



PCIe/PXIe-5500 Series Family of Multi-functional Data Acquisition Boards User Manual



User Manual Version: V2.2

Revision Date: Dec. 25, 2020

Table of Contents

1.	Introduction	1
1.1	Overview	1
1.2	Main Features	1
1.3	Abbreviations	2
1.4	Learn by Example.....	3
2.	Hardware Specifications.....	6
2.1	System Diagram.....	6
2.2	Analog Input Specifications	7
2.3	Analog Output Specifications	8
2.4	Digital IO Specifications.....	9
2.5	Counter/Timer Specifications	9
2.6	PFI Specifications	9
2.7	Front Panel and Pin Definition.....	10
3.	Performance and Tests.....	15
3.1	AI Accuracy.....	15
3.2	AI Bandwidth	16
3.3	System Noise	17
3.4	Temperature Drift.....	17
3.5	CMRR	18
3.6	AO Accuracy	19
4.	Software	20
4.1	System Requirements	20
4.2	System Software	20
4.3	C# Programming Language.....	21
4.4	PCIe/PXIe-5500 Series Hardware Driver	21
4.5	Install the SeeSharpTools from JYTEK	21
4.6	Running C# Programs in Linux	21
5.	Operating PCIe/PXIe-5500.....	23
5.1	Quick Start.....	23
5.2	Data Acquisition Methods.....	23
5.2.1	Continuous Acquisition.....	25
5.2.2	Finite Acquisition.....	26
5.2.3	Single Point Acquisition	26
5.2.4	Record Acquisition	26
5.3	Analog Input Terminal Type	26
5.3.1	Differential Mode	26
5.3.2	RSE Mode.....	28
5.3.3	NRSE Mode	29
5.4	Trigger Source	30
5.4.1	Immediate trigger	30

5.4.2	Software Trigger	31
5.4.3	External Analog Trigger.....	32
5.4.4	External Digital Trigger	38
5.5	Trigger Mode.....	40
5.5.1	Start Trigger	40
5.5.2	Reference Trigger	41
5.5.3	ReTrigger	42
5.6	AO Operations.....	45
5.6.1	Finite Output.....	45
5.6.2	Continuous NoWrapping Output.....	47
5.6.3	Continuous Wrapping Output	51
5.7	Digital I/O Operations	53
5.7.1	Static DI/DO	53
5.7.2	Dynamic DI/DO	55
5.8	Counter Input Operations	55
5.8.1	Edge Counting.....	56
5.8.2	Pulse Measurement	61
5.8.3	Frequency Measurement.....	65
5.8.4	Period Measurement.....	68
5.8.5	Two-Edge Separation.....	70
5.8.6	Quadrature Encoder.....	74
5.8.7	Two-Pulse Encoder.....	78
5.9	Counter Output Operations	83
5.9.1	Single Pulse Output	83
5.9.2	Finite Pulse Output.....	85
5.9.3	Continuous Pulse Output.....	86
5.10	System Synchronization Interface (SSI)	89
5.11	DIP Switch in PCIe-5500	89
6.	Calibration	91
7.	Using PCIe/PXle-5500 in Other Software	92
7.1	Python.....	92
7.2	C++.....	92
7.3	LabVIEW.....	92
8.	About JYTEK.....	93
8.1	JYTEK China	93
8.2	JYTEK Korea and JYTEK In Other Countries	93
8.3	JYTEK Hardware Products	93
8.4	JYTEK Software Platform	94
8.5	JYTEK Warranty and Support Services.....	94
9.	Appendix 1 Common Analog Measurement Issues	95
9.1	Floating Signals and Ground Referenced Signals	95

9.2 Differential, NRSE, RSE Modes.....	95
9.3 Reducing the Common Mode Voltage Effect.....	96
10. Statement.....	97

Figure 1 PCIe/PXle-5500 Series System Block Diagram.....	6
Figure 2 PXle/PCIe 5510 Front Panel.....	10
Figure 3 AI Measurement Error.....	16
Figure 4 AI Error due to System Random Noise.....	17
Figure 5 AI Gain Error due to Temperature Change.....	17
Figure 6 AI Offset Error due to Temperature Change.....	18
Figure 7 Common Mode Rejection Ratio.....	18
Figure 8 AO Output Error.....	19
Figure 9 Sample Rate and Internal AD Conversion.....	24
Figure 10 RSE Mode for Floating Signals.....	29
Figure 11 NRSE Mode for Grounding Signals.....	29
Figure 12 Differential Mode for Grounding Signals.....	27
Figure 13 Above high.....	33
Figure 14 Below low.....	34
Figure 15 Rising hysteresis.....	35
Figure 16 Falling hysteresis.....	35
Figure 17 Window comparator.....	36
Figure 18 External Digital Trigger.....	38
Figure 19 Start Trigger.....	41
Figure 20 Reference Trigger.....	42
Figure 21 Re-Trigger.....	42
Figure 22 Counter Terminal.....	55
Figure 23 Simple Edge Counting in Single Mode.....	57
Figure 24 Bufferd Edge Counting with Explicit Sample Clock.....	57
Figure 25 Simple Edge Counting with Implicit SampleClk.....	58
Figure 26 Count Direction.....	58
Figure 27 Pulse Measurement in Single Mode.....	62
Figure 28 Pulse Measurement with Explicit SampleClk.....	62
Figure 29 Pulse Measurement with Implicit SampleClk.....	62
Figure 30 Frequency Measurement with Explicit Sample Clock.....	66
Figure 31 Two-Edge Separation in Single Mode.....	71
Figure 32 Two-Edge Separation with Explicit Sample Clock.....	71
Figure 33 Two-Edge Separation with Implicit Sample Clock.....	72
Figure 34 Quadrature Encoder x1 Mode.....	74
Figure 35 Quadrature Encoder x2 Mode.....	75

Figure 36 Quadrature Encoder x4 mode.....	75
Figure 37 Quadrature Encoder x4 with Explicit Sample Clock.....	76
Figure 38 Quadrature Encoder x4 with Implicit Sample Clock.....	77
Figure 39 Two-Pulse Encoder in Single Mode.....	79
Figure 40 Two-Pulse Encoder with Explicit Sample Clock.....	79
Figure 41 Two-Pulse Encoder with Implicit Sample Clock.....	80
Figure 42 Single Pulse Output.....	83
Figure 43 Finite Pulse Output.....	85
Figure 44 Continuous Pulse Output.....	87
Figure 45 SSI Connector in PCIe-5500.....	89
Figure 46 Slide Switch in PCIe-5500.....	90
Figure 47 Six Measurement Scenarios.....	95
Figure 48 Using Resister to Reduce Common Mode Voltage Effect.....	96

Table 1 5500 Family and Main Features.....	1
Table 2 5500 on Different Bus.....	1
Table 3 Analog Input Specifications.....	7
Table 4 AI Absolute Accuracy.....	8
Table 5 Analog Output Specifications.....	8
Table 6 Analog Output Accuracy.....	8
Table 7 Digital IO Specifications.....	9
Table 8 Counter/Timer Specifications.....	9
Table 9 PFI Specifications.....	9
Table 10 5510 / 5511 Pin Definition.....	11
Table 11 5510 / 5511 Counter Pin Define.....	12
Table 12 5515 / 5516 Pin Definition.....	13
Table 13 5515 / 5516 Counter Pin Define.....	14
Table 14 Notes to Legend.....	14
Table 15 Typical AI Accuracy.....	15
Table 16 AI Bandwidth.....	16
Table 17 Typical AO Accuracy.....	19
Table 18 Supported Linux Versions.....	20
Table 19 SSI Connector Pin Assignment for PCIe-5500.....	89
Table 20 Relationship between switch position and slot number.....	90

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

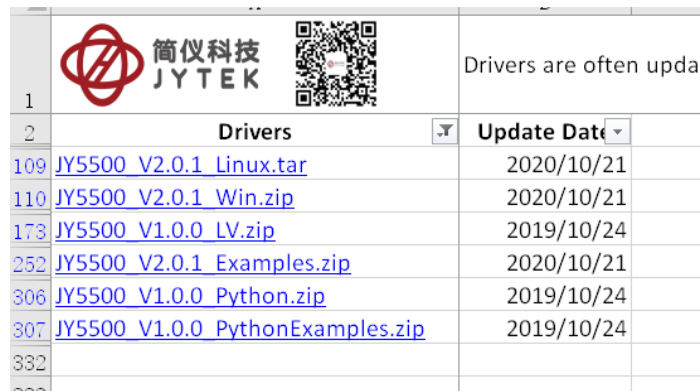
NRSE: Non-Referenced Single-Ended

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 and install the sample programs for this device. You can visit our web www.jytek.com to search for the example file or you can download a [JYPEDIA](#) excel file from our web. Open JYPEDIA and search for 5500 in the driver sheet, select **JY5500_V2.0.1 Examples.zip**. This will lead you to download the sample program for this device. 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		Update Date
109	JY5500_V2.0.1_Linux.tar	2020/10/21
110	JY5500_V2.0.1_Win.zip	2020/10/21
173	JY5500_V1.0.0_LV.zip	2019/10/24
252	JY5500_V2.0.1_Examples.zip	2020/10/21
306	JY5500_V1.0.0_Python.zip	2019/10/24
307	JY5500_V1.0.0_PythonExamples.zip	2019/10/24

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 1.

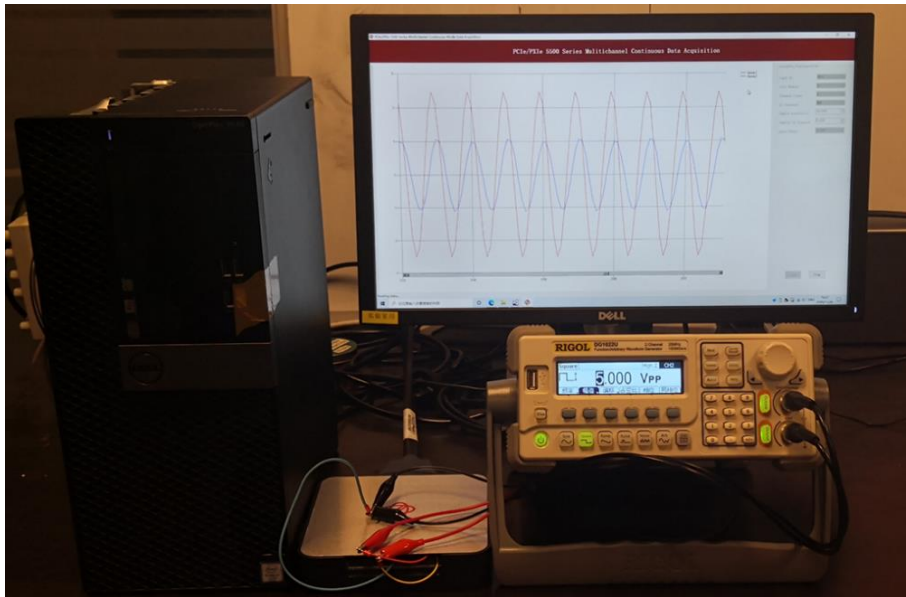


Figure 1 Experiment Setup

The TB-68 has 4 terminal columns, J1 – J4 and is shown below as Figure 2. In the rest of this manual, the wire connection in each **Learn by Example** section will be given by the pin numbers only.

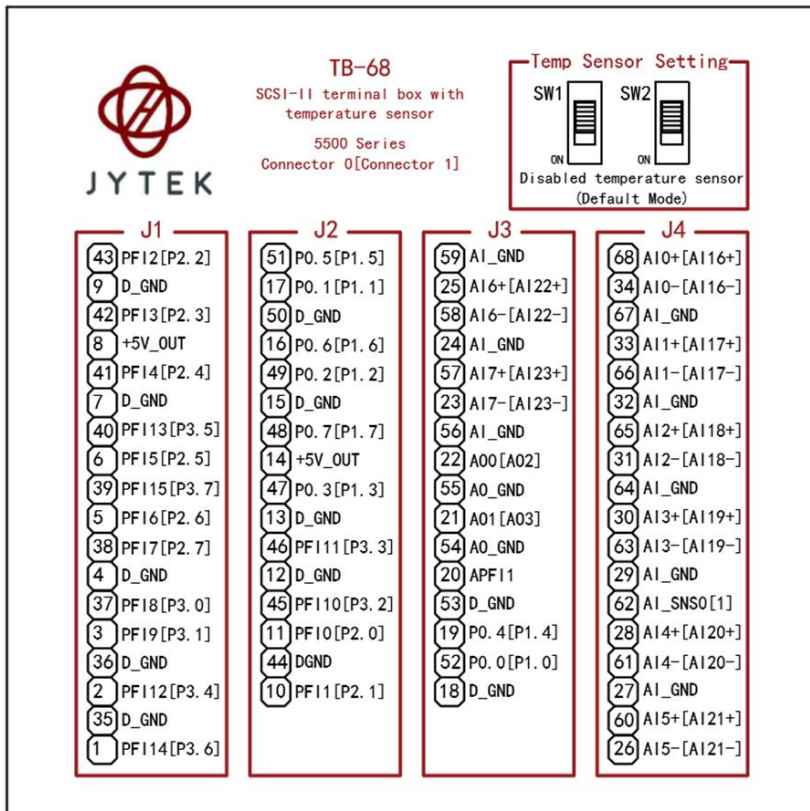


Figure 2 TB-68 Terminal Block

*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 Specifications

2.1 System Diagram

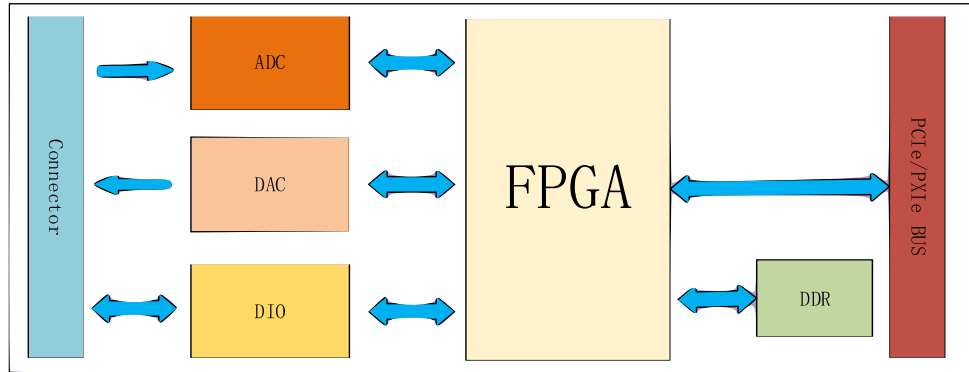


Figure 3 PCIe/PXIe-5500 Series System Block Diagram

The system block diagram of PCIe/PXIe-5500 series is shown in Figure 3. The system is mainly composed of ADC module, DAC module, DIO module and FPGA control module. It realizes ADC, DAC, digital IO and counter operations. JYTEK proprietary FPGA based driver code provides stable and efficient PCIe/PXIe/USB interfaces.

2.2 Analog Input 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)	$\pm 10/\pm 5/\pm 2/\pm 1/\pm 0.5/\pm 0.2/\pm 0.1$			
Input mode	RSE / NRSE / Differential			
Input impedance	$>1\text{ G}\Omega 100\text{ pF}$			
Input coupling	DC			
Overvoltage protection	$\pm 25\text{ V}$			
CMRR	85 dB			
Crosstalk	Adjacent Channel : -80 dB Non-adjacent Channel : -95 dB Differential: -65 dB			
DNL	$<1\text{ LSB}$			
INL	70 ppm of Range Typical			
Operating Temperature	$0^\circ\text{C} \sim 50^\circ\text{C}$			
Input FIFO	64M Samples			
Trigger type	Digital, Analog, Software			
Trigger mode	StartTrigger, ReferenceTrigger, ReTrigger			
Analog trigger voltage range	$\pm 10\text{V}$ Software Programmable			
Overvoltage Protection	Continuous : 20m A, $\pm 25\text{ V}$ Instantaneous : 40 mA, $\pm 25\text{ V}$			

Table 3 Analog Input Specifications

AI Absolute Accuracy

Nominal Range Full Scale (V)	Absolute Accuracy at Full Scale(μV)
± 10	1253
± 5	792
± 2	363
± 1	189
± 0.5	95
± 0.2	51
± 0.1	21

Table 4 AI Absolute Accuracy

2.3 Analog Output 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 5 Analog Output Specifications

AO Absolute Accuracy

Nominal Range Full Scale (V)	Absolute Accuracy at Full Scale(μ V)
± 10	1380
± 5	535

Table 6 Analog Output Accuracy

2.4 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 Duty cycle of instantaneous current pulse does not exceed 15%	

Table 7 Digital IO Specifications

2.5 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 8 Counter/Timer Specifications

2.6 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 9 PFI Specifications

2.7 Front Panel and Pin Definition

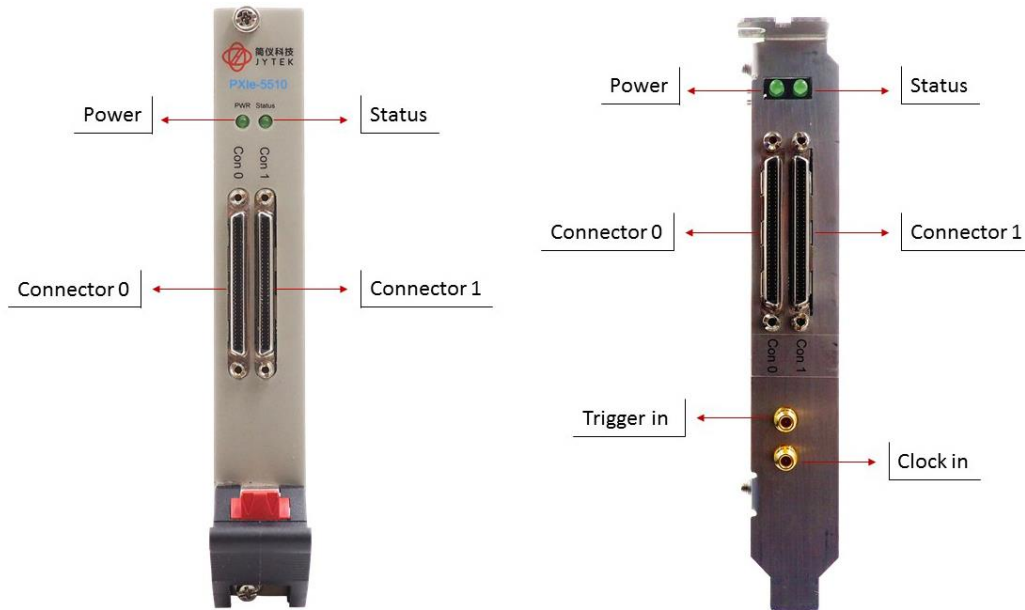


Figure 4 PXIe/PCIe 5510 Front Panel

PCIe/PXIe-5500 series boards are connected to outside signals by either two 68-pin cables for the 32 channel configurations or one the 68-pin cable for 16 channel configurations. Table 10, Table 11, Table 12, Table 13 show the pin definition 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!

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	DGND	10	P2.1	44	DGND
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 10 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 11 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 12 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 13 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 14 Notes to Legend

3. Performance and Tests

3.1 AI Accuracy

The maximum AI accuracy is limited by three factors: Total Gain Error, Total Offset Error and the Noise Uncertainty as shown in the following table. The PCIe/PXIe-5500 device is first calibrated. The test temperature is 5° C from the calibration temperature.

Typical Tested 5500 AI Accuracy							
	Nominal Range						
Nominal Range Positive Full Scale (V)	10.0	5.0	2.0	1.0	0.5	0.2	0.1
Temperature Change From Calibration (°C)	5.0	5.0	5.0	5.0	5.0	5.0	5.0
Gain Temperature Coefficient (PPM of Reading per °C)	6.1	5.6	5.5	5.2	5.0	5.7	7.0
Residual Gain Error Coefficient (PPM of Reading)	0.9	2.7	1.1	3.1	7.8	6.8	17.0
Total Gain Error Coefficient (PPM of Reading)	31.4	30.7	28.6	29.1	32.8	35.3	52.0
Total Gain Error (µV)	314.0	153.5	57.2	29.1	16.4	7.1	5.2
Offset Temperature Coefficient (PPM of Range per °C)	1.2	1.3	1.5	2.0	3.2	7.4	15.2
Residual Offset Error Coefficient (PPM of Range)	2.3	2.4	0.7	4.0	3.0	7.2	16.6
INL Error (PPM of Range)	7.1	10.1	4.2	9.2	19.2	40.1	58.9
Total Offset Error Coefficient (PPM of Range)	15.4	19.0	12.4	23.2	38.2	84.3	151.5
Total Offset Error (µV)	154.0	95.0	24.8	23.2	19.1	16.9	15.2
Random Noise,σ (µVrms)	93.4	51.7	26.4	16.4	14.9	14.1	13.9
Noise Uncertainty (µV)	2.8	1.6	0.8	0.5	0.4	0.4	0.4
Absolute Accuracy at Full Scale (µV)	470.8	250.1	82.8	52.8	35.9	24.3	20.8

Table 15 Typical AI Accuracy

A typical test of different input range is shown below. The input voltage is normalized to -1.0V and 1.0V.

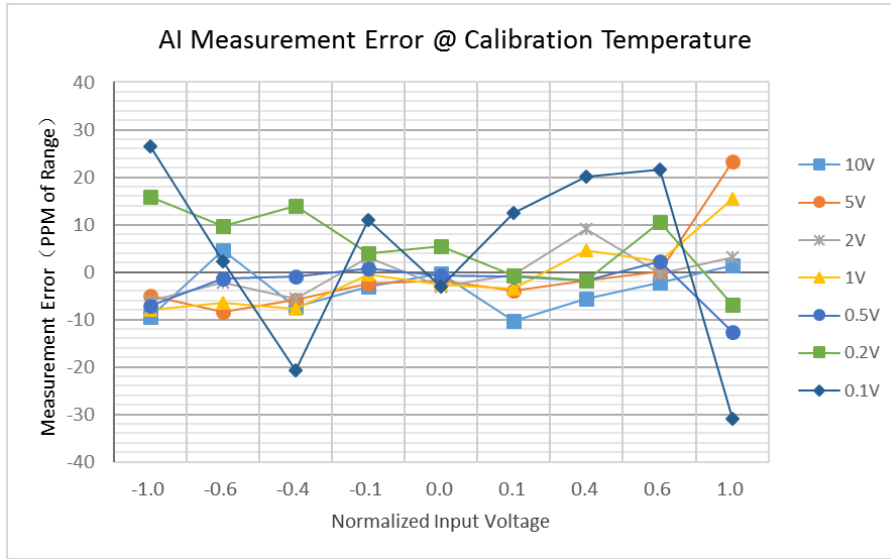


Figure 5 AI Measurement Error

3.2 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 16 AI Bandwidth

3.3 System Noise

A typical system noise is show below.

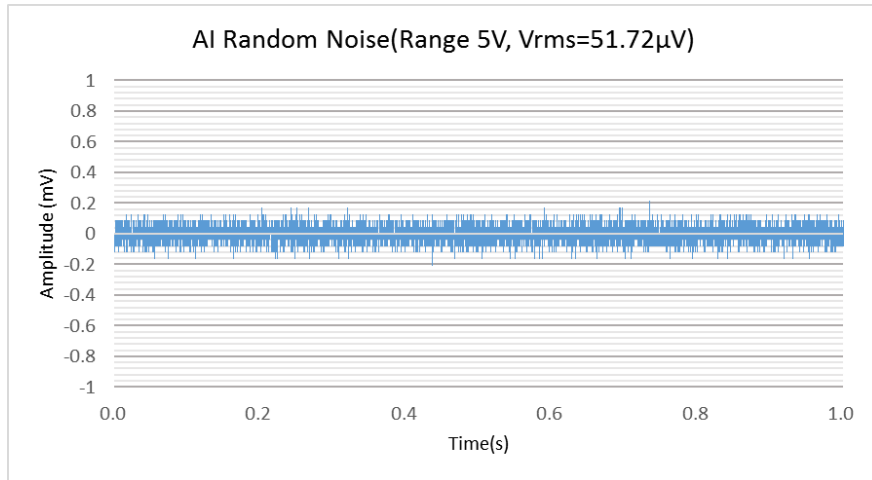


Figure 6 AI Error due to System Random Noise

3.4 Temperature Drift

The temperature change will affect the AI accuracy by the gain error and offset error. The typical gain error and offset error of PCIe/PXIe-5500 are show below.

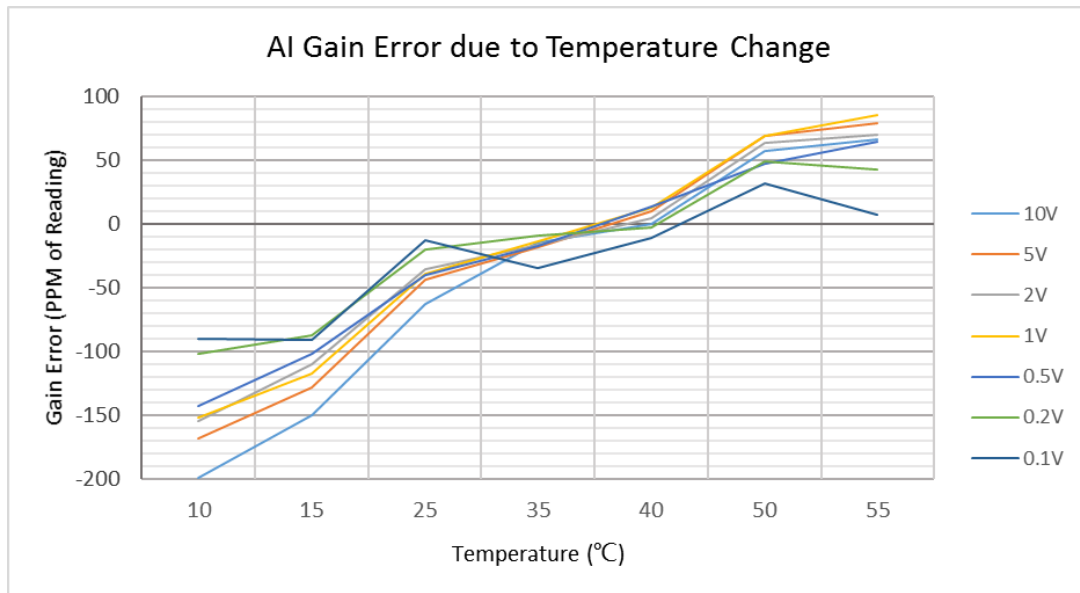


Figure 7 AI Gain Error due to Temperature Change

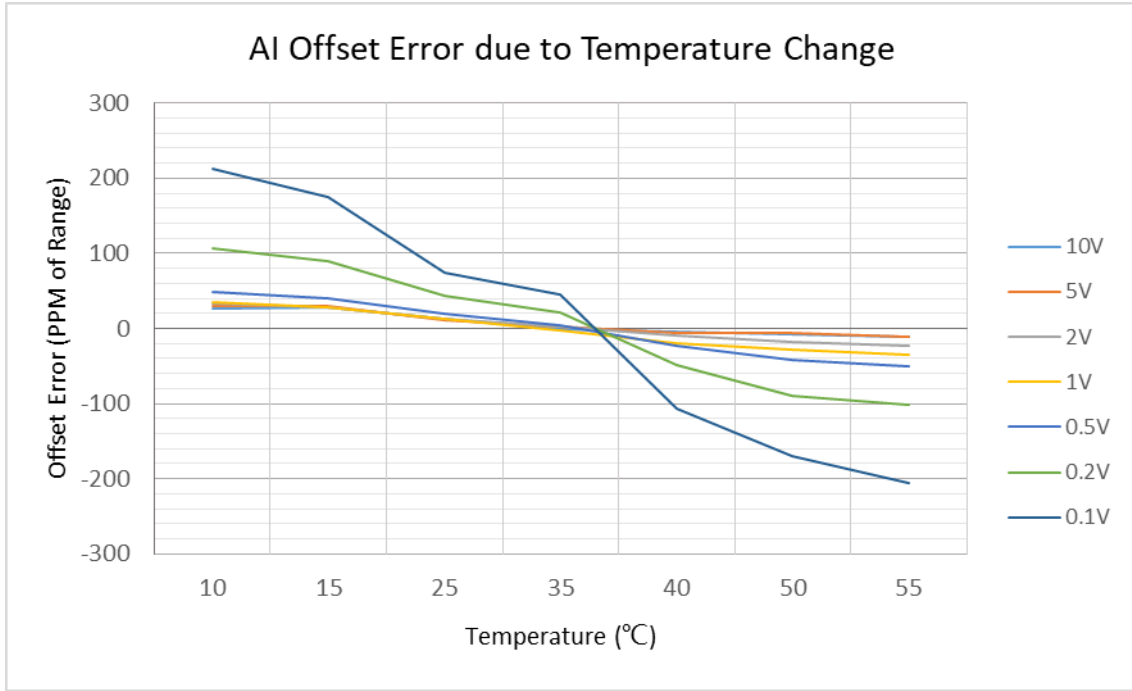


Figure 8 AI Offset Error due to Temperature Change

3.5 CMRR

A typical CMRR performance is shown below.

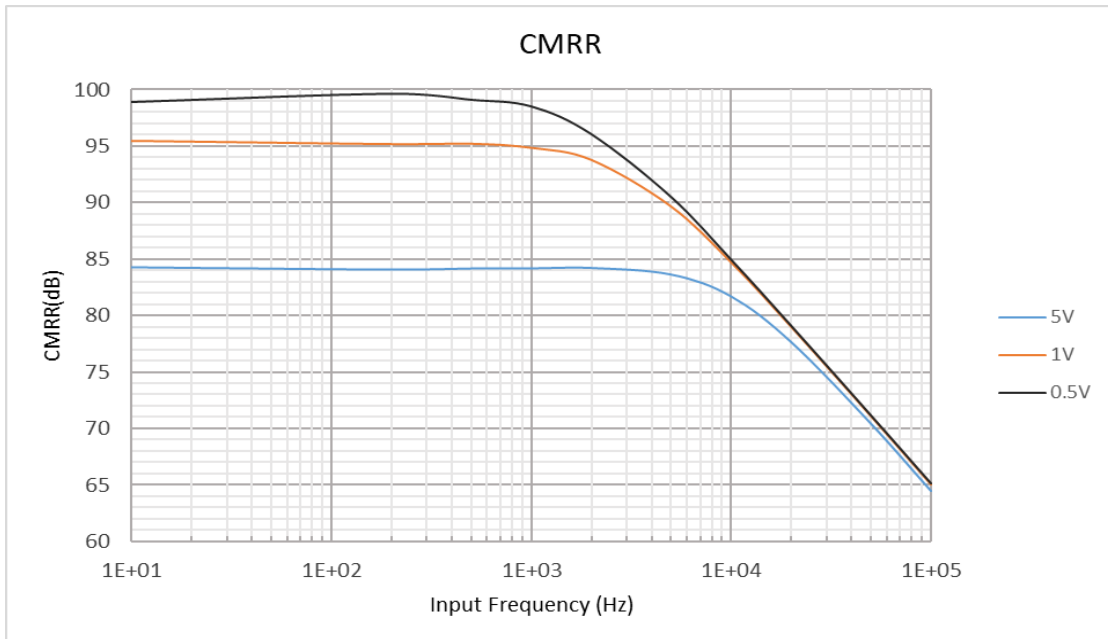


Figure 9 Common Mode Rejection Ratio

3.6 AO Accuracy

The maximum AO accuracy is limited by two factors: Total Gain Error, Total Offset Error as shown in the following table. The PCIe/PXIe-5500 device is first calibrated. The test temperature is 2°C from the calibration temperature.

Typical Tested 5500 AO Accuracy		
Nominal Range Positive Full Scale (V)	10	5
Gain Temperature Coefficient (PPM of Reading per °C)	10	7
Residual Gain Error Coefficient (PPM of Reading)	10	8
Offset Temperature Coefficient (PPM of Range per °C)	2	2
Residual Offset Error Coefficient (PPM of Range)	8	4
INL Error (PPM of Range)	60	50
Temperature Change (C)	2	2
Gain Error	30	22
Offset Error	72	58
Absolute Accuracy at Full Scale (µV)	1020	400

Table 17 Typical AO Accuracy

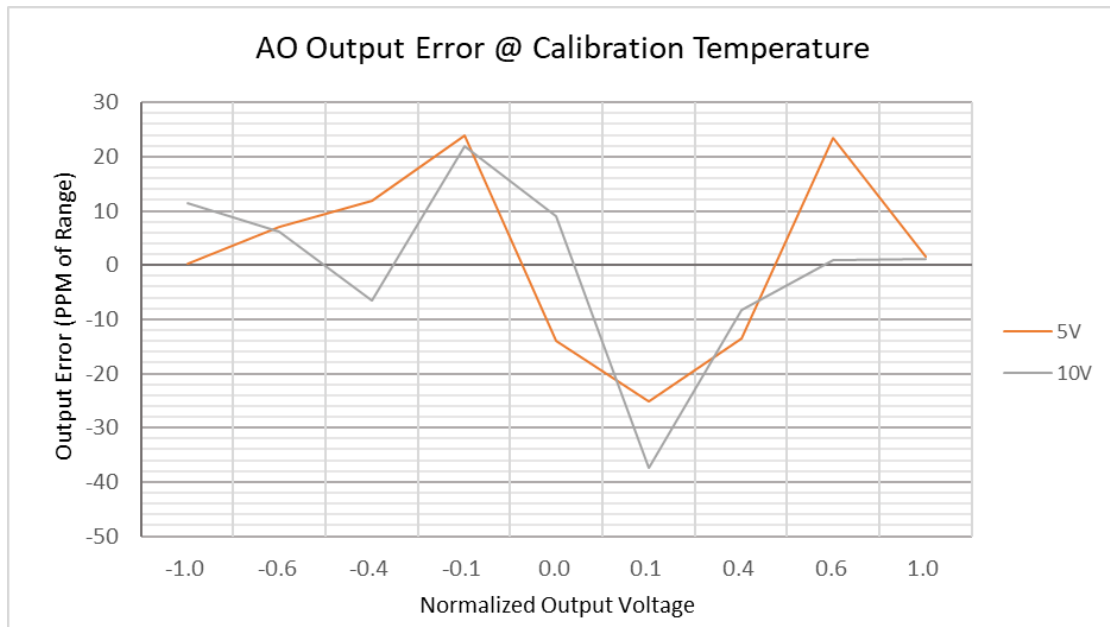


Figure 10 AO Output Error

4. Software

4.1 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.e17.x86_64

Table 18 Supported Linux Versions

4.2 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.

4.3 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.

4.4 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.

4.5 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.

4.6 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.

5. 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 his own application. In many cases, an example can also be a good starting point for a user application.

5.1 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.

5.2 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*, and *Analog Input Terminal Type*.

AI Acquisition mode (AIMode): PCIe/PXIe-5310 provides 4 acquisition modes, **Continuous**, **Finite**, **Single Point**, **Record** described in details in Section 5.2.1-5.2.4.

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 5.2.1; in the finite acquisition mode, it is the total number of samples to capture, please see Section 5.2.2.

Channel Count: how many channels do 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 11. In this particular case, **Channel Count** is 4.

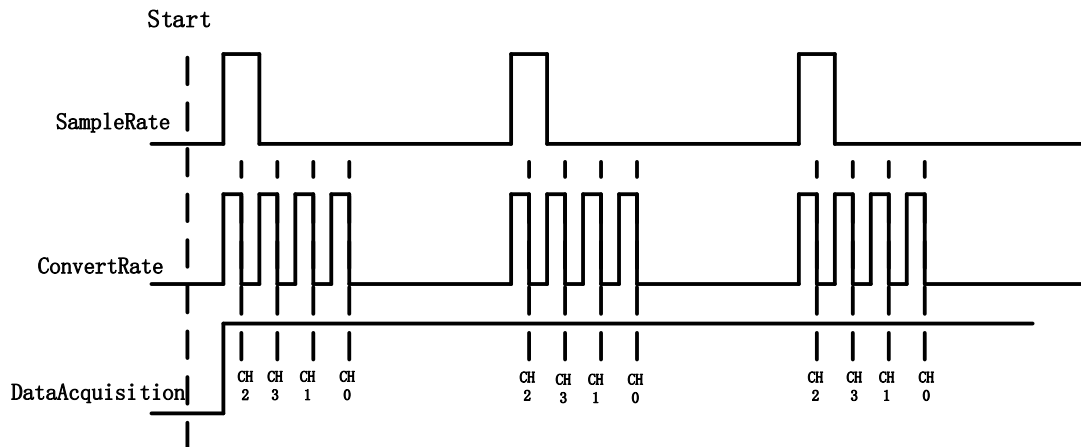


Figure 11 Sample Rate and Internal AD Conversion

ConvertRate is not a parameter in our software and is here to help you understand how ADC works. **ConvertRate** is total number of data acquired for all channels. For example, in Figure 11, if the **Sample Rate** is 3Hz, 3 samples per second per channel, and there are a total of four channels, then **ConvertRate** = 3*4 = 12 samples/s.

$$\text{ConvertRate} = \text{Sample Rate} * \text{Channel Count}$$

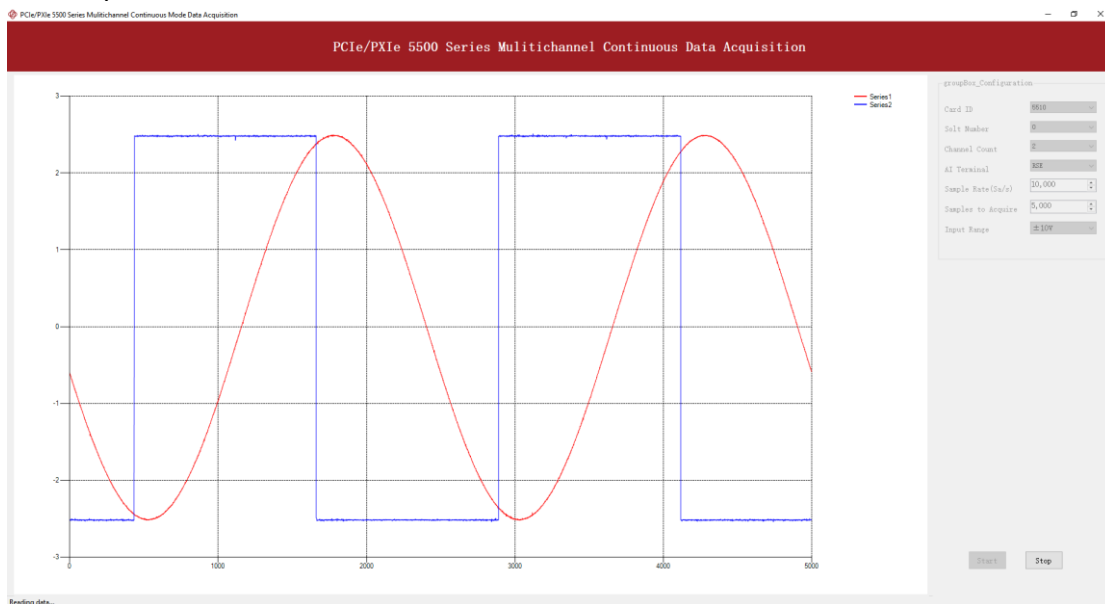
Learn by Example 5.2

- 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 1 and Figure 2. (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

- *SampleRate* is set by **Sample Rate**
- **Samples to Acquire** is the total samples to be acquired for all 2 channels 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.



5.2.1 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.

5.2.2 Finite Acquisition

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

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

5.2.3 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.

5.2.4 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.

5.3 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) and

The DIFF connection is recommended for ground-reference signal sources and it is usually better in rejecting the common-mode noise. The RSE and NRSE are recommended when the input signal sources are floating signals. These floating signal sources all share the same ground reference. Because of it, the RSE and NRSE modes can acquire twice as many channels than the DIFF mode. Appendix 9 has more details on these 3 modes.

5.3.1 Differential 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 12. The common noise appears on the 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.

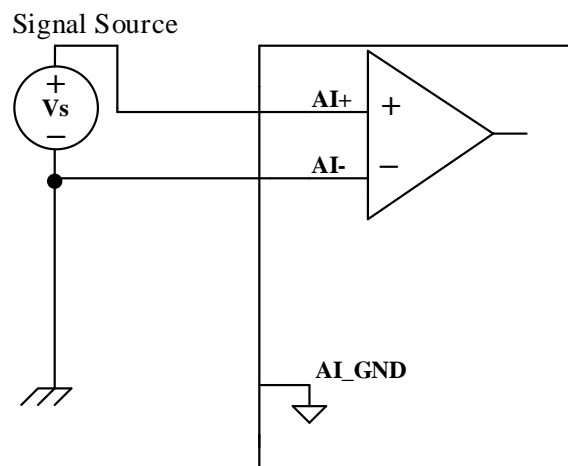


Figure 12 Differential Mode for Grounding Signals

Learn by Example 5.3.15.3

- 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 1 and Figure 2. (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	±10V

Start Stop

5.3.2 RSE Mode

In the RSE mode, all input signals' negative sides are connected to the AI ground of Instrumentation Amplifier, as shown in Figure 13. 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.

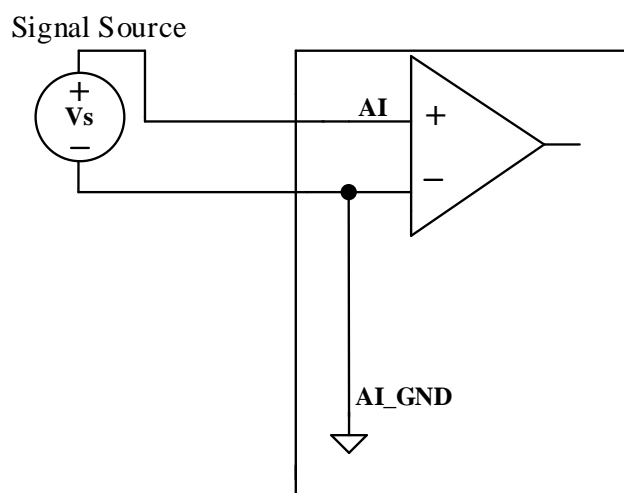


Figure 13 RSE Mode for Floating Signals

The **Learn by Example** 5.2 sample program operates under the RSE mode.

5.3.3 NRSE Mode

The NRSE mode is recommended for the measurement of ground-referenced signals, as shown in Figure 14. 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.

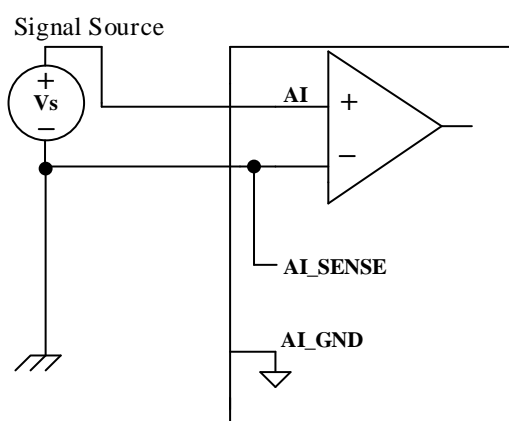


Figure 14 NRSE Mode for Grounding Signals

Learn by Example 5.3.15.3

- 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 1 and Figure 2. (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

5.4 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.

5.4.1 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 5.4.1

- Use the same program and connection as in **Learn by Example 5.2**.

groupBox_Configuration

Card ID	5510
Solt Number	0
Channel Count	2
AI Terminal	RSE
Sample Rate (Sa/s)	10,000
Samples to Acquire	5,000
Input Range	±10V

Start Stop

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

5.4.2 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 5.4.2

- 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 1 and Figure 2. (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.

groupBox_Configuration

Card ID: 5510

Solt Number: 0

Channel ID: 0

AI Terminal: RSE

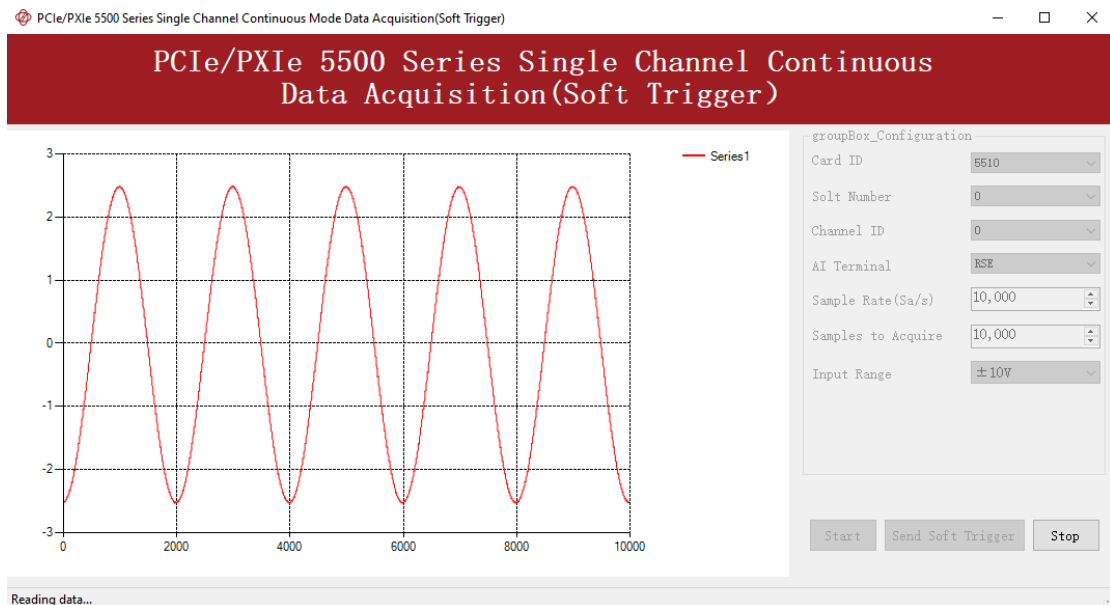
Sample Rate(Sa/s): 10,000

Samples to Acquire: 10,000

Input Range: ±10V

Start Send Soft Trigger Stop

- 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:



5.4.3 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 15

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 16

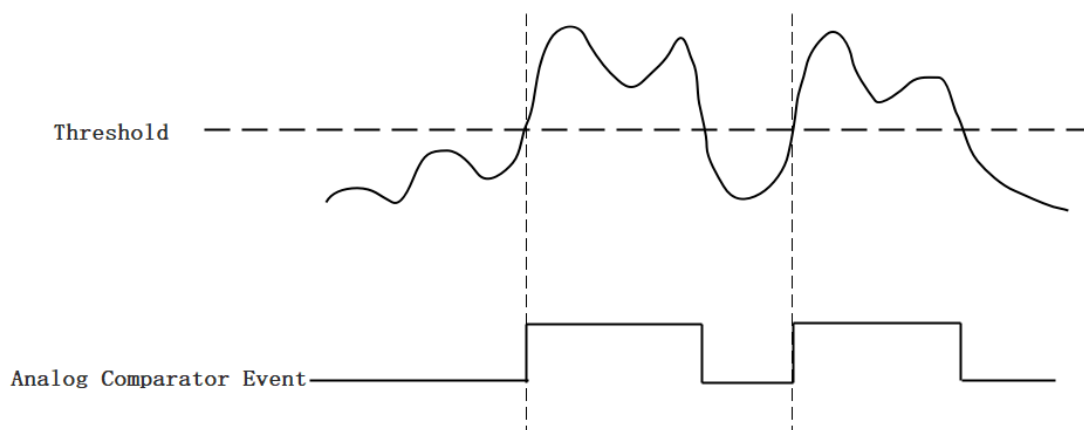


Figure 15 Rising Slope Trigger

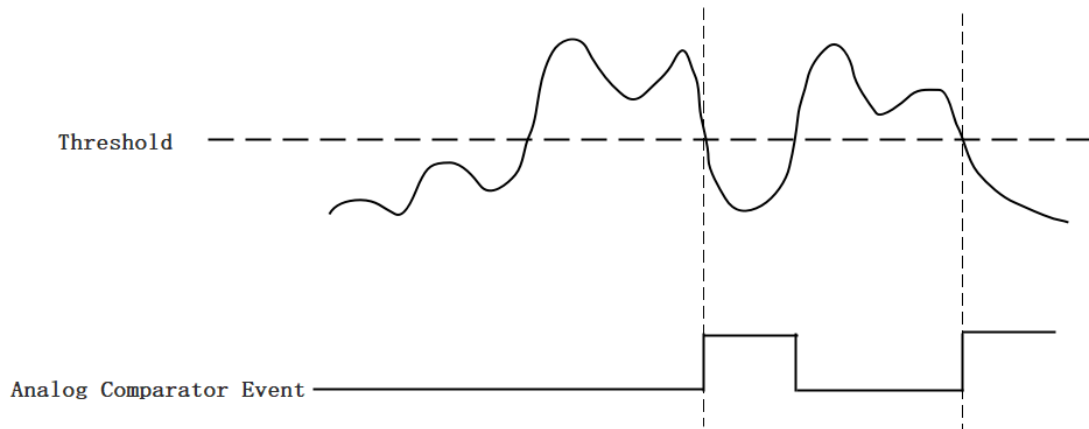


Figure 16 Falling Slope Trigger

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 17.

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 18.

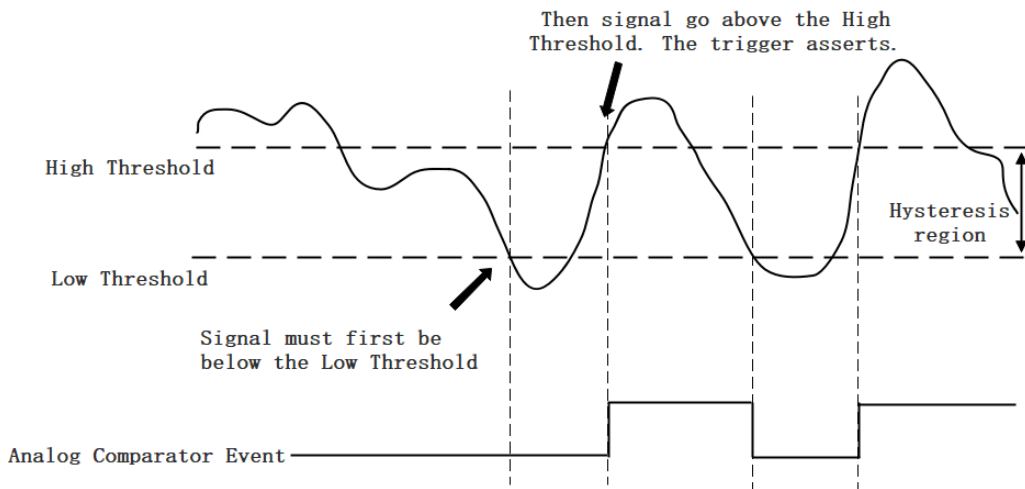


Figure 17 Hysteresis with Rising Slope Trigger

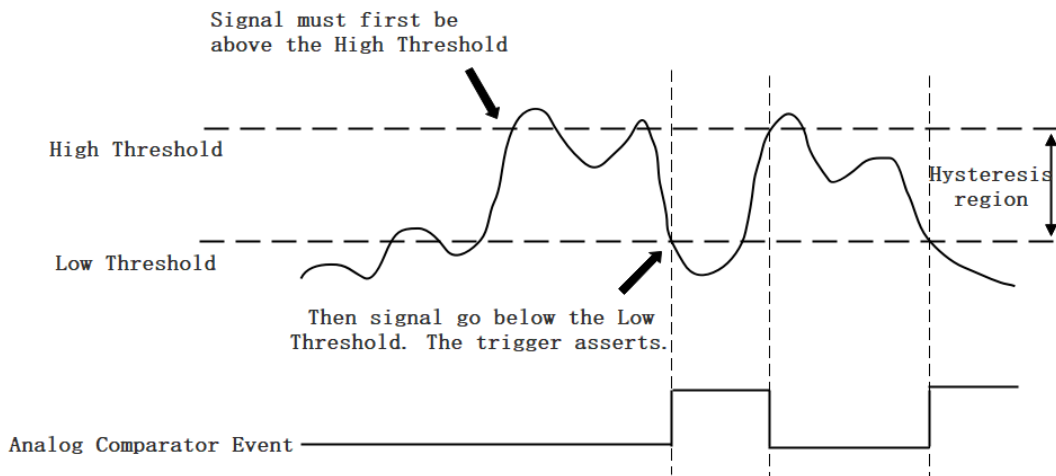


Figure 18 Hysteresis with Falling Slope Trigger

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 19.

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 20 Leaving Window Trigger

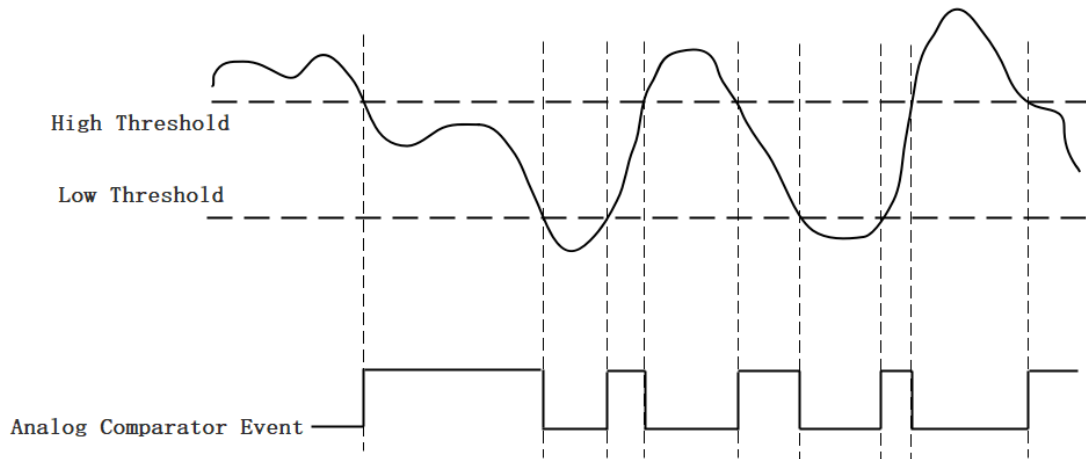


Figure 19 Entering Window Trigger

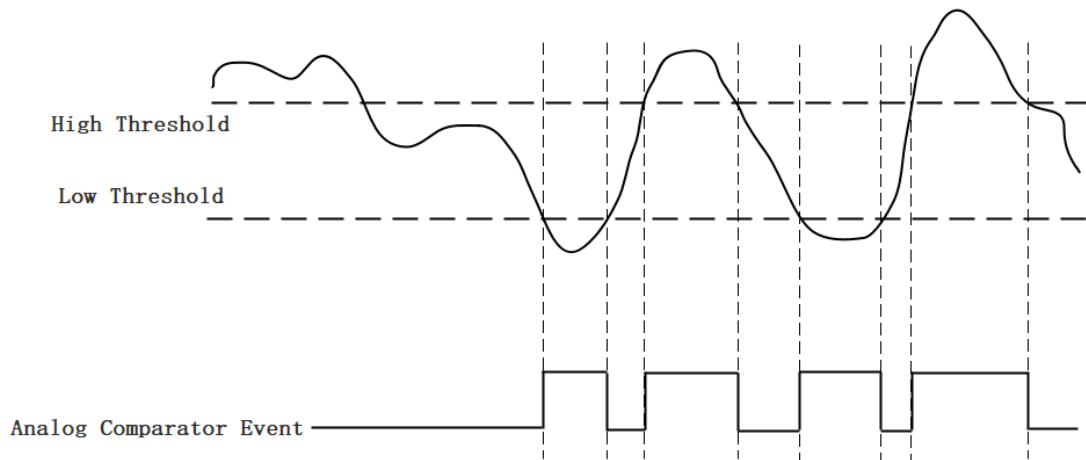


Figure 20 Leaving Window Trigger

Learn by Example 5.4.3

- Connect the signal source's positive terminal to PCIe/PXle-5500 AI Ch0 (AI0+, Pin#68), the negative terminal to the ground (AI_GND, Pin#67) as shown in Figure 1 and Figure 2. (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

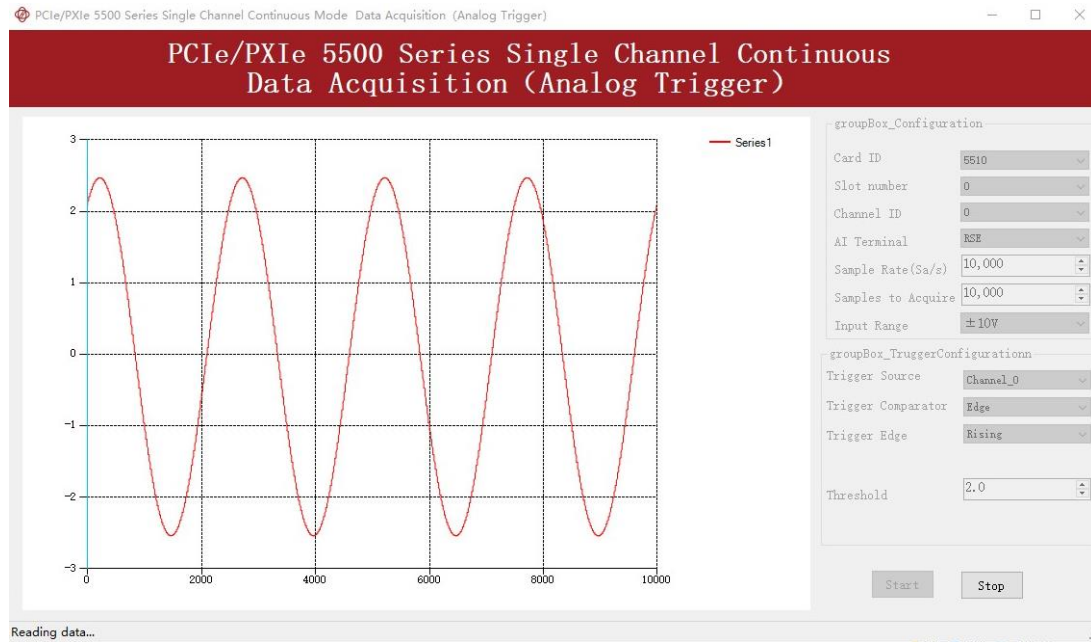
Start Stop

- 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

- 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 **5.4.3**.

- The result is shown below:



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

5.4.4 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 21



Figure 21 External Digital Trigger

Learn by Example 错误!未找到引用源。

- Connect the signal source two positive terminals to PCIe/PXIe-5500 AI Ch0,

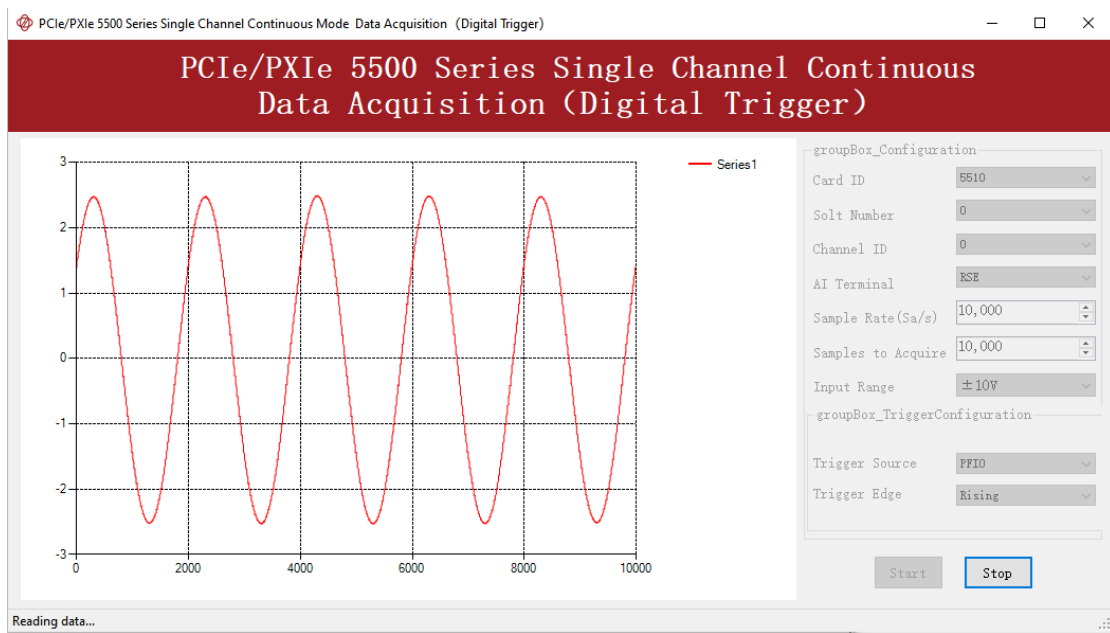
(AI0+, Pin #68) and digital trigger source (PFI 0, Pin#11), two negative terminals to the ground of analog input (AI_GND,Pin#67)and the ground of digital input/output (DGND,Pin#44) as shown in Figure 1 and Figure 2. (AI0+,AI_GND) consists of a RSE input. (PFI0, DGND) provides the trigger signal.

- 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

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



- 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.

5.5 Trigger Mode

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

5.5.1 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 22.

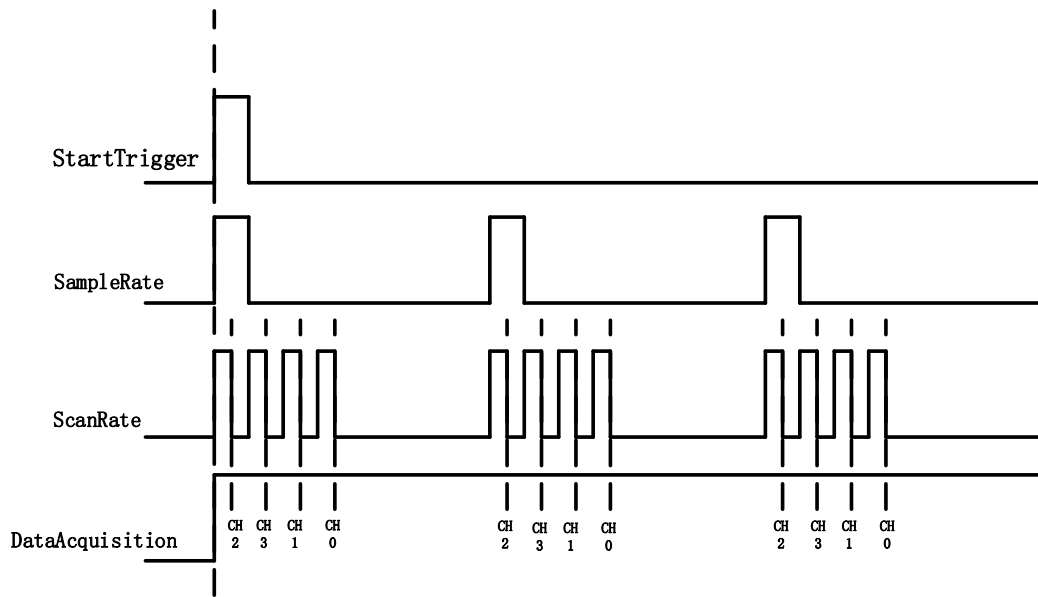


Figure 22 Start Trigger

5.5.2 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 23.

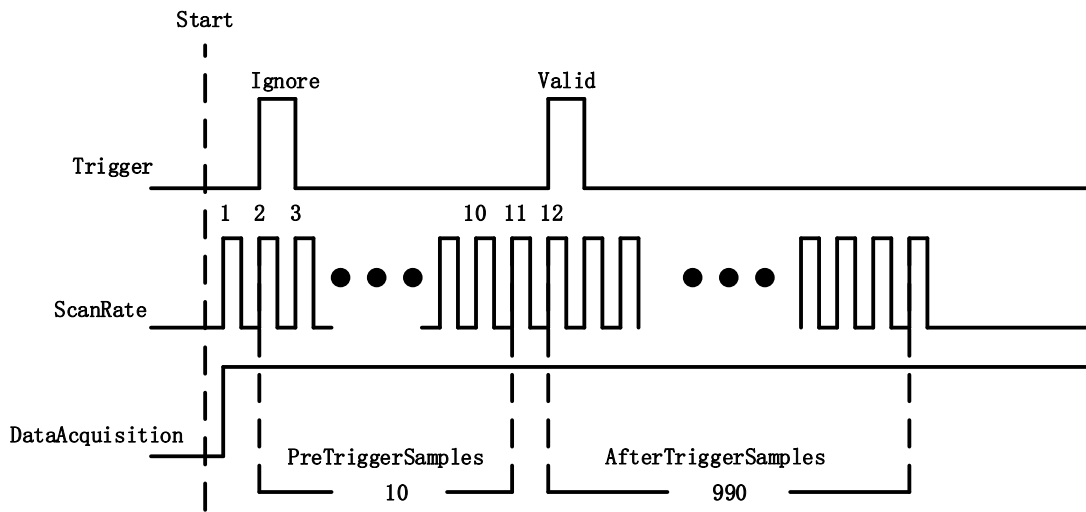


Figure 23 Reference Trigger

5.5.3 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 24.

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

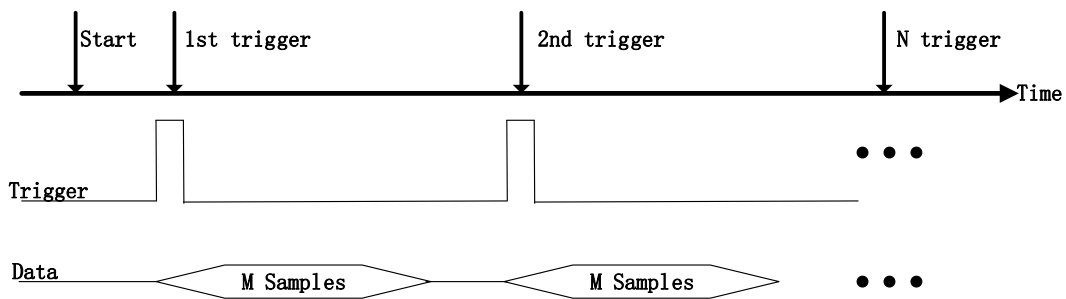


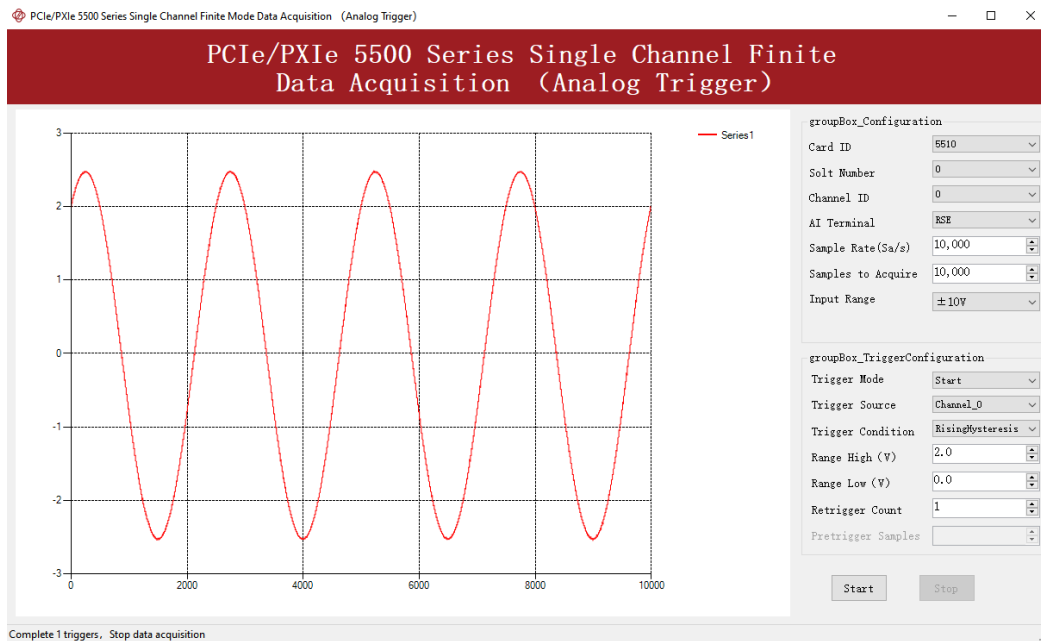
Figure 24 ReTrigger

Learn by Example 5.5

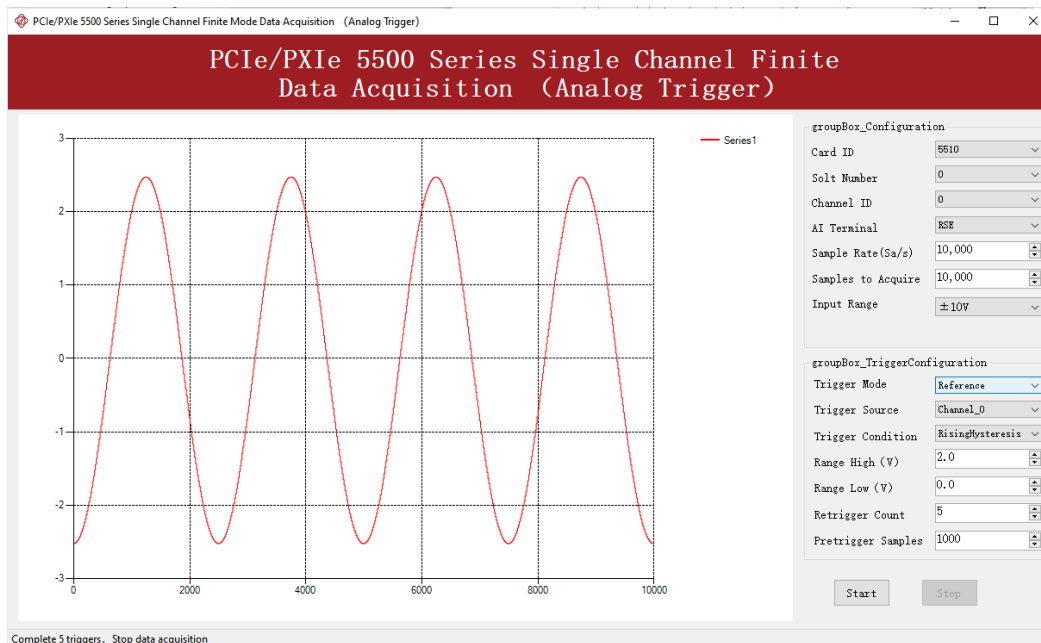
- Connect the signal source's positive terminal to PCIe-5500PCIe/PXIe-5500 AI Ch0 (AI0+, Pin#68), the negative terminal to the ground (AI_GND, Pin#67) as shown in Figure 1 and Figure 2. (AI0+, AI_GND) consists of an RSE input.
- Set a sinewave signal ($f=4\text{Hz}$, $V_{pp}=5\text{v}$).
- Open **Analog 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
<input type="button" value="Start"/> <input type="button" value="Stop"/>	

- You can use three different kinds of triggers in this program as mentioned in **5.5**. *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:



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



- 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

-
- It shows the acquisition process through every trigger signal.

5.6 AO Operations

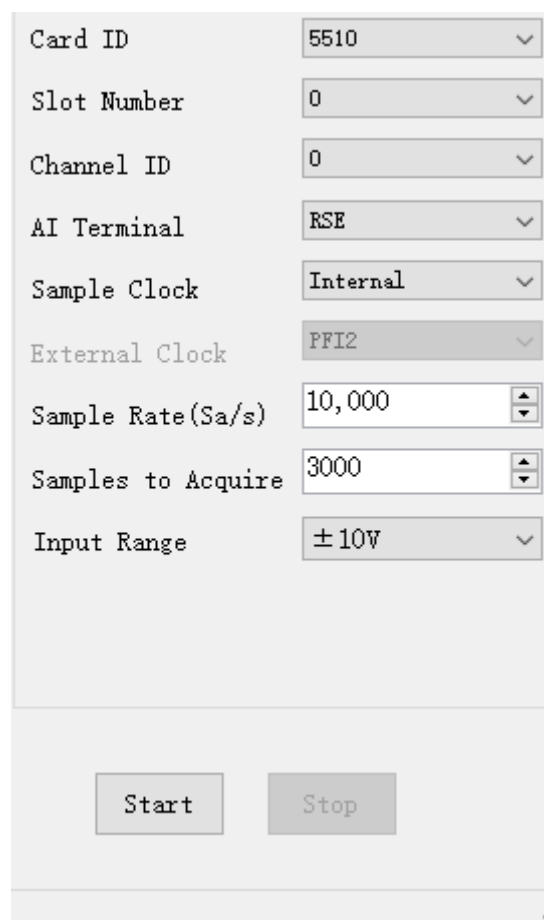
The PCIe/PXIe-5500 AO provides 16-bit simultaneous outputs. The analog output has three modes of operation: finite output, continuous acyclic output, and continuous cycle output.

5.6.1 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 5.6.1

- 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

- 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

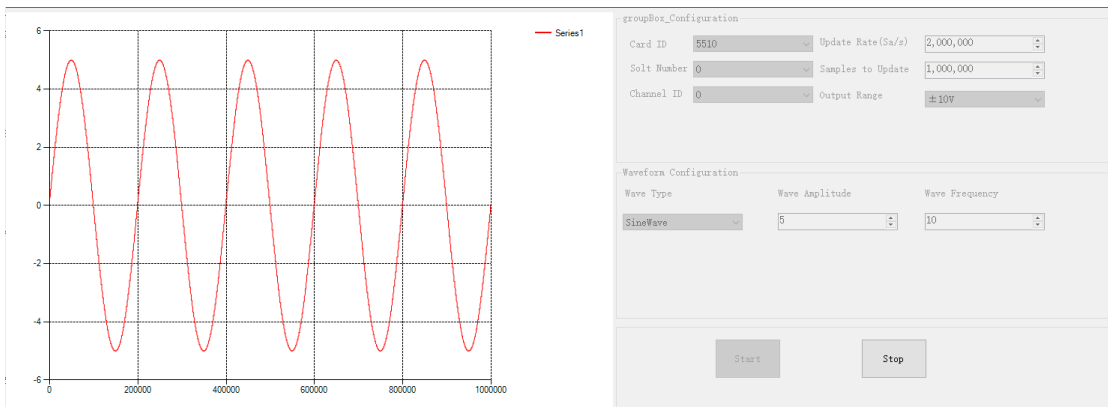
Waveform Configuration

Wave Type	Wave Amplitude	Wave Frequency
SineWave	5	10

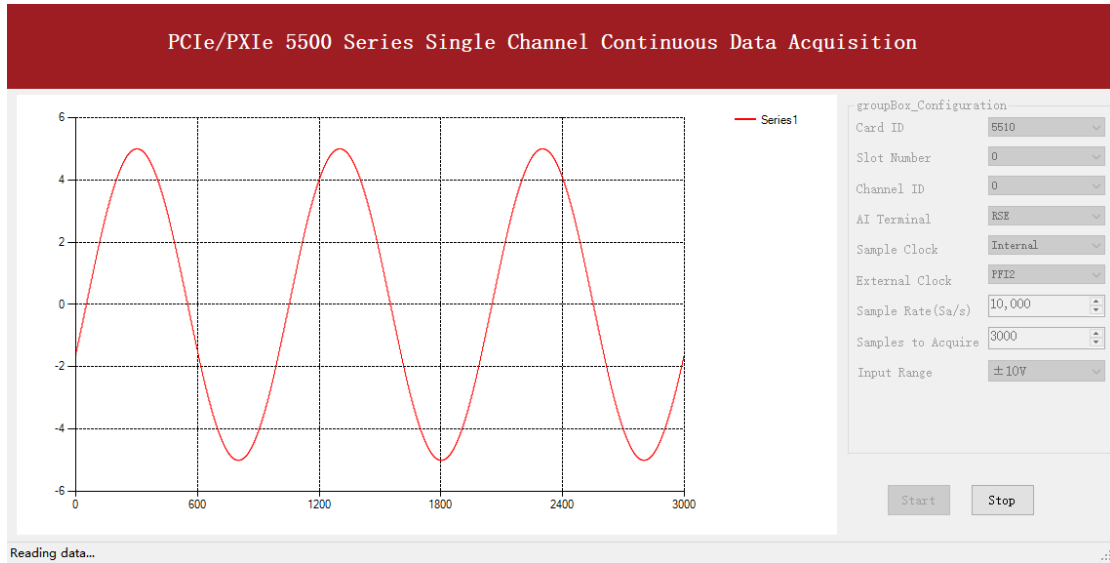
Start

Stop

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



- And the received signal is shown below.



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

5.6.2 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 5.6.2

- 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

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

Card ID: 5510 Update Rate(Sa/s): 1,000,000

Solt Number: 0 Output Range: ±10V

Channel ID: 0

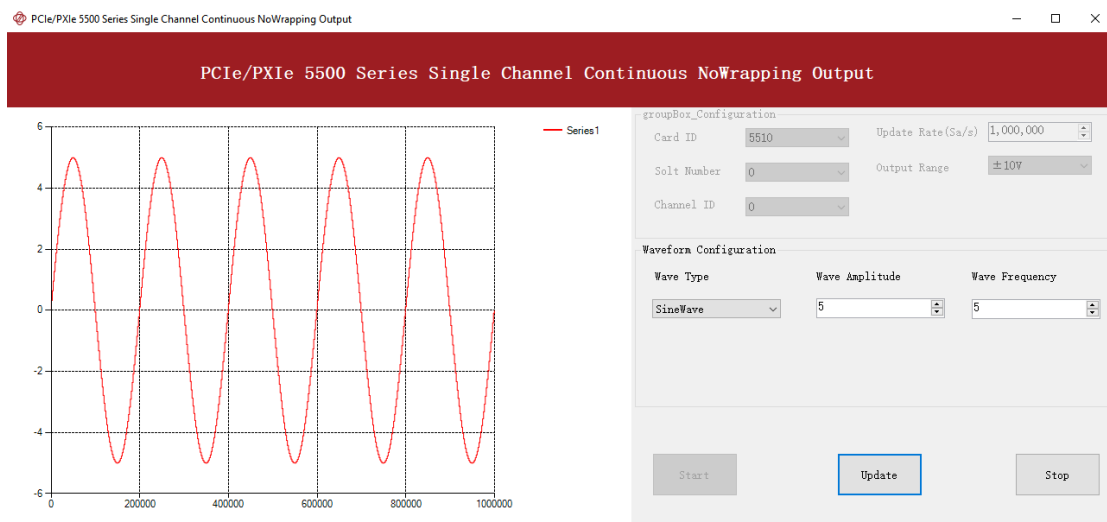
Waveform Configuration

Wave Type	Wave Amplitude	Wave Frequency
SineWave	5	10

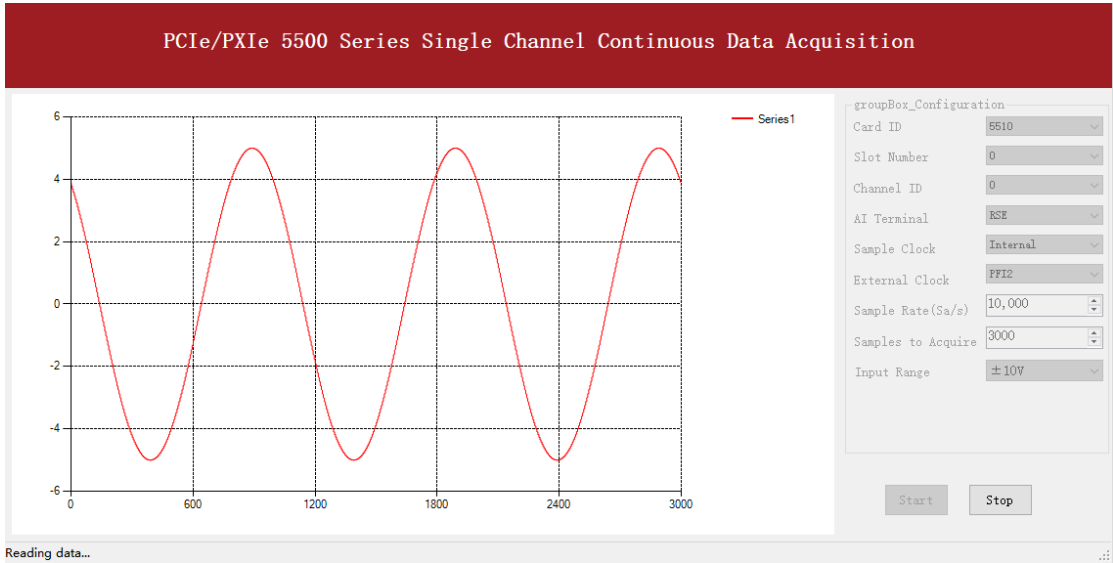
Start Update Stop

➤ 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.

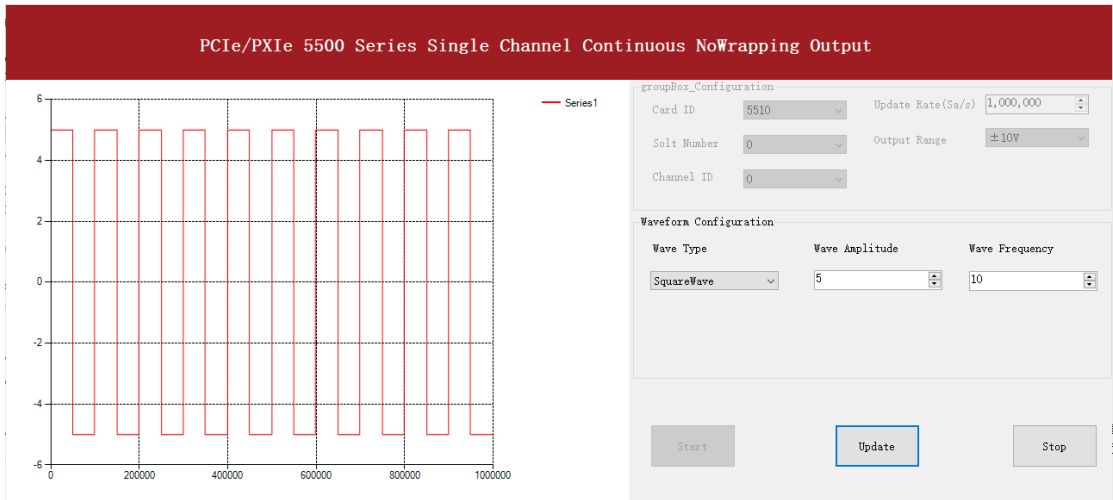
■ Click **Start** to generate a sinewave first. The result is shown below.



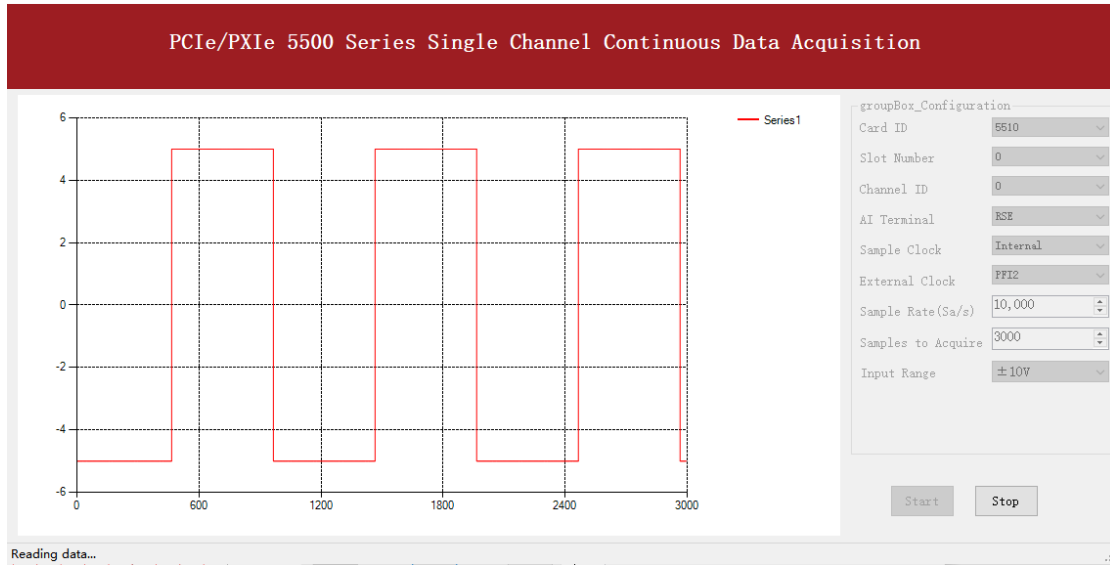
■ And the received signal is shown below.



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



- And the received signal is shown below.



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

5.6.3 Continuous Wrapping Output

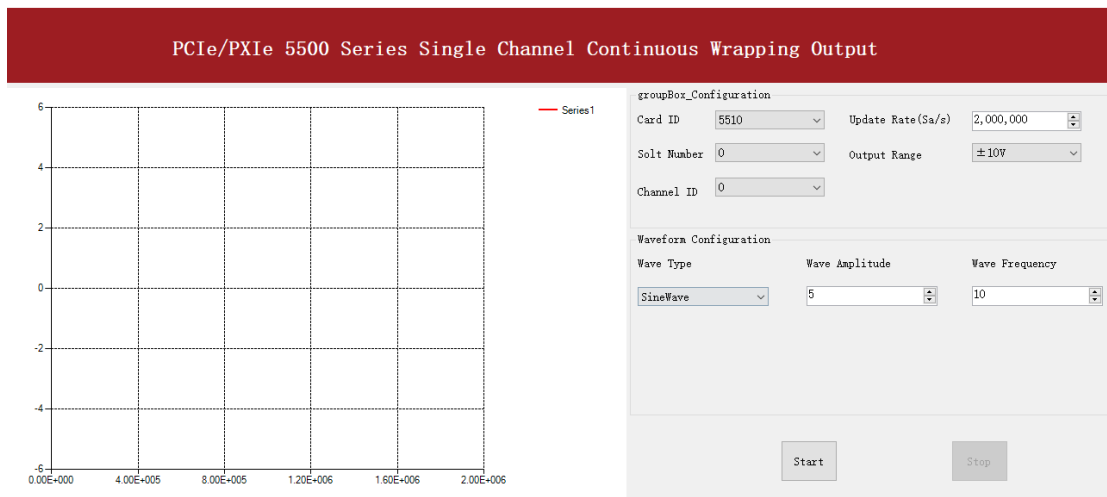
The continuous loop output first writes a piece of data before starting the AO. After the AO starts, the board will repeatedly output this data until user sends a stop command.

Learn by Example 5.6.3

- 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

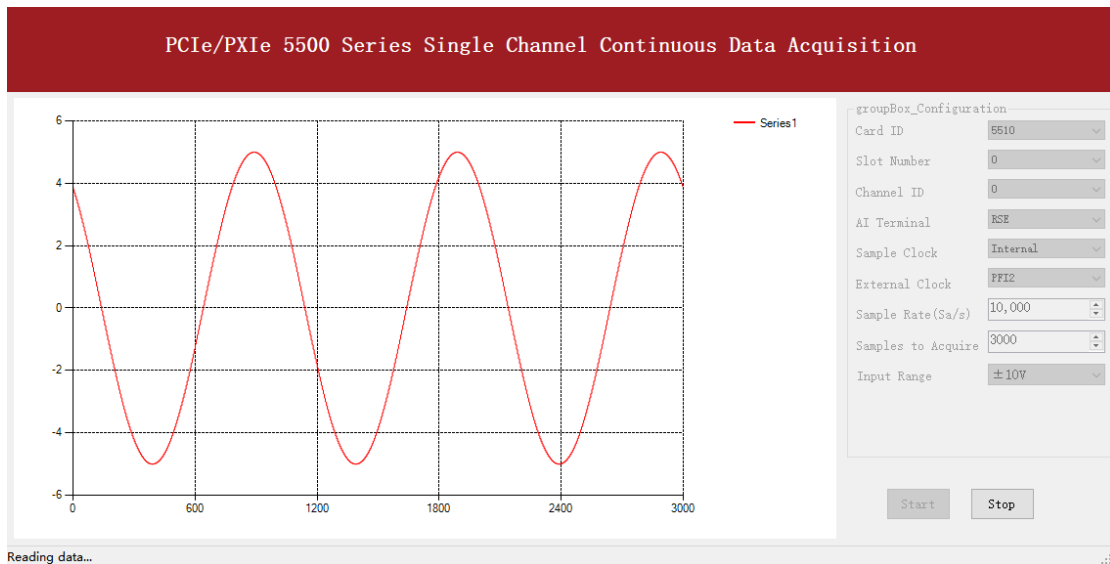
- Click **Start** to start the data acquisition.
- Open **Analog Output-->Winform AO Continuous Wrapping**, set the numbers as shown.



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



■ And the received signal is shown below.



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

5.7 Digital I/O Operations

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

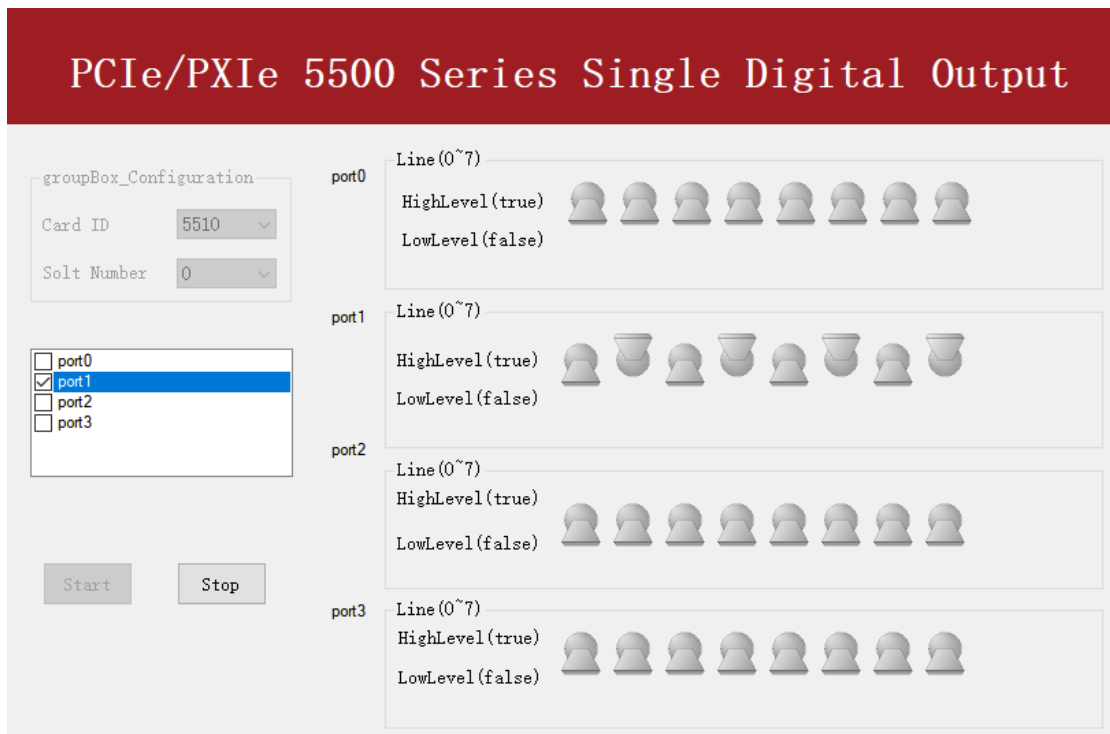
5.7.1 Static DI/DO

Programmable I/O supports static TTL, 6 ports (0,1,2,3,4,5) which is 48 channel digital I/O. User can poll these I / O information through software polling.

Learn by Example 5.7.1

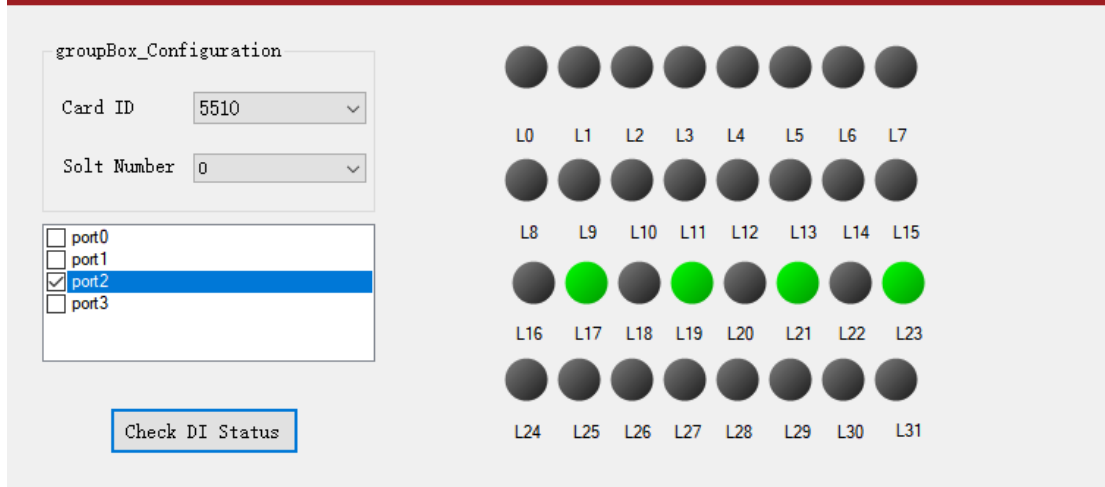
■ In this example PCIe/PXIe-5500 outputs a digital signal by its DO function and reads it back by its DI function.

- Connect Connector1 of PCIe/PXIe-5500 to the TB-68 terminal block according to **错误!未找到引用源。**
- Connect Port 1/Line 0~7 (P1.0~P1.7) to Port 2/Line 0~7 (P2.1~2.7). PCIe/PXIe-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 in High-Level positions, make sure all other lines are in Low-Level positions. Click **Start** to generate the High-Levels as shown.



- 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.

PCIe/PXIE 5500 Series Single Digital Signal Input



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

5.7.2 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.

5.8 Counter Input Operations

The PCIe/PXIE-5500 has four or two identical 32 bits timer/counters.

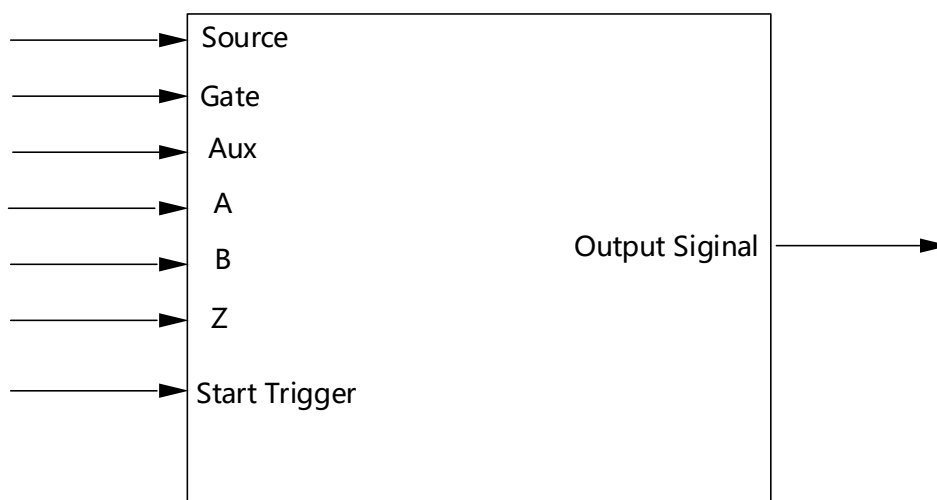


Figure 25 Counter Terminal

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

5.8.1 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 26.

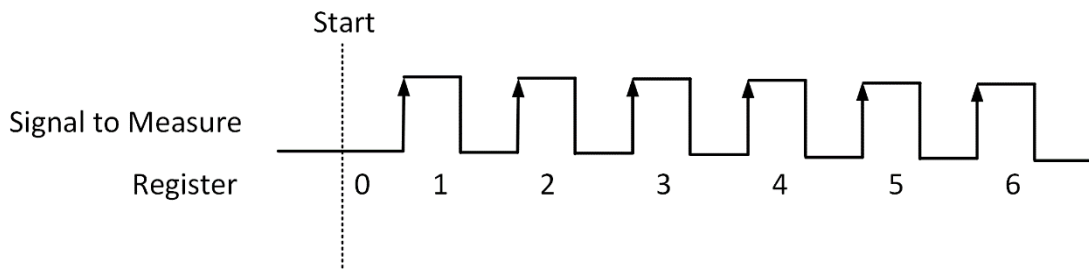


Figure 26 Simple Edge Counting in Single Mode

2) Finite/Continuous Mode with Explicit 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 27.

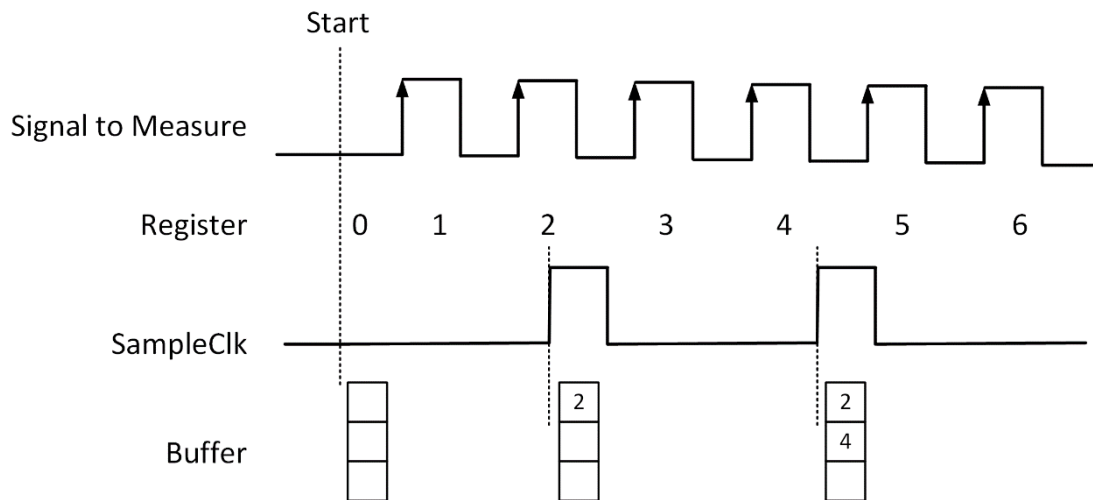


Figure 27 Buffered Edge Counting with Explicit Sample Clock

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 28.

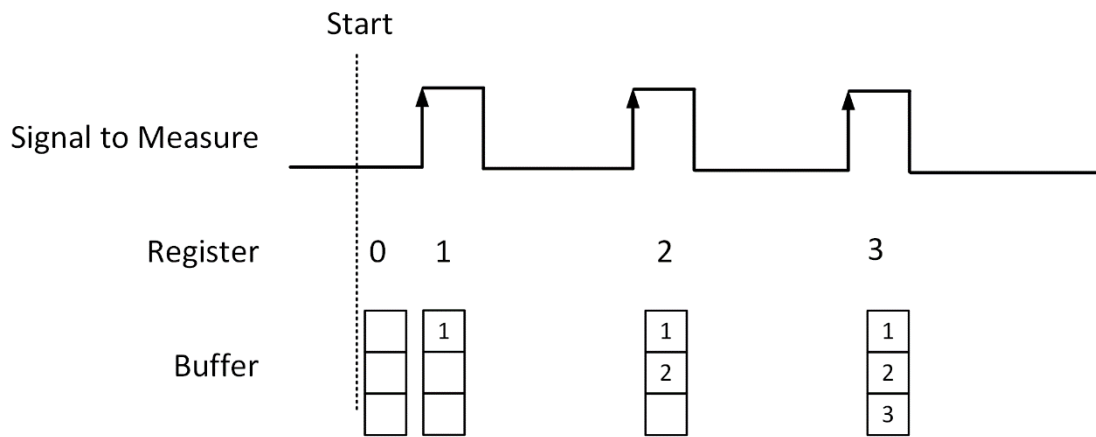


Figure 28 Simple Edge Counting with Implicit SampleClk

Counting Direction

User can control the counting direction through software configuration or by an input signal with Gate 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 29.

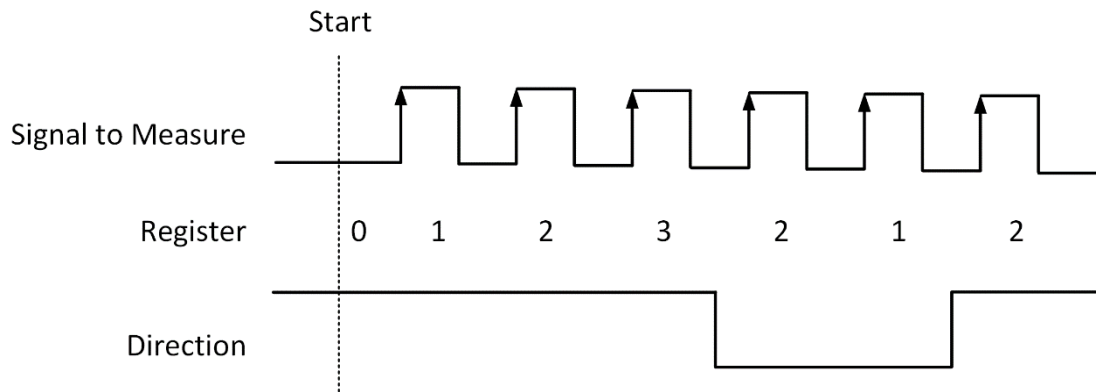


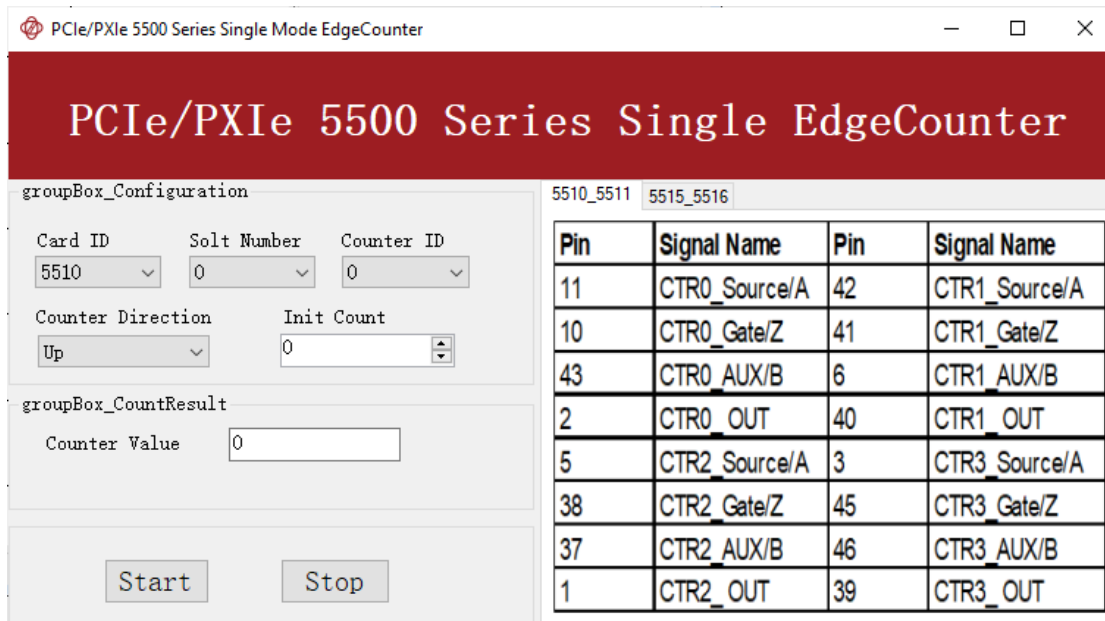
Figure 29 Count Direction

Learn by Examples 5.8.1

- 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 1 and Figure 2. (CTR0_Source, DGND) consists of an edge counting counter input and they share the same ground.
- Set a sinewave signal (f=1Hz, Vpp=5v).

Single Mode

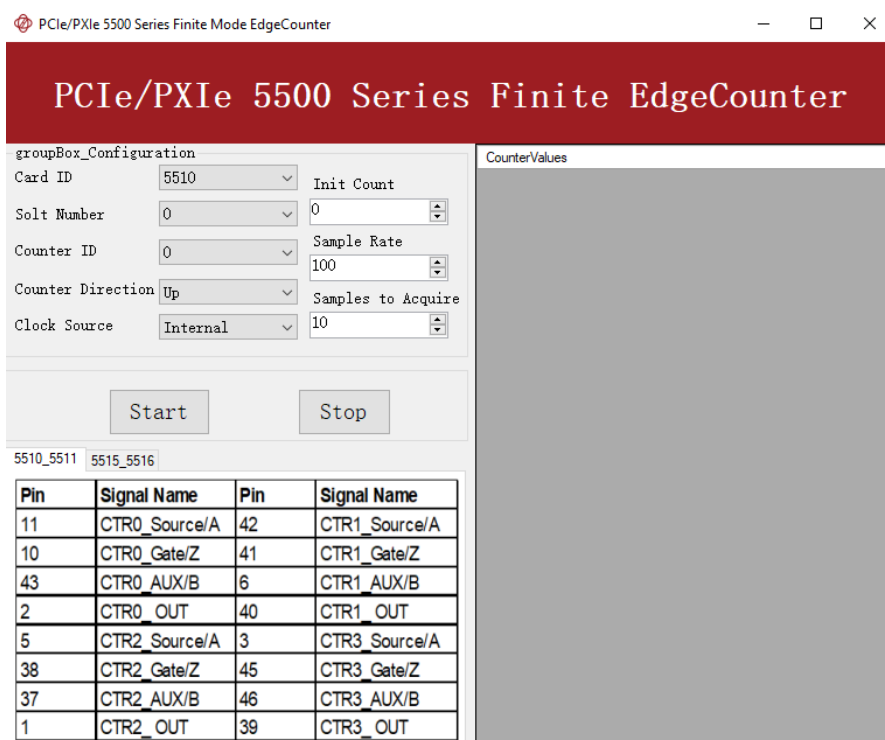
- Open **Counter Input-->Winform CI Single EdgeCounter**, set the following numbers as shown:



- *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 EdgeCounter**, set the following numbers as shown:



- 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/PXIe-5500 *Explicit* and *Implicit*: as mentioned in 错误!未找到引用源。 . *Explicit* includes two modes: **internal** and external. PCIe/PXIe-5500 only supports the **internal** mode. This example uses **Internal** implicit 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

- 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

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

5.8.2 Pulse Measurement

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

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 30.

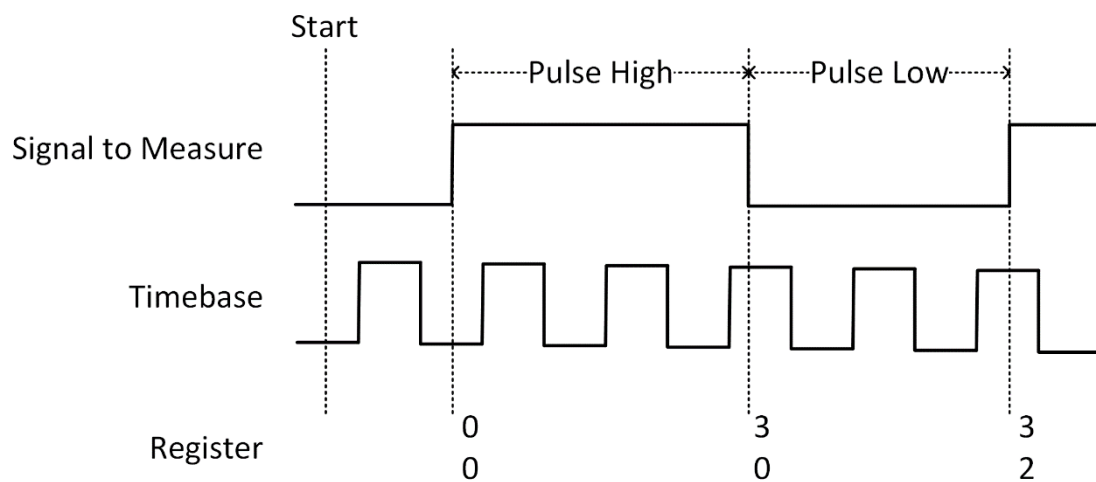


Figure 30 Pulse Measurement in Single Mode

2) Finite/Continuous Mode with Explicit Sample Clock

The count value of the duration of the high or low level is stored into the buffer on each rising edge of the sample clock, as shown in Figure 31.

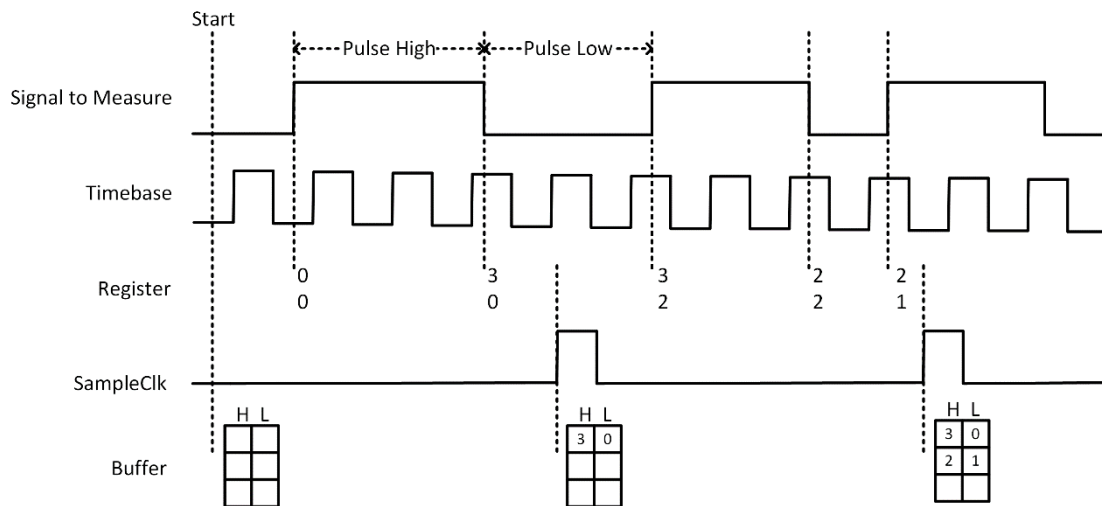


Figure 31 Pulse Measurement with Explicit SampleClk

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 32.

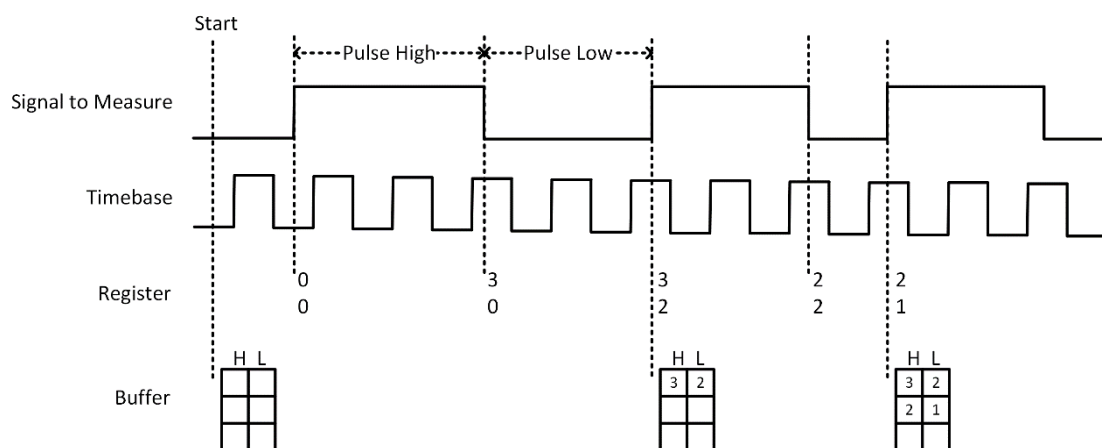


Figure 32 Pulse Measurement with Implicit SampleClk

Learn by Examples 5.8.2

- 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 1 and Figure 2. (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:

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

➤ The table in the sample program is a connection diagram for your convenience.

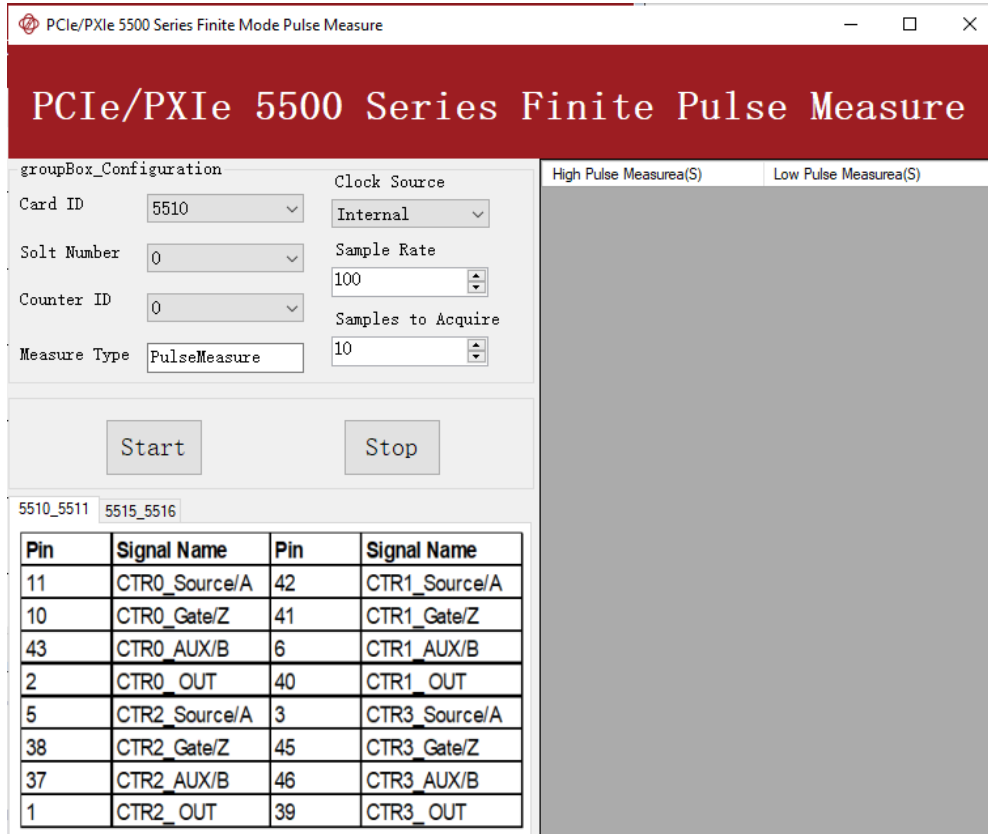
- Click **Start** to start measuring the pulses. The result is shown by **High Pulse Measure(S)** and **Low Pulse Measure(S)**:

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

- 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:



- 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

- 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 5.8.1 Finite/Continuous Mode** about the difference between *Explicit* and *Implicit*.

5.8.3 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 5.8.2 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 JY5211CITask.Mode to CIMode.Single.

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

Between every two rising edges of the sample clock, the counter counts the he numbers of full periods ($T1$) of the signal to measure, and the number of rising edges

of timebase (T_2) during those full periods. These two values are stored into the buffer on each rising edge of the sample clock, as shown in Figure 33.

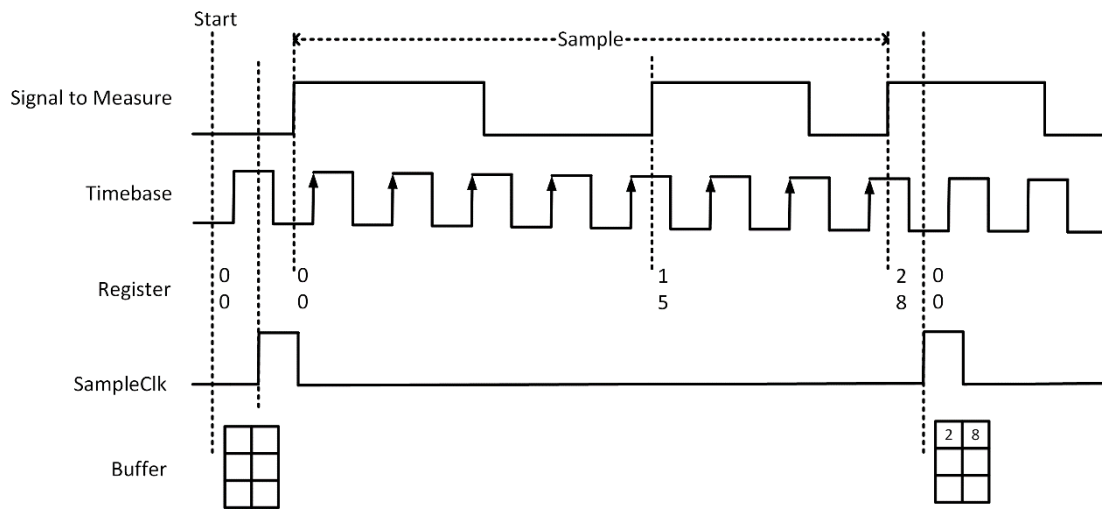


Figure 33 Frequency Measurement with Explicit Sample Clock

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.

$$f_x = f_{base} \times \frac{T_1}{T_2}$$

3) Finite/Continuous Mode with Implicit Sample Clock

Frequency Measurement with implicit sample clock is actually using Pulse Measurement internally. refer to chapter 5.8.2 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 5.8.3

■ 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 1 and Figure 2. (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

- 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**.

PCIe/PXIE 5500 Series Continuous Frequency Measure

groupBox_Configuration

Card ID: Sample Rate:

Solt Number: Samples to Acquire:

Counter ID: Measure Type:

Clock Source:

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

FrequencyMeasurea(Hz)

0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0

- The table in the sample program is a connection diagram for your convenience.
- *Explicit* 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
50.0009281422286
50.0009343924615

5.8.4 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 5.8.3 for more information.

Learn by Examples 5.8.4

- 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 1 and Figure 2. (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)**:

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

- 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:

PeriodMeasurea(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

- 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.

5.8.5 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 34.

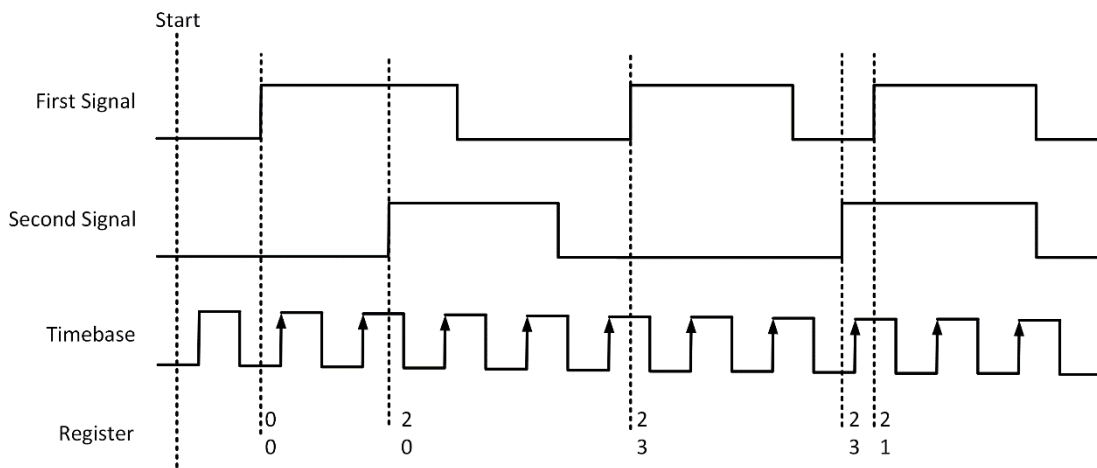


Figure 34 Two-Edge Separation in Single Mode

2) Finite/Continuous Mode with Explicit 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 35.

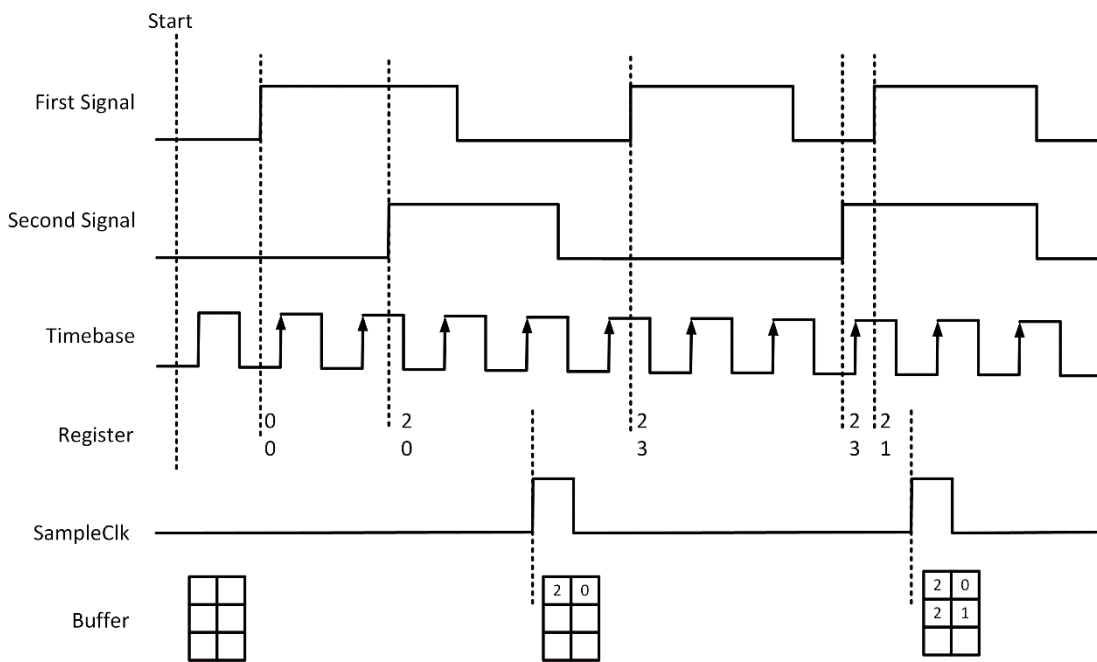


Figure 35 Two-Edge Separation with Explicit Sample Clock

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 36.

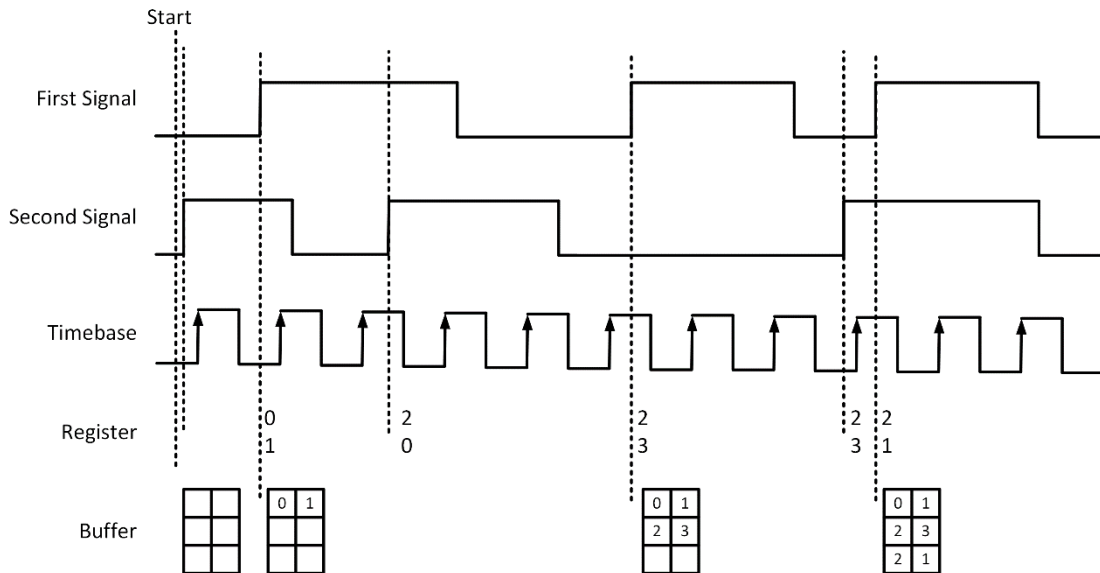


Figure 36 Two-Edge Separation with Implicit Sample Clock

Learn by Examples 5.8.5

- 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 1 and Figure 2. (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 sinewave signal ($f=1\text{Hz}$, $\text{Phase}=0^\circ$) and a squarewave signal ($f=1\text{Hz}$, $\text{Phase}=135^\circ$).

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

- 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:

PCIe/PXIe 5500 Series Finite TwoEdgeSeparation Measure

groupBox_Configuration

Card ID: Measure Type:

Solt Number: Sample Rate:

Counter ID: Samples to Acquire:

Clock Source:

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

First to Second Measure(S)	Second to First Measure(S)
0	0
0.37548128	0.62450008
0.37548127	0.624500085
0.37548127	0.624500085
0.37548128	0.62450009
0.37548127	0.624500085
0.37548128	0.624500085
0.37548127	0.624500085
0.37548127	0.624500085
0.37548127	0.624500085
0.37548128	0.624500085

- 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.

5.8.6 Quadrature Encoder

The quadrature encoder includes three encoding type: 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 37.

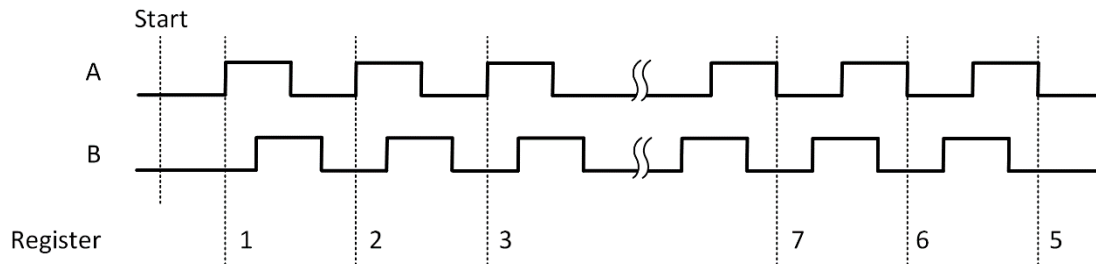


Figure 37 Quadrature Encoder x1 Mode

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 38.

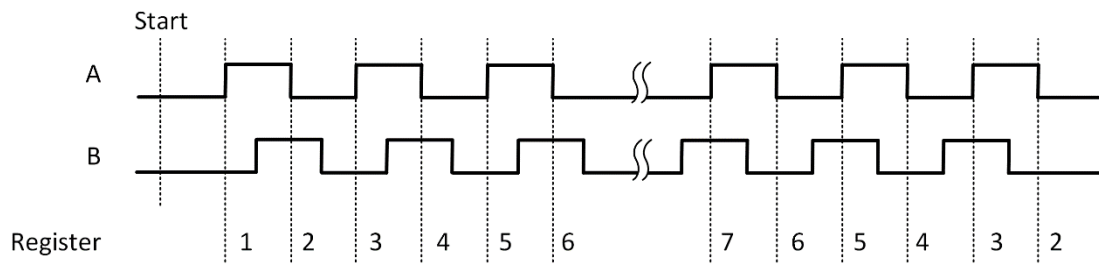


Figure 38 Quadrature Encoder x2 Mode

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 39.

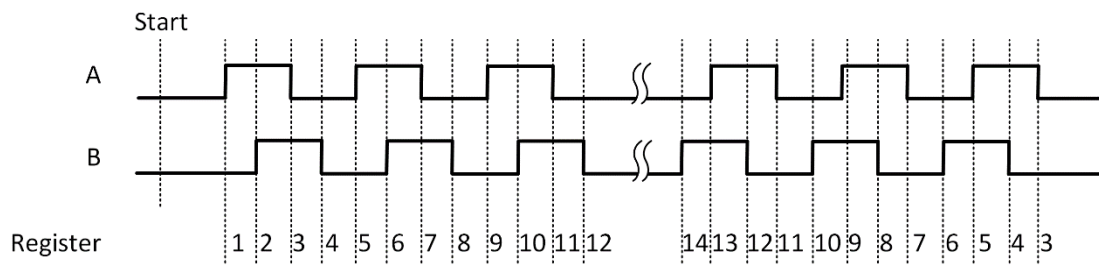


Figure 39 Quadrature Encoder x4 mode

Channel Z Behavior

The reload phase is 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 26.

To configure the counter to work in this mode, set JY5211CITask. Mode to CIMode.Single.

2) Finite/Continuous Mode with Explicit Sample Clock

The count value is stored into the buffer on each rising edge of the sample clock, as shown in Figure 40.

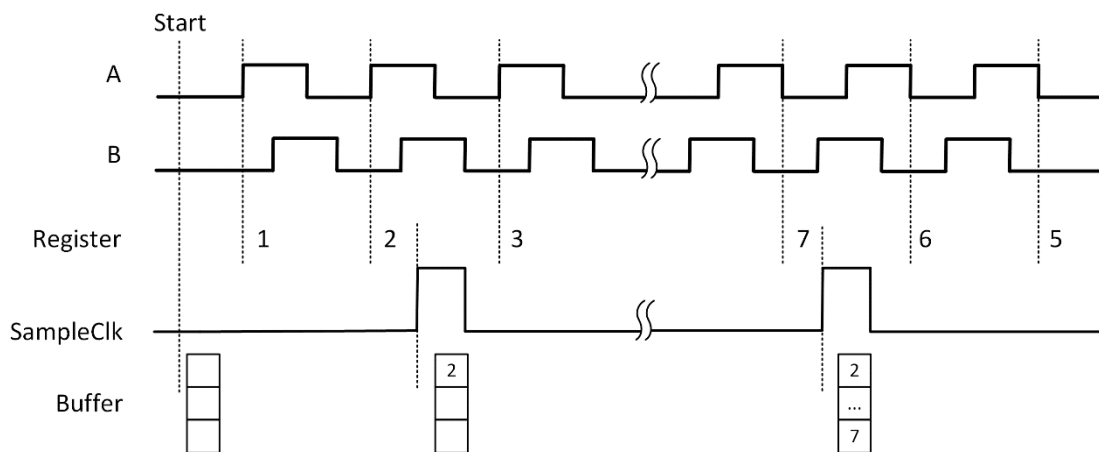


Figure 40 Quadrature Encoder x4 with Explicit Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

The count value is stored into the buffer every time it changes, the as shown in Figure 41.

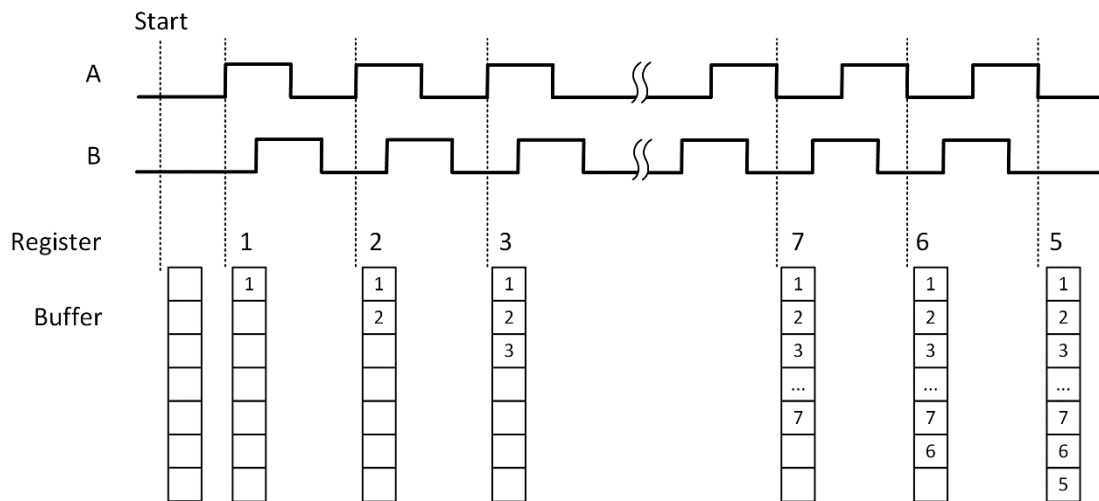


Figure 41 Quadrature Encoder x4 with Implicit Sample Clock

Learn by Examples 5.8.6

- 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 1 and Figure 2. (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 ($f=10\text{Hz}$, $\text{Phase}=90^\circ$) and a squarewave signal ($f=10\text{Hz}$, $\text{Phase}=0^\circ$).

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 5.8.6:

PCIe/PXIE 5500 Series Single QuadEncoder

groupBox_Configuration

Card ID: 5510 Slot Number: 0 Counter ID: 0

Encode Type: X1

groupBox_CountResult

CounterValue: 22

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

- 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**.

PCIe/PXIE 5500 Series Continuous QuadEncoder

groupBox_Configuration

Card ID: 5510 Sample Rate: 10

Solt Number: 0 Samples to Acquire: 10

Counter ID: 0

Encode Type: X1

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

1
2
3
4
5
6
7
8
9
10

- 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.

5.8.7 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 42.

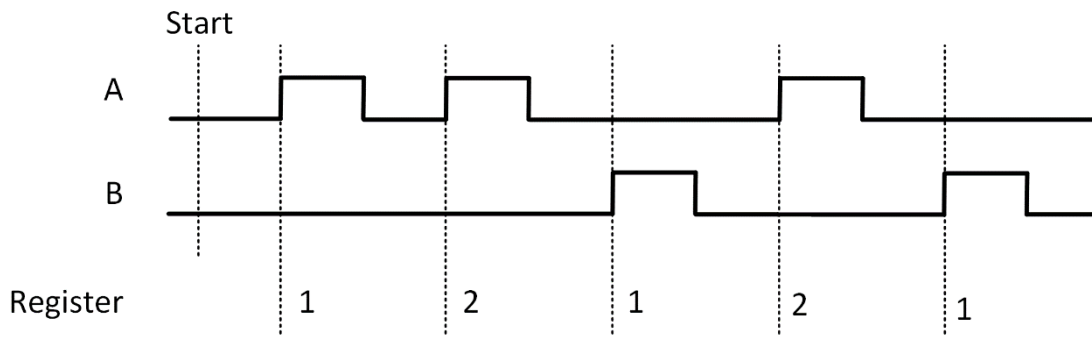


Figure 42 Two-Pulse Encoder in Single Mode

2) Finite/Continuous Mode with Explicit Sample Clock

The count value is stored into the buffer on each rising edge of the sample clock, as shown in Figure 43.

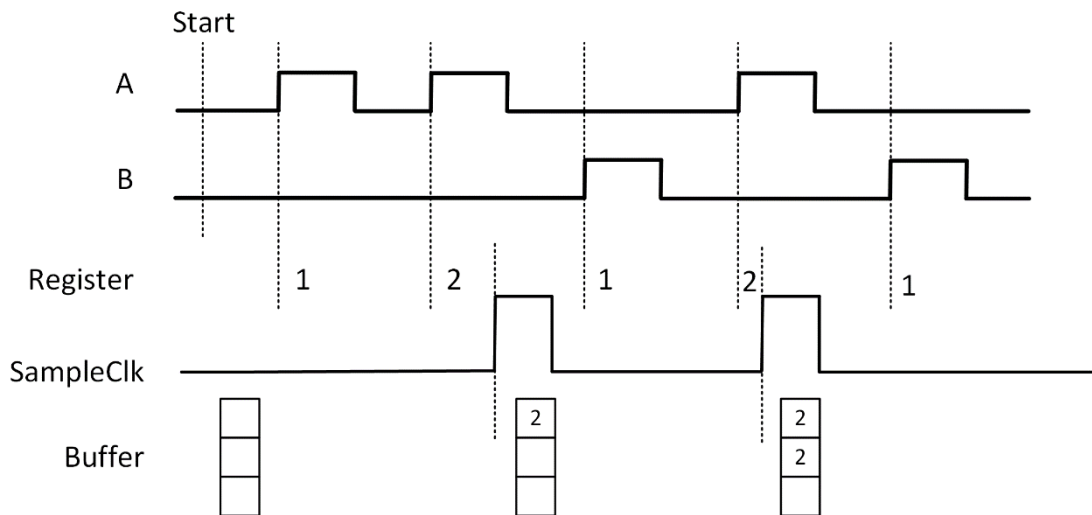


Figure 43 Two-Pulse Encoder with Explicit Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

The count value is stored into the buffer every time it changed, as shown in Figure 44.

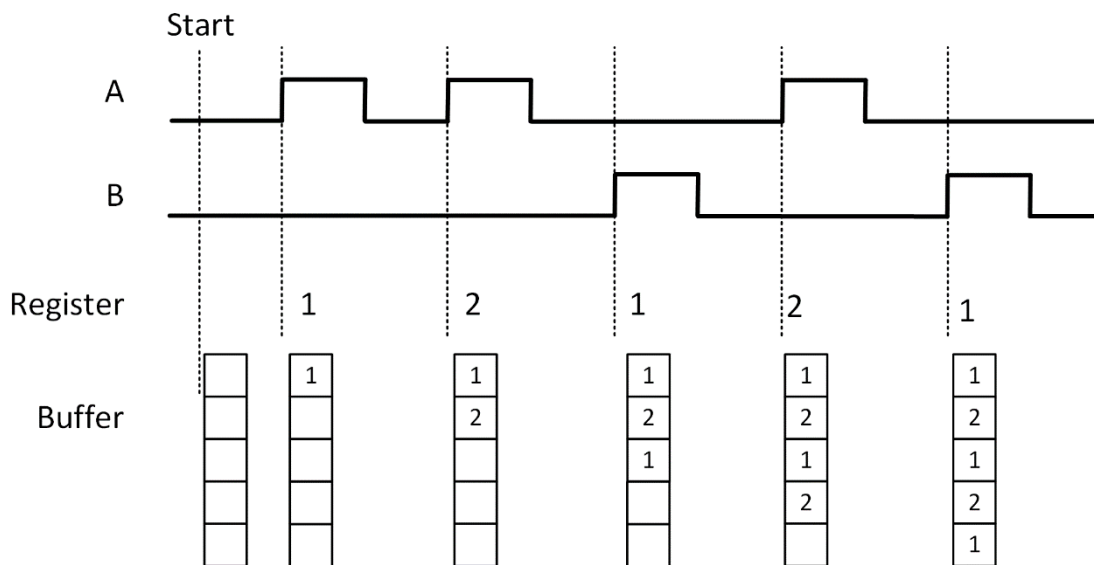


Figure 44 Two-Pulse Encoder with Implicit Sample Clock

Learn by Examples 5.8.7

- 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 1 and Figure 2. (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=40Hz) and a squarewave signal (f=40Hz).

Single Mode

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

PCIe/PXLe 5500 Series Single EncoderTwoPulse

groupBox_Configuration

Card ID: 5510 Solt Number: 0 Counter ID: 0

Encoder Type: EncoderTwoPulse

groupBox_CountResult

Counter Value: 0

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

- 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 [错误!未找到引用源。](#)
- 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.

PCIe/PXIE 5500 Series Finite TwoPulseEncoder

groupBox_Configuration

Card ID: Sample Rate:

Solt Number: Samples to Acquire:

Counter ID: Clock Source:

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

1
2
2
3
4
5
6
6
7
8

- 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 [错误!未找到引用源。](#).

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/PXLe 5500 Series Continuous TwoPulseEncoder

PCIe/PXLe 5500 Series Continuous TwoPulseEncoder

groupBox_Configuration

Card ID: Sample Rate:

Solt Number: Samples to Acquire:

Counter ID: Clock Source:

CounterValues

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

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

- 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 [错误!未找到引用源。](#).

5.9 Counter Output Operations

5.9.1 Single Pulse Output

The PCIe/PXLe-5500 timer/counter can output a single pulse with a specified pulse width. The timing diagram of the pulse output is shown in Figure 45.

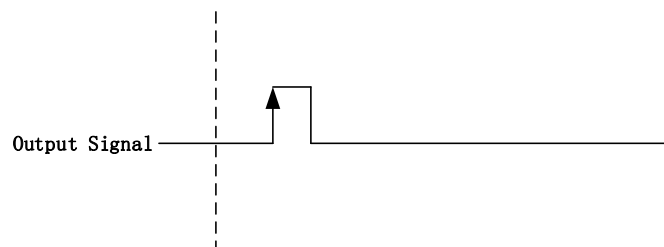


Figure 45 Single Pulse Output

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 5.9.1

- To see the signal that PCIe/PXle-5500 Counter Output generates, it is recommended to connect PCIe/PXle-5500 Counter Output (CTR0_OUT, Pin#2) to PCIe/PXle-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/PXle 5500 Series Single Pulse Generation

groupBox_Parameter

Card ID: 5510 Pulse Delay: 0

Solt Number: 0

Counter ID: 0

groupBox_PulseParameter

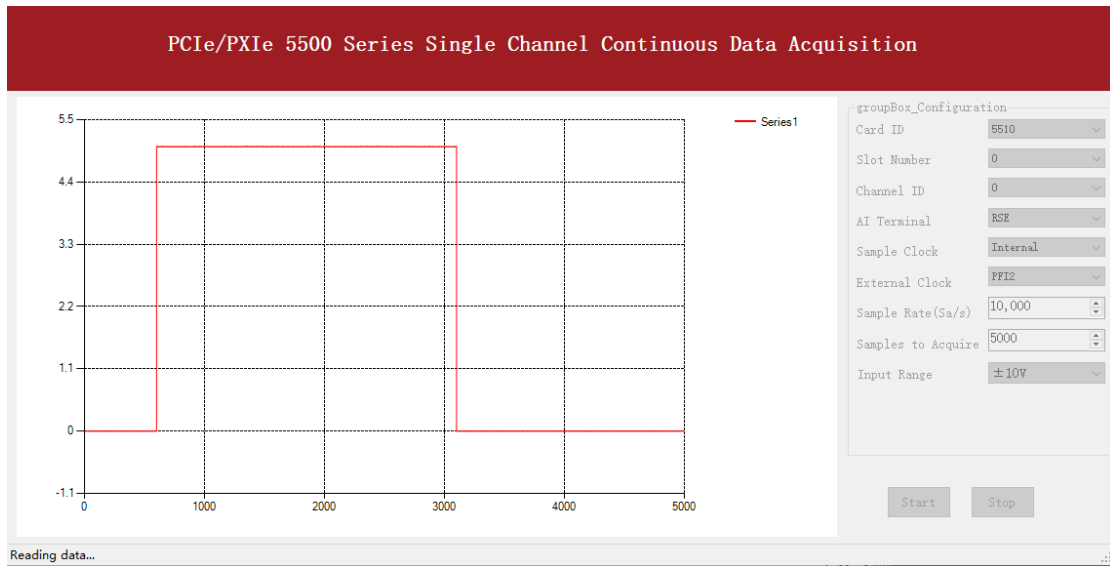
outputPulse Type: DutyCycleFrequency Frequency: 2.000

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

- 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 5.2 to configure an analog input to receive the signal from Counter Output.
- Click **Start** to generate a single pulse as shown.



5.9.2 Finite Pulse Output

The pulse output timing is as shown in Figure 46.

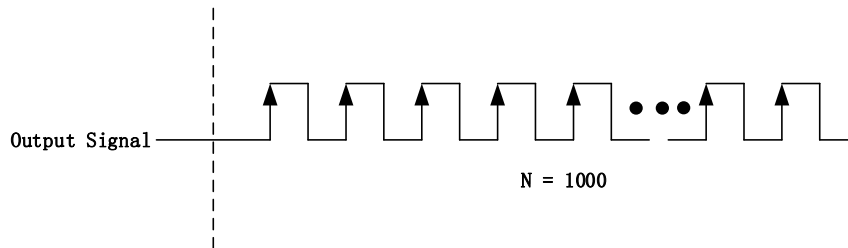


Figure 46 Finite Pulse Output

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.

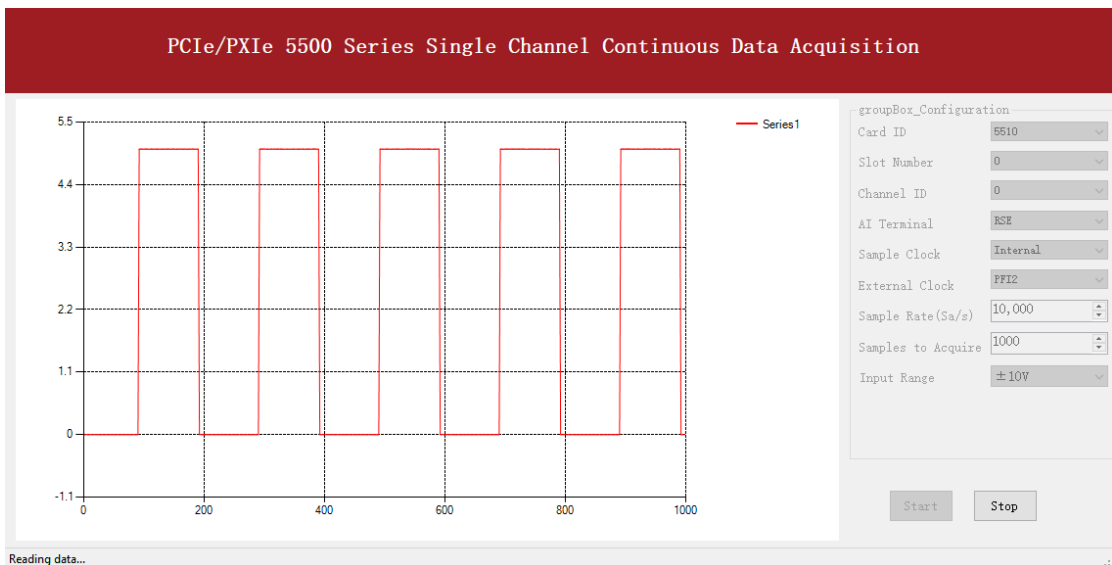
Learn by Example 5.9.2

- 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:

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

- 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 5.2 to configure an analog input to receive the signal from Counter Output.
- Click **Start** to generate the pulse shown below.



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

5.9.3 Continuous Pulse Output

The pulse output timing is shown in Figure 47 below.

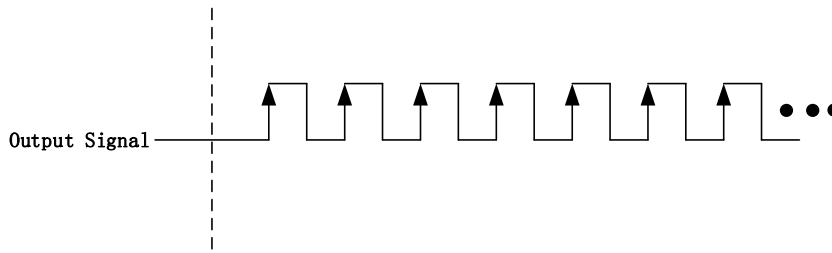


Figure 47 Continuous Pulse Output

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 5.9.3

- To see the signal that PCIe/PXle-5500 Counter Output generates, it is recommended to connect PCIe/PXle-5500 Counter Output (CTR0_OUT, Pin#2) to PCIe/PXle-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/PXle 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

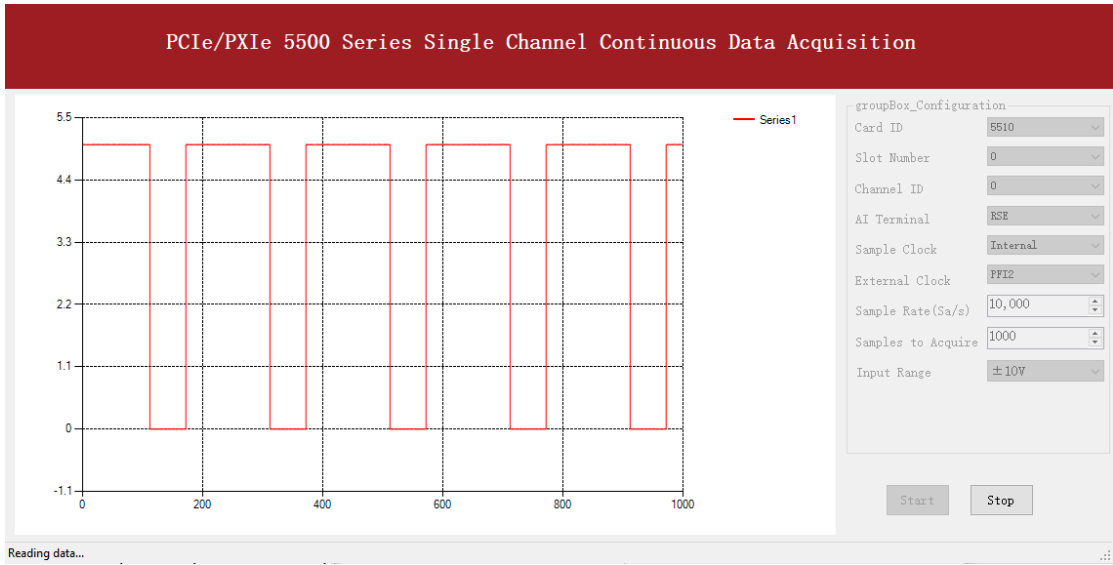
Idle State: LowLevel

Duty Cycle: 0.500

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

- 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.



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

5.10 System Synchronization Interface (SSI)

SSI (System Synchronization Interface) provides the PCIe-5500 timing synchronization between multiple cards. SSI could replace the PXITrigger Bus in PXIe cards. In the PCIe-5500, we designed a bi-directional SSI I/O to provide flexible connection between cards and allow one SSI master to output the signal and up to three slaves to receive the SSI signal. Note that the SSI signals are designed for card synchronization only, not for external devices.

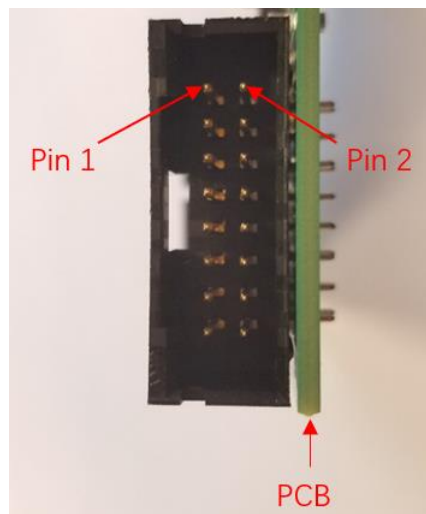


Figure 48 SSI Connector in PCIe-5500

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 19 SSI Connector Pin Assignment for PCIe-5500

5.11 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, when the board is inserted in slot 3, position 2 and 1 of the DIP switch turns to the ON position and the others to OFF. See below for details.

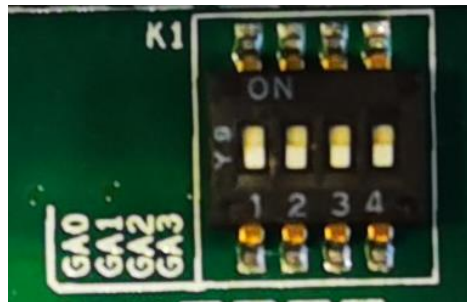


Figure 49 DIP Switch in PCIe-5500

	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 20 Relationship between switch position and slot number

6. 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.

7. 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.

7.1 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.

7.2 C++

JYTEK internally uses our C++ drivers to design the C# drivers. We recommend our customers to use C# drivers because C# platform deliver much better efficiency and performance in most situations. We also make our C++ drivers available. 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.

7.3 LabVIEW

LabVIEW is a software product from National Instruments. JYTEK provides LabVIEW interface to PCIe/PXIe-5500 boards. You can down load the LabVIEW drivers from our website. While JYTEK does not support LabVIEW applications, we may recommend you to a third party who can assist you to interface your LabVIEW with our PCIe/PXIe-5500 boards. We can also recommend you if you want to convert your LabVIEW applications to C# based applications.

8. About JYTEK

8.1 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.

8.2 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.

8.3 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.

8.4 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.

8.5 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 2-year warranty. For technical consultation, pre-sale and after-sales support, please contact JYTEK of your country.

9. Appendix 1 Common Analog Measurement Issues

9.1 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.

9.2 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.

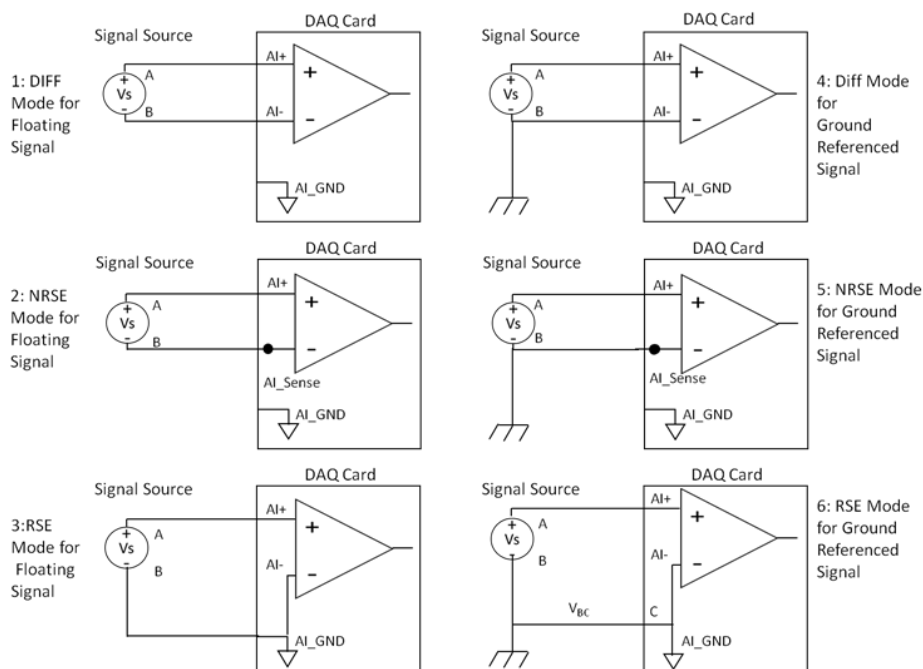


Figure 50 Six Measurement Scenarios

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.

9.3 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 100K Ω . At this level, R has very little impact on the measurement.

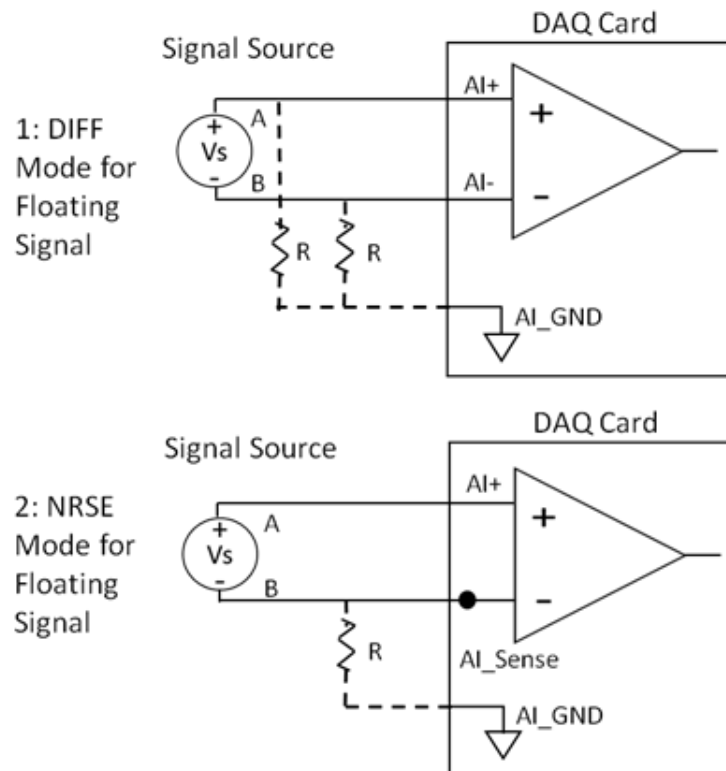


Figure 51 Using Resistor to Reduce Common Mode Voltage Effect

10. 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/PXle-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 JYTEK Co., Ltd.

Address: Building 3, No. 300 Fangchun Road, Pudong New District, Shanghai.

Post Code: 201203

Tel: 021-5047 5899

Website: www.jytek.com