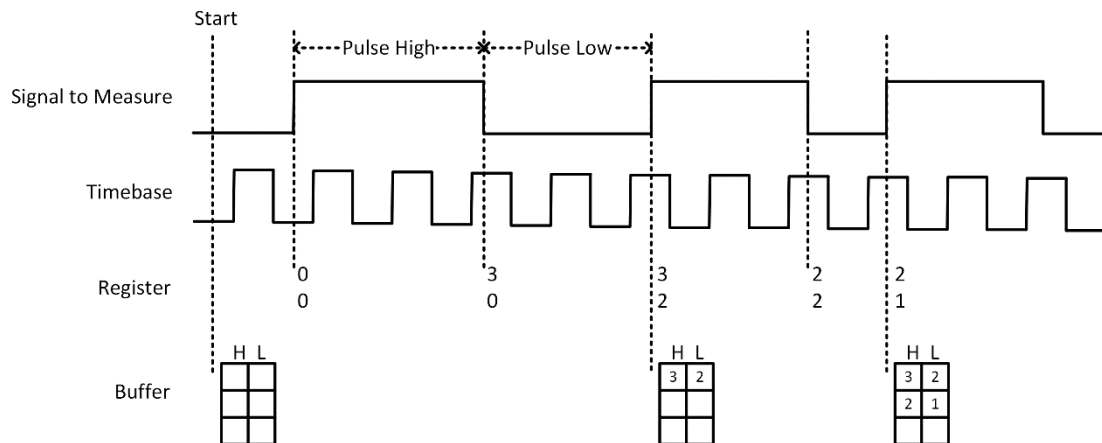


Figure 68 Pulse Measurement with Implicit SampleClk

Learn by Examples 4.8.2

- Connect the signal source's positive terminal to PCIe-5510 counter0's pulse measure source (CTR0_Gate/Z, Pin#10), negative terminal to the ground (DGND, Pin#44) as shown in Figure 2 and Figure 3. (CTR0_Gate/Z, DGND) consists of a pulse measure counter input and they share the same ground.
- Set a squarewave signal (f=1Hz, Duty Cycle=50%, Vpp=5V).

Single Mode

- Open **Counter Input-->Winform CI Single PulseMeasure**, set the following numbers as shown:

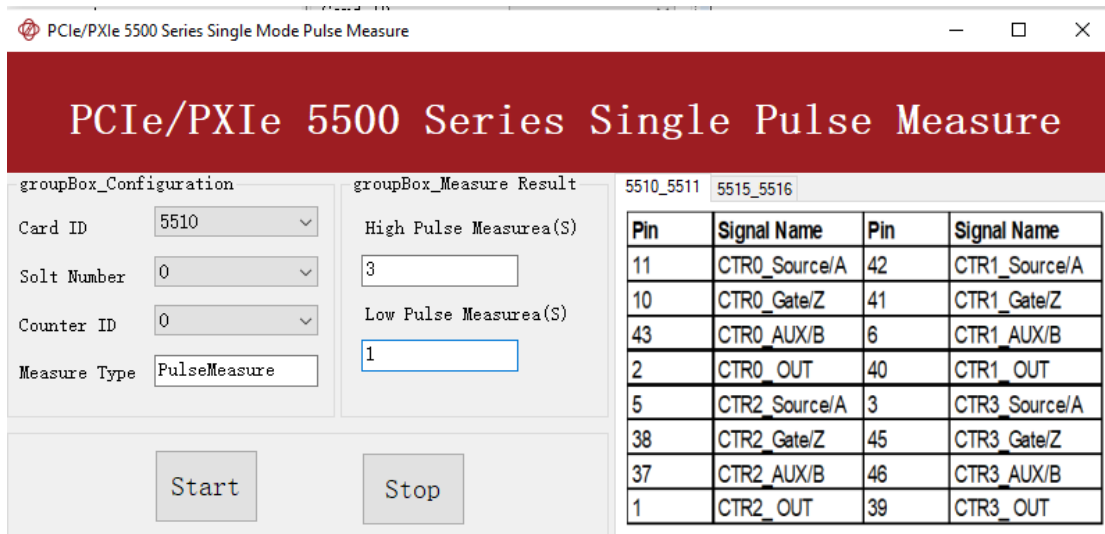


Figure 69 Pulse Measure For Single Mode

- The table in the sample program is a connection diagram for your convenience.
- Click **Start** to start measuring the pulses. The result is shown by **High Pulse Measure(S)** and **Low Pulse Measure(S)**:

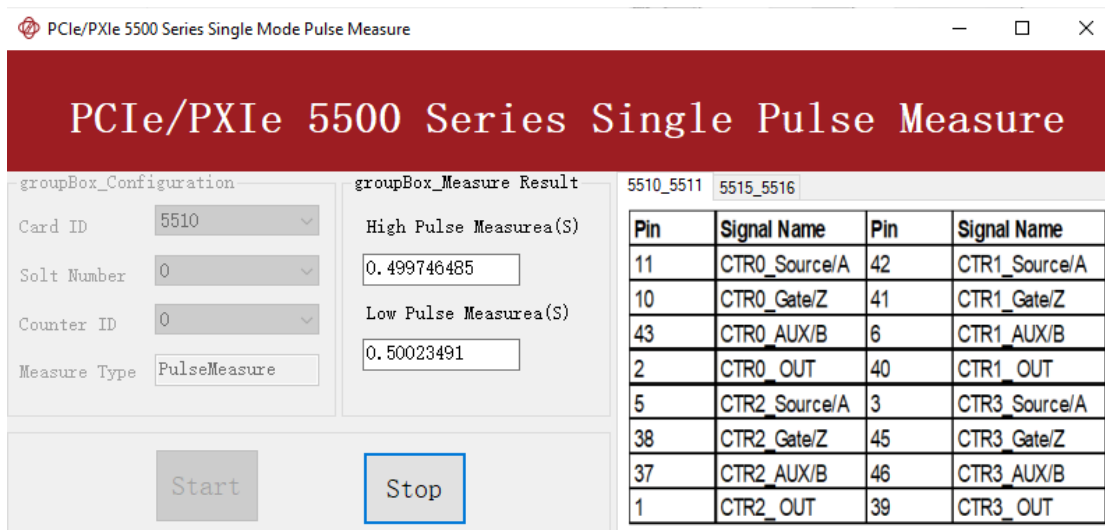


Figure 70 Pulse Measure Value For Single Mode

- The numbers show the duration of **High/Low Pulse** in one signal period and match the duty cycle set before.

Finite/Continuous Mode

- Change the frequency of Squarewave to 50 Hz.

- Open **Counter Input-->Winform CI Finite/Continuous PulseMeasure**, set the following numbers as shown:

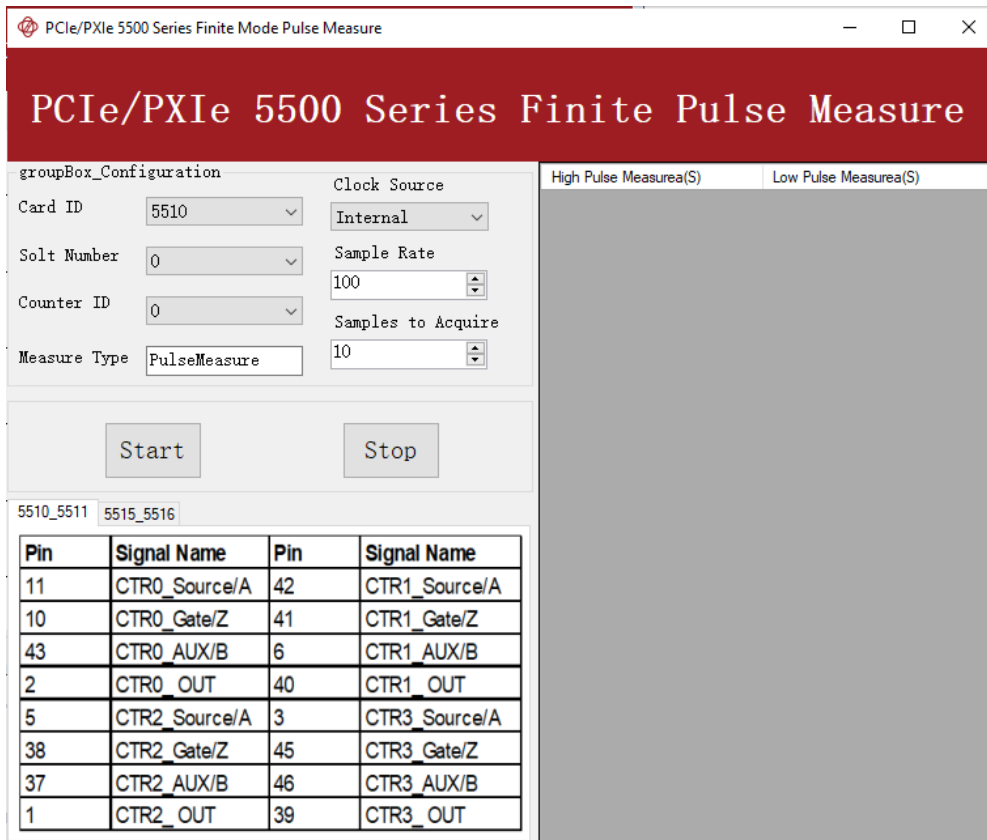


Figure 71 Pulse Measure For Finite Mode

- The table in the sample program is a connection diagram for your convenience.
- Click Start to begin the finite/continuous pulse measurement. The result is shown below:

High Pulse Measurea(S)	Low Pulse Measurea(S)
0	0
0.01000478	0
0.01000478	0.00999485
0.01000478	0.00999485
0.01000478	0.00999485
0.01000477	0.00999485
0.01000477	0.00999486
0.01000477	0.00999486
0.01000477	0.00999486
0.01000477	0.00999486
0.01000477	0.00999486

Figure 72 Pulse Measure Values For Finite Mode

-
- The numbers show the duration of **High/Low Pulse** in one signal period and match the duty cycle set before.
 - Please refer to **Learn by Examples 4.8.1 Finite/Continuous Mode** about the difference between Internal and Implicit.

4.8.3 Frequency Measurement

The counter measures the frequency of signal to measure.

Timing

1) Single Mode

Frequency Measurement without sample clock is actually using Pulse Width Measurement internally, refer to chapter 4.8.2 for more information.

Every time the user reads the data, driver will automatically calculate the frequency (f_x) according to the HighTick ($tick_h$), LowTick ($tick_l$) values and known frequency of the timebase (f_{base}) according to the formula 1 and return the result to the user.

$$f_x = f_{base} \times \frac{1}{tick_h + tick_l}$$

To configure the counter to work in this mode, set JY5500CITask.Mode to CIMode.Single.

2) Finite/Continuous Mode with Internal Sample Clock (*Averaging*)

Between every two rising edges of the sample clock, the counter counts the numbers of full periods ($T1$) of the signal to measure, and the number of rising edges of timebase ($T2$) during those full periods. These two values are stored into the buffer on each rising edge of the sample clock, as shown in Figure 73.

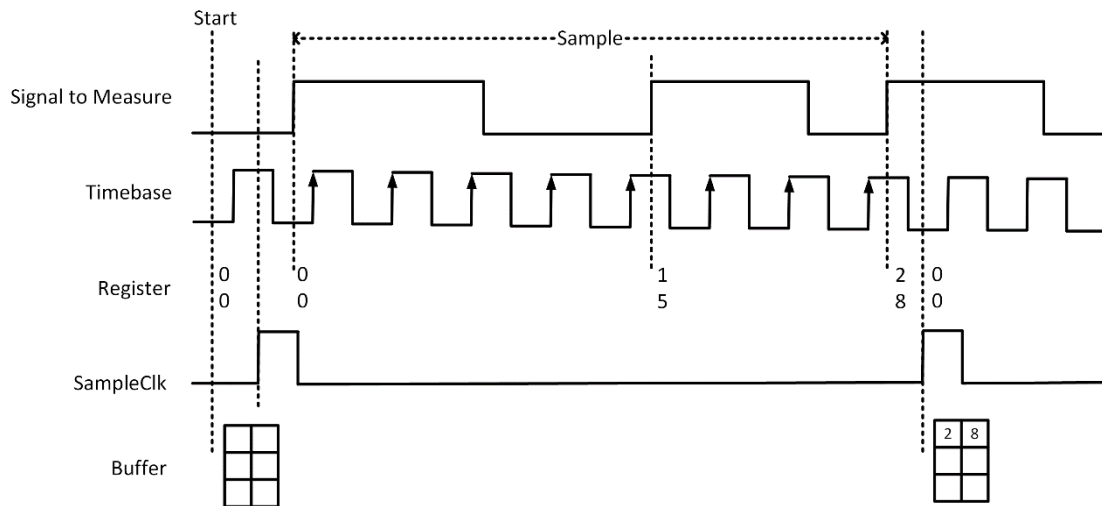


Figure 73 Frequency Measurement with Internal Sample Clock

Every time the user reads the data, driver will automatically calculate the frequency (f_x) according to the buffered values and known frequency of the timebase (f_{base}) by using following formula and return the result to user.

$$f_x = f_{base} \times \frac{T1}{T2}$$

3) Finite/Continuous Mode with Implicit Sample Clock

Frequency Measurement with implicit sample clock is actually using Pulse Measurement internally. refer to chapter 4.8.2 for more information.

Every time the user reads the data, driver will automatically calculate the frequency (f_x) according to the HighTick (T_h) and LowTick (T_l) values according to the formula 1 and return the result to the user.

$$f_x = \frac{1}{T_h + T_l}$$

Learn by Examples 4.8.3

- Connect the signal source's positive terminal to PCIe-5510 counter0's frequency measure source (CTR0_Gate/Z, Pin#10), negative terminal to the ground (DGND, Pin#44) as shown in Figure 2 and Figure 3. (CTR0_Gate/Z, DGND) consists of a frequency measure counter input and they share the same ground.

- Set a squarewave signal (f=50Hz, Duty Cycle=50%, Vpp=5V).

Single Mode

- Open Counter Input-->Winform CI Single Frequency Measure and click Start. The result is shown below by Frequency Measure (Hz):

Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
2	CTR0_OUT	40	CTR1_OUT
5	CTR2_Source/A	3	CTR3_Source/A
38	CTR2_Gate/Z	45	CTR3_Gate/Z
37	CTR2_AUX/B	46	CTR3_AUX/B
1	CTR2_OUT	39	CTR3_OUT

Figure 74 Frequency Measure For Single Mode

- The table in the sample program is a connection diagram for your convenience.
- The result matches the frequency set before.

Finite/Continuous Mode

- Open Counter Input-->Winform CI Finite/Continuous Frequency Measure.

Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
2	CTR0_OUT	40	CTR1_OUT
5	CTR2_Source/A	3	CTR3_Source/A
38	CTR2_Gate/Z	45	CTR3_Gate/Z
37	CTR2_AUX/B	46	CTR3_AUX/B
1	CTR2_OUT	39	CTR3_OUT

Figure 75 Frequency Measure For Continuous Mode

- The table in the sample program is a connection diagram for your convenience.
- Internal and Implicit Sample Clocks are set by **Clock Source** as before. (Please refer to Finite/Continuous Mode for more information.)
- Click Start and it will show the frequency 50 as set in the signal resource.

FrequencyMeasurea(Hz)
50.0009281422286
50.0009281422286
50.0009343924615
50.0009312673449
50.0009343924615
50.0009281422286
50.0009343924615
50.0009281422286
50.0009281422286
50.0009343924615

Figure 76 Frequency Measure Values

4.8.4 Period Measurement

The counter measures the period of signal to measure. Period Measurements is using Frequency Measurement internally and returns the inverse result of Frequency Measurement. Refer to chapter 4.8.3 for more information.

Learn by Examples 4.8.4

- Connect the signal source's positive terminal to PCIe-5510 counter0's period measure source (CTR0_Gate/Z, Pin#10), negative terminal to the ground (DGND, Pin#44) as shown in Figure 2 and Figure 3. (CTR0_Gate/Z, DGND) consists of a period measure counter input and share the same ground.
- Set a squarewave signal (f=200Hz, Duty Cycle=50%, Vpp=5V).

Single Mode

- Open **Counter Input-->Winform CI Single Period Measure** and click **Start**. The result is shown below by **Period Measure(S)**:

PCIe/PXIE 5500 Series Single Peroid Measure

Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
2	CTR0_OUT	40	CTR1_OUT
5	CTR2_Source/A	3	CTR3_Source/A
38	CTR2_Gate/Z	45	CTR3_Gate/Z
37	CTR2_AUX/B	46	CTR3_AUX/B
1	CTR2_OUT	39	CTR3_OUT

Figure 77 Peroid Measure For Single Mode

- The table in the sample program is a connection diagram for your convenience.
- The result of **Period Measure(S)** shows the correspond to the frequency set before.

Finite/Continuous Mode

- Open **Counter Input-->Winform CI Finite/Continuous Period Measure** and click **Start**. The result is shown below by **PeriodMeasure (S)**.

PCIe/PXIE 5500 Series Continuous Period Measure

groupBox_Configuration

Card ID: Sample Rate:

Solt Number: Counter ID:

Measure Type: Samples to Acquire:

Clock Source:

PeroidMeasurea(S)

0.00499990578947368
0.00499990578947368
0.00499990578947368
0.00499990578947368
0.00499990578947368
0.00499990578947368
0.00499990578947368
0.00499990578947368
0.00499990578947368
0.00499990578947368
0.00499990578947368
0.00499990578947368

5510_5511 5515_5516

Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
2	CTR0_OUT	40	CTR1_OUT
5	CTR2_Source/A	3	CTR3_Source/A
38	CTR2_Gate/Z	45	CTR3_Gate/Z
37	CTR2_AUX/B	46	CTR3_AUX/B
1	CTR2_OUT	39	CTR3_OUT

Figure 78 Peroid Measure For Continuous Mode

- The table in the sample program is a connection diagram for your convenience.
- The result of **Period Measure(S)** shows the correspond to the frequency set before.

4.8.5 Two-Edge Separation

The counter measures the separation between the rising edges of two signals.

Timing

1) Single Mode

The number of rising edges of timebase between the rising edge of the first signal and the rising edge of the second signal is written to the register on each rising edge of the second signal.

The number of rising edges of timebase between previous rising edge of the second signal and current rising edge of the first signal is written to the register on each rising edge of the first signal as shown in Figure 79.

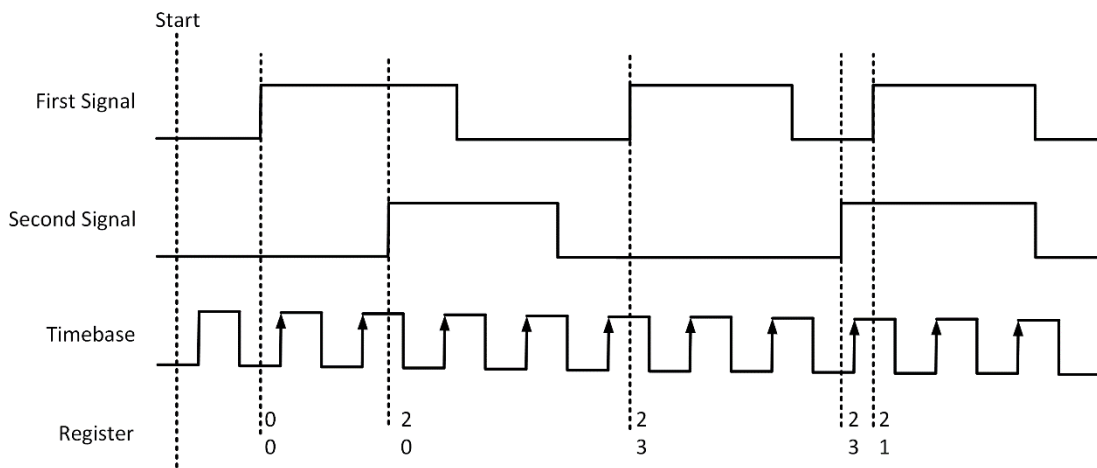


Figure 79 Two-Edge Separation in Single Mode

2) Finite/Continuous Mode with Internal Sample Clock:

The count values of rising edges of timebase between first signal and second signal are stored into buffer on each rising edge of the sample clock, as shown in Figure 80.

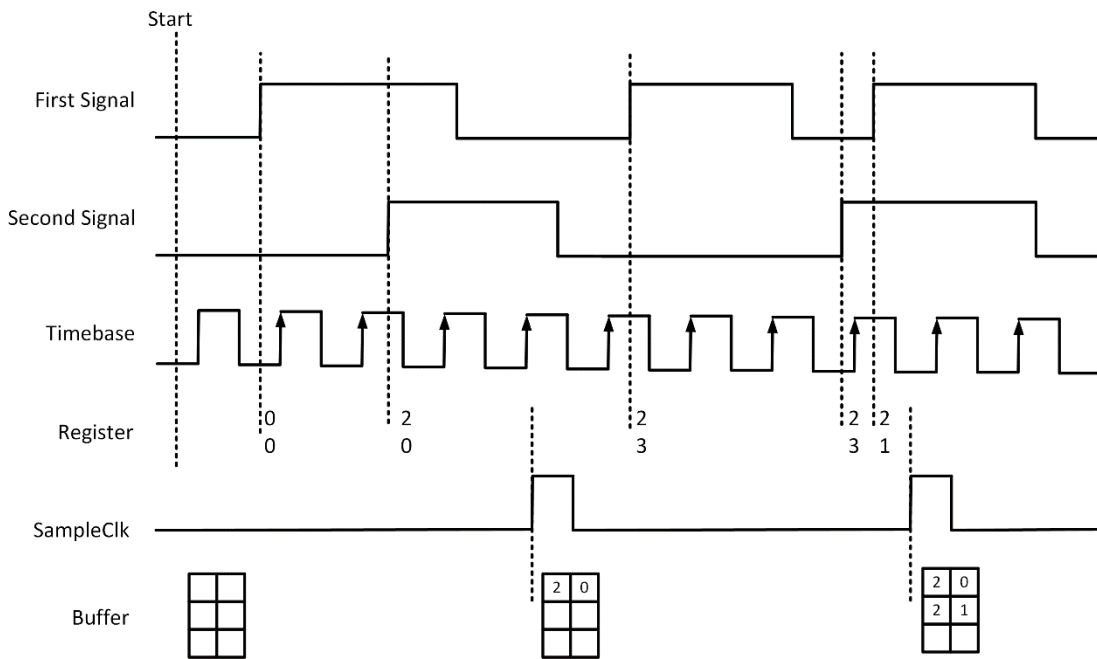


Figure 80 Two-Edge Separation with Internal Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

The count values of rising edges of timebase between first signal and second signal are stored into buffer on each rising edge of the first signal, as shown in Figure 81.

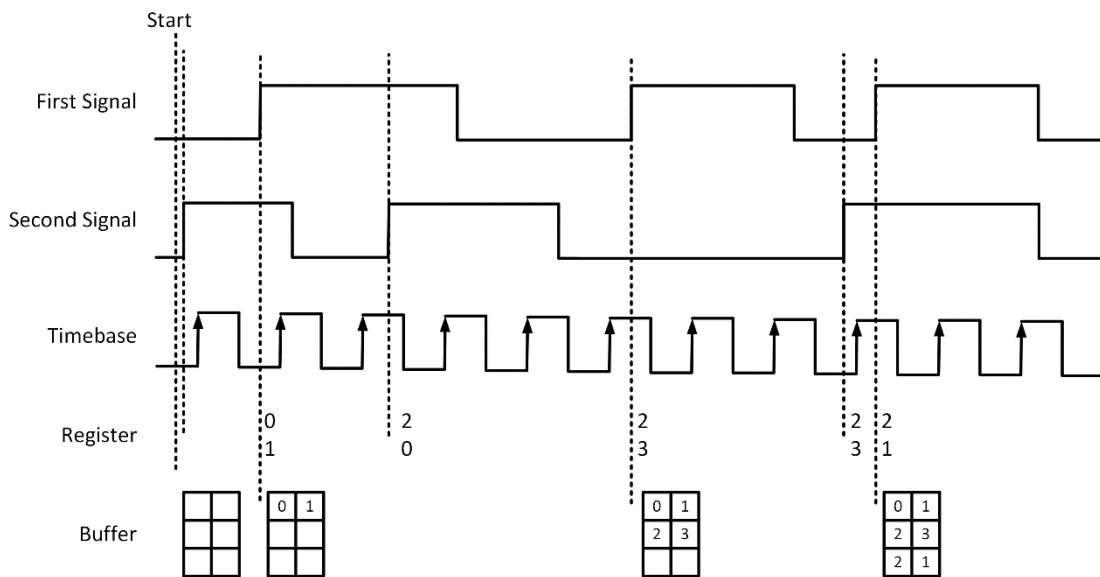


Figure 81 Two-Edge Separation with Implicit Sample Clock

Learn by Examples 4.8.5

- Connect the signal source's two positive terminals to PCIe-5510 first signal input (squarewave, CTR0_Gate/Z, Pin #10) and second signal input (squarewave, CTR0_AUX/B, Pin#43), two negative terminals to the ground (DGND, Pin#44) and (D_GND, Pin#9) as shown in Figure 2 and Figure 3.
- Set a squarewave signal (f=1Hz, Phase=0°) and a squarewave signal (f=1Hz, Phase=135°).

Single Mode

- Open **Counter Input-->Winform CI Single TwoEdgeSeparation Measure** and click **Start**. The result is shown below by **First to Second(S)** and **Second to First(S)**, which represent the time difference between the rising edges of the two signals:

PCIe/PXIE 5500 Series Single TwoEdgeSeparation Measure

groupBox_Configuration

Card ID:

Solt Number:

Counter ID:

Measure Type:

groupBox_MeasureResult

First to Second(S):

Second to First(S):

5510_5511 | 5515_5516

Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
2	CTR0_OUT	40	CTR1_OUT
5	CTR2_Source/A	3	CTR3_Source/A
38	CTR2_Gate/Z	45	CTR3_Gate/Z
37	CTR2_AUX/B	46	CTR3_AUX/B
1	CTR2_OUT	39	CTR3_OUT

Figure 82 Two-EdgeSeparation Measure For Single Mode

- The table in the sample program is a connection diagram for your convenience.
- Due to the phase-difference between First Signal and Second Signal, **First to Second** and **Second to First** are different and summarize as 1.

Finite/Continuous Mode

- Open **Counter Input-->Winform CI Finite/Continuous TwoEdge Separation Measure** and click **Start**. The result is shown below by **First to Second(S)** and **Second to First(S)**, which represent the time difference between the rising edges of the two signals:

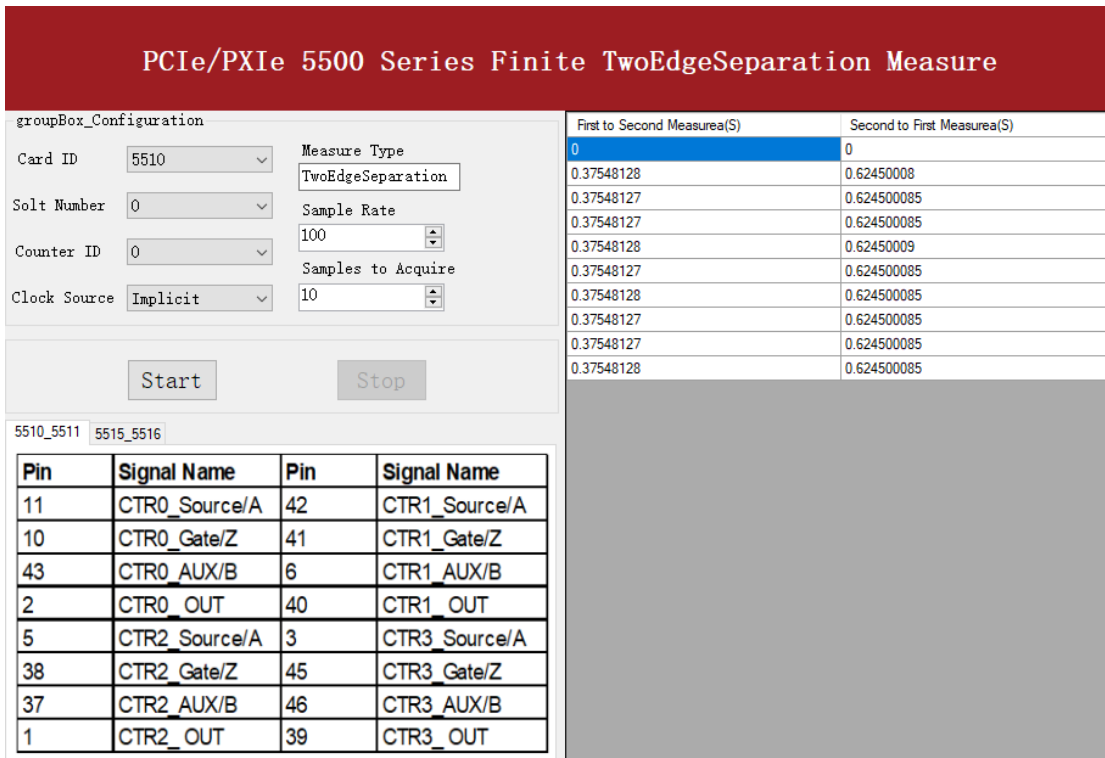


Figure 83 Two-EdgeSeparation Measure For Finite Mode

- The result in this picture is similar to the result in **Single Mode** before.
- The table in the sample program is a connection diagram for your convenience.

4.8.6 Quadrature Encoder

The quadrature encoder includes three encoding types: x1, x2, and x4.

Encoding Type

1) x1 Encoding

When A leads B, the count increase occurs on the rising edge of A; when B leads A, the count decrease occurs on the falling edge of A as shown in Figure 84.

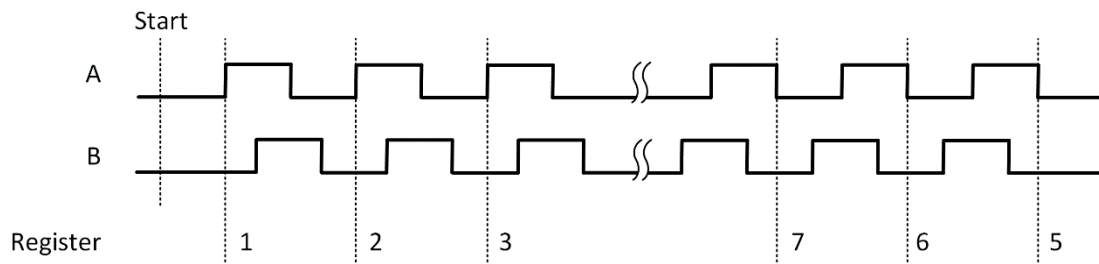


Figure 84 Quadrature Encoder x1 Mode

2) x2 Encoding

When A leads B, the count increase occurs on the rising edge and the falling edge of A; when B leads A, the count reduction occurs on the rising edge and falling edge of A as shown in Figure 85.

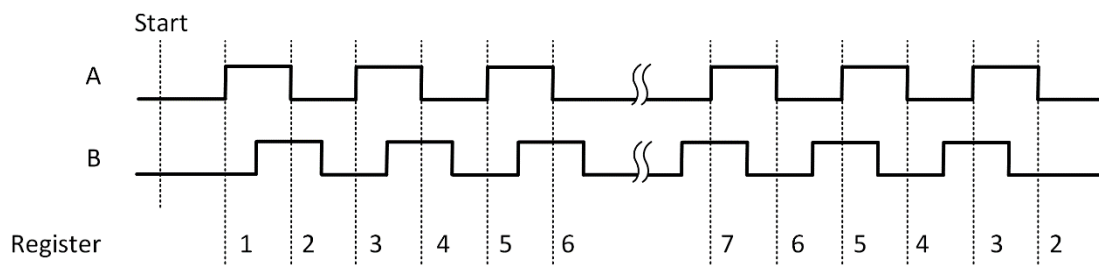


Figure 85 Quadrature Encoder x2 Mode

3) x4 Encoding

When A leads B, the increase of count occurs on the rising and falling edges of A and B. When B leads A, the decrease in count occurs on the rising and falling edges of A and B. As shown in Figure 86.

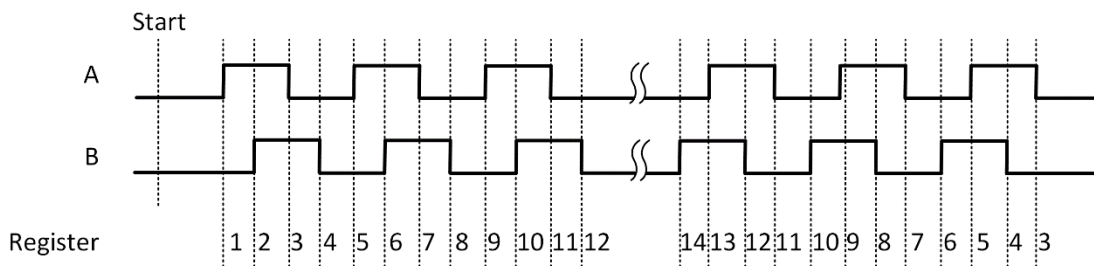


Figure 86 Quadrature Encoder x4 mode

Channel Z Behavior

The phase is reloaded when channel Z is high, A and B are low.

Timing

Take Encoding x1 mode as an example.

1) Single Mode

The count value is written to the register on each rising edge of the signal A, as shown in Figure 58.

To configure the counter to work in this mode, set JY5500CITask. Mode to CIMode.Single.

2) Finite/Continuous Mode with Internal Sample Clock

The count value is stored into the buffer on each rising edge of the sample clock, as shown in Figure 87.

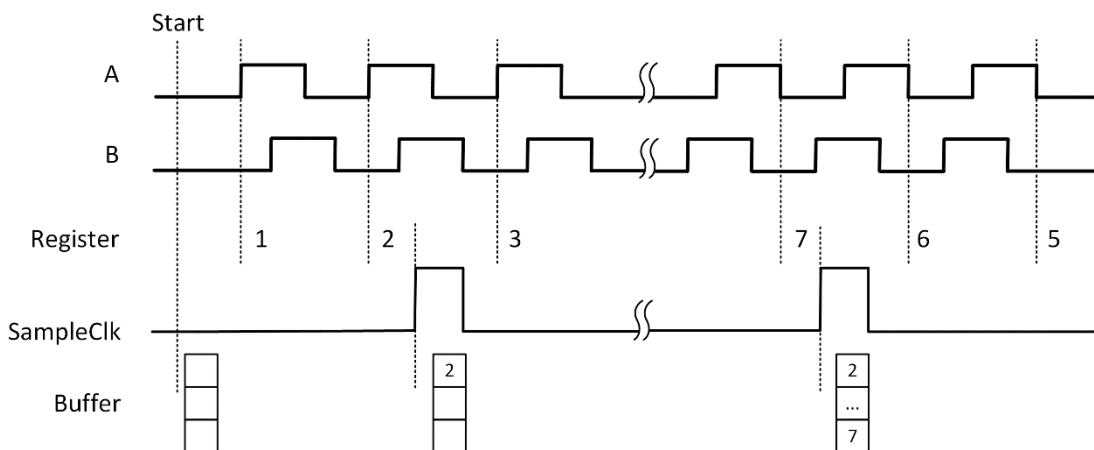


Figure 87 Quadrature Encoder x1 with Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

The count value is stored into the buffer every time it changes, as shown in Figure 88.

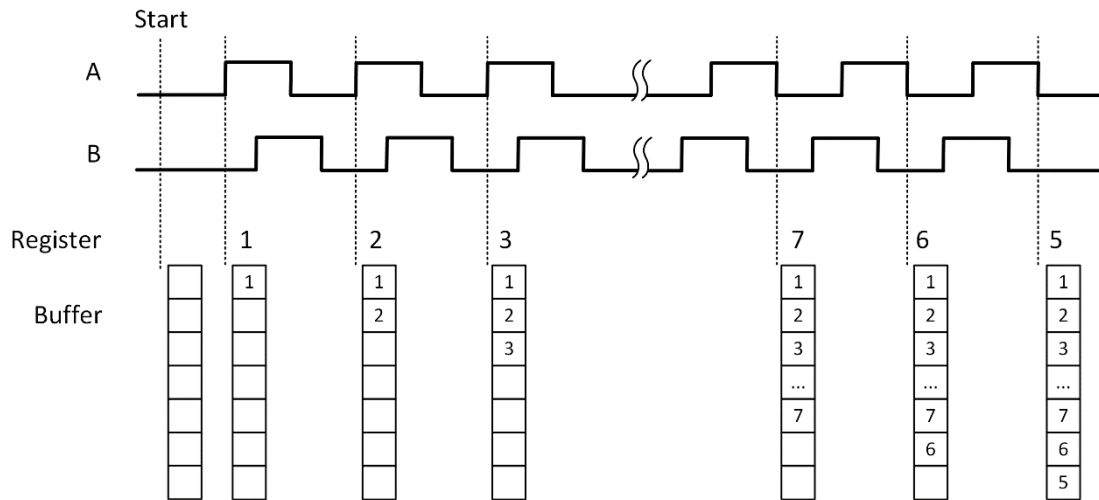


Figure 88 Quadrature Encoder x4 with Implicit Sample Clock

Learn by Examples 4.8.6

- Connect the signal source's two positive terminals to PCIe-5510 first signal input (sinewave, CTR0_Source/A, Pin #11) and second signal input (squarewave , CTR0_AUX/B, Pin#43), two negative terminals to the ground (DGND, Pin#44) and (D_GND, Pin#9) as shown in Figure 2 and Figure 3. (CTR0_Source/A, DGND) consists of the first signal to be measured; (CTR0_AUX/B, D_GND) consists of the second signal to be measured.
- Set a squarewave signal (f=10Hz, Phase=90°) and a squarewave signal (f=10Hz, Phase=0°).

Single Mode

- Open **Counter Input--> Winform CI Single QuadEncoder** and click **Start**. The result is shown below by **CounterValue** according to the counting rules explained in 4.8.6:

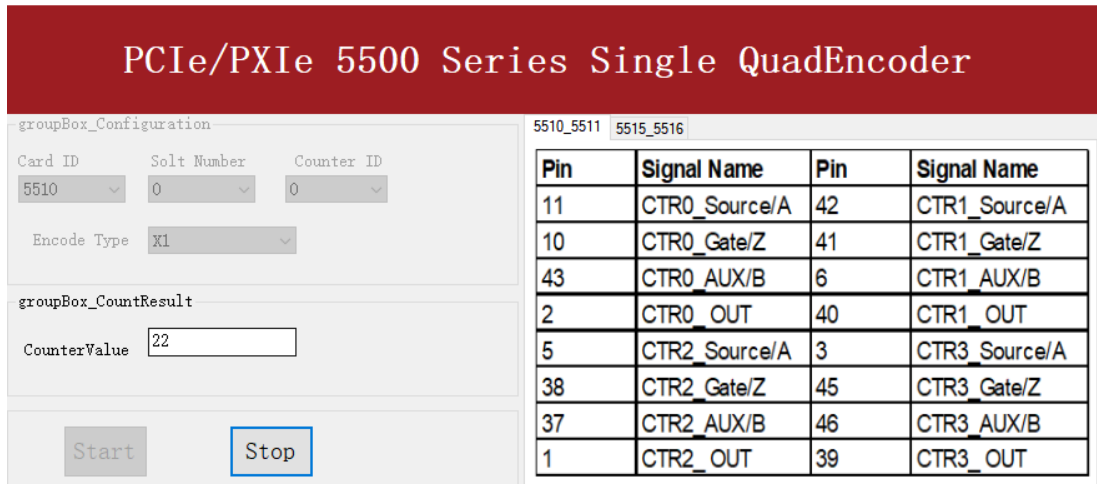


Figure 89 QuadEncoder For Single Mode

- The table in the sample program is a connection diagram for your convenience.
- *Encoding Type* is set by **Encode Type (x1, x2, x4)**.
- When the *encode type* is changed from x1 to x2 and x4, you can see the rising speed of **CounterValue** is twice and four times than x1Mode.

Continuous Mode

- Open **Counter Input--> Winform CI Continuous QuadEncoder** and click **Start**. The result is shown below by **CounterValues**.

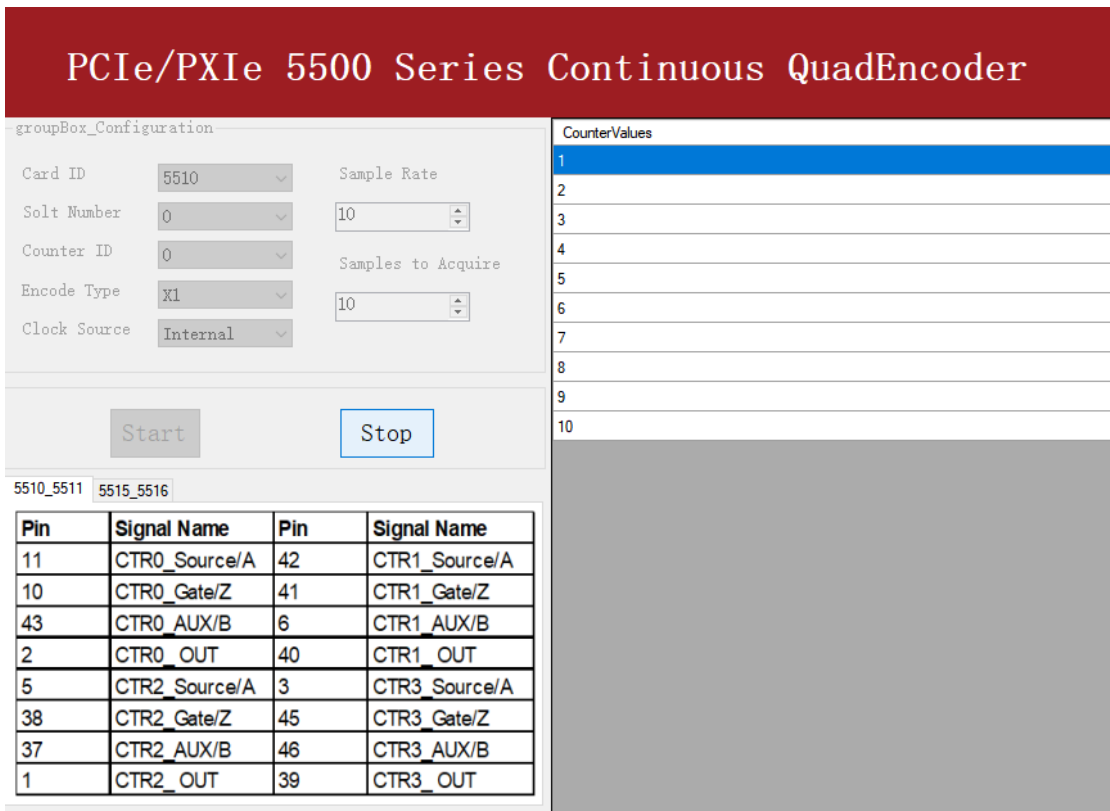


Figure 90 QuadEncoder For Continuous Mode

- The table in the sample program is a connection diagram for your convenience.
- *Encoding Type* is set by **Encode Type (x1, x2, x4)**.
- When the *encode type* is changed from x1 to x2 and x4, you can see the rising speed of **CounterValue** is twice and four times than x1Mode.

4.8.7 Two-Pulse Encoder

The count value increases on the rising edge of A and decreases on the rising edge of B.

Timing

1) Single Mode

The count value is written to the register on each rising edge of the signal A, and signal B, as shown in Figure 91.

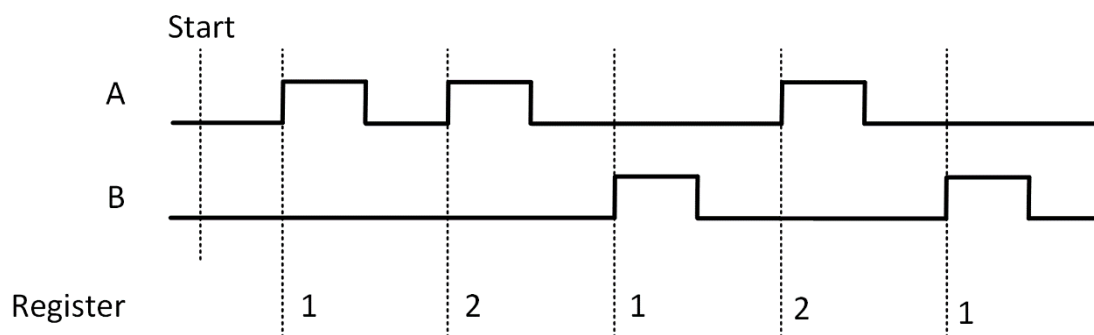


Figure 91 Two-Pulse Encoder in Single Mode

2) Finite/Continuous Mode with Internal Sample Clock

The count value is stored into the buffer on each rising edge of the sample clock, as shown in Figure 92.

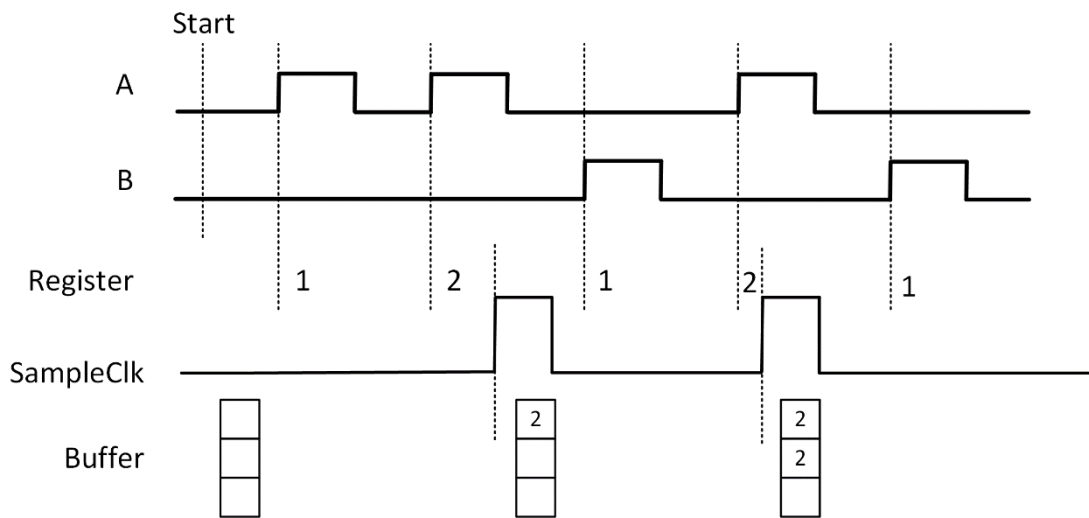


Figure 92 Two-Pulse Encoder with Internal Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

The count value is stored into the buffer every time it changed, as shown in Figure 93.

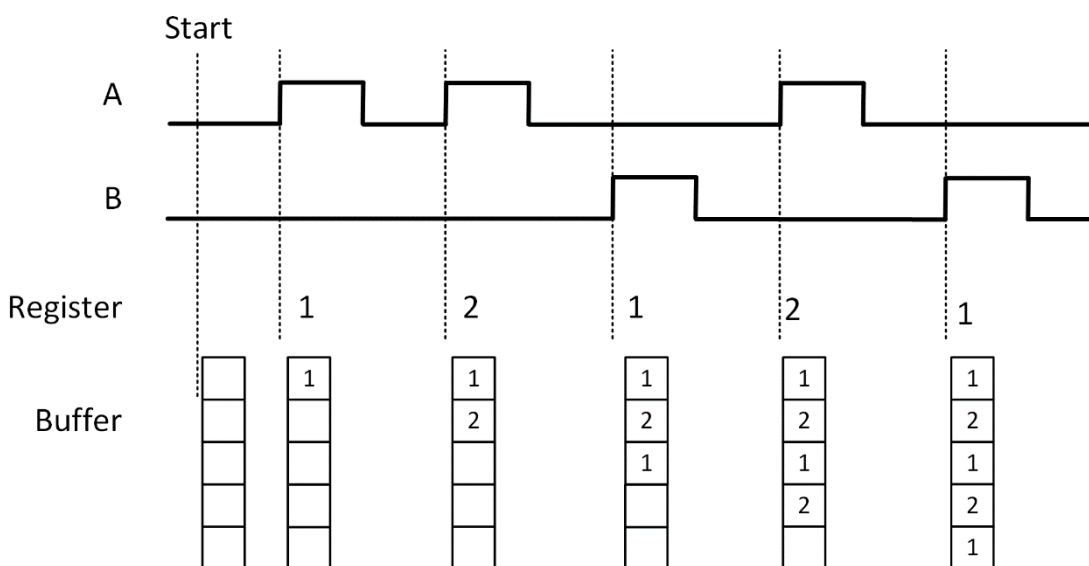


Figure 93 Two-Pulse Encoder with Implicit Sample Clock

Learn by Examples 4.8.7

- Connect the signal source's two positive terminals to PCIe-5510 first signal input (squarewave, CTR0_Source/A, Pin #11) and second signal input (squarewave, CTR0_AUX/B, Pin#43), two negative terminals to the ground (DGND, Pin#44) and (D_GND, Pin#9) as shown in Figure 2 and Figure 3.

- Set a squarewave signal (f=40Hz) and a squarewave signal (f=40Hz).

Single Mode

- Open **Counter Input-->Winform CI Single Two PulseEncoder** and set the numbers as shown.

Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
2	CTR0_OUT	40	CTR1_OUT
5	CTR2_Source/A	3	CTR3_Source/A
38	CTR2_Gate/Z	45	CTR3_Gate/Z
37	CTR2_AUX/B	46	CTR3_AUX/B
1	CTR2_OUT	39	CTR3_OUT

Figure 94 Two-PulseEncoder For Single Mode

- The table in the sample program is a connection diagram for your convenience.
- Click **Start** to start counting. You can see a continuously rising of the **Counter Value**, which follows the counting rules explained in this chapter.

Finite Mode

- Change the frequency of the second channel squarewave from 50Hz to 120Hz.
- Open **Counter Input-->Winform CI Finite Two PulseEncoder** and set the numbers as shown.

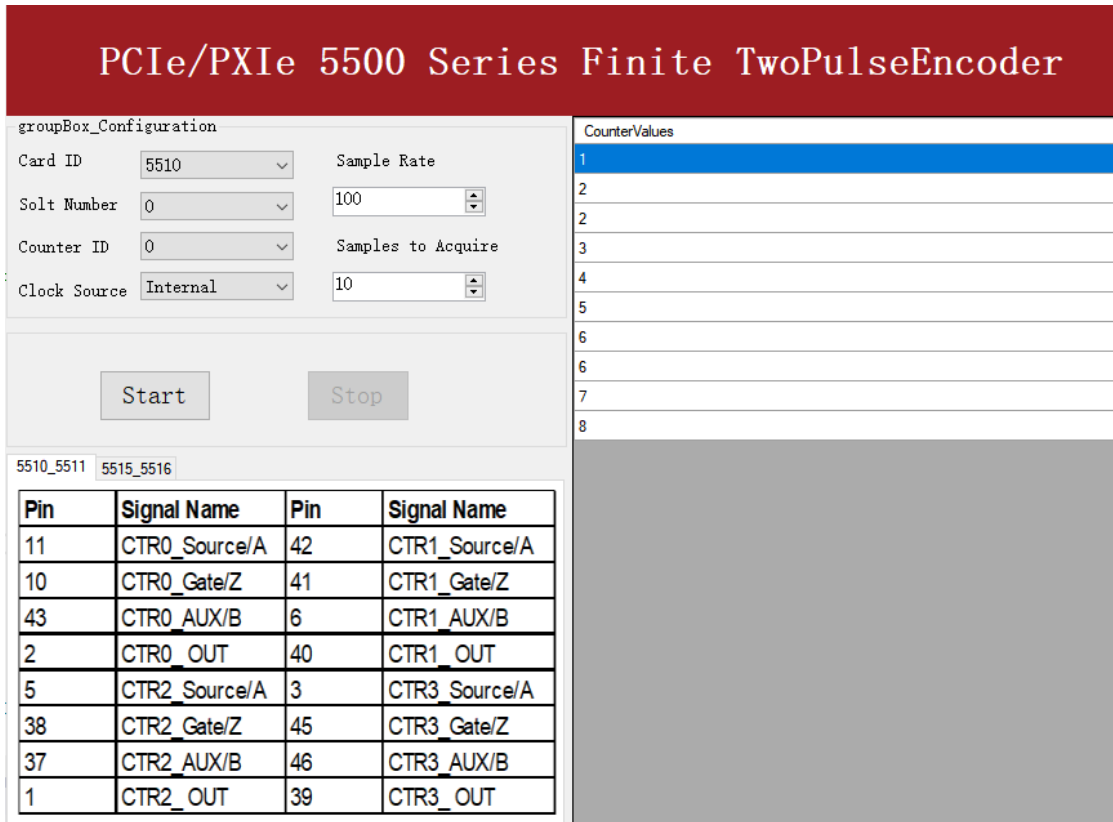


Figure 95 Two-PulseEncoder For Finite Mode

- The table in the sample program is a connection diagram for your convenience.
- Click **Start** and the result is shown above by **Counter Value** in the right list, which follows the counting rules explained in this chapter.

Continuous Mode

- Change the frequency of the second squarewave back to 50 Hz.
- Open **Counter Input-->Winform CI Continuous Two PulseEncoder** and set the numbers as shown.

PCIe/PXIE 5500 Series Continuous TwoPulseEncoder

groupBox_Configuration

Card ID	5510	Sample Rate	400
Solt Number	0	Samples to Acquire	10
Counter ID	0	Clock Source	Internal

5510_5511 5515_5516

Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
2	CTR0_OUT	40	CTR1_OUT
5	CTR2_Source/A	3	CTR3_Source/A
38	CTR2_Gate/Z	45	CTR3_Gate/Z
37	CTR2_AUX/B	46	CTR3_AUX/B
1	CTR2_OUT	39	CTR3_OUT

CounterValues

10
10
11
11
10
10
10
10
10
10
10
10

Figure 96 Two-PulseEncoder For Continuous Mode

- The table in the sample program is a connection diagram for your convenience.
- Click **Start** and you can see a group of rising numbers in **CounterValues**, which follows the counting rules explained in this chapter.

4.9 Counter Output Operations

4.9.1 Single Pulse Output

The PCIe/PXIe-5500 timer/counter can output a single pulse with a specified pulse width. The timing diagram of the pulse output is shown in Figure 97.

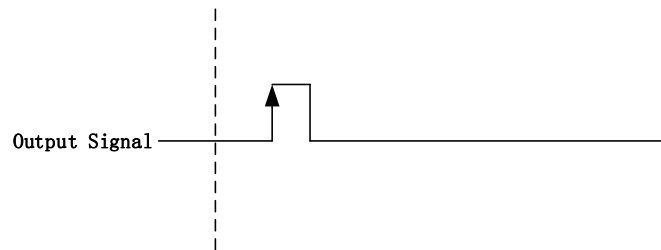


Figure 97 Single Pulse Output

In single pulse output mode, the user could set up the pulse width by configuring the frequency and duty cycle.

If you want to generate a single pulse with 1 ms pulse width, the parameter, frequency should be setup 500Hz and the duty cycle is 50%. Here is the formula for frequency setting:

$$\text{Frequency} = 1 / (1\text{ms} / 0.5) = 500\text{Hz}$$

Learn by Example 4.9.1

- To see the signal that PCIe/PXIe-5500 Counter Output generates, it is recommended to connect PCIe/PXIe-5500 Counter Output (CTR0_OUT, Pin#2) to PCIe/PXIe-5500 AI Ch0 input (AI0+, Pin#68). Please note Counter Output and AI Ch0 input share the same ground so only one connection is needed.
- Open **Counter Output-->Winform CO Single** and click **Start** and set the numbers as follow:

PCIe/PXIE 5500 Series Single Pulse Generation

groupBox_Parameter

Card ID: 5510 Pulse Delay: 0

Slot Number: 0

Counter ID: 0

groupBox_PulseParameter

outputPulse Type: DutyCycleFrequency Frequency: 2.000

Idle State: LowLevel Duty Cycle: 0.500

5510_5511 5515_5516

Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
2	CTR0_OUT	40	CTR1_OUT
5	CTR2_Source/A	3	CTR3_Source/A
38	CTR2_Gate/Z	45	CTR3_Gate/Z
37	CTR2_AUX/B	46	CTR3_AUX/B
1	CTR2_OUT	39	CTR3_OUT

Figure 98 Single Pulse Generation

- The table in the sample program is a connection diagram for your convenience.
- The frequency and duty cycle of the pulse are set by **Frequency** and **Duty Cycle**.
- Please refer Learn by Example 4.2 to configure an analog input to receive the signal from Counter Output.
- Click **Start** to generate a single pulse as shown.

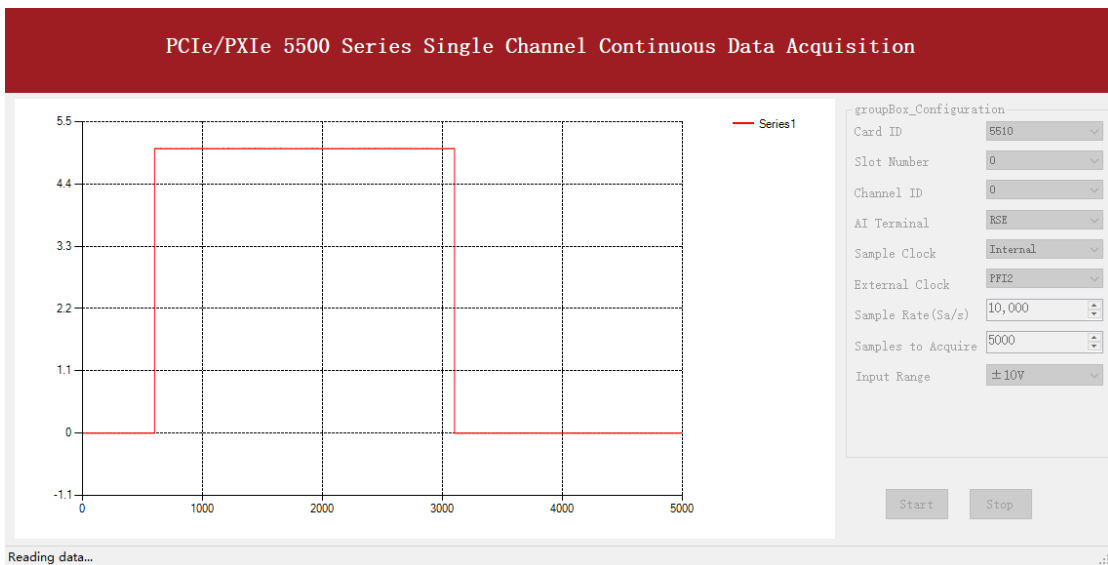


Figure 99 AI Acquisition Single Pulse

4.9.2 Finite Pulse Output

The pulse output timing is as shown in Figure 100.

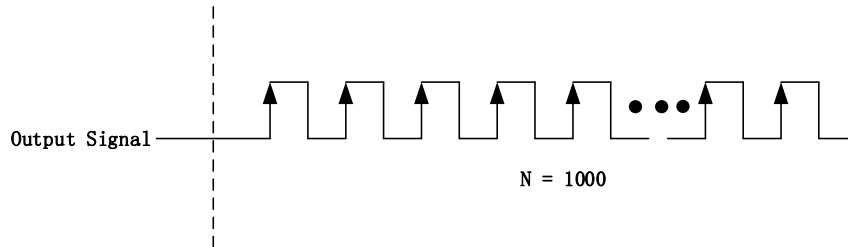


Figure 100 Finite Pulse Output

In finite pulse output mode, the user is required to configure the output frequency, duty cycle and the number of output pulses.

Assuming that the pulse width to be output by the user is 1ms, the frequency calculated according to the duty cycle of 50% is as follows:

$$\text{Set frequency} = 1 / (1\text{ms} / 0.5) = 500\text{Hz}$$

That is to say, when the user sets the frequency as 500Hz and the duty cycle as 0.5, a limited pulse of 1ms pulse width will be obtained.

Learn by Example 4.9.2

- To see the signal that PCIe/PXIe-5500 Counter Output generates, it is recommended to connect PCIe/PXIe-5500 Counter Output (CTR0_OUT, Pin#2) to PCIe/PXIe-5500 AI Ch0 input (AI0+, Pin#68). Please note Counter Output and AI Ch0 input share the same ground so only one connection is needed.
- Open **Counter Output-->Winform CO Finite** and click **Start** and set the numbers as follow:

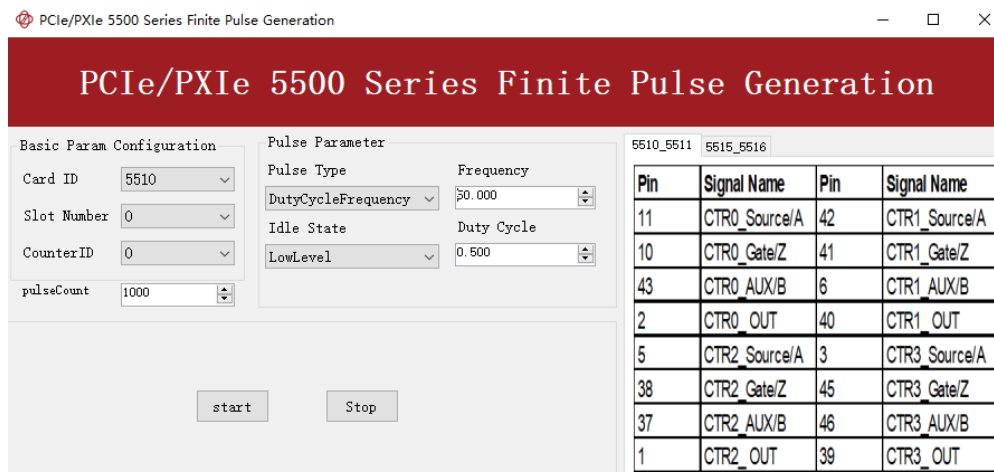


Figure 101 Finite Pulses Generation

- The table in the sample program is a connection diagram for your convenience.
- The frequency and duty cycle of the pulse are set by **Frequency** and **Duty Cycle**.
- Please refer Learn by Example 4.2 to configure an analog input to receive the signal from Counter Output.
- Click **Start** to generate the pulse shown below.

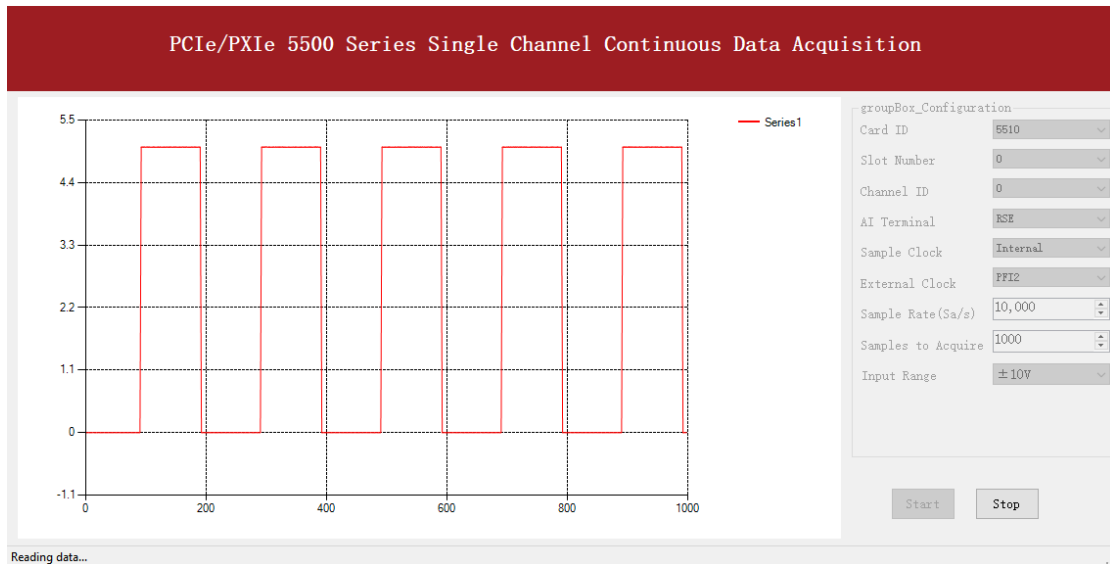


Figure 102 AI Acquisition Finite Pulse

- According to the picture, the *duty cycle* is 0.5 as set before.

4.9.3 Continuous Pulse Output

The pulse output timing is shown in Figure 103 below.

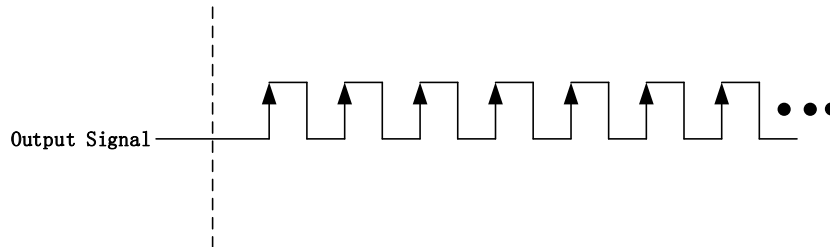


Figure 103 Continuous Pulse Output

In continuous output mode, you need to configure the output frequency and duty cycle. After starting the output, the pulse signal with fixed frequency and duty cycle will be output continuously.

Learn by Example 4.9.3

- To see the signal that PCIe/PXIe-5500 Counter Output generates, it is recommended to connect PCIe/PXIe-5500 Counter Output (CTR0_OUT, Pin#2) to PCIe/PXIe-5500 AI Ch0 input (AI0+, Pin#68). Please note Counter Output and AI Ch0 input share the same ground so only one connection is needed.
- Open **Counter Output-->Winform CO Continuous** and click **Start** and set the numbers as follow:

**PCIe/PXIe 5500 Series Continuous Pulse Generation
(Real-time Modification of Frequency And Duty Cycle)**

groupBox_Configuration

Card ID: 5510

Solt Number: 0

Counter ID: 0

groupBox_PulseParameter

Pulse Type: DutyCycleFrequenc

Frequency: 50.000

Idle State: LowLevel

Duty Cycle: 0.500

start modification of frequency and duty Stop

5510_5511 5515_5516

Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
2	CTR0_OUT	40	CTR1_OUT
5	CTR2_Source/A	3	CTR3_Source/A
38	CTR2_Gate/Z	45	CTR3_Gate/Z
37	CTR2_AUX/B	46	CTR3_AUX/B
1	CTR2_OUT	39	CTR3_OUT

Figure 104 Continuous Pulse Generation

- The table in the sample program is a connection diagram for your convenience.
- The frequency and duty cycle of the pulse are set by **Frequency** and **Duty Cycle**.

■ Change the **Duty Cycle** to 0.7 for instance. The result is shown below.

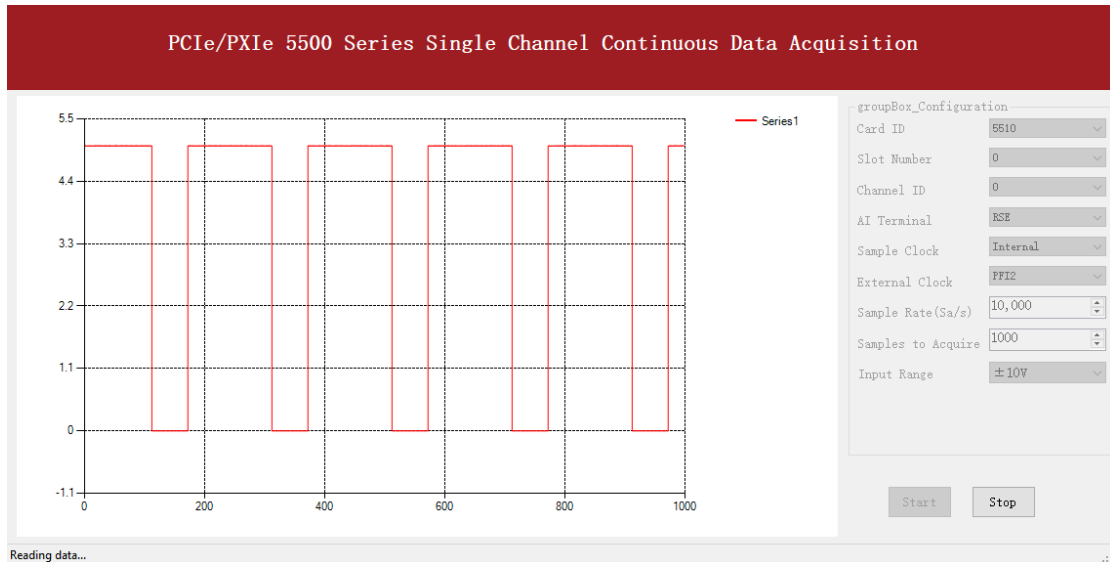


Figure 105 AI Acquisition Continuous Pulse

- According to the picture, the **duty cycle** is 0.7 as set before.

4.10 System Synchronization Interface (SSI) for PCIe Modules

The synchronization between PCIe modules are handled differently from the PXIe synchronization, it is implemented by the system synchronization interface (SSI). SSI is designed as a bidirectional bus and it can synchronize up to four PCIe modules. One PCIe module is designated as the master module and the other PCIe modules are designated as the slave modules.

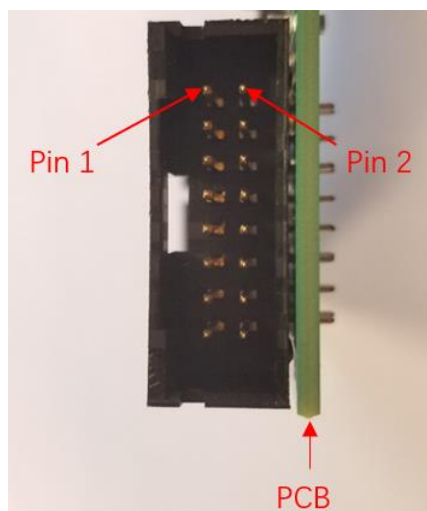


Figure 106 SSI Connector in PCIe-5500

Pin	Signal Name	Signal Name	Pin
1	PXI_TRIG0	GND	2
3	PXI_TRIG1	GND	4
5	PXI_TRIG2	GND	6
7	PXI_TRIG3	GND	8
9	PXI_TRIG4	GND	10
11	PXI_TRIG5	GND	12
13	PXI_TRIG6	GND	14
15	PXI_TRIG7	GND	16

Table 25 SSI Connector Pin Assignment for PCIe-5500

4.11 DIP Switch in PCIe-5500

PCIe-5500 series modules have a DIP switch. The card number can be adjusted manually by changing the DIP switch setting, which is used to identify the boards with different slot positions.

For example, if you want to set the card number to 3, you could turn the position 2 and 1 of the DIP switch to the ON position and the others to OFF. See below for details.

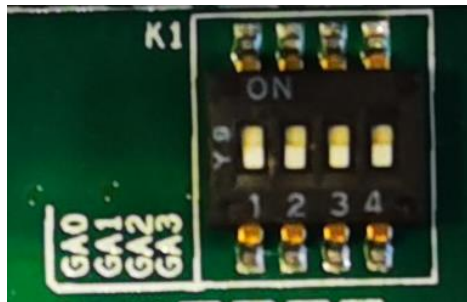


Figure 107 DIP Switch in PCIe-5500

	Position 4 (GA3)	Position 3 (GA2)	Position 2 (GA1)	Position 1 (GA0)
Slot 0	0	0	0	0
Slot 1	0	0	0	1
Slot 2	0	0	1	0
Slot 3	0	0	1	1
Slot 4	0	1	0	0
Slot 5	0	1	0	1
Slot 6	0	1	1	0
Slot 7	0	1	1	1
Slot 8	1	0	0	0
Slot 9	1	0	0	1
Slot 10	1	0	1	0
Slot 11	1	0	1	1
Slot 12	1	1	0	0
Slot 13	1	1	0	1
Slot 14	1	1	1	0
Slot 15	1	1	1	1

Note: OFF=0/ ON=1

Table 26 Relationship between switch position and slot number

5. Calibration

PCIe/PXIe-5500 Series boards are precalibrated before the shipment. We recommend you recalibrate PCIe/PXIe-5500 board periodically to ensure the measurement accuracy. A commonly accepted practice is one year. If for any reason, you need to recalibrate your board, please contact JYTEK.

6. Using PCIe/PXIe-5500 in Other Software

While JYTEK's default application platform is Visual Studio, the programming language is C#, we recognize there are other platforms that are either becoming very popular or have been widely used in the data acquisition applications. Among them are Python, C++ and LabVIEW. This chapter explains how you can use PCIe/PXIe-5500 DAQ card using one of this software.

6.1 Python

JYTEK provides and supports a native Python driver for PCIe/PXIe-5500 boards. There are many different versions of Python. JYTEK has only tested in CPython version 3.5.4. There is no guarantee that JYTEK python drivers will work correctly with other versions of Python.

If you want to be our partner to support different Python platforms, please contact us.

6.2 C++

We recommend our customers to use C# drivers because C# platform deliver much better efficiency and performance in most situations. We also provide C++ drivers and examples in the Qt IDE, which can be downloaded from web. However, due to the limit of our resources, we do not actively support C++ drivers. If you want to be our partner to support C++ drivers, please contact us.

6.3 LabVIEW

LabVIEW is a software product from National Instruments. JYTEK does not support LabVIEW and will no longer provide LabVIEW interface to PCIe/PXIe-5500 boards. Our third-party partners may have LabVIEW support to PCIe/PXIe-5500 boards. We can recommend you if you want to convert your LabVIEW applications to C# based applications.

7. About JYTEK

7.1 JYTEK China

Founded in June, 2016, JYTEK China is a leading Chinese test & measurement company, providing complete software and hardware products for the test and measurement industry. The company is a joint venture between Adlink Technologies and a group of experienced professionals from the industry. JYTEK independently develop the software and hardware products and is entirely focused on the Chinese market. Our Shanghai headquarters and production service center have regular stocks to ensure timely supply; we have R&D centers in Xi'an and Chongqing to develop new products; we also have highly trained direct technical sales representatives in Shanghai, Beijing, Tianjin, Xi'an, Chengdu, Nanjing, Wuhan, Haerbin, and Changchun. We also have many partners who provide system level support in various cities.

7.2 JYTEK Korea and JYTEK In Other Countries

JYTEK Korea was the first JYTEK enterprise outside China to promote JYTEK products. Together with Adlink Technologies and JYTEK China, JYTEK is expanding to more countries. Each JYTEK location is an independently owned and operated franchise. It shares JYTEK's philosophy and business approach. Together JYTEK entities promote the JYTEK brand, technology, and products.

7.3 JYTEK Hardware Products

According to JYTEK's agreement with our equity partner Adlink Technologies, JYTEK's hardware is manufactured by the state-of-art manufacturing facility located in Shanghai Zhangjiang Hi-Tech Park. Adlink has over 20 years of the world-class low-volume and high-mix manufacturing expertise with ISO9001-2008, China 3C, UL, ROHS, TL9000, ISO-14001, ISO-13485 certifications. Its 30,000 square meters facilities and three high-speed Panasonic SMT production lines can produce 60,000 pieces boards/month; it also has full supply chain management - planning, sweeping, purchasing, warehousing and distribution. Adlink's manufacturing excellence ensures JYTEK's hardware has word-class manufacturing quality.

One core technical advantage is JYTEK's pursue for the basic and fundamental technology excellence. JYTEK China has developed a unique PCIe, PXIe, USB hardware driver architecture, FirmDrive, upon which many our future hardware will be based.

In addition to our own developed hardware, JYTEK also rebrands Adlink's PXI product lines. In addition, JYTEK has other rebranding agreements to increase our hardware coverage. It is our goal to provide the complete product coverage in PXI and PCI modular instrumentation and data acquisition.

7.4 JYTEK Software Platform

JYTEK has developed a complete software platform, SeeSharp Platform, for the test and measurement applications. We leverage the open sources communities to provide the software tools. Our platform software is also open sourced and is free, thus lowering the cost of tests for our customers. We are the only domestic vendor to offer complete commercial software and hardware tools.

7.5 JYTEK Warranty and Support Services

With our complete software and hardware products, JYTEK is able to provide technical and sales services to wide range of applications and customers. In most cases, our products are backed by a 1-year warranty. For technical consultation, pre-sale and after-sales support, please contact JYTEK of your country.

8. Appendix 1 Common Analog Measurement Issues

8.1 Floating Signals and Ground Referenced Signals

Signals to be measured often fall into two categories: floating and ground referenced. The floating signals include battery output, isolated output, thermocouples etc; the ground referenced signals include most instrumentation output signals. Some instruments also offered isolated floating output.

8.2 Differential, NRSE, RSE Modes

The DAQ boards have three measurement modes: differential (DIFF), non-referenced singled end (NRSE), and the referenced single end (RSE). The NRSE mode is also referred as the pseudo differential mode. Under the NRSE mode, the DAQ card provides a common connecting terminal, referred as AI_Sensing. The negative ends of input signal and the DAQ boards are all connected to this terminal, making it look like the differential mode. Thus, the NRSE mode can handle twice as many channels as the DIFF mode.

The three measurement modes and the two types of input signals, floating and ground referenced, form 6 different measurement scenarios as shown in the following.

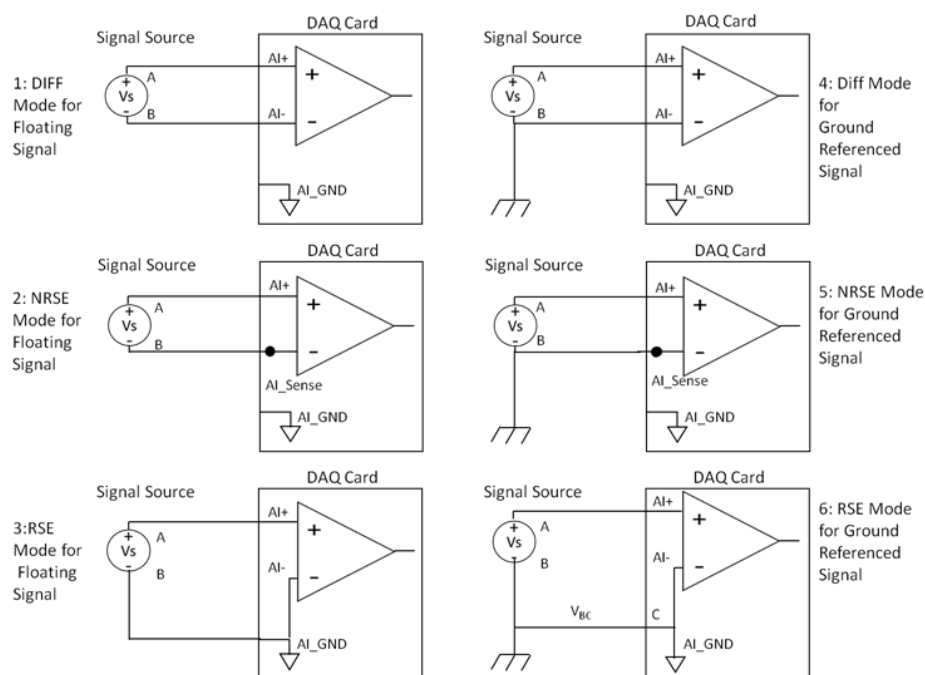


Figure 108 Six Measurement Scenarios

In the first 5 scenarios, V_{AB} is measured voltage. But in the 6th scenario, both the measured signal and the DAQ have own grounds. The two ground may have a voltage difference V_{BC} . The actual measurement is $V_{AC}=V_{AB}+V_{BC}$, not V_{AB} . Due to the ground noise, V_{BC} is quite noisy. This affects the measurement accuracy. The caution must be taken using 6th mode.

8.3 Reducing the Common Mode Voltage Effect

In the first 2 modes, the measured signal is floating. It is quite often that the common mode voltage will appear. To reduce this effect on the measurement accuracy, a resistor can be added as shown. The value of this resistor depends on the impedance of the signal source. As a rule of thumb, R should be 1000 times of the signal source output impedance, roughly 10K to 100KΩ. At this level, R has very little impact on the measurement.

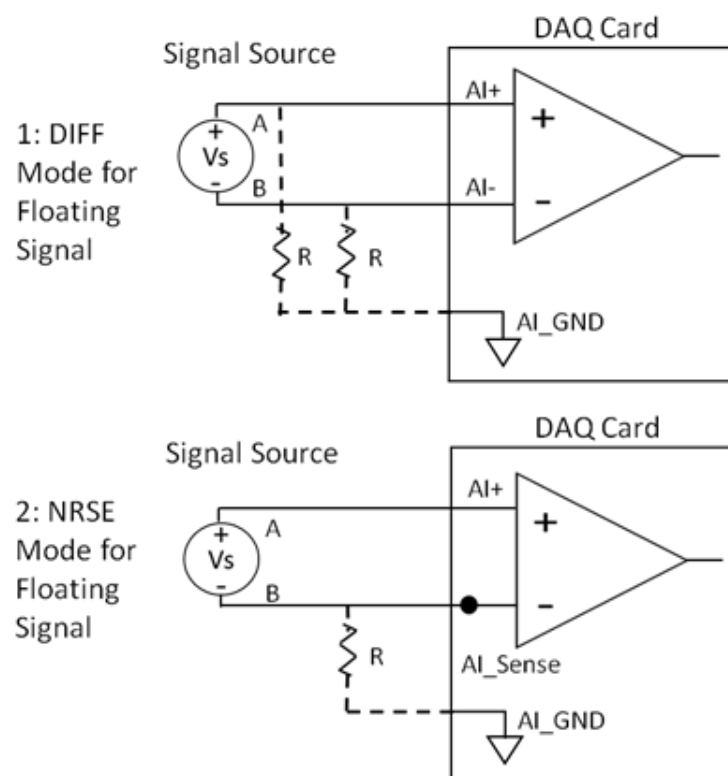


Figure 109 Using Resistor to Reduce Common Mode Voltage Effect

8.4 DC, AC and DSA Mode

Figure 110 shows three different measurement modes: DC, AC and DSA. It is important to know what type of the measurement you are making. Table 27 shows differences and features in these three modes.

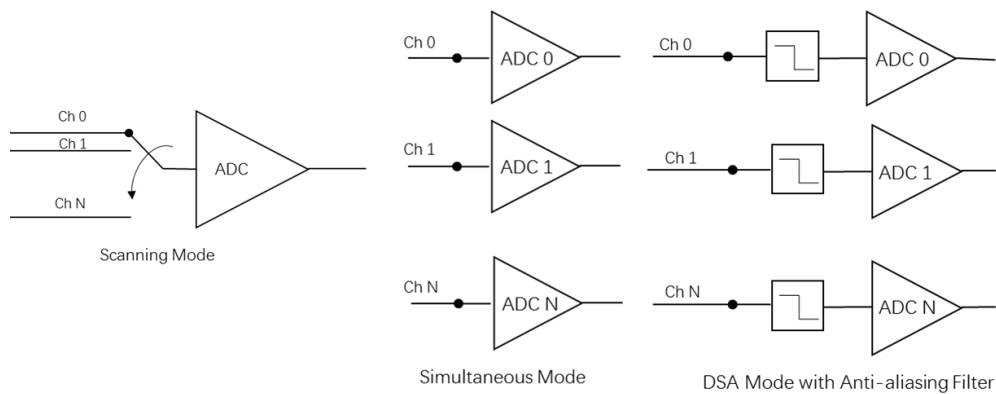


Figure 110 DC, AC and DSA Mode

	DC	AC	DSA
Signal Frequency (f)	$f=0$, or $f \leq \epsilon * f_s / 10$	$f_s > 5f$	$f_s > 2.5f$
Anti-aliasing Filter	No	No	Yes
Measurements			
Single point voltage accuracy	Yes	Maybe	No
Power Spectrum	No	Maybe	Yes
Rising/falling edges	No	Yes	No
Averaging	Time	Frequency	Frequency
ADC Mode			
Scanning	Optional	Optional	No
Scanning Interval (T)	$T \ll 1/f$	$T \ll 1/f$	N/A
Simutaneous	Optional	Optional	Yes
fs: channel sampling rate; ϵ : total accuracy;			

Table 27 DC, AC, DSA Measurements

8.4.1 DC Mode

In a DC mode, the signal frequency f should be zero or very small. Many times, engineers use averaging to reduce the noise effect. But inappropriate use of averaging will not reduce the noise effect but introduce the error. Given the **Total Accuracy** ϵ , from Sections 2.1.2, 2.1.3, 2.1.4, the maximum source signal frequency f should be bounded by:

$$f \leq \frac{\varepsilon}{10} f_s$$

where f_s is the sample rate. This formula can be used in both the DAQ and the DS Mode. This formula suggests that a faster sampling device such as PCIe/PXIe-5500 can allow bigger signal changes and still achieve excellent accuracy.

8.4.2 AC Mode

The AC mode traditionally measures power line voltage of 50Hz or 60Hz, but has been extended to other frequencies. Due to the alternating nature of the AC signal, the average cannot be done in the time domain. If averaging must be used, it is used in the frequency domain when measuring the power spectrum.

If you use the 1M/N maximum channel sample rate of PCIe/PXIe-5500 in the multi-channel mode, the channel switching error cannot be ignored, as seen in Table 6. If you need better accuracy, you should consider using the simultaneous DAQ devices such as JYTEK PXIe-5315. These devices do not use scanning mode. Rather, each channel is serviced by a dedicated ADC to avoid the channel switching error. This ensures better AC accuracy as well as better synchronization.

Another use of the AC measurement is to analyze the signal's change. In this case, the sample rate must be sufficiently higher than the signal frequency to catch the changing nature of the signal. As a rule of thumb, 5 times of the signal frequency is often used.

8.4.3 DSA Mode

The DSA (Dynamic Signal Analysis) mode mostly measures the signal frequency spectrum. In order to reduce the noise and increase the dynamic range, an anti-aliasing filter is used. Similar to the simultaneous mode, each channel is serviced by a dedicated ADC. To meet the sampling theorem, the sample rate f_s should be at least 2.5 times of the signal frequency.

PCIe/PXIe-5500 boards are not designed for the DSA measurements. There is no anti-aliasing filter in a PCIe/PXIe-5500 board.

9. Statement

The hardware and software products described in this manual are provided by JYTEK China, or JYTEK in short.

This manual provides the product review, quick start, some driver interface explanation for JYTEK PCIe/PXIe-5500 Series family of multi-function data acquisition boards. The manual is copyrighted by JYTEK.

No warranty is given as to any implied warranties, express or implied, including any purpose or non-infringement of intellectual property rights, unless such disclaimer is legally invalid. JYTEK is not responsible for any incidental or consequential damages related to performance or use of this manual. The information contained in this manual is subject to change without notice.

While we try to keep this manual up to date, there are factors beyond our control that may affect the accuracy of the manual. Please check the latest manual and product information from our website.

Shanghai Jianyi Technology Co., Ltd.

Address: Room 201, Building 3, NO.300 Fangchun Road, Shanghai.

Post Code: 201203

Tel: 021-5047 5899

Website: www.jytek.com