JY5500 Series

Family of Multi-functional Data Acquisition Boards

User Manual

User Manual Version: V2.5.5

Revision Date: July 12, 2023

Table of Contents

1.Introduction 1
1.1 Overview 1
1.2 Main Features 1
1.3 Abbreviations 2
1.4 Learn by Example 2
2.Hardware 5
2.1 Analog Input 5
2.1.1 General Specifications 5
2.1.2 DC Accuracy 5
2.1.3 AC Accuracy 11
2.1.4 Dynamic Performance 12
2.2 Analog Output 13
2.2.1 General Specifications 13
2.2.2 Basic AO Accuracy 14
2.2.3 Additional AO Accuracy Adjustment 15
2.3 Digital IO Specifications 16
2.4 Counter/Timer Specifications 16
2.5 PFI Specifications 16
2.6 Additional Specifications 17
2.6.1 Front Panel and Pin Definition 17
2.6.2 5510/5511 Connector Pin Definition 18
2.6.3 5510/5511 Counter Pin Definition (Connector 0) 19
2.6.4 5515/5516 Connector Pin Definition 20
2.6.5 5515/5516 Counter Pin Definition 21
2.6.6 Power Specification 22
2.6.7 Physical and Environment 22
3.Software 23
3.1 System Requirements 23
3.2 System Software 23
3.3 C# Programming Language 24
3.4 JY5500 Series Hardware Driver 24
3.5 Install the SeeSharpTools from JYTEK 24
3.6 Running C# Programs in Linux 25
4.Operating JY5500 26
4.1 Quick Start 26
4.2 Data Acquisition Methods 26
4.2.1 Continuous Acquisition 29
4.2.2 Finite Acquisition 29
4.2.3 Single Point Acquisition 29
4.2.4 Record Acquisition 29

4.3 Analog Input Terminal Type 29
4.3.1 DIFF Mode 30
4.3.2 RSE Mode 32
4.3.3 NRSE Mode 32
4.4 Trigger Source 35
4.4.1 Immediate trigger 35
4.4.2 Software Trigger 36
4.4.3 External Analog Trigger 37
4.4.4 External Digital Trigger 42
4.5 Trigger Mode 44
4.5.1 Start Trigger 44
4.5.2 Reference Trigger 45
4.5.3 ReTrigger 46
Learn by Example 4.5 46
4.6 AO Operations 49
4.6.1 Finite Output 49
4.6.2 Continuous NoWrappping Output 52
4.6.3 Continuous Wrapping Output 56
4.7 Digital I/O Operations 59
4.7.1 Static DI/DO 59
4.7.2 Dynamic DI/DO 60
4.8 Counter Input Operations 63
4.8.1 Edge Counting 64
4.8.2 Pulse Measurement 69
4.8.3 Frequency Measurement 73
4.8.4 Period Measurement 76
4.8.5 Two-Edge Separation 78
4.8.6 Quadrature Encoder 83
4.8.7 Two-Pulse Encoder 87
4.9 Counter Output Operations 92
4.9.1 Single Pulse Output 92
4.9.2 Finite Pulse Output 94
4.9.3 Continuous Pulse Output 96
4.10 System Synchronization Interface（SSI）for PCIe Modules 98
4.11 DIP Switch in PCIe-5500 98
5.Calibration 100
6.Using JY5500 in Other Software 101
6.1 Python 101
6.2 C++ 101
6.3 LabVIEW 101
7.About JYTEK 102

7.1 JYTEK China 102
7.2 JYTEK Hardware Products 102
7.3 JYTEK Software Platform 102
7.4 JYTEK Warranty and Support Services 103
8.Appendix 1 Common Analog Measurement Issues 104
8.1 Floating Signals and Ground Referenced Signals 104
8.2 Differential, NRSE, RSE Modes 104
8.3 Reducing the Common Mode Voltage Effect 105
8.4 DC, AC and DSA Mode 106
8.4.1 DC Mode 106
8.4.2 AC Mode 107
8.4.3 DSA Mode 107
9.Statement 108

Figure 1 JYPEDIA Information 3
Figure 2 PCIe-5500 experiment 3
Figure 3 TB-68 Terminal Block 4
Figure 4 Averaging in DC Measurements 6
Figure 6 PXIe/PCIe 5510 Front Panel 17
Figure 7 Sample Rate and Internal AD Conversion 27
Figure 8 Continuous MultiChannel Paraments 28
Figure 9 MultiChannel Continuous Acquisition 28
Figure 10 Differential Mode for Grounding Signals 30
Figure 11 Choose Differential In AI Terminal 31
Figure 12 RSE Mode for Floating Signals 32
Figure 13 NRSE Mode for Grounding Signals 33
Figure 14 Choose NRSE In AI Terminal 34
Figure 15 Immediate trigger Paraments 35
Figure 16 Software trigger Paraments 36
Figure 17 Software trigger Acquisition 37
Figure 18 Rising Slope Trigger 38
Figure 19 Falling Slope Trigger 38
Figure 20 Hysteresis with Rising Slope Trigger 39
Figure 21 Hysteresis with Falling Slope Trigger 39
Figure 22 Entering Window Trigger 40
Figure 23 Leaving Window Trigger 40
Figure 24 Analog Trigger Paraments 41
Figure 25 Waiting For Trigger 41
Figure 26 Analog Trigger Acquisition 42

Figure 27 External Digital Trigger....42
Figure 28 Digital Trigger Paraments....43
Figure 29 Digital Trigger Acquisition....44
Figure 30 Start Trigger....45
Figure 31 Reference Trigger....46
Figure 32 ReTrigger....46
Figure 33 Retrigger Paraments....47
Figure 34 Retrigger In Start Trigger Mode....48
Figure 35 Retrigger In Reference Trigger Mode....48
Figure 36 Complete Retrigger Count....49
Figure 37 AI Continuous Paraments....50
Figure 38 AO Finite Output Paraments....51
Figure 39 AO Finite Signal....51
Figure 40 AI Acquisition Signal....52
Figure 41 AI Continuous Paraments....53
Figure 42 AO ContinuousNoWrapping Output Paraments....54
Figure 43 AO ContinuousNoWrapping Signal....54
Figure 44 AI Acquisition AO Sin Signal....55
Figure 45 Update AO Square Signal....55
Figure 46 AI Acquisition AO Square Signal....56
Figure 47 AI Continuous Paraments....57
Figure 48 AO Continuous Wrapping Paraments....57
Figure 49 AO Continuous Wrapping Signal....58
Figure 50 AI Acquisition AO Signal....58
Figure 51 Single Digital Output....59
Figure 52 Single Digital Input....60
Figure 53 DI Continuous Paraments....61
Figure 54 DO ContinuousNoWrapping Output....61
Figure 55 DI Continuous Acquisition....62
Figure 56 Counter Terminal....63
Figure 57 Counter Signal Wiring Instruction....64
Figure 58 Simple Edge Counting in Single Mode....64
Figure 59 Buffered Edge Counting with Internal Sample Clock....65
Figure 60 Simple Edge Counting with Implicit SampleClk....65
Figure 61 Count Direction....66
Figure 62 EdgeCounting For Single Mode....66
Figure 63 EdgeCounting For Finite Mode....67
Figure 64 Counter Values For Internal Clock....68
Figure 65 Counter Values For Implicit Clock....68
Figure 66 Pulse Measurement in Single Mode....69
Figure 67 Pulse Measurement with Internal SampleClk....69

Figure 68 Pulse Measurement with Implicit SampleClk 70
Figure 69 Pulse Measure For Single Mode 71
Figure 70 Pulse Measure Value For Single Mode 71
Figure 71 Pulse Measure For Finite Mode 72
Figure 72 Pulse Measure Values For Finite Mode 72
Figure 73 Frequency Measurement with Internal Sample Clock 74
Figure 74 Frequency Measure For Single Mode 75
Figure 75 Frequency Measure For Continuous Mode 75
Figure 76 Frequency Measure Values 76
Figure 77 Peroid Measure For Single Mode 77
Figure 78 Peroid Measure For Continuous Mode 78
Figure 79 Two-Edge Separation in Single Mode 79
Figure 80Two-Edge Separation with Internal Sample Clock 79
Figure 81 Two-Edge Separation with Implicit Sample Clock 80
Figure 82 Two-EdgeSeparation Measure For Single Mode 81
Figure 83 Two-EdgeSeparation Measure For Finite Mode 82
Figure 84 Quadrature Encoder x1 Mode 83
Figure 85 Quadrature Encoder x2 Mode 83
Figure 86 Quadrature Encoder x4 mode 84
Figure 87 Quadrature Encoder x1 with Sample Clock 84
Figure 88 Quadrature Encoder x4 with Implicit Sample Clock 85
Figure 89 QuadEncoder For Single Mode 86
Figure 90 QuadEncoder For Continuous Mode 86
Figure 91 Two-Pulse Encoder in Single Mode 87
Figure 92 Two-Pulse Encoder with Internal Sample Clock 88
Figure 93 Two-Pulse Encoder with Implicit Sample Clock 88
Figure 94 Two-PulseEncoder For Single Mode 89
Figure 95 Two-PulseEncoder For Finite Mode 90
Figure 96 Two-PulseEncoder For Continuous Mode 91
Figure 97 Single Pulse Output 92
Figure 98 Single Pulse Generation 93
Figure 99 AI Acquisition Single Pulse 93
Figure 100 Finite Pulse Output 94
Figure 101 Finite Pulses Generation 94
Figure 102 AI Acquisition Finite Pulse 95
Figure 103 Continuous Pulse Output 96
Figure 104 Continuous Pulse Generation 96
Figure 105 AI Acquisition Continuous Pulse 97
Figure 106 SSI Connector in PCIe-5500 98
Figure 107 DIP Switch in PCIe-5500 99
Figure 108 Six Measurement Scenarios 104

Figure 109 Using Resister to Reduce Common Mode Voltage Effect 105
Figure 110 DC, AC and DSA Mode.....106

Table 1 JY5500 Family and Main Features.....1
Table 2 JY5500 on Different Bus 1
Table 3 General Analog Input Specifications 5
Table 4 Basic Accuracy in DAQ Mode 7
Table 5 Additional Accuracy Adjustment due to Temperature, DAQ Mode 8
Table 6 Additional Accuracy Adjustment due to Multi-Channel Mode.....10
Table 7 Basic AI Accuracy in DSMode 11
Table 8 AC Accuracy for One Channel 12
Table 9 AI Bandwidth 12
Table 10 Common Mode Rejection Ratio Specifications.....13
Table 11 Crosstalk Specifications.....13
Table 12 Analog Output Specifications.....14
Table 13 Basic AO Accuracy 15
Table 14 Additional AO Accuracy.....15
Table 15 Digital IO Specifications.....16
Table 16 Counter/Timer Specifications 16
Table 17 PFI Specifications.....17
Table 18 5510 / 5511 Pin Definition 18
Table 19 5510 / 5511 Counter Pin Define.....19
Table 20 5515 / 5516 Pin Definition 20
Table 21 5515 / 5516 Counter Pin Define.....21
Table 22 Notes to Legend 21
Table 23 Power Specification.....22
Table 24 Physical and Environment.....22
Table 25 Supported Linux Versions 23
Table 26 SSI Connector Pin Assignment for PCIe-5500 98
Table 27 Relationship between switch position and slot number 99
Table 28 DC, AC, DSA Measurements.....106

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 JY5500 Series is a family of multifunction data acquisition boards, which can run on PCIe, PXIe, TXI(Thunderbolt) and USB buses. Depending on the model number, a JY5500 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 JY5500 Family and Main Features

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

Table 2 JY5500 on Different Bus

Please check with JYTEK for the latest JY5500 series offering.

1.2 Main Features

- Up to 0.012% of 24 Hour full scale accuracy or 0.024% of 90 days full scale accuracy
- 32 single-ended or 16 differential 18-bit analog input channels
- 18 bits of resolution
- 7 voltage ranges: ±0.1V/±0.2V/±0.5V/±1V/±2V/±5V/±10V
- 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
- DAQ Mode: Common Data Acquisition Mode
- DS Mode: Digital Signal Processing Mode

1.4 Learn by Example

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

Drivers	Update Date	Category
JY5500_V4.0.4_Linux.tar	2022/2/18	Driver
JY5500_V4.0.3_Win.zip	2022/2/18	Driver
JY5500_V4.0.3_Examples.zip	2022/2/18	Example
JY5500_V4.0.3_C++Examples.zip	2022/2/18	Example
JY5500_V1.0.2_Python.zip	2021/10/29	Driver
JY5500_V1.0.2_PythonExamples.zip	2021/10/29	Example
JY5500_V4.0.0_LV.zip	2021/9/10	Driver

Figure 1 JYPEDIA Information

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

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

Figure 2 PCIe-5500 experiment

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

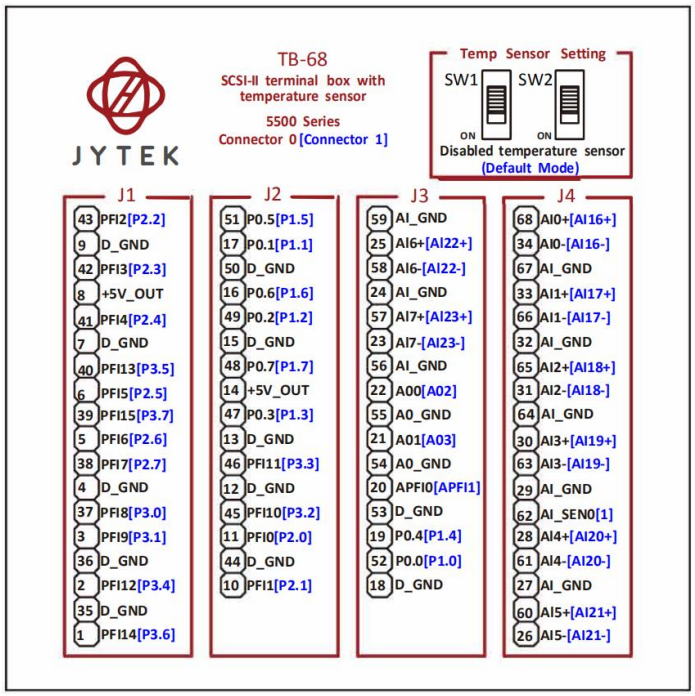


Figure 3 TB-68 Terminal Block

Tip: JY5500 also has the analog output capability. If you do not have a signal source, you can use the outputs of JY5500 as the signal source. In this case you need first run example program **Analog Output-->Winform AO Continuous Wrapping Multichannel** *to generate the output.*

2. Hardware

2.1 Analog Input

2.1.1 General Specifications

Analog Input	5510	5511	5515	5516
Number of channels	32 SE / 16 DIFF		16 SE / 8 DIFF	
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)	±0.1/±0.2/±0.5/±1/±2/±5/±10			
Input mode	RSE / NRSE / Differential			
Input impedance	>1 GΩ||100 pF			
Input coupling	DC			
Input Bandwidth(-3db)*	up to 2.01MHz			
CMRR (dc-60Hz)*	up to 98db			
Crosstalk*	up to -80 dB			
Input FIFO	64M Samples			
Trigger type	Digital, Analog, Software			
Trigger mode	StartTrigger, ReferenceTrigger, ReTrigger			
External Sampling Clock	<= 1MHz			
Analog trigger voltage range	±10V Software Programmable			
Overvoltage protection	±25 V			
Overvoltage Protection	Continuous : 20m A, ±25 V			
	Instantaneous : 40 mA, ±25 V			
Maximum Working	±11 V (ref. AIGND)			
*: see detailed section descriptions				

Table 3 General Analog Input Specifications

2.1.2 DC Accuracy

Basic DC Accuracy, DAQ Mode

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

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

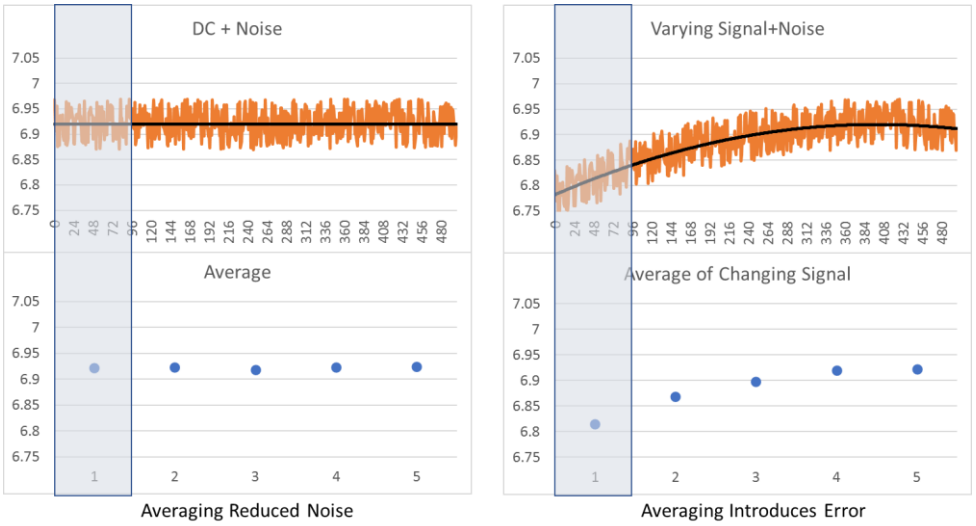


Figure 4 Averaging in DC Measurements

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

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

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

The DAQ mode is the normal data acquisition mode commonly found in commercial DAQ hardware. The basic AI DC accuracy Table 4 of the DAQ mode provides accuracy entries when JY5500 operates in the single channel mode and within the indicated calibration temperature range. Please note that this accuracy is valid for every single point regardless how many sample points you acquire.

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

following fomular:

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

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

$$\pm(0.005\% * 0.02 + 0.065\% * 0.1) = \pm 0.000066\text{V} = \pm 66\text{uV}$$

The basic accuracy table also provides full-scale accuracy entries for a quick and convenient look-up. For example, the full-scale accuracy for the 0.1V range and the 24-Hour calibration column is 70 uV.

JY5500 Basic Accuracy = ±(% Reading+% Range),DAQ Mode

Nominal Range (V)	Resolution (18-bits) (uV)	24 Hour Tcal ±1C°	90 Days Tcal ± 5°C	24 Hr Full Scale Accuracy	90 Days Full Scale Accuracy	Full Scale Accuracy (%)	Max Input Frequency @2MHz Fs (Hz)
0.1	0.8	0.005 + 0.065	0.016 + 0.093	70 uV	110 uV	0.109	140
0.2	1.5	0.003 + 0.031	0.011 + 0.046	70 uV	110 uV	0.057	70
0.5	3.8	0.001 + 0.013	0.007 + 0.019	70 uV	130 uV	0.026	30
1	7.6	0.001 + 0.008	0.006 + 0.010	90 uV	160 uV	0.016	18
2	15.3	0.001 + 0.006	0.006 + 0.008	140 uV	270 uV	0.014	14
5	38.1	0.003 + 0.007	0.008 + 0.009	480 uV	840 uV	0.017	19
10	76.3	0.003 + 0.005	0.009 + 0.006	760 uV	1500 uV	0.015	15

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

Table 4 Basic Accuracy in DAQ Mode

Temperature Adjustment

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

JY5500 Temperature Accuracy Adjustment = ±(% Reading+% Range)				
Nominal Range (V)	Temperature Coefficients (/ °C)			Full-Scale Temp Adjustment (uV/ °C)
0.1	0.0010	+	0.0025	3 uV
0.2	0.0007	+	0.0013	4 uV
0.5	0.0004	+	0.0006	5 uV
1	0.0002	+	0.0003	6 uV
2	0.0002	+	0.0002	9 uV
5	0.0005	+	0.0002	30 uV
10	0.0006	+	0.0002	70 uV
For all sample rates All accuracy data in this table are tested with 1m shield cable. Specs subject to minor changes when more tests become available				

Table 5 Additional Accuracy Adjustment due to Temperature, DAQ Mode

For example, at the 0.1V range and 2° C outside the 24 Hour Tcal ± 1° C range, the basic accuracy entry (0.005%, 0.065%) must be adjusted to

(0.005%+0.0010%*2, 0.065%+0.0025%*2) = (0.007%, 0.07%).

The accuracy of the same 0.02V reading value is

$$\pm(0.007\% * 0.02 + 0.07\% * 0.1) = \pm 0.0000714\text{V} = \pm 71\text{uV}$$

The full-scale accuracy numbers are also provided in the additional accuracy table. For example, at the 0.1V range, and 2°C outside 24 Hour Tcal ± 1° C range, there is a 3uV/°C adjustment. You will need to add 2*3uV = 6uV to the basic accuracy, and the total accuracy is 70+2*3=76 uV.

Multi-Channel Adjustment

When JY5500 operates at the multi-channel mode, the channel switching can incur additional errors because it takes time for the voltage to discharge from one channel before the next channel starts acquistion. Because JY5500's 32 channels share one ADC, the channel sample rate of each channel is dedermined by

ChannelSample Rate = Sample Rate/N

where

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

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

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

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

JY5500 Multi-Channel Accuracy Adjustment (uV)		
Nominal Range (V)	Sample Rate (Hz) per Channel	Full-Scale Multi-Channel Adjustment (uV)
0.1	<=200K/N	0
0.1	500K/N	45 uV
0.1	1M/N	98 uV
0.2	<=200K/N	0
0.2	500K/N	23 uV
0.2	1M/N	136 uV
0.5	<=200K/N	0
0.5	500K/N	32 uV
0.5	1M/N	314 uV
1	<=200K/N	0
1	500K/N	70 uV
1	1M/N	683 uV
2	<=200K/N	0
2	500K/N	109 uV
2	1M/N	904 uV
5	<=200K/N	0
5	500K/N	693 uV
5	1M/N	2,904 uV
10	<=200K/N	0
10	500K/N	2,375 uV
10	1M/N	5,713 uV
N: Number of channels from 2 to 32 Use the next higher sample rate for the not listed sample rates 1M/N not recommended for a DC measurement. Source Impedence: <=25Ω TB-68 Terminal Block, ACL-2006868-1 1m shield cable Preliminary, subject to changes.		

Table 6 Additional Accuracy Adjustment due to Multi-Channel Mode

Basic DC Accuracy, DS Mode

JY 5510 and 5515 is capable to make very accurate DC measurement by using the DS Mode operation. Please note the DS mode is currently available for one channel measurement only. The reason is that the channel switching induces errors that affect the measurement accuracy. The DS mode is not intended for high frequency measurement at all. We strongly recommend the DS Mode measurement within the calibration temperature range to keep the maximum accuracies.

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

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

JY5510/5515 only provide the accuracy entries for several predetermined sample rates. If you choose a different sample rate, you need to use the accuracy entries in the next higher sample rate. For instance, if you choose a 5K sample rate, you need to use the accuracy of the 10K sample rate.

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

JY5510, 5515 Basic Accuracy = ±(% Reading+% Range),DSMode

Nominal Range (V)	Max Sample Rate for 1 Channel	Resolution (uV)	24 Hour Tcal ±1C°	90 Days Tcal ± 5°C	24 Hr Full Scale Accuracy	90 Days Full Scale Accuracy	Full Scale Accuracy (%)	Max Input Frequency Fs (Hz)
0.1	1K	0.8	0.0024 + 0.0087	0.0052 + 0.0148	11 uV	20 uV	0.020	0.011
0.1	10K	0.8	0.0024 + 0.0106	0.0052 + 0.0166	13 uV	22 uV	0.022	0.130
1	1K	7.6	0.0008 + 0.0013	0.0029 + 0.0021	22 uV	50 uV	0.005	0.002
1	10K	7.6	0.0008 + 0.0015	0.0029 + 0.0023	23 uV	52 uV	0.005	0.023
10	1K	76.3	0.0007 + 0.0009	0.0031 + 0.0014	161 uV	453 uV	0.005	0.002
10	10K	76.3	0.0007 + 0.0010	0.0031 + 0.0015	171 uV	463 uV	0.005	0.017

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

Table 7 Basic AI Accuracy in DSMode

2.1.3 AC Accuracy

AC Accuracy for One Channel

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

Table 8 gives the total AC accuracy specification for JY5500. It is for one channel only. There is no AC accuracy specification when the sinusoidal input is greater than 200KHz. The reason is we need at least 5 sample points per cycle to calculate a time domain sinusoidal waveform.

JY5500 Total Absolute AC Accuracy for One Channel							
	0.1V	0.2V	0.5V	1V	2V	5V	10V
[10Hz,50K)	0.28%	0.24%	0.23%	0.22%	0.17%	0.16%	0.15%
[50K,100K)	0.79%	0.66%	0.57%	0.74%	0.48%	0.39%	0.38%
[100K,200K]	2.42%	2.09%	1.90%	2.12%	1.28%	1.00%	1.00%
Valid for one channel measurement of sinusoidal input 90 days, Tcal±5ºC. For one year AC accuracy, add 20%. Preliminary Sample Rate>= 1.25M. No specs for Sample Rate<1.25MHz. DC Coupling 10 V range: valid for ±9.5V Source impedence <=100Ω All accuracy data in this table are tested with 1m shield cable. 95% confidence level							

Table 8 AC Accuracy for One Channel

AC Accuracy for Multiple Channels

In the case where the multi-channel sampling rate is less than 200k/N, the single-channel AC precision result is applicable.

Temperature Adjustment for AC Accuracy

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

2.1.4 Dynamic Performance

AI Bandwidth

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

Table 9 AI Bandwidth

CMRR

The CMRR performance at is shown below.

Input Voltage	CMRR (DC-60Hz) (db)
0.5	**98**
1	**95**
5	**84**

Table 10 Common Mode Rejection Ratio Specifications

Crosstalk

The crosstalk is the interference from one channel wire to another channel wire. It depends on the connection wires and internal circuit layout. In JY5500, each differential channel has three wires, AI+, AI-, and GND. The two wires, AI+ and AI-, are interwound together. Therefore, the interferenc from AI+ to AI- within the same differential group has the largest crosstalk interference. Table 11 shows the crosstalk specifications of between different wire configurations. Please see the pin definition in Section 2.6 to determine which wires belong to the same differential pair. To reduce the crosstalk interference, select the wires that belong to the different differential pairs whenever possible.

Wires	Crosstalk (db)
Wires within a differential pair	-65
Wires belong to different differential pairs	-80
TB-68 Terminal Block, ACL-2006868-1 1m shield cable	

Table 11 Crosstalk Specifications

Example, from Table 18 , Pin 30 and Pin 63 are AI3+ and AI3-, they from a pair of differential pair. The crosstalk between these two wires is -65db.

2.2 Analog Output

2.2.1 General Specifications

Analog Output	5510	5511	5515	5516
Number of channels	4		2	
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)	±5, ±10			
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 12 Analog Output Specifications

2.2.2 Basic AO Accuracy

The AO output accuracy of JY5500 Series when using the analog output function can be calculated according to the corresponding parameters in the following table

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

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

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

$$\pm(0.003\% * 2 + 0.006\% * 5) = \pm 0.00036\text{V} = \pm 360\text{uV}$$

The basic accuracy table also provides full-scale accuracy entries for a quick and convenient look-up. For example, the full-scale accuracy for the 5V range and the 24-Hour calibration column is 470 uV.

JY5500 Basic AO Accuracy = ±(% of Output+% of Range)							
Nominal Range (V)	Resolution (16-bits) (uV)	24 Hour Tcal ±1C°	90 Days Tcal ± 5°	24 Hr Full-Scale Accuracy	90 Days Full-Scale Accuracy	Full Scale Accuracy (%)	Max Update Rate (S/s)
5	153	0.003 + 0.006	0.009 + 0.008	470 uV	840 uV	0.017	2.86M
10	305	0.004 + 0.005	0.013 + 0.005	810 uV	1800 uV	0.018	2.86M
Valid for all update rates. Add accuracy adjustment if temperature is ouside calibration temperature range. Add 20% to Gain and Offset Errors From 91 Days to 1 Year. Preliminary. Maximum update rates(simultaneous) 1 Ch: 2.86M; 2 Ch: 2M; 3 Ch: 1.54M; 4 Ch: 1.25M All accuracy data in this table are tested with 1m shield cable. Specs subject to minor changes when more tests become available.							

Table 13 Basic AO Accuracy

2.2.3 Additional AO Accuracy Adjustment

Table 14 is the additional AO accuracy adjustment table. It provides additional accuracy adjustments: temperature adjustment.

JY5500 AO Additional Accuracy Adjustment			
Nominal Range (V)	Resolution (18-bits) (uV)	Temperature Coefficients (/ °C)	Full-Scale Temp Adjustment (uV/ °C)
5	153	0.0025 + 0.0008	160 uV
10	305	0.0028 + 0.0006	340 uV
Valid for all update rates. All accuracy data in this table are tested with 1m shield cable. Specs subject to minor changes when more tests become available.			

Table 14 Additional AO Accuracy

The temperature adjustment is needed when the operating temperature is outside the calibration temperature range, such as the Tcal $\pm 1^{o}$ C range, or the Tcal $\pm 5^{o}$ C range etc. The temperature coefficients, both the gain and offset coefficients, are provided. These cofficients must be added to the gain and offset coefficients in the basic accuracy table to calculate the accuracy. For example, at the 5V output and 2^{o} C outside the 24 Hour Tcal $\pm 1^{o}$ C range, the basic accuracy entry (0.003%, 0.006%) must be adjusted to

(0.003%+0.0025%*2, 0.006%+0.0008%*2) = (0.008%, 0.0076%).

The accuracy of the same 2V output value is

$$\pm(0.008\% * 2 + 0.0076\% * 5) = \pm 0.00054\text{V} = \pm 540\text{uV}$$

The full-scale accuracy numbers are also provided in the additional accuracy table. For example, at the 5V range, and 2°C outside 24 Hour Tcal $\pm 1^{o}$ C range, there is a 45uV/°C adjustment. You will need to add 2*160uV = 320uV to the basic accuracy, and the total accuracy is 470+320=790 uV.

2.3 Digital IO Specifications

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

Table 15 Digital IO Specifications

2.4 Counter/Timer Specifications

CI/CO	**5510**	**5511**	**5515**	**5516**
Number of channels	4		2	
Resolution	32 bits			
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			

Table 16 Counter/Timer Specifications

2.5 PFI Specifications

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

Table 17 PFI Specifications

2.6 Additional Specifications

2.6.1 Front Panel and Pin Definition

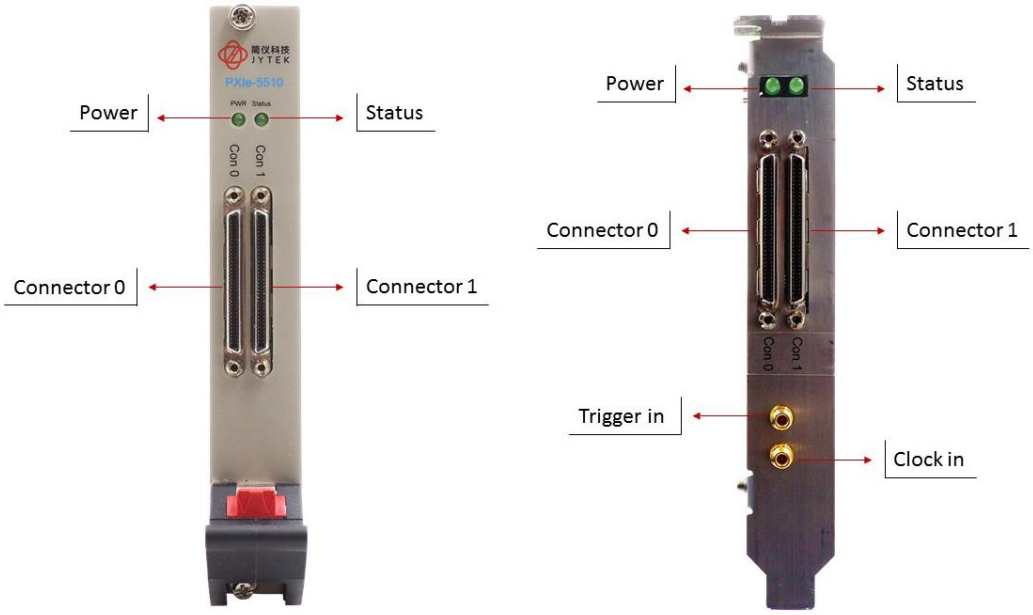


Figure 5 PXIe/PCIe 5510 Front Panel

JY5500 series boards are connected to outside signals by either two 68-pin cables for the 32 channel configurations or one 68-pin cable for 16 channel configurations. Table 18, Table 19, Table 20, Table 21 show the pin definitions for 32 channels and 16 channels of JY5500 series boards respectively. Please note that pin definition of connector 0 and 1 is different!

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

2.6.2 5510/5511 Connector Pin Definition

Connector 0			
Pin	**Signal Name**	**Pin**	**Signal Name**
1	PFI 14 /P5.6/DO_ECLK	35	D_GND
2	PFI 12/P5.4	36	D_GND
3	PFI 9/P5.1	37	PFI 8/P5.0
4	D_GND	38	PFI 7/P4.7
5	PFI 6 /P4.6/ AO_ECLK	39	PFI 15/P5.7
6	PFI 5/P4.5	40	PFI 13/P5.5
7	D_GND	41	PFI 4/P4.4
8	+5V_OUT	42	PFI 3/P4.3
9	D_GND	43	PFI 2 /P4.2/AI_ECLK
10	PFI 1/P4.1	44	D_GND
11	PFI 0/P4.0	45	PFI10 /P5.2/DI_ECLK
12	D_GND	46	PFI 11/P5.3
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	NC*	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 0
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+）

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

* NC: Not Connected

Table 18 5510 / 5511 Pin Definition

2.6.3 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 19 5510 / 5511 Counter Pin Define

2.6.4 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	D_GND
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	NC*	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+）

* NC: Not Connected

Table 20 5515 / 5516 Pin Definition

2.6.5 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 21 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 22 Notes to Legend

2.6.6 Power Specification

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

USB Power Supply

Power	USB-5510	USB-5511	USB-5515	USB-5516
12V	4A			

Table 23 Power Specification

2.6.7 Physical and Environment

Operating Environment

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

Storage Environment

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

Table 24 Physical and Environment

3. Software

3.1 System Requirements

JY5500 boards can be used in a Windows or a Linux operating system.

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

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

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

Table 25 Supported Linux Versions

3.2 System Software

When using the JY5500 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 JY5500 be with .NET Framework 4.0 with Microsoft Visual Studio 2015. JYTEK relies on Microsoft to maintain the compatibility for the newer versions.

3.3 C# Programming Language

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

3.4 JY5500 Series Hardware Driver

After installing the required application development environment as described above, you need to install the JY5500 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 JY5500 function. JYTEK has standardized the ways which JYTEK and other vendor's DAQ boards are used by providing a consistent user interface, using the methods, properties and enumerations in the object-oriented programming environment. Once you get yourself familiar with how one JYTEK DAQ card works, you should be able to know how to use all other DAQ hardware by using the same methods.

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

3.5 Install the SeeSharpTools from JYTEK

To efficiently and effectively use JY5500 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 JY5500 hardware. Please register and down load the latest SeeSharpTools from our website, www.jytek.com.

3.6 Running C# Programs in Linux

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

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

4. Operating JY5500

This chapter provides the operation guides for JY5500, 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 JY5500 board. JYTEK strongly recommends you go through these examples before writing your own application. In many cases, an example can also be a good starting point for a user application.

4.1 Quick Start

After you have installed the driver software and the SeeSharpTools, you are ready to use Microsoft Visual Studio C# to operate the JY5500 products.

If you are already familiar with Microsoft Visual Studio C#, the quickest way to use JY5500 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 JY5500.

4.2 Data Acquisition Methods

JY5500 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 JY5500 driver software. The most important parameters are *Data Acquisition mode*, *Sample Rate*, *SamplesToAcquire*, *Channel Count*, *ChannelRange* and *Analog Input Terminal Type*.

AI Acquisition mode **(***AIMode***)**: JY5500 provides 4 acquisiontion modes, **Continuous**, **Finite**, **Single Point**, **Record,** which will be described in details in Section 4.2.1-4.2.4.

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

SamplesToAcquire: This parameter behaves differently in the different AI acquisition modes. In the continuous acquisition mode, *SamplesToAcuire* is the buffer size used

in the AI acquisition task, please see Section 4.2.1; in the finite acquisition mode, it is the total number of samples to capture, please see Section 4.2.2.

Channel Count: how many channels you want to collect data. You can set up the channels in different orders, for instance 2,3,1,0. The acquired data will be arranged in the way you specify as shown in Figure 6. In this particular case, *Channel Count* is 4.

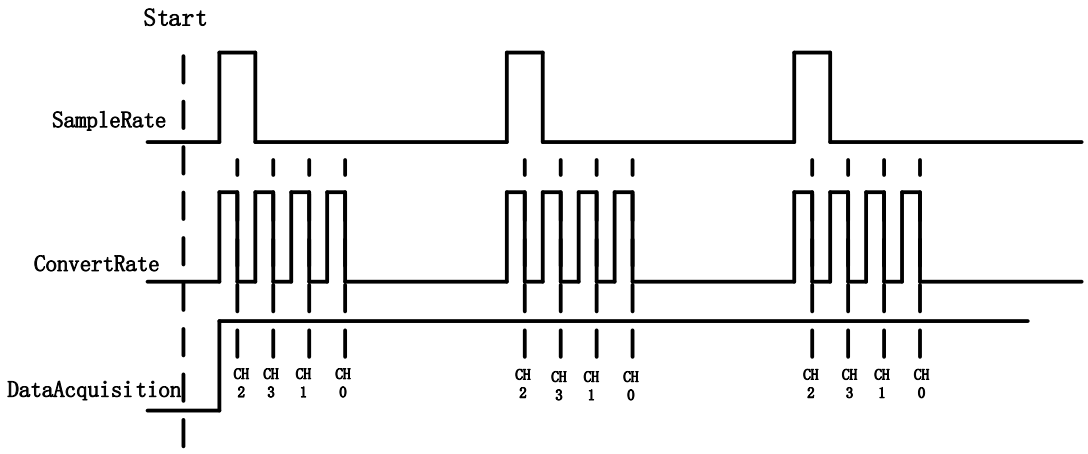


Figure 6 Sample Rate and Internal AD Conversion

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

*Multichannel maximum sample rate (aggregate) >=ConvertRate >= SampleRate * ChannelCount.*

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

Learn by Example 4.2

- Connect the two signal source's positive outputs to JY5500 AI Ch0 (AI0+, Pin #68) and AI Ch1 (AI1+, Pin#33), two negative terminals to the ground (AI_GND, Pin#67) as shown in Figure 2 and Figure 3. (AI0+, AI_GND) and (AI1+, AI_GND) consist of two channels of RSE inputs and they share the same GND.
- Set a sinewave signal (f=4Hz, Vpp=5V) and a squarewave signal (f=4Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous MultiChannel**, set the following numbers as shown. This sample program will continuously acquire data from multiple channels.

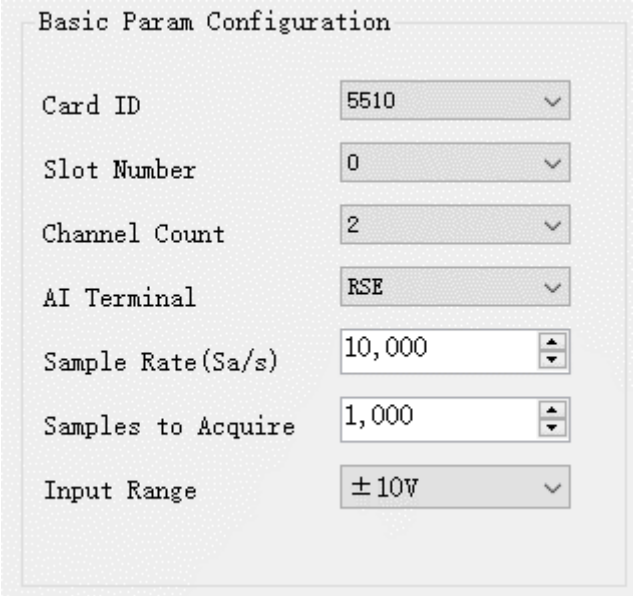


Figure 7 Continuous MultiChannel Paraments

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

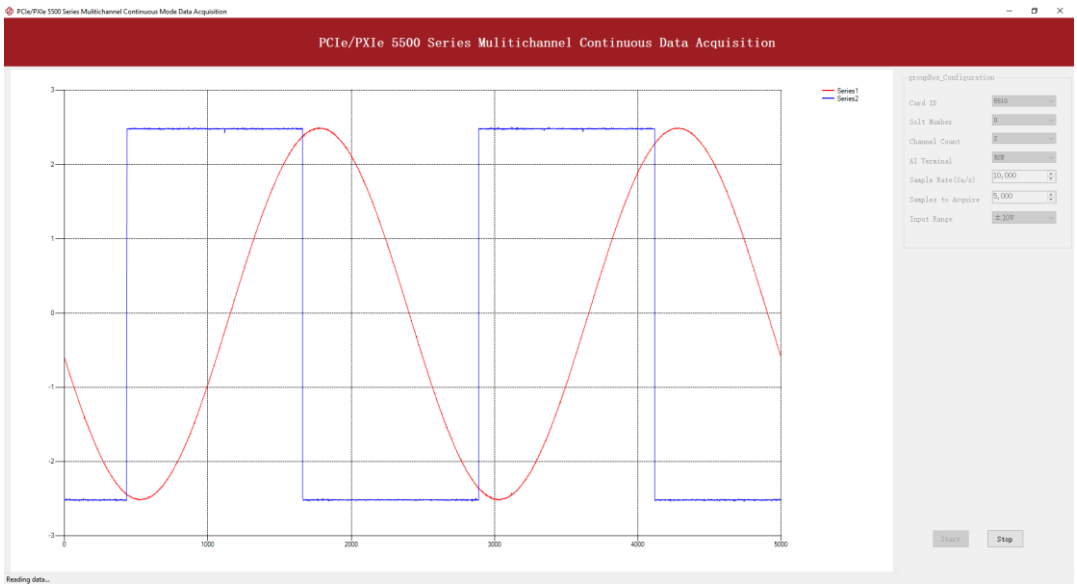


Figure 8 MultiChannel Continuous Acquisition

4.2.1 Continuous Acquisition

An AI acquisition task will acquire the data continuously until the task is stopped. The JY5500 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 erro message.

Tip: User buffer's length 1/10th to 1/4th *SampleRate* is a good start.

4.2.2 Finite Acquisition

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

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

4.2.3 Single Point Acquisition

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

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

4.2.4 Record Acquisition

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

4.3 Analog Input Terminal Type

The JY5500 provide 3 analog input terminal types:

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

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

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

4.3.1 DIFF Mode

The DIFF mode connects signal's positive side to AI's positive input, signal's grounded negative side to AI's negative input as shown in Figure 9. The common noise appears on both positive and negative terminals of the differential amplifier, thus it will be cancelled out. Therefore, the DIFF mode has better signal-to-noise ratio (SNR). Please see Appendix 1 Common Analog Measurement Issues for more explainations.

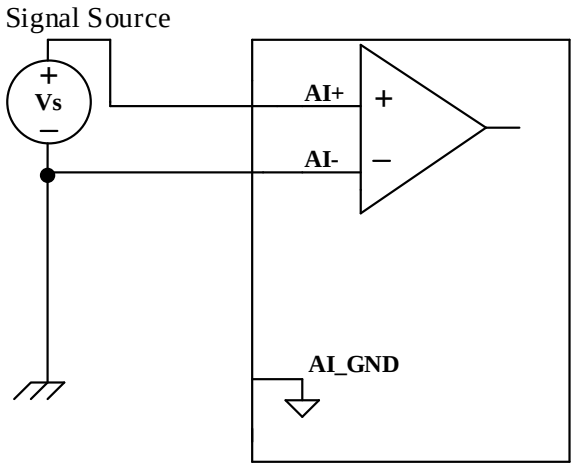


Figure 9 Differential Mode for Grounding Signals

Learn by Example 4.3.1

- Open the program **Analog Input-->Winform AI Continuous MultiChannel**
- Connect the two signal source's positive outputs to JY5500 AI Ch0 (AI0+, Pin #68) and AI Ch1 (AI1+, Pin#33), two negative terminals to AI Ch0 negative (AI0-, Pin#34) and AI Ch1 negative (AI1-, Pin#66) as shown in Figure 2 and Figure 3. (AI0+, AI0-) and (AI1+, AI1-) consist of two pairs of DIFF inputs;
- Choose Differential in **AI Terminal**;
- Set other numbers as shown and click **start**.

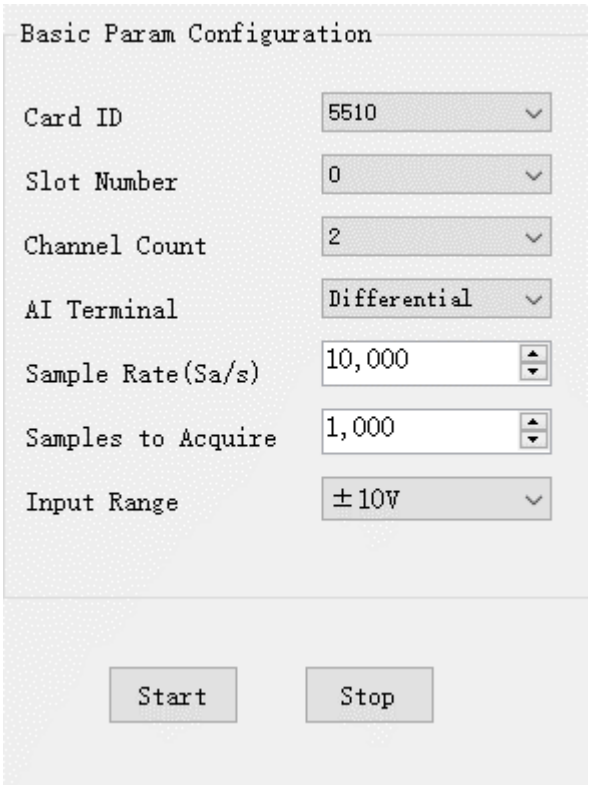


Figure 10 Choose Differential In AI Terminal

4.3.2 RSE Mode

In the RSE mode, all input signals' negative sides are connected to the AI ground of Instrumentation Amplifier, as shown in Figure 11. 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 conneted 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 explainations.

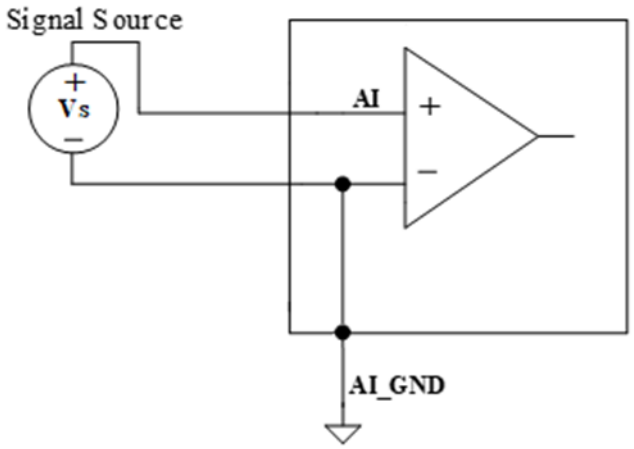


Figure 11 RSE Mode for Floating Signals

4.3.3 NRSE Mode

The NRSE mode is recommended for the measurement of ground-referenced signals, as shown in Figure 12. NRSE is also called the pseudo differential mode, because it looks very similar to a DIFF connection. In this mode, the JY5500 device offers a special reference point, AI SENSE. Instead of connecting two gounds 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 JY5500is 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 explainations.

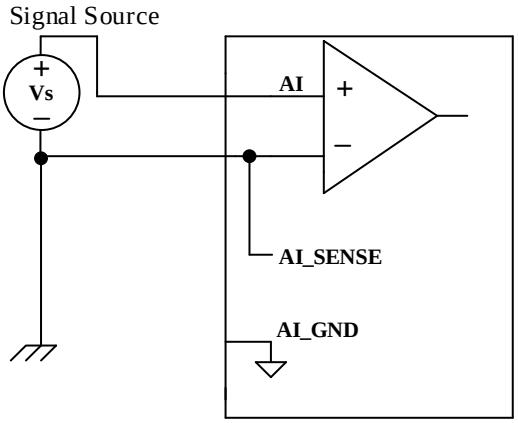


Figure 12 NRSE Mode for Grounding Signals

Learn by Example 5.3.3

- Open the program **Analog Input-->Winform AI Continuous MultiChannel**.
- This Example needs two TB-68 terminal blocks, Connector0 and Connector1 and two cables, which are connected to JY5500. Connect the two signal source's positive outputs to PCIe-5510 AI Ch0 (AI0+, Pin #68) and AI Ch1 (AI1+, Pin#33), two negative terminals to AI_SENSE 0 (Pin#62) of the first TB-68 and AI_SENSE 1 (Pin#62) of the second TB-68 as shown in Figure 2 and Figure 3. (AI0+, AI_SENSE 0) and (AI1+, AI_SENSE 1) consist of two channels of NRSE inputs.
- Choose the NRSE in **AI Terminal**
- Set other numbers as shown and click **start**.

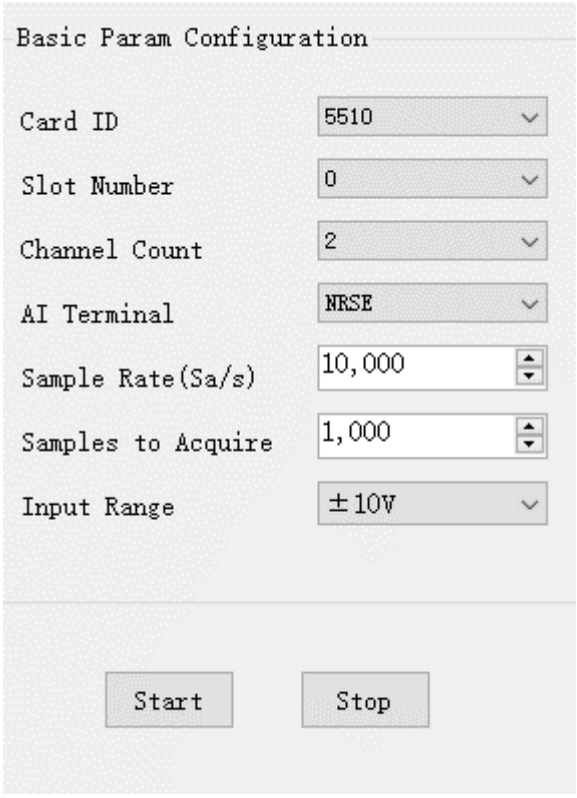


Figure 13 Choose NRSE In AI Terminal

4.4 Trigger Source

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

4.4.1 Immediate trigger

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

Learn by Example 4.4.1

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

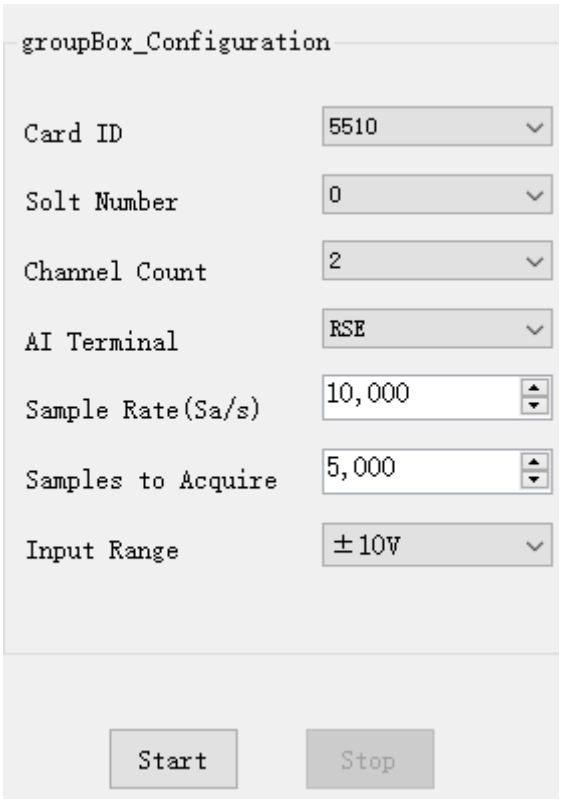


Figure 14 Immediate trigger Paraments

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

4.4.2 Software Trigger

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

Learn by Example 4.4.2

- Connect the signal source's positive terminal to JY5500 AI Ch0 (AI0+, Pin#68), the negative terminal to the ground (AI_GND, Pin#67) as shown in Figure 2 and Figure 3. (AI0+, AI_GND) consists of a RSE input.
- Set a sinewave signal (f=4Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous Soft Trigger**, set the following numbers as shown.
- Click **Start** to run the task.

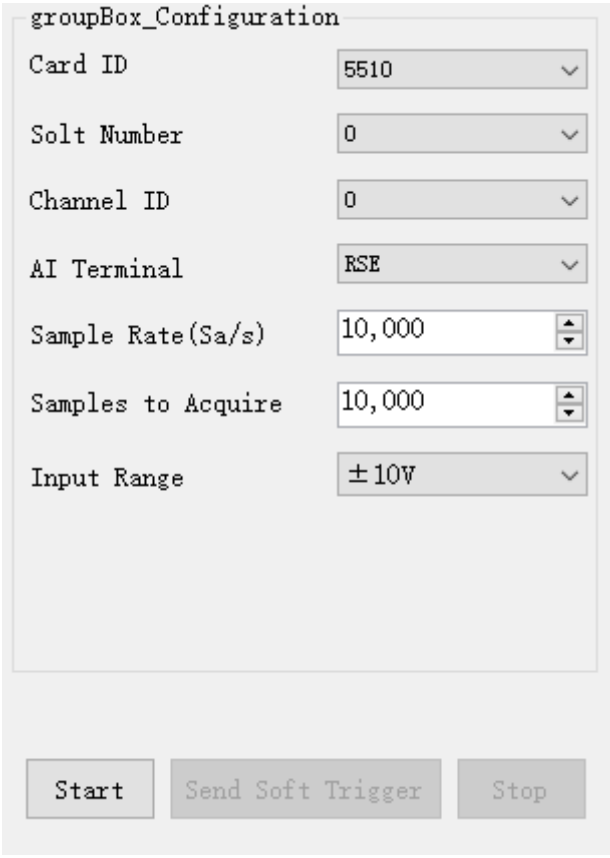


Figure 15 Software trigger Paraments

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

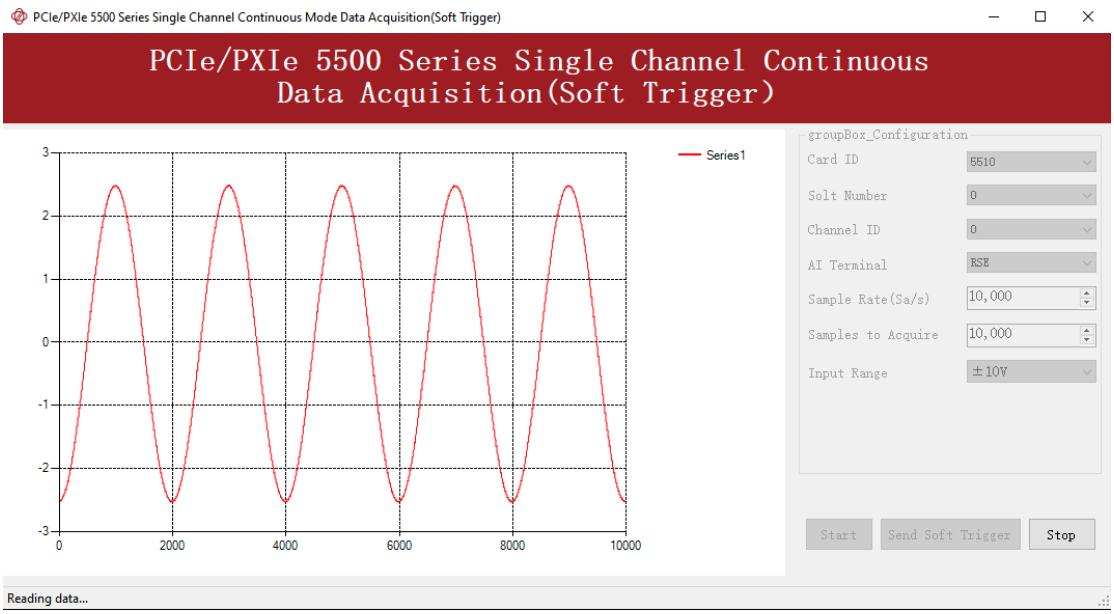


Figure 16 Software trigger Acquisition

4.4.3 External Analog Trigger

You can assign one of measurement channels as the analog trigger source. JY5500 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 17.

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 18.

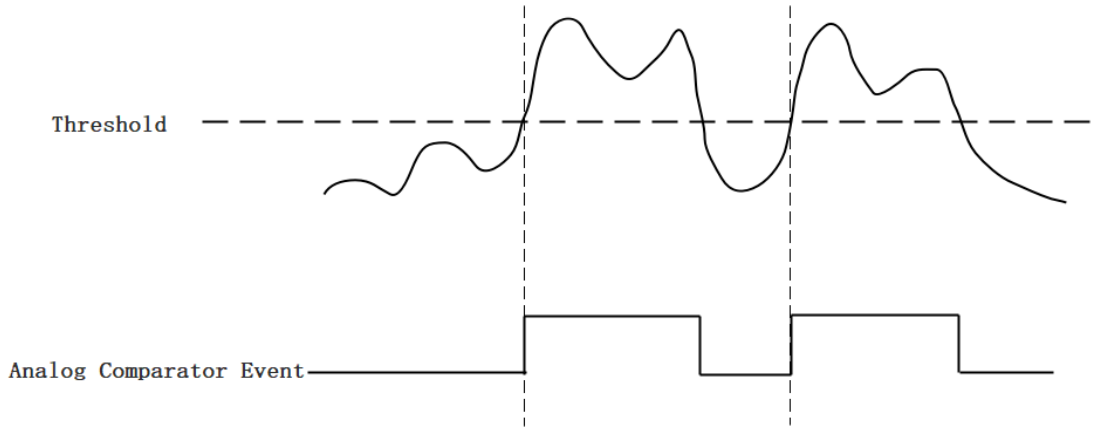


Figure 17 Rising Slope Trigger

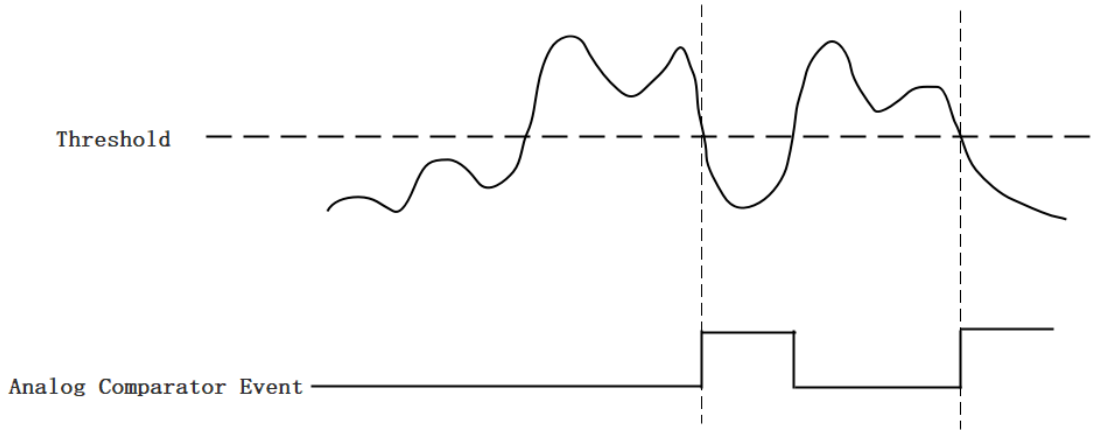


Figure 18 Falling Slope Trigger

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 19.

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 20.

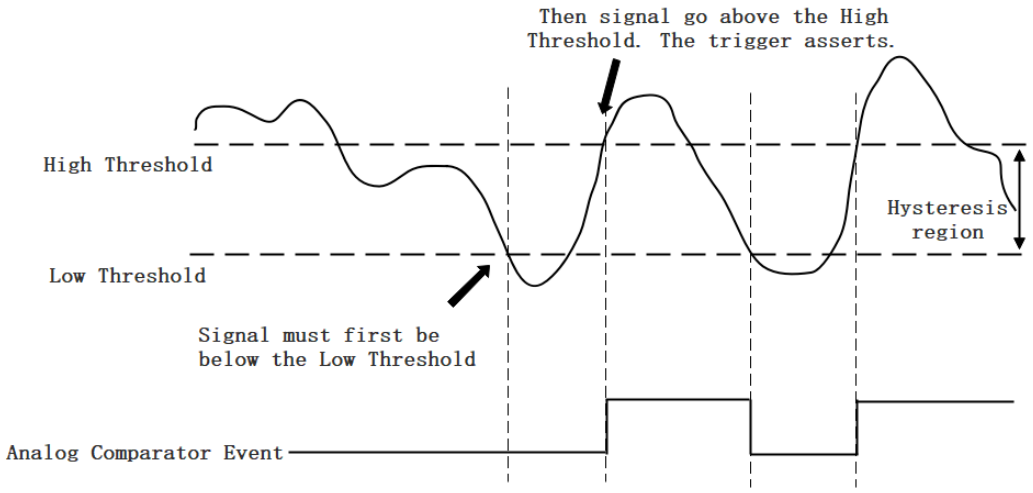


Figure 19 Hysteresis with Rising Slope Trigger

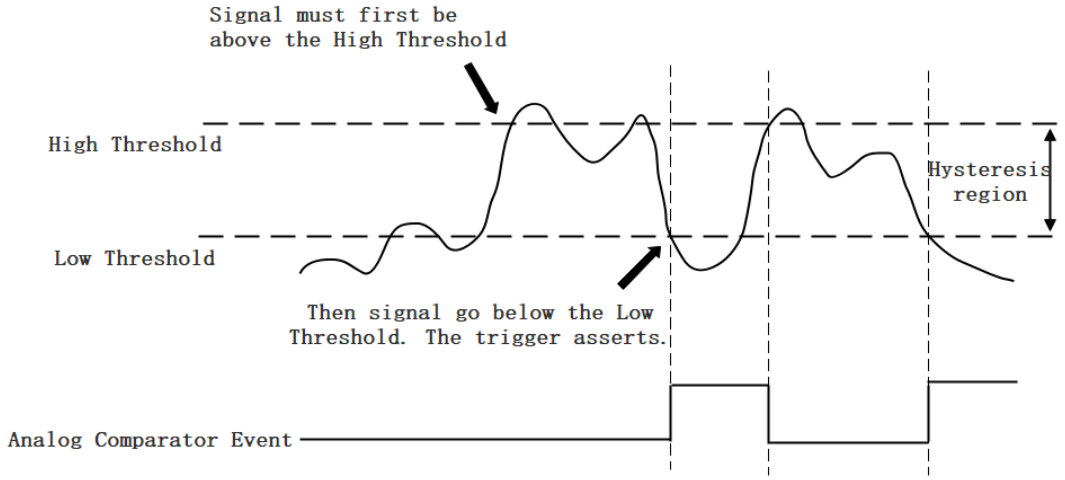


Figure 20 Hysteresis with Falling Slope Trigger

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 21.

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 22 Leaving Window Trigger.

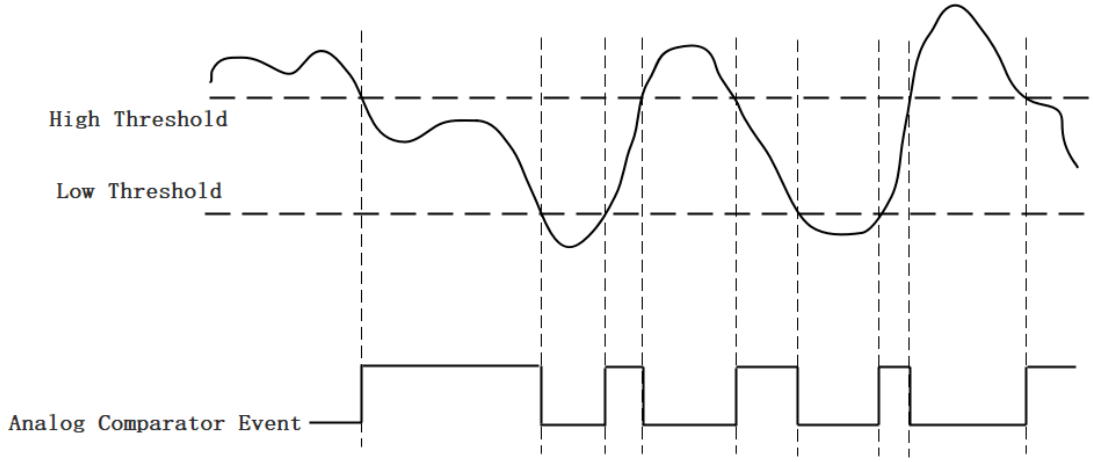


Figure 21 Entering Window Trigger

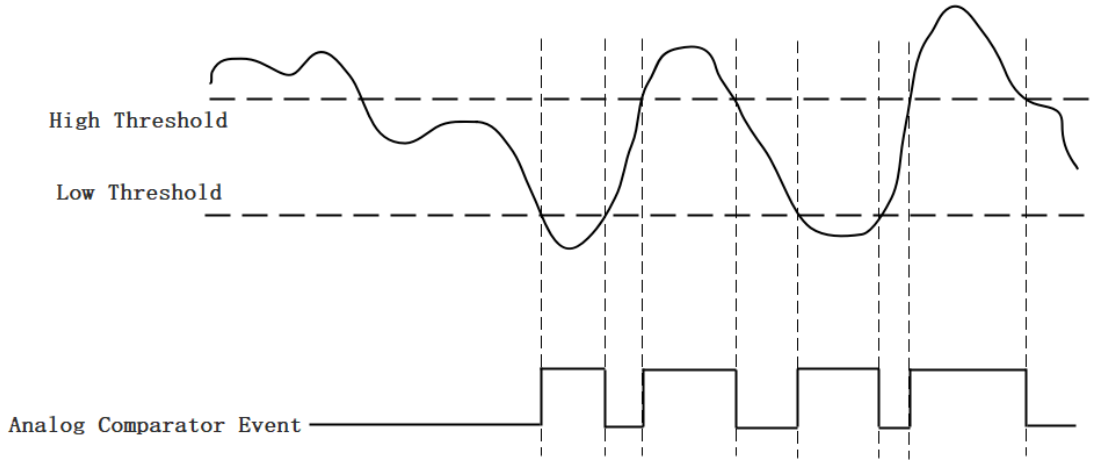


Figure 22 Leaving Window Trigger

Learn by Example 4.4.3

- Connect the signal source's positive terminal to JY5500 AI Ch0 (AI0+, Pin#68), the negative terminal to the ground (AI_GND, Pin#67) as shown in Figure 2 and Figure 3. (AI0+, AI_GND) consists of a RSE input.
- Set a sinewave signal (f=4Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous Analog Trigger**, set the following numbers as shown.

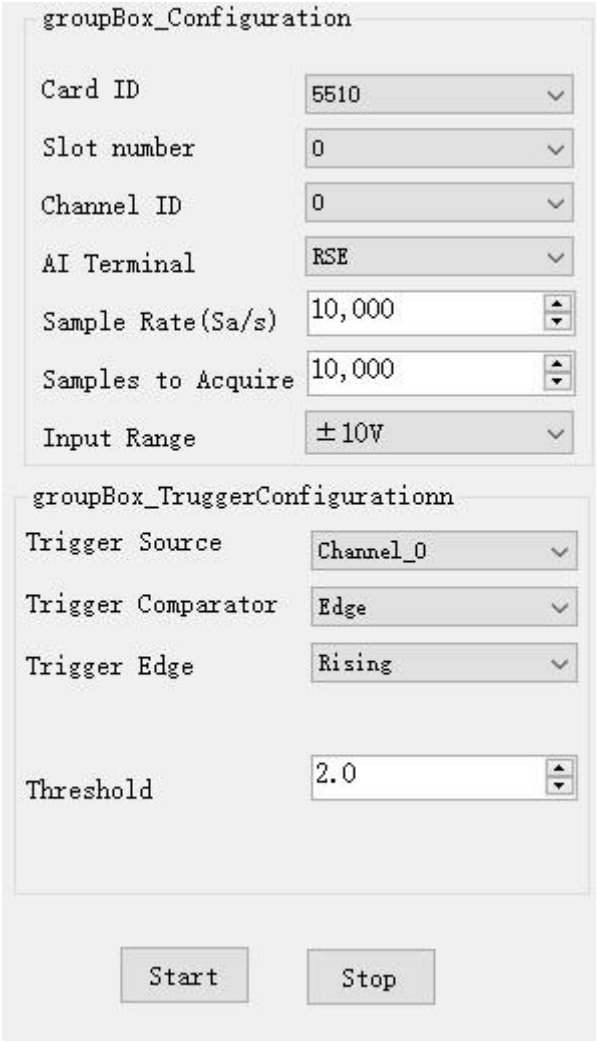


Figure 23 Analog Trigger Paraments

- 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 JY5500 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:

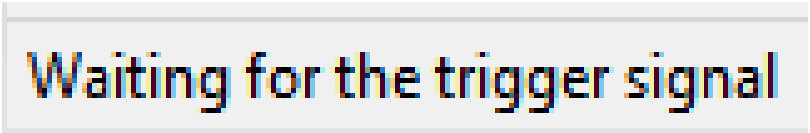


Figure 24 Waiting For Trigger

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

■ The result is shown below:

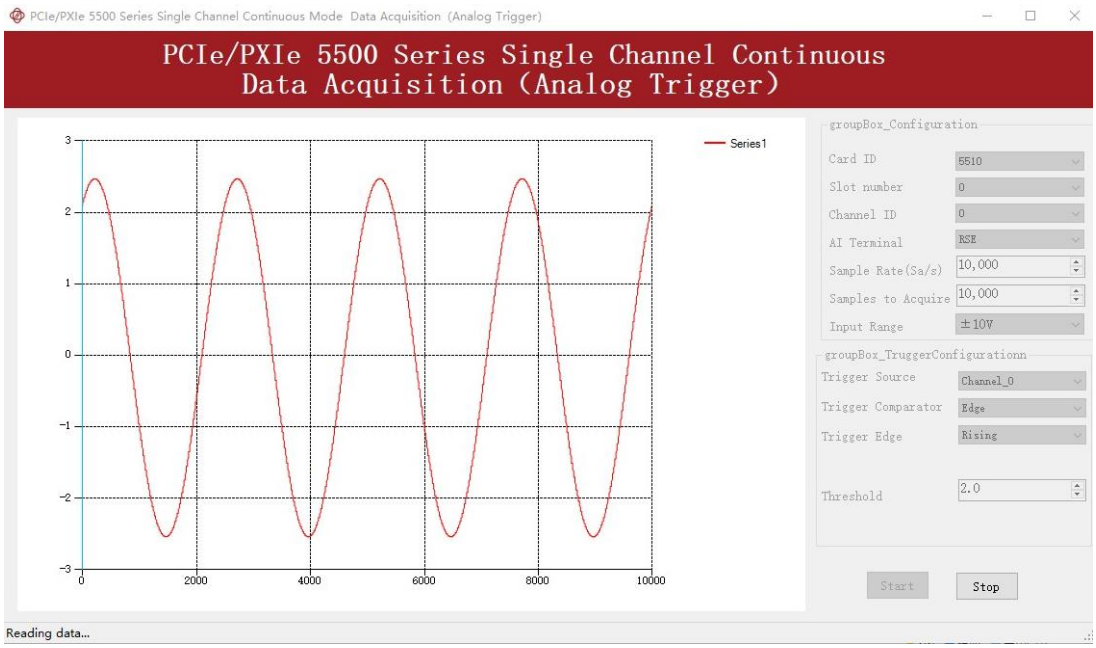


Figure 25 Analog Trigger Acquisition

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

4.4.4 External Digital Trigger

JY5500 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 26

Figure 26 External Digital Trigger

Learn by Example 4.4.4

- Connect the signal source two positive terminals to JY5500 AI Ch0, (AI0+, Pin #68) and digital trigger source (PFI 0, Pin#11), two negative terminals to the ground of analog input (AI_GND, Pin#67) and the ground of digital input/output (DGND, Pin#44) as shown in Figure 2 and Figure 3. (AI0+, AI_GND) consists of a RSE input. (PFI0, DGND) provides the trigger signal.
- Set a sinewave signal (f=4Hz, Vpp=5V) and a squarewave signal (f=4Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous Digital Trigger**, set the following numbers as shown.

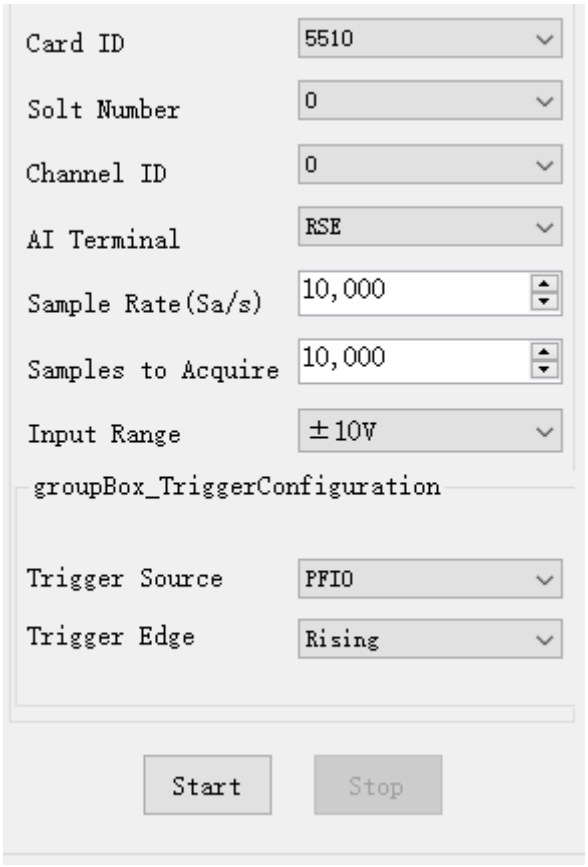


Figure 27 Digital Trigger Paraments

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

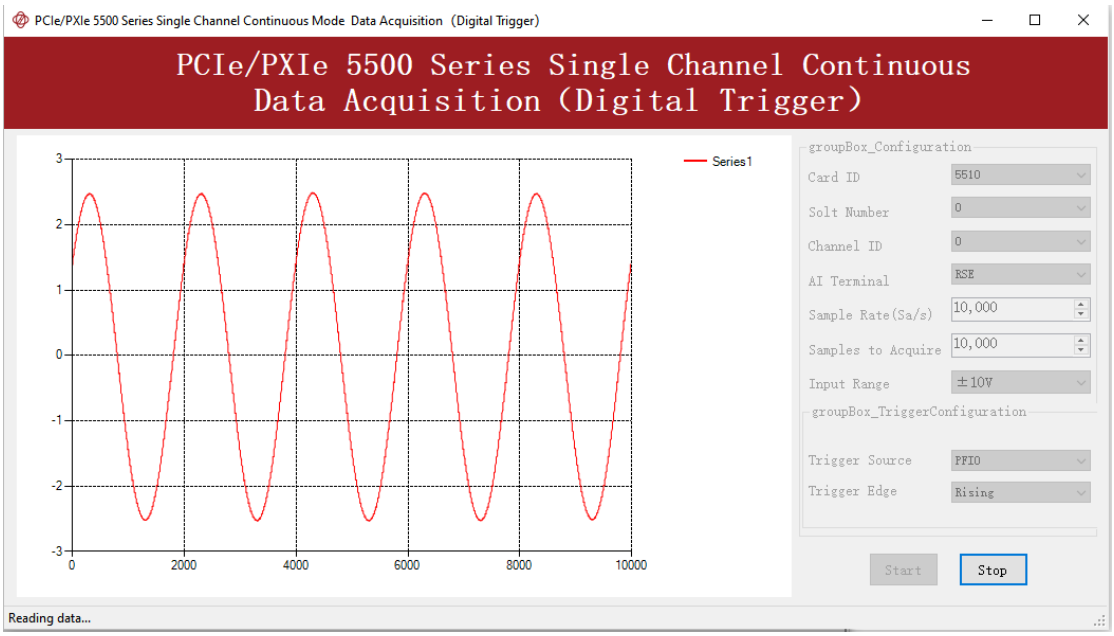


Figure 28 Digital Trigger Acquisition

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

4.5 Trigger Mode

The JY5500's analog inputs support several trigger modes: start trigger, reference trigger, and re-trigger.

4.5.1 Start Trigger

In this mode, data acquisition begins immediately after the trigger. This trigger mode is suitable for continuous acquisition and finite acquisition. As shown in Figure 29.

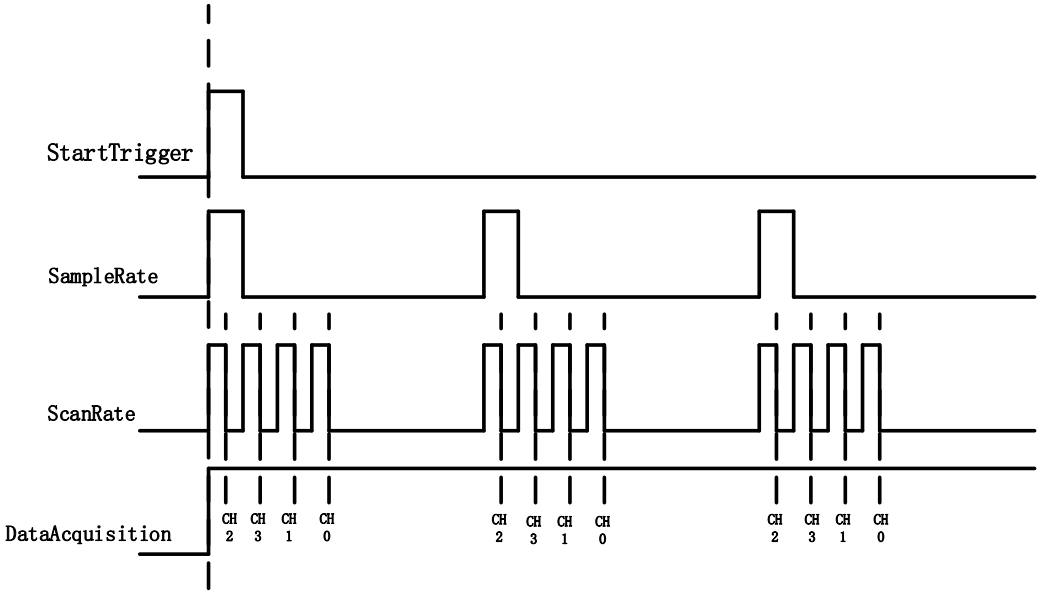


Figure 29 Start Trigger

4.5.2 Reference Trigger

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

Example

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

The principle is shown in Figure 30.

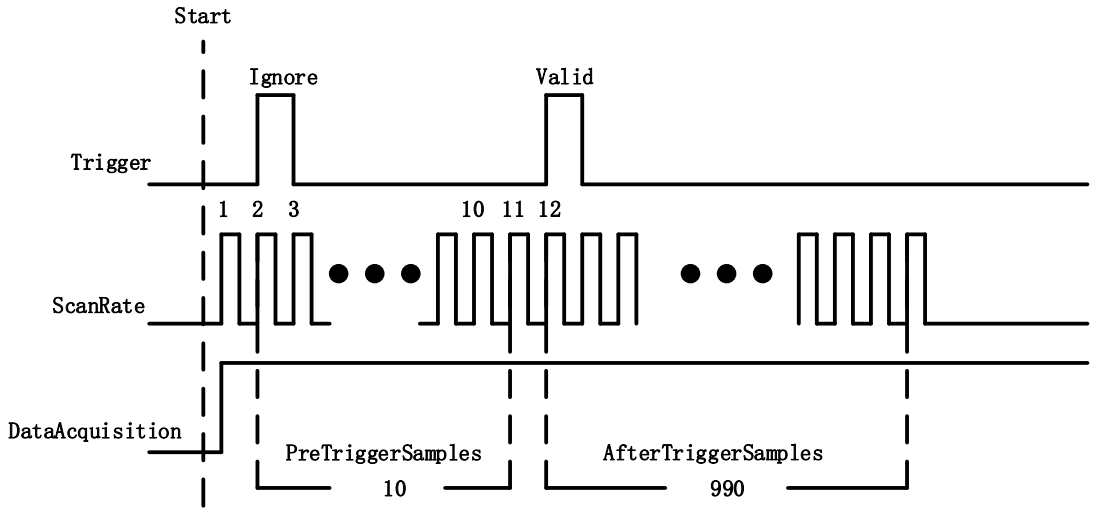


Figure 30 Reference Trigger

4.5.3 ReTrigger

JY5500 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 * channelcount. Show in Figure 31.

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

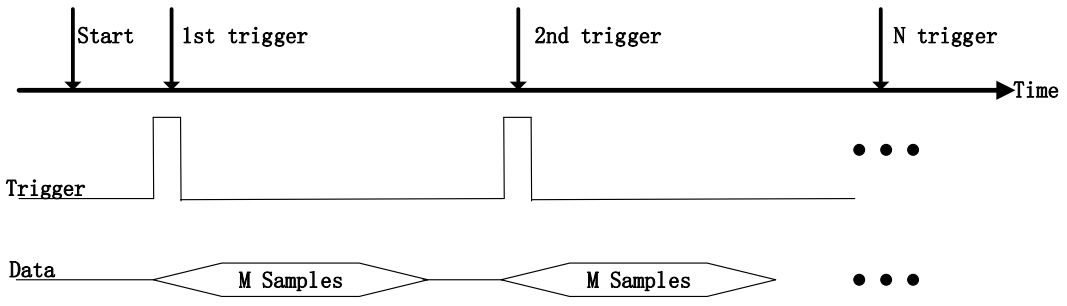


Figure 31 ReTrigger

Learn by Example 4.5

- Connect the signal source's positive terminal to JY5500 AI Ch0 (AI0+, Pin#68), the negative terminal to the ground (AI_GND, Pin#67) as shown in Figure 2 and Figure 3. (AI0+, AI_GND) consists of an RSE input.
- Set a sinewave signal (f=4Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Finite Analog Trigger**, set the following

numbers as shown.

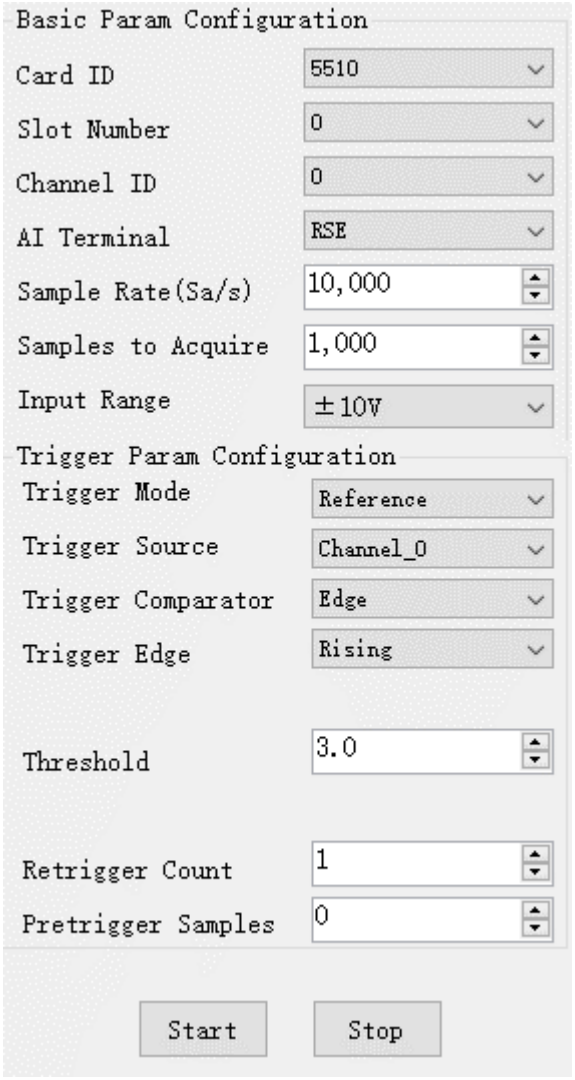


Figure 32 Retrigger Paraments

- You can use three different kinds of triggers in this program as mentioned in **4.5**. *Start Trigger* and *Reference Trigger* can be set by **Trigger Mode**. For *ReTrigger* can be used by changing the numbers in **Retrigger Count**.
- PretriggerSamples is set by **Pretrigger Samples**.

■ Now the trigger is a **Start Trigger**. Click **Start** to begin the data acquisition, the result is shown below:

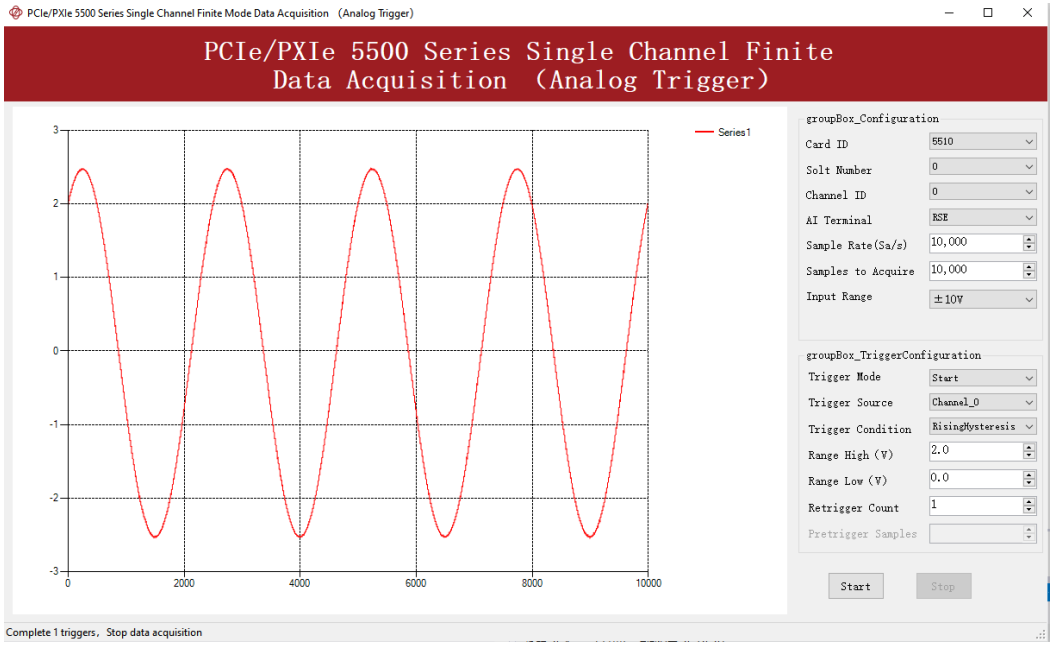


Figure 33 Retrigger In Start Trigger Mode

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

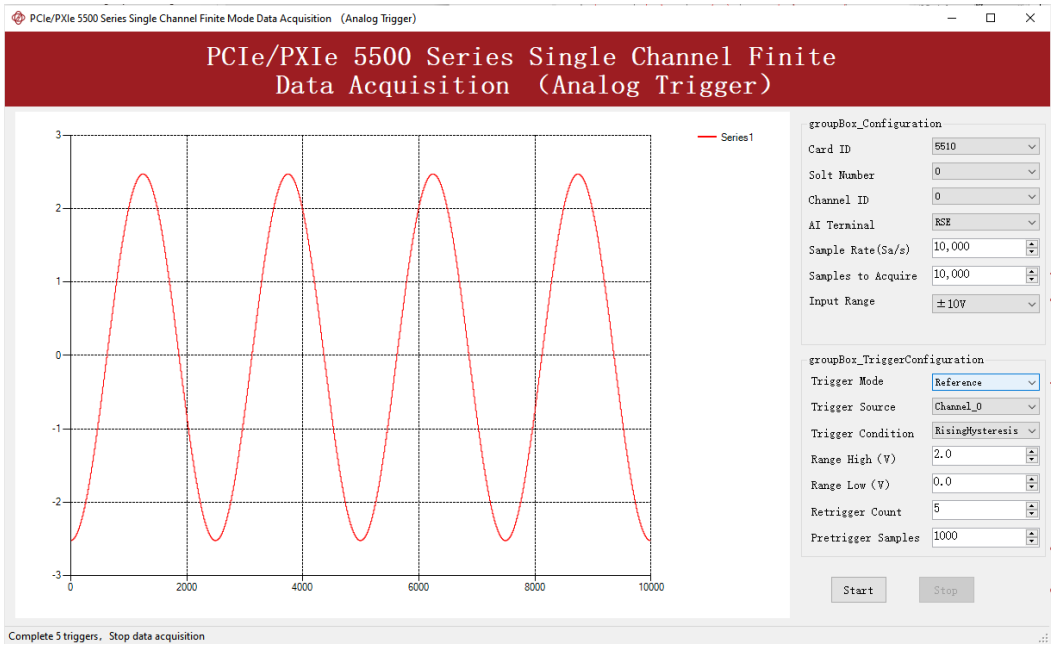


Figure 34 Retrigger In Reference Trigger Mode

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

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

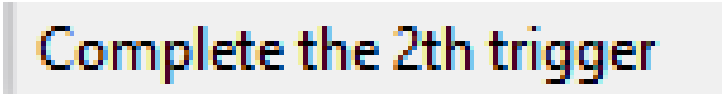


Figure 35 Complete Retrigger Count

➢ It shows the acquisition process through every trigger signal.

4.6 AO Operations

The JY5500 AO provides 16-bit simultaneous outputs. The analog output has three modes of operation: Finite, ContinuousWrapping, and ContinuousNoWrapping.

4.6.1 Finite Output

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

Learn by Example 4.6.1

- Connect JY5500 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.
  JY5500 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.

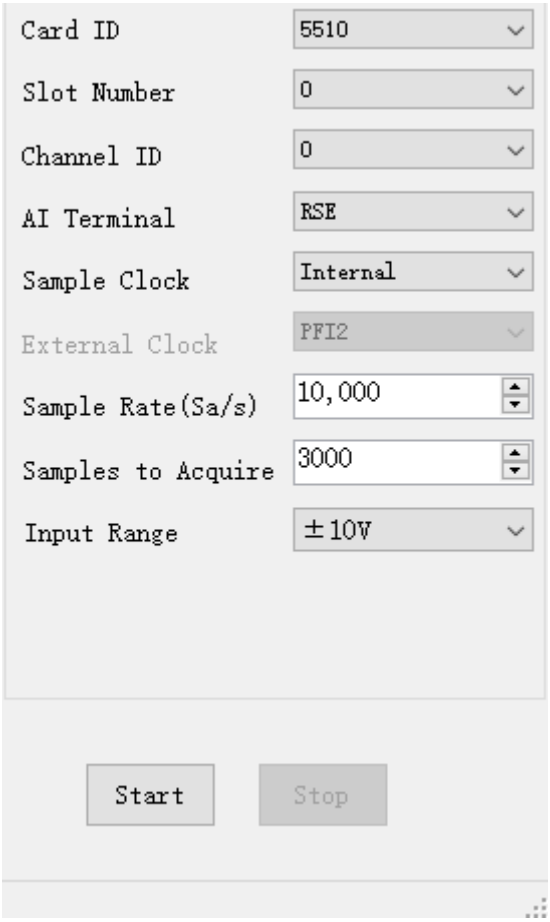


Figure 36 AI Continuous Paraments

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

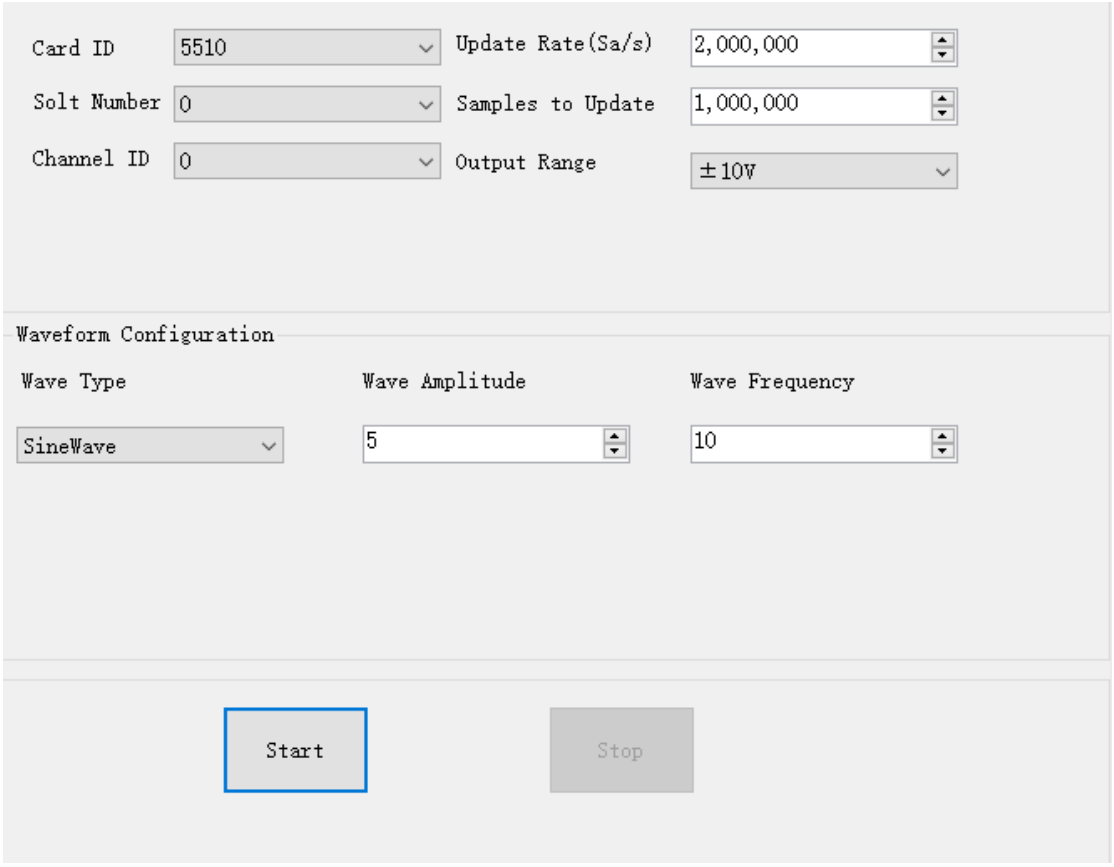


Figure 37 AO Finite Output Paraments

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

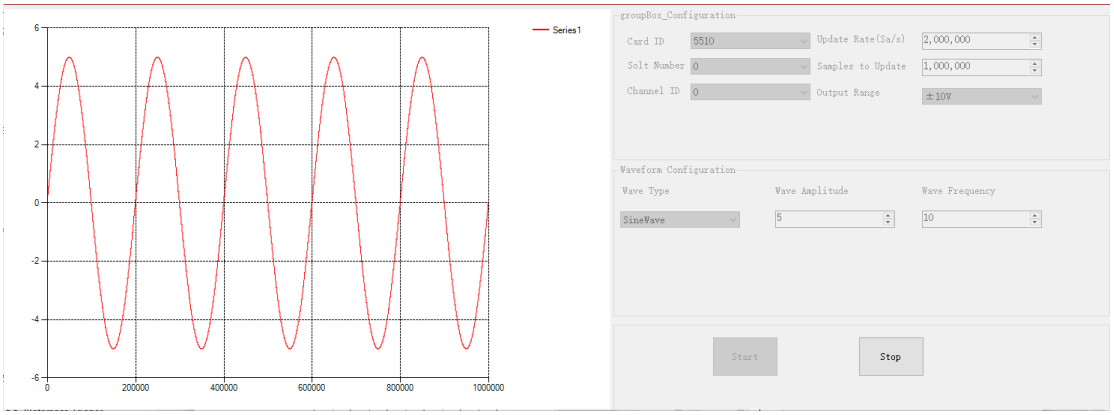


Figure 38 AO Finite Signal

- And the received signal is shown below.

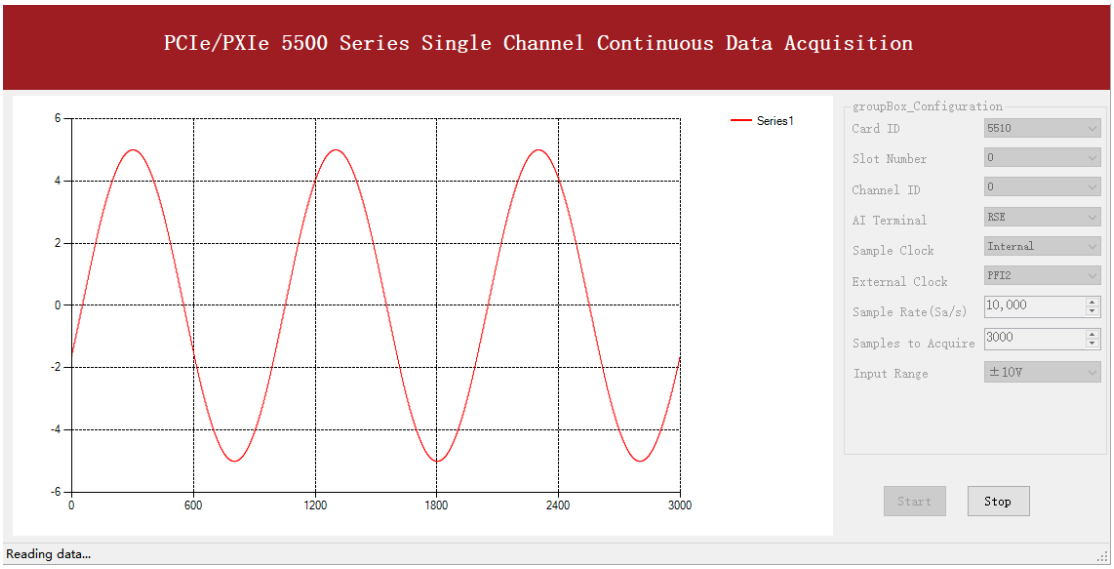


Figure 39 AI Acquisition Signal

➢ The analog signal is successfully generated and received by JY5500.

4.6.2 Continuous NoWrappping Output

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

Learn by Example 4.6.2

- Connect JY5500 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.
- JY5500 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.

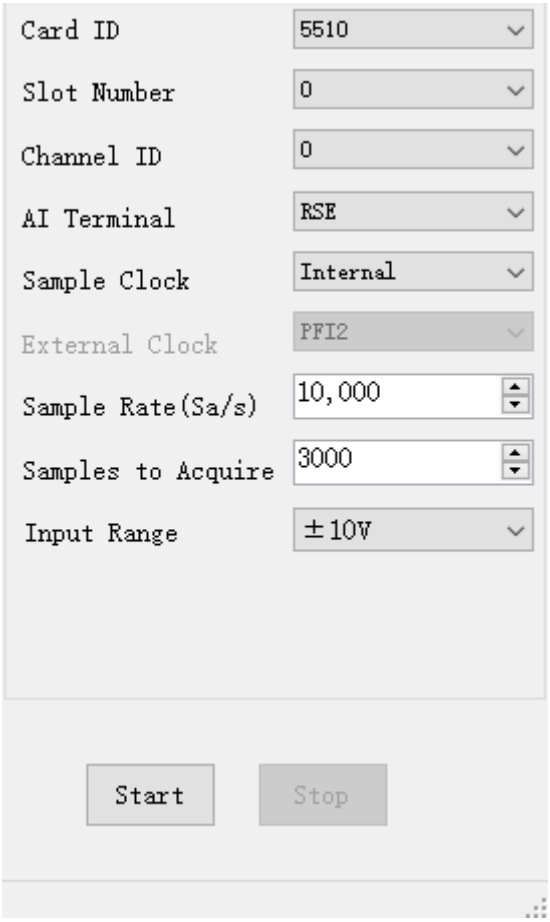


Figure 40 AI Continuous Paraments

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

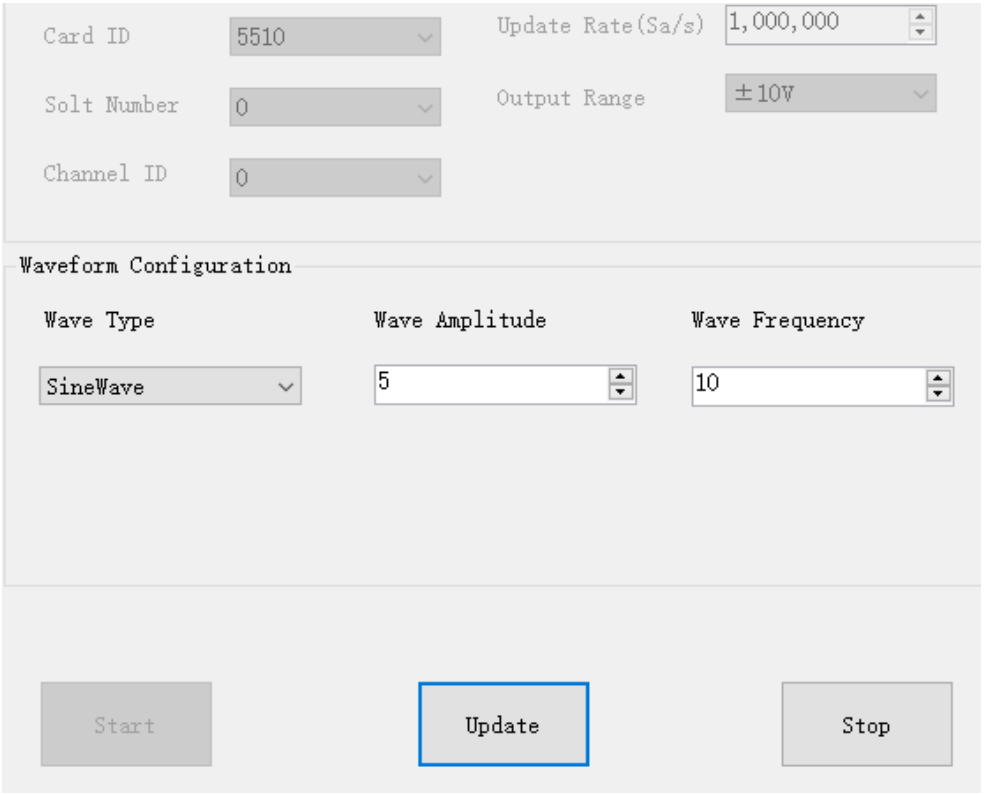


Figure 41 AO ContinuousNoWrapping Output Paraments

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

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

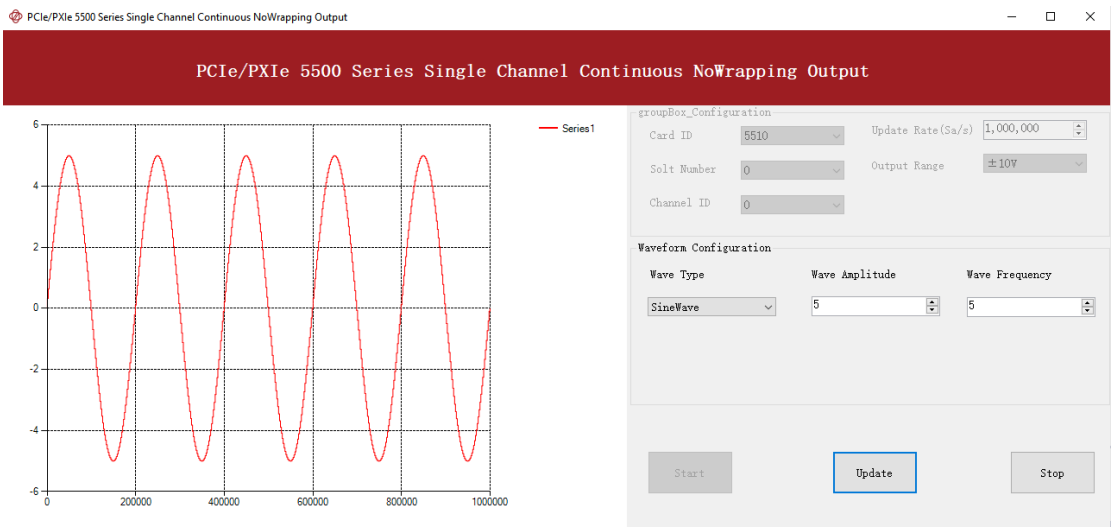


Figure 42 AO ContinuousNoWrapping Signal

- And the received signal is shown below.

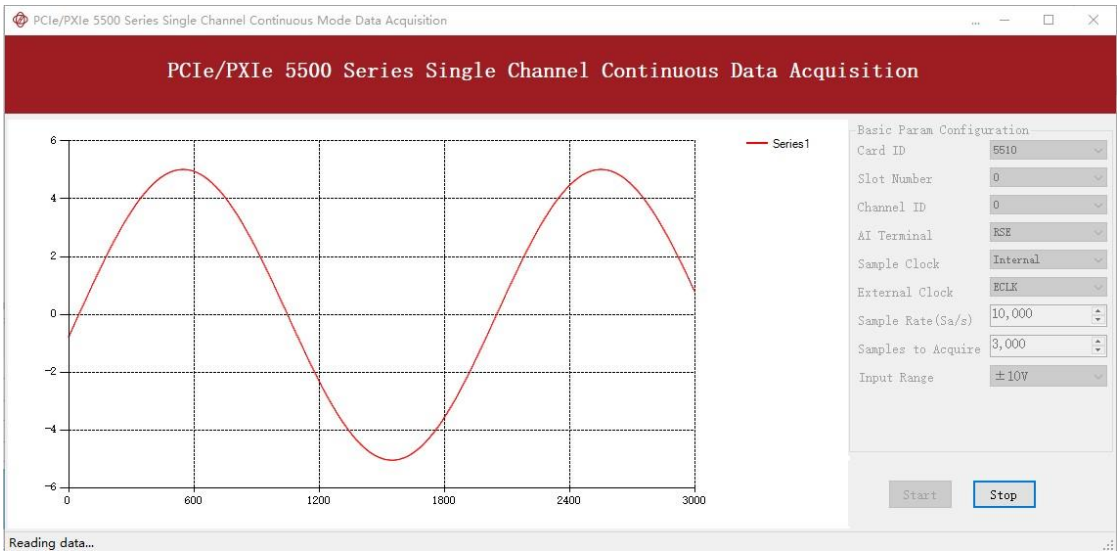


Figure 43 AI Acquisition AO Sin Signal

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

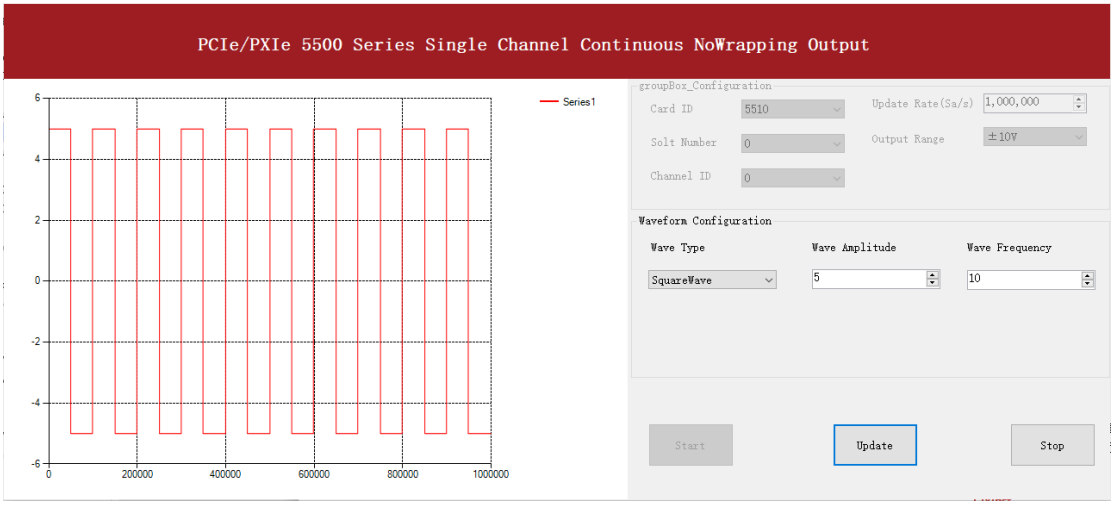


Figure 44 Update AO Square Signal

- And the received signal is shown below.

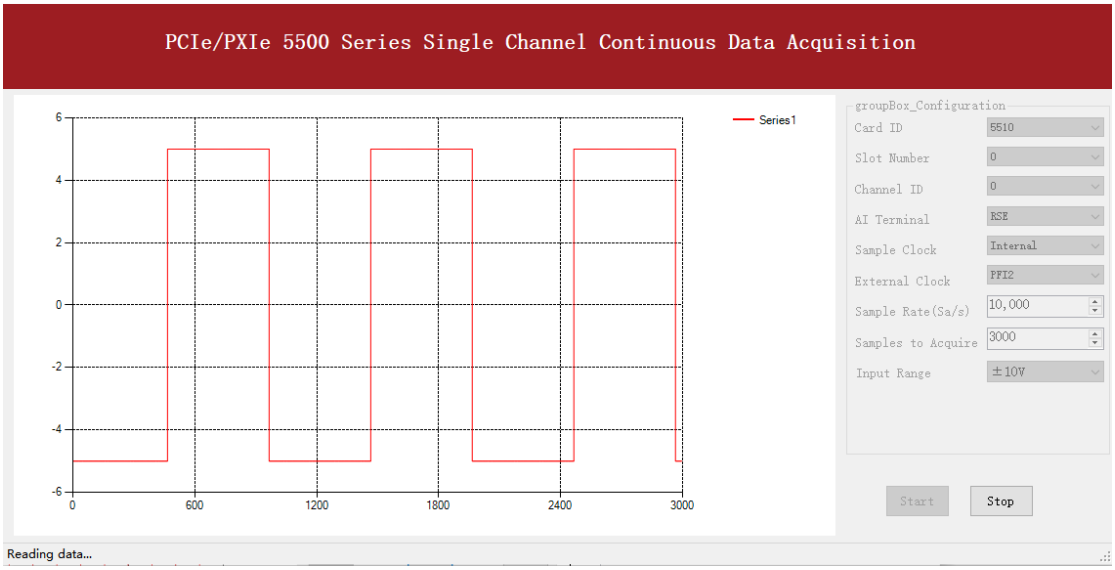


Figure 45 AI Acquisition AO Square Signal

➢ The analog signal is successfully generated and received by JY5500.

4.6.3 Continuous Wrapping Output

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

Learn by Example 4.6.3

- Connect JY5500 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.
- JY5500 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.

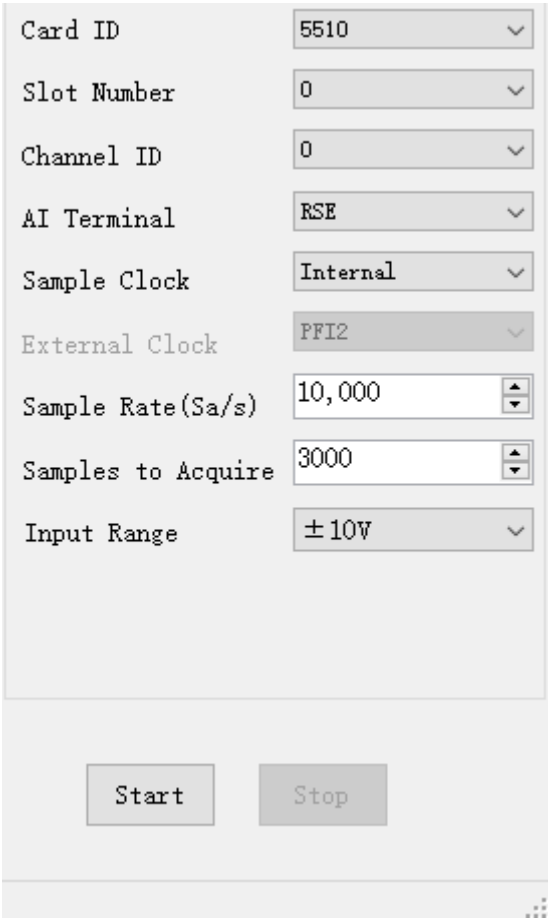


Figure 46 AI Continuous Paraments

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

Figure 47 AO Continuous Wrapping Paraments

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

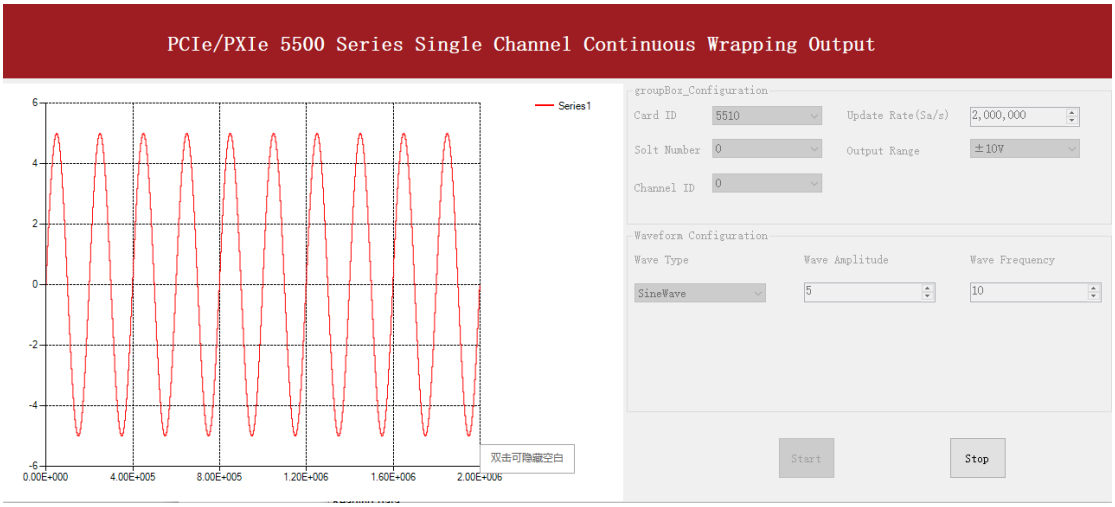


Figure 48 AO Continuous Wrapping Signal

■ And the received signal is shown below.

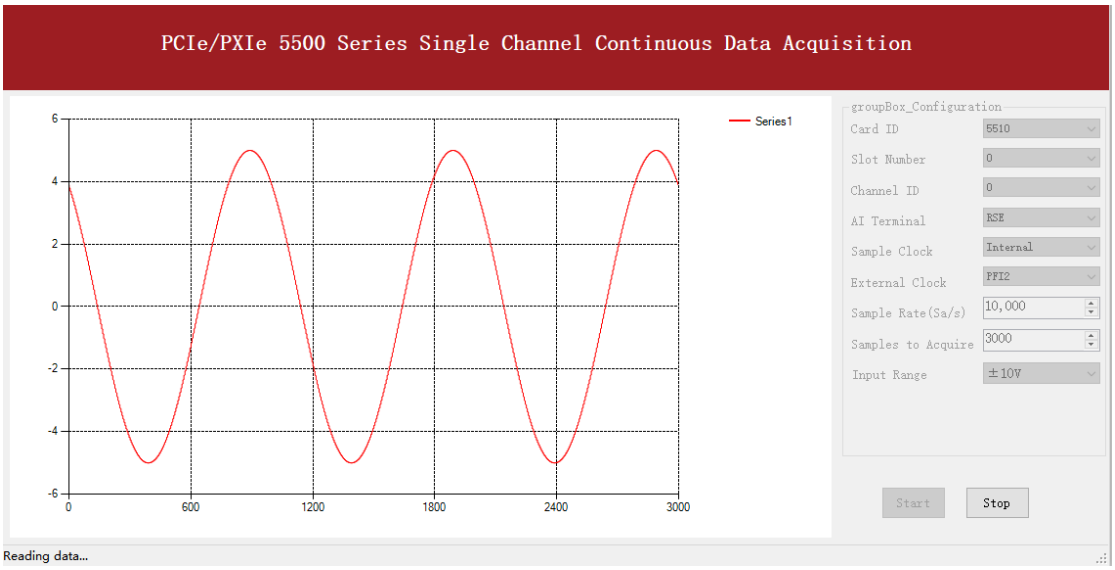


Figure 49 AI Acquisition AO Signal

➢ The analog signal is successfully generated and received by JY5500.

4.7 Digital I/O Operations

The JY5500 provides powerful programmable digital I/O functions.

4.7.1 Static DI/DO

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

Learn by Example 4.7.1

- In this example JY5500 outputs a digital signal by its DO function and reads it back by its DI function.
- Connect Connector1 of JY5500 to the TB-68 terminal block according to Figure 3.
- Connect Port 1/Line 0~7 (P1.0~P1.7) to Port 2/Line 0~7 (P2.0~2.7). JY5500 sends a digital signal through Port 1 and reads the signal back from Port 2.
- Open the first program **Digital Output-->Winform DO SinglePoint.**
- Select **port 1** for **Digital Output**, Set Line 1,3,5,7 in High-Level positions, make sure all other lines are in Low-Level positions. Click **Start** to generate the High-Levels as shown.

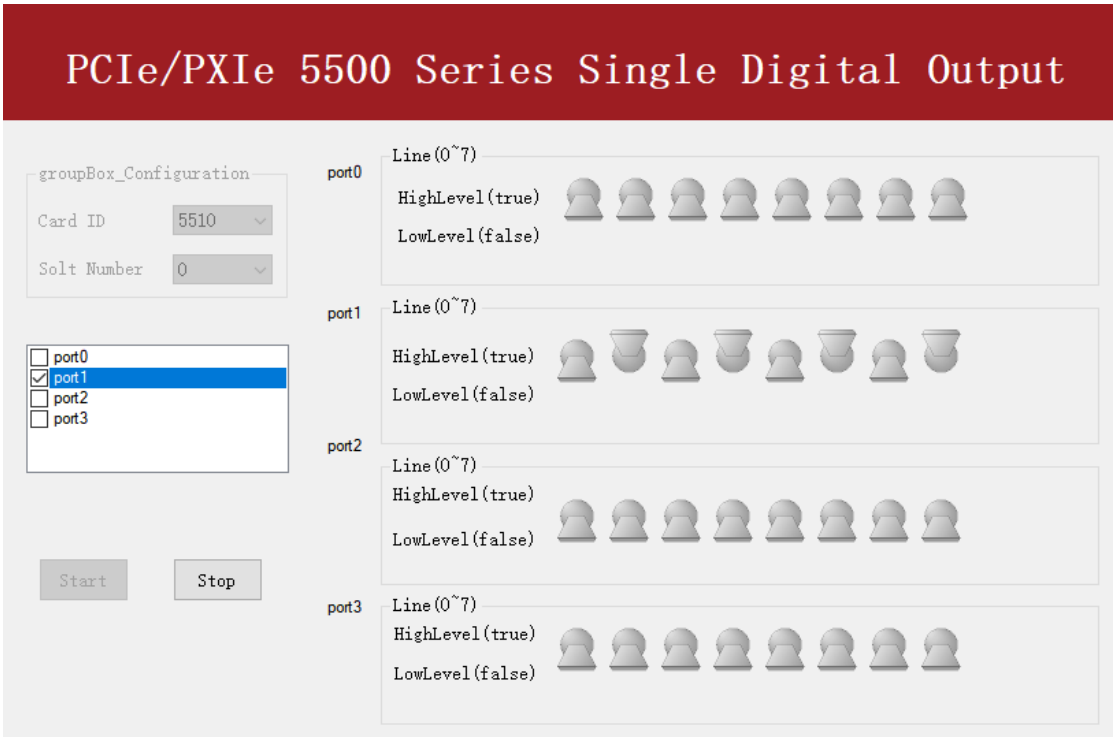


Figure 50 Single Digital Output

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

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

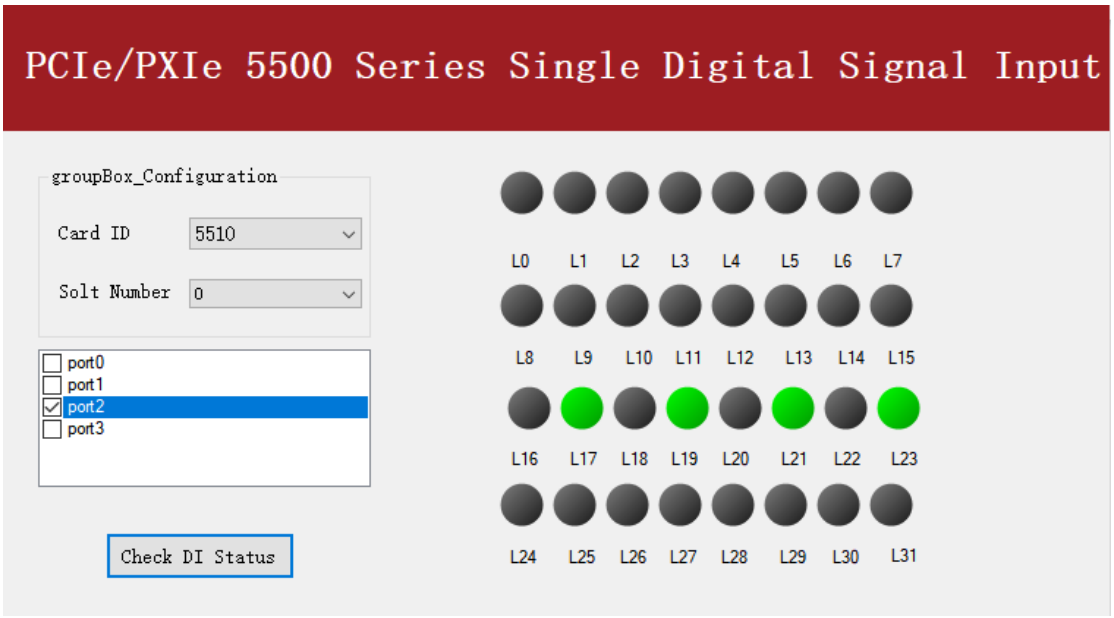


Figure 51 Single Digital Input

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

4.7.2 Dynamic DI/DO

The JY5500 supports both dynamic DI/DO operation with a maximum sample rate (update rate) of up to 10MHz. User can acquire or output digital waveforms in this way.

Learn by Example 4.7.2

- In this example JY5500 outputs a squarewave by its DO function and reads it back by its DI function.
- Connect Connector1 of PCIe/PXIe-5510/5511 or Connector0 of PCIe/PXIe-5515/5516 to the TB-68 terminal block according to Figure 3.
- Connect PCIe/PXIe-5510/5511 Port 1/Line 0~7 (P1.0~P1.7) to Port 2/Line 0~7 (P2.0~2.7) or PCIe/PXIe-5515/5516 Port 0/Line 0~7 (P0.0~P0.7) to Port 1/Line 0~7 (P1.0~1.7). JY5500 sends digital signals through Port 1/Port 0 and reads them back from Port 2/Port 1.
- Open **Digital Input-->Winform DI Continuous** and set the numbers as shown. Select **port 2**(PCIe/PXIe-5510/5511) or **port 1**(PCIe/PXIe-5515/5516).

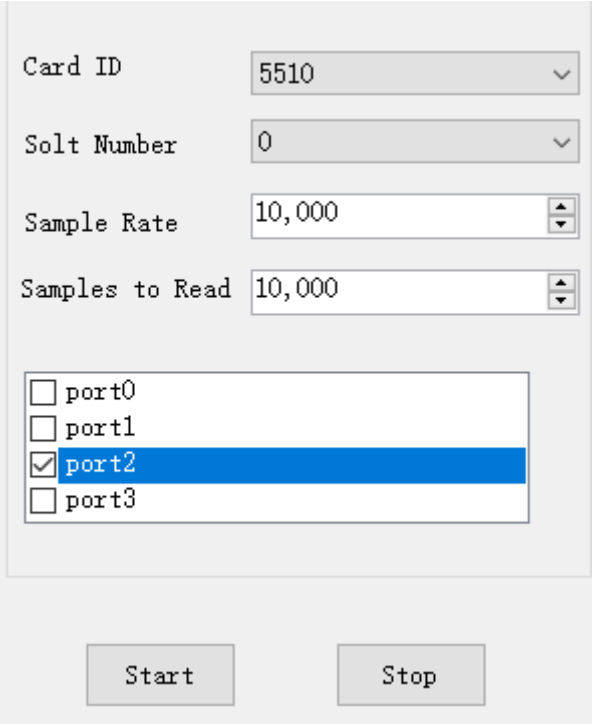


Figure 52 DI Continuous Paraments

- Click **Start** to begin the data acquisition.

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

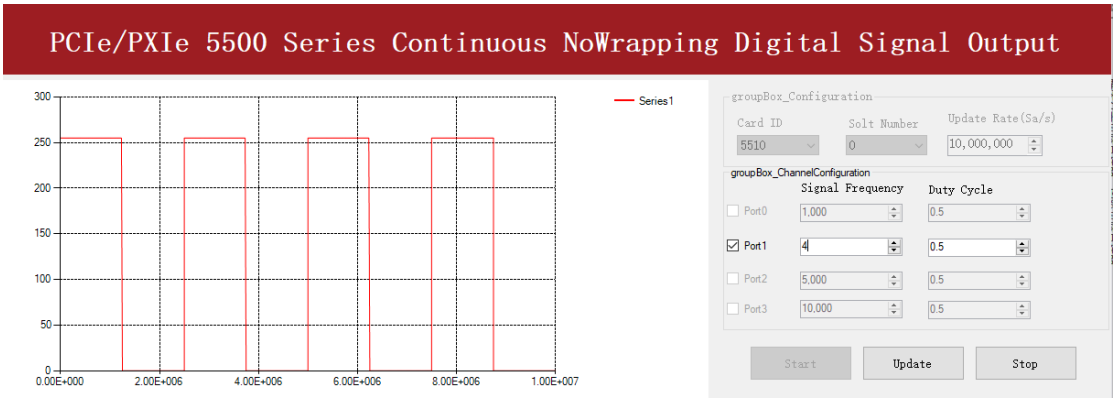


Figure 53 DO ContinuousNoWrapping Output

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

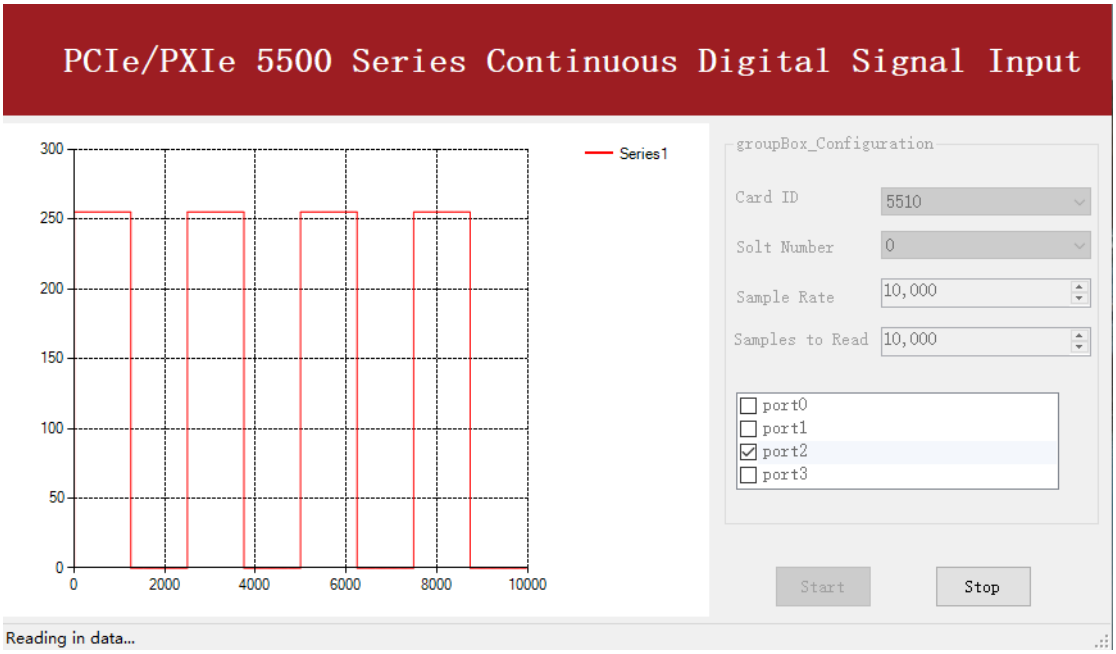


Figure 54 DI Continuous Acquisition

- The digital signal is successfully generated and acquired by JY5500.

4.8 Counter Input Operations

The JY5500 has four or two identical 32 bits timers/counters.

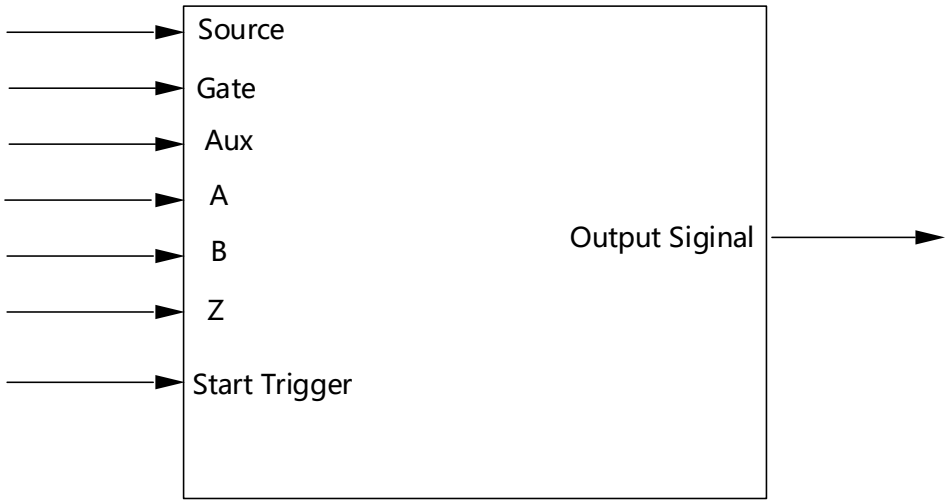


Figure 55 Counter Terminal

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

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

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

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

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

Figure 56 Counter Signal Wiring Instruction

4.8.1 Edge Counting

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

Timing

1) Single Mode

The count value is written to the register on each rising edge or falling edge of the signal to measure as shown in Figure 57.

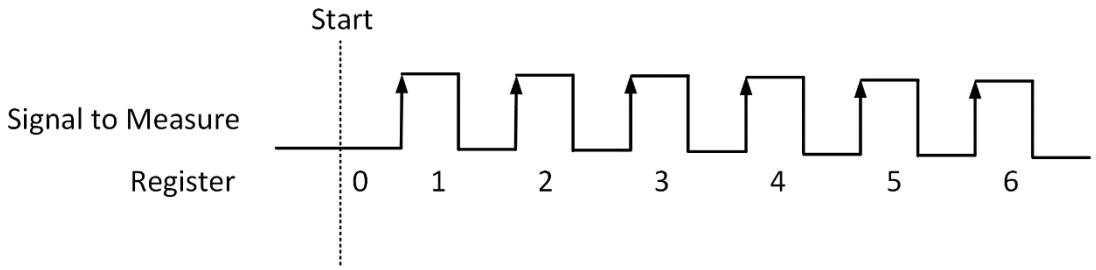


Figure 57 Simple Edge Counting in Single Mode

2) Finite/Continuous Mode with Internal Sample Clock

The count value is stored into the buffer on each rising edge or falling edge of the sample clock as shown in Figure 58.

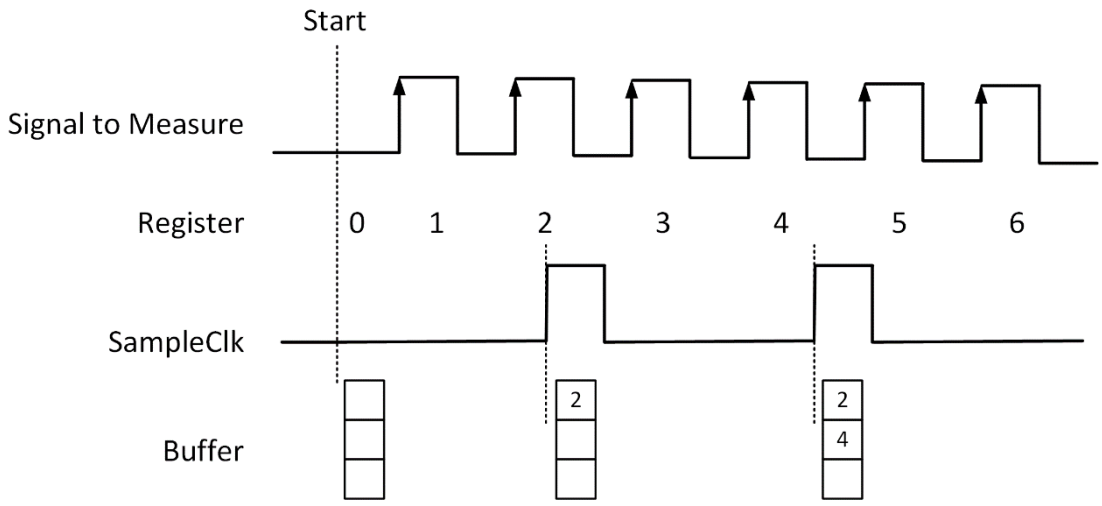


Figure 58 Buffered Edge Counting with Internal Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

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

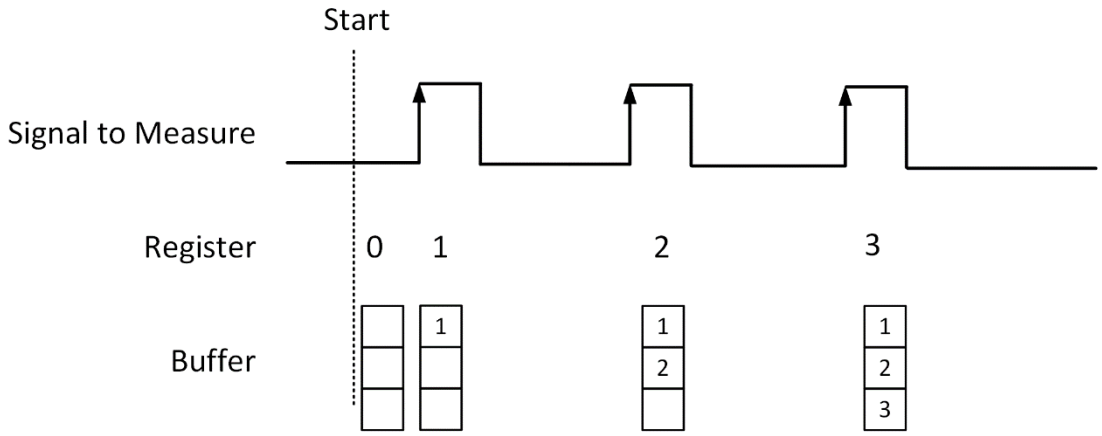


Figure 59 Simple Edge Counting with Implicit SampleClk

Counting Direction

User can control the counting direction through software configuration or by an input signal with AUX terminal. When using an input signal to control the counting direction, the counter counts up when the signal is high and counts down when the signal is low as shown in Figure 60.

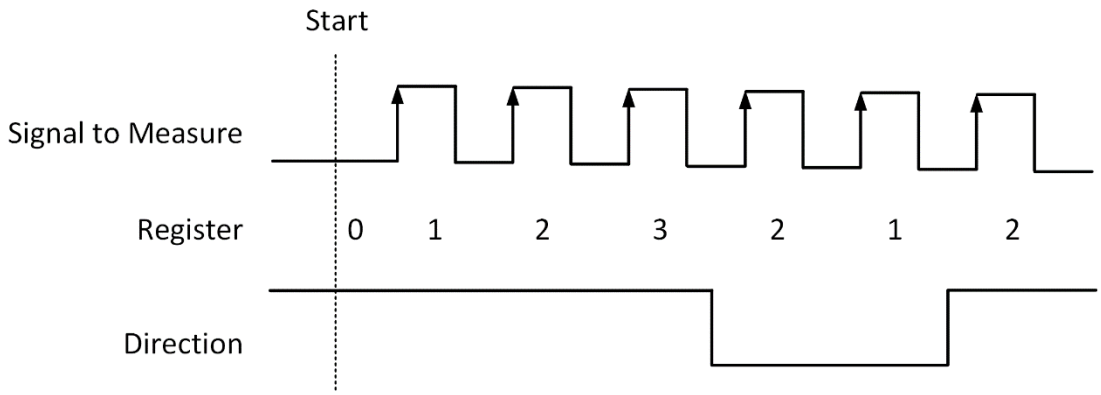


Figure 60 Count Direction

Learn by Examples4.8.1

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

Single Mode

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

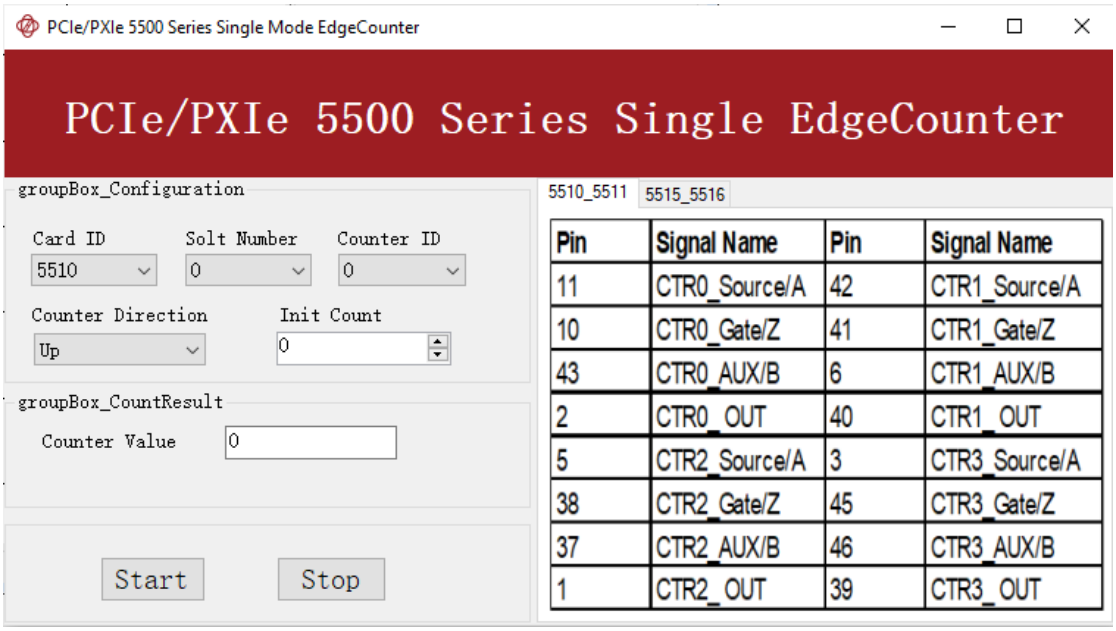


Figure 61 EdgeCounting For Single Mode

➢ Counter Direction is set by **Counter Direction**.

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

- The *rising edge counter* works when **Start** is clicked.
- The result is shown by **Counter Value**. In this example the **Counter Value** increases by 1 every second for a 1Hz sinewave.

Finite/Continuous Mode

- Change the squarewave frequency to 50 Hz.

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

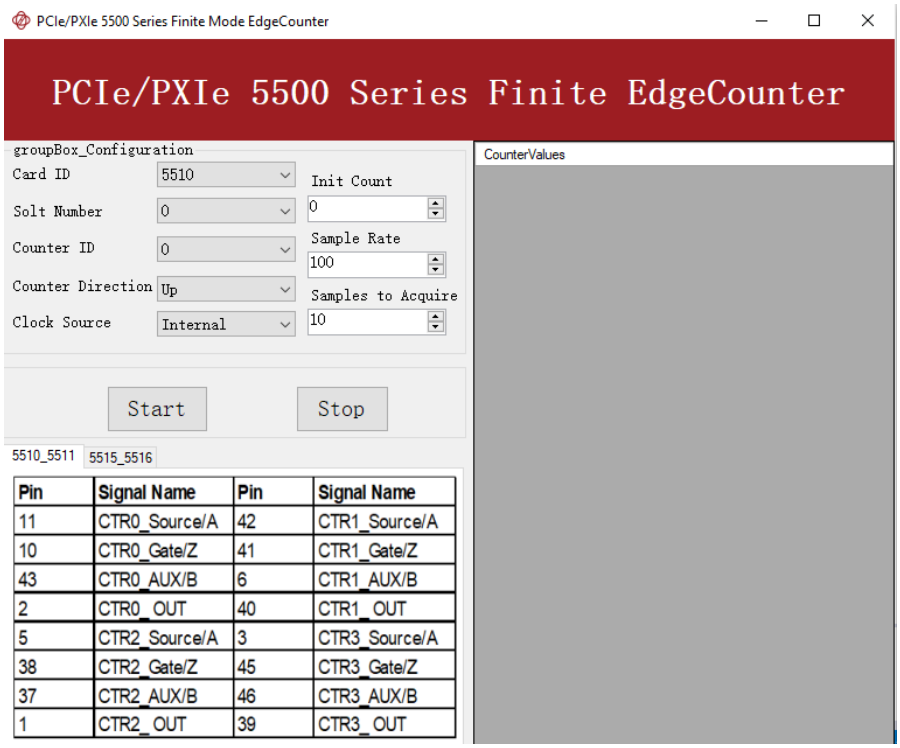


Figure 62 EdgeCounting For Finite Mode

➢ The table in the sample program is a connection diagram for your convenience.
➢ Counter Direction is set by **Counter Direction**.
➢ There are two clock sources in JY5500 *Internal* and *Implicit*: This example uses **Internal** mode set by **Clock Source**.

- Click **Start** to start counting by rising edge. The result is shown below:

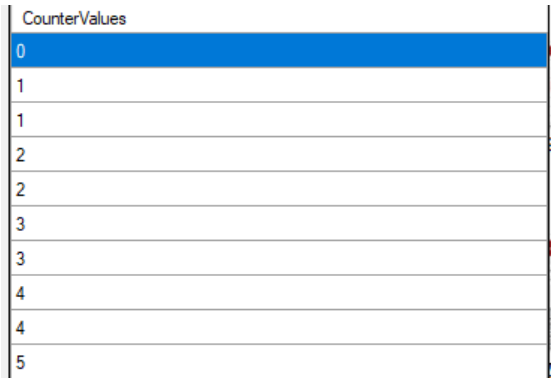


Figure 63 Counter Values For Internal Clock

- The numbers are stored in a buffer **CounterValues**.

■ Change the **Clock Source** to **Implicit**:

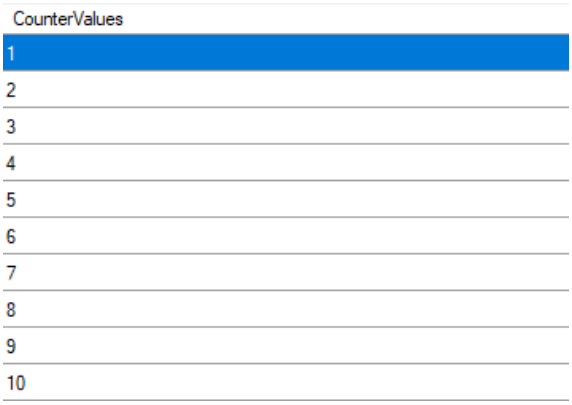


Figure 64 Counter Values For Implicit Clock

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

4.8.2 Pulse Measurement

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

Timing

1) Single Mode

The count value of the duration of the high-level or low-level is written to the register on each rising or falling edge of the pulse to measure, as shown in Figure 65.

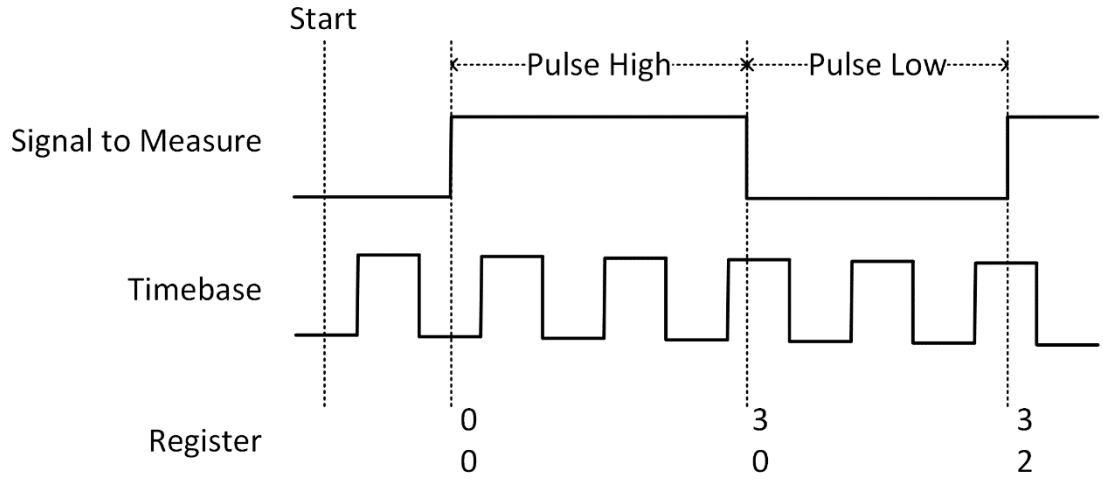


Figure 65 Pulse Measurement in Single Mode

2) Finite/Continuous Mode with Internal Sample Clock

The count value of the duration of the high or low level is stored into the buffer on each rising or falling edge of the sample clock, as shown in Figure 66.

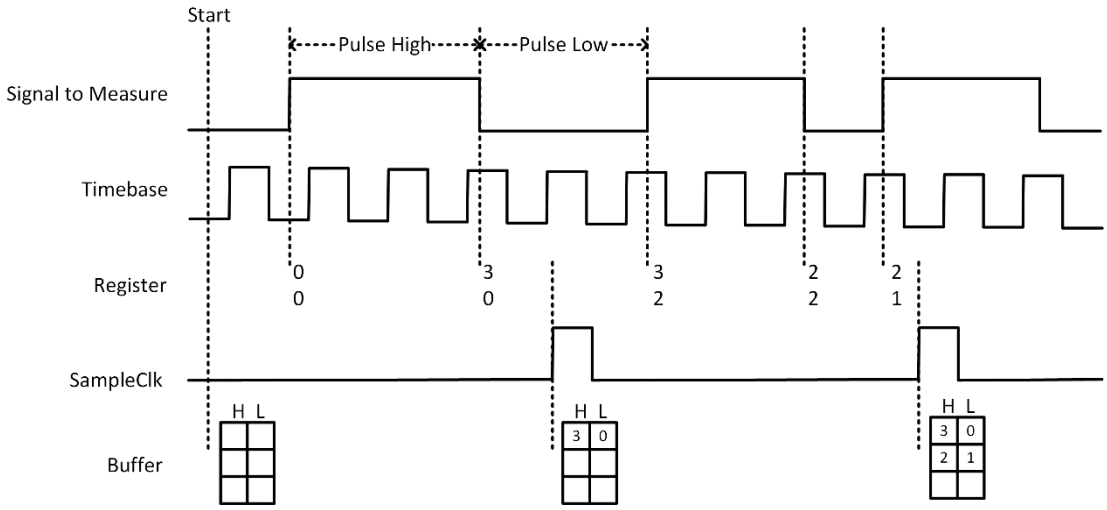


Figure 66 Pulse Measurement with Internal SampleClk

3) Finite/Continuous Mode with Implicit Sample Clock

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

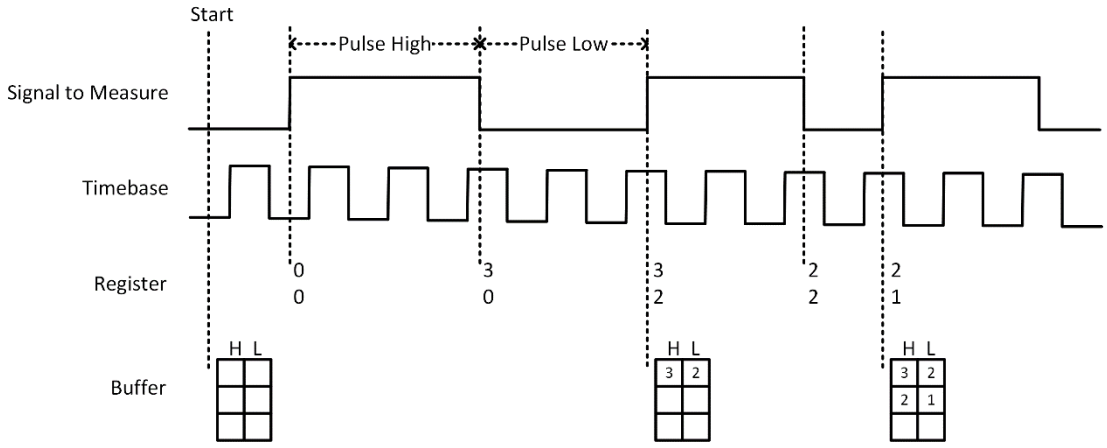


Figure 67 Pulse Measurement with Implicit SampleClk

Learn by Examples 4.8.2

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

Single Mode

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

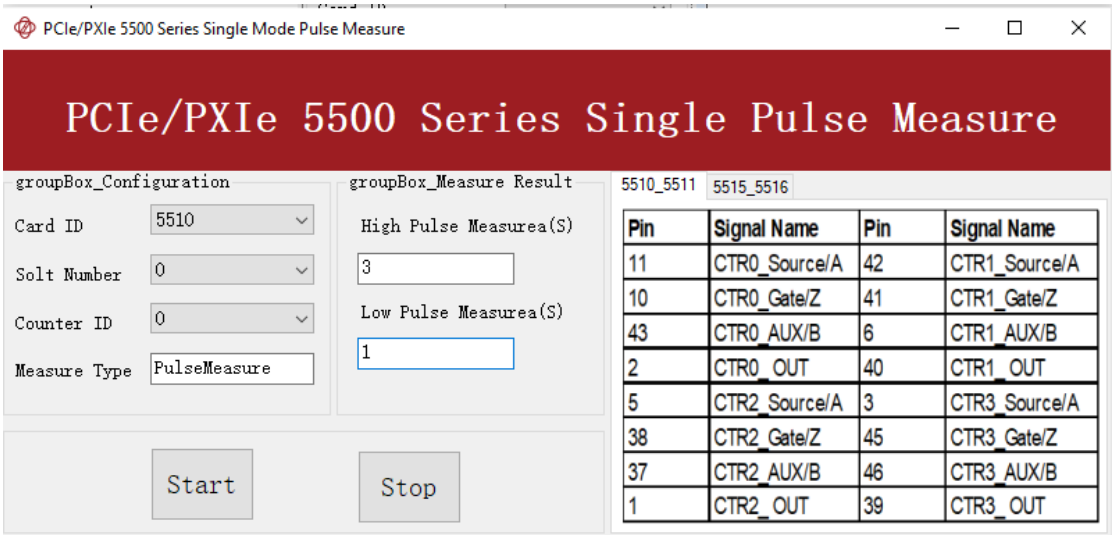


Figure 68 Pulse Measure For Single Mode

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

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

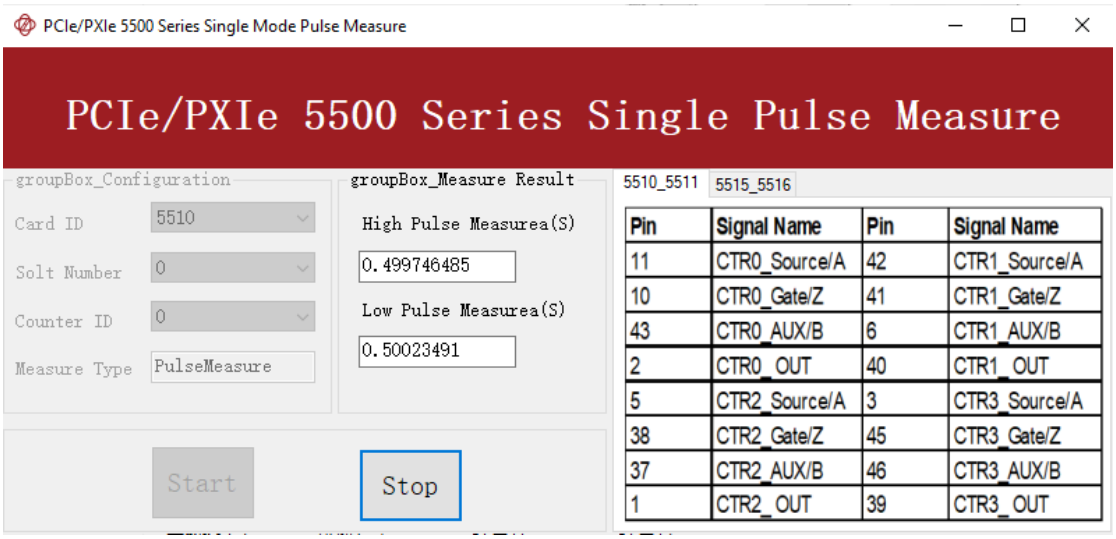


Figure 69 Pulse Measure Value For Single Mode

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

Finite/Continuous Mode

- Change the frequency of Squarewave to 50 Hz.

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

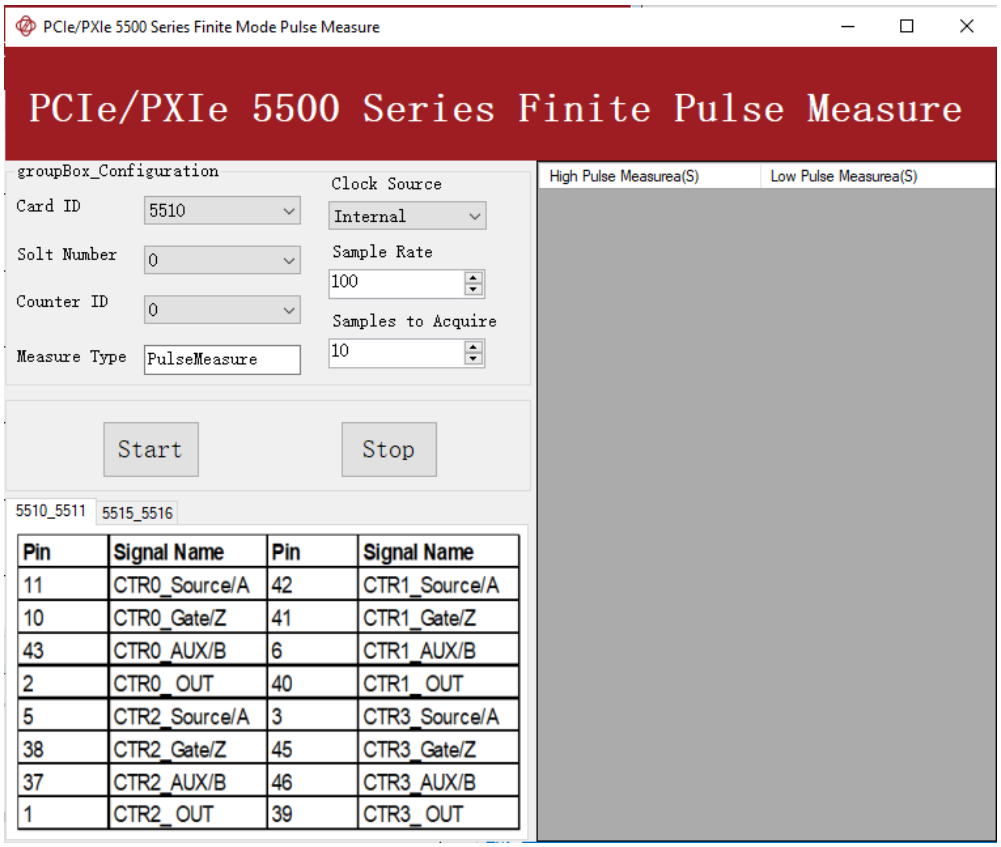


Figure 70 Pulse Measure For Finite Mode

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

■ Click Start to begin the finite/continuous pulse measurement. The result is shown below:

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

Figure 71 Pulse Measure Values For Finite Mode

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

- Please refer to **Learn by Examples4.8.1 Finite/Continuous** Mode about the difference between Internal and Implicit.

4.8.3 Frequency Measurement

The counter measures the frequency of signal to measure.

Timing

1) Single Mode

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

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

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

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

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

Between every two rising edges of the sample clock, the counter counts the numbers of full periods ($T1$) of the signal to measure, and the number of rising edges of timebase ($T2$) during those full periods. These two values are stored into the buffer on each rising edge of the sample clock, as shown in Figure 72.

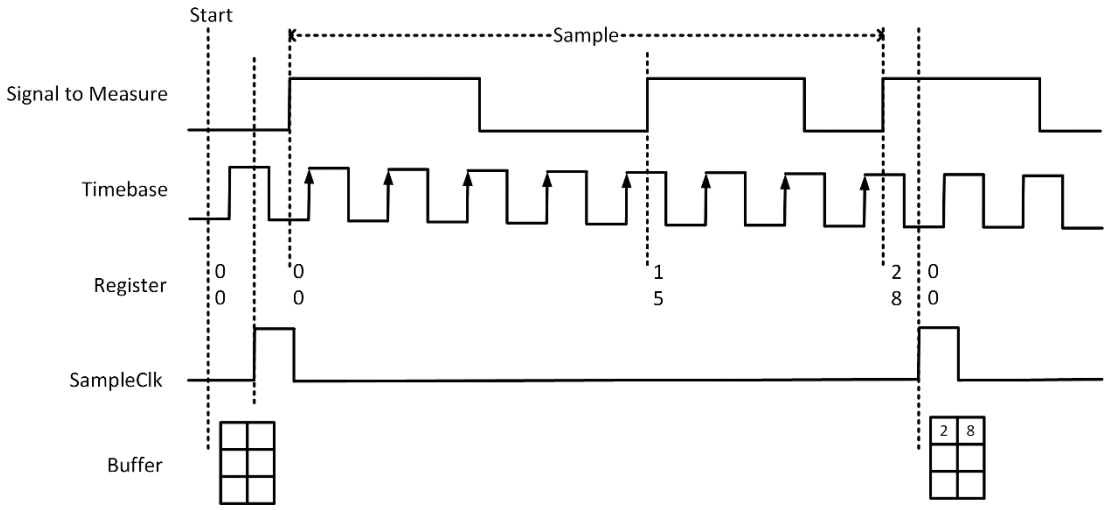


Figure 72 Frequency Measurement with Internal Sample Clock

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

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

3) Finite/Continuous Mode with Implicit Sample Clock

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

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

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

Learn by Examples 4.8.3

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

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

Single Mode

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

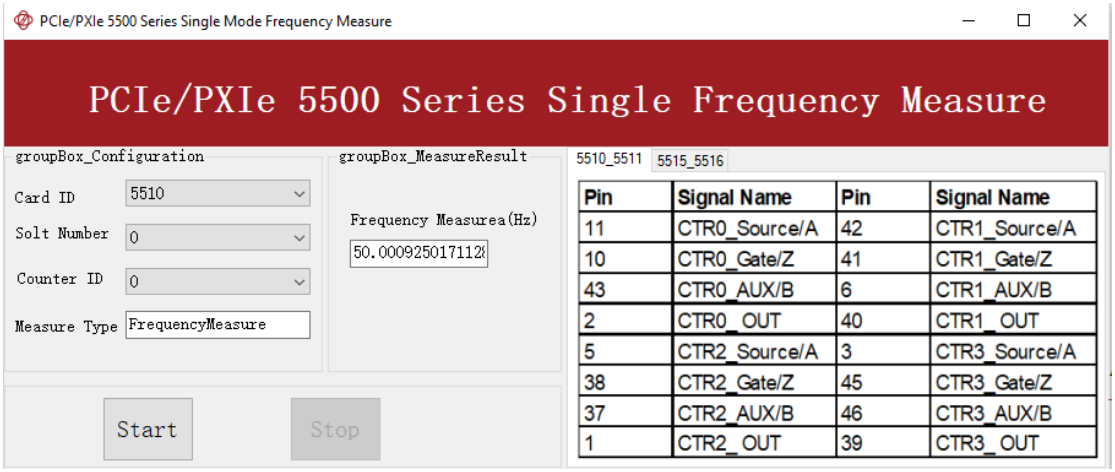


Figure 73 Frequency Measure For Single Mode

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

Finite/Continuous Mode

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

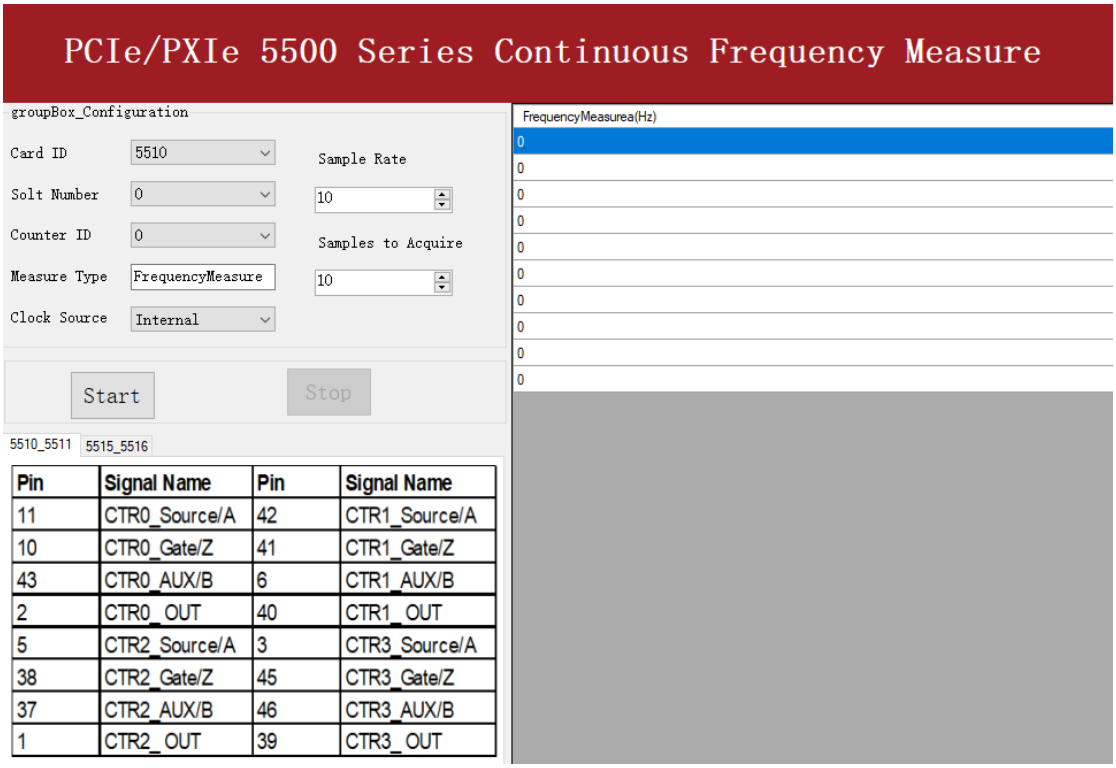


Figure 74 Frequency Measure For Continuous Mode

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

- Click Start and it will show the frequency 50 as set in the signal resource.

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

Figure 75 Frequency Measure Values

4.8.4 Period Measurement

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

Learn by Examples 4.8.4

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

- Set a squarewave signal (f=200Hz, Duty Cycle=50%, Vpp=5V).

Single Mode

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

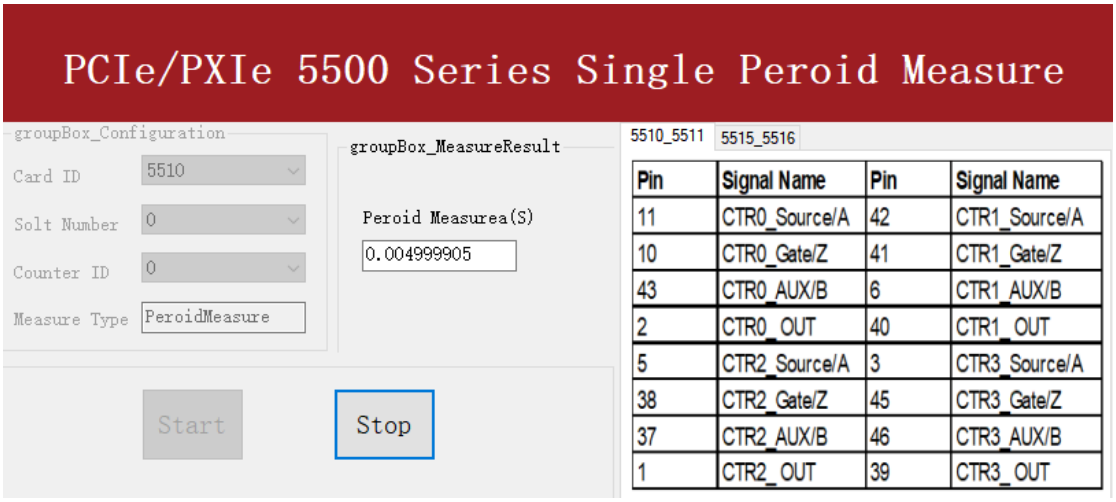


Figure 76 Peroid Measure For Single Mode

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

Finite/Continuous Mode

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

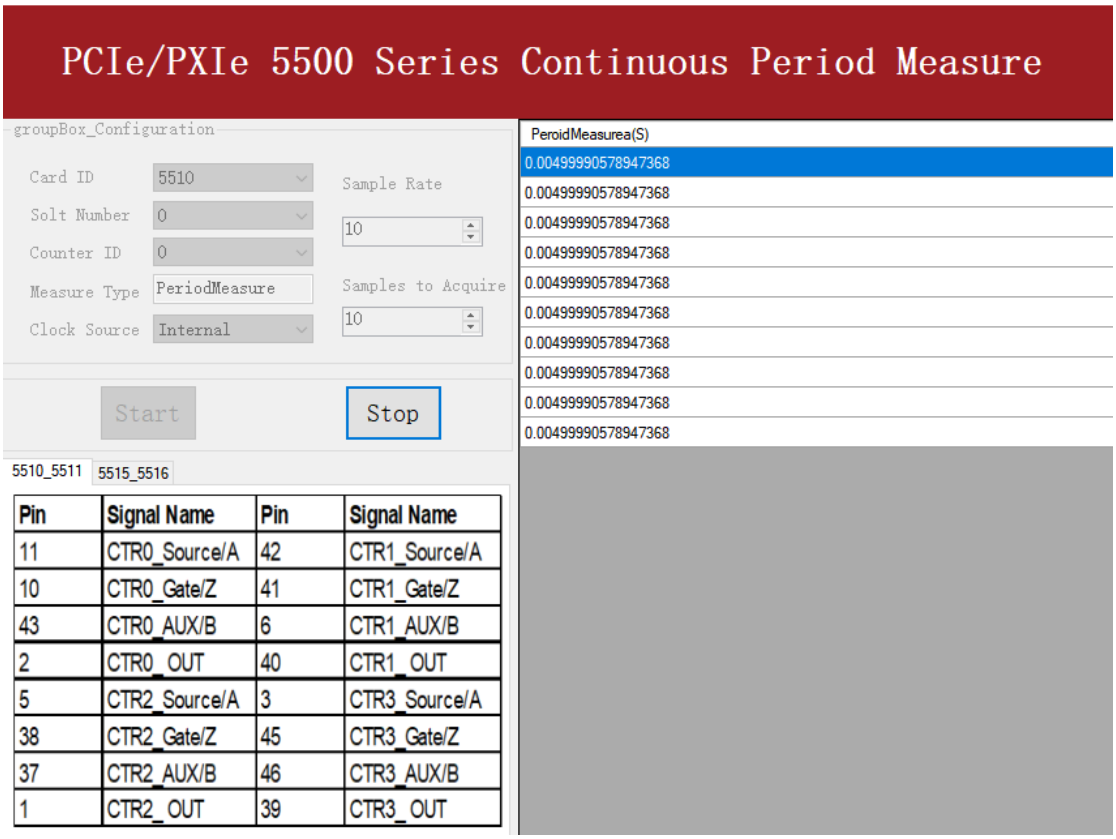


Figure 77 Peroid Measure For Continuous Mode

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

4.8.5 Two-Edge Separation

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

Timing

1) Single Mode

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

The number of rising edges of timebase between previous rising edge of the second signal and current rising edge of the first signal is written to the register on each rising edge of the first signal as shown in Figure 78.

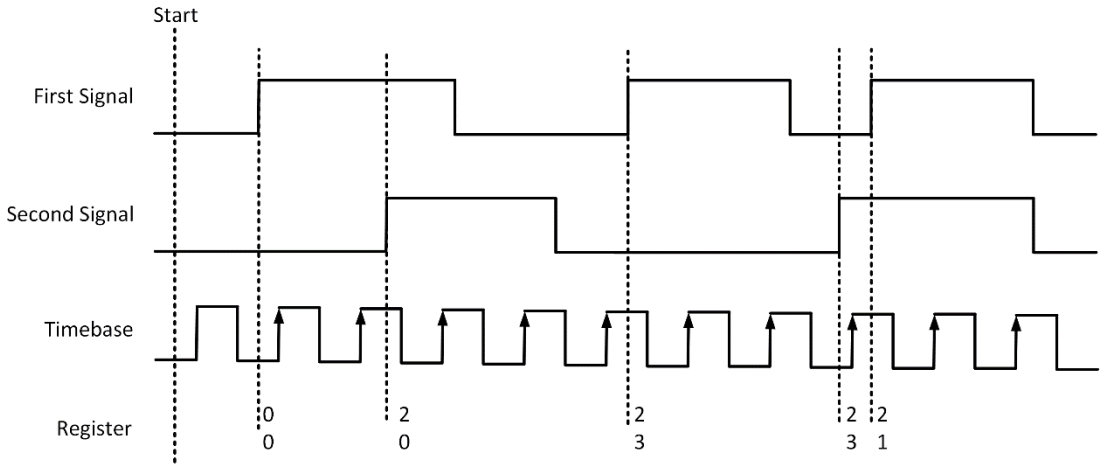


Figure 78 Two-Edge Separation in Single Mode

2) Finite/Continuous Mode with Internal Sample Clock:

The count values of rising edges of timebase between first signal and second signal are stored into buffer on each rising edge of the sample clock, as shown in Figure 79.

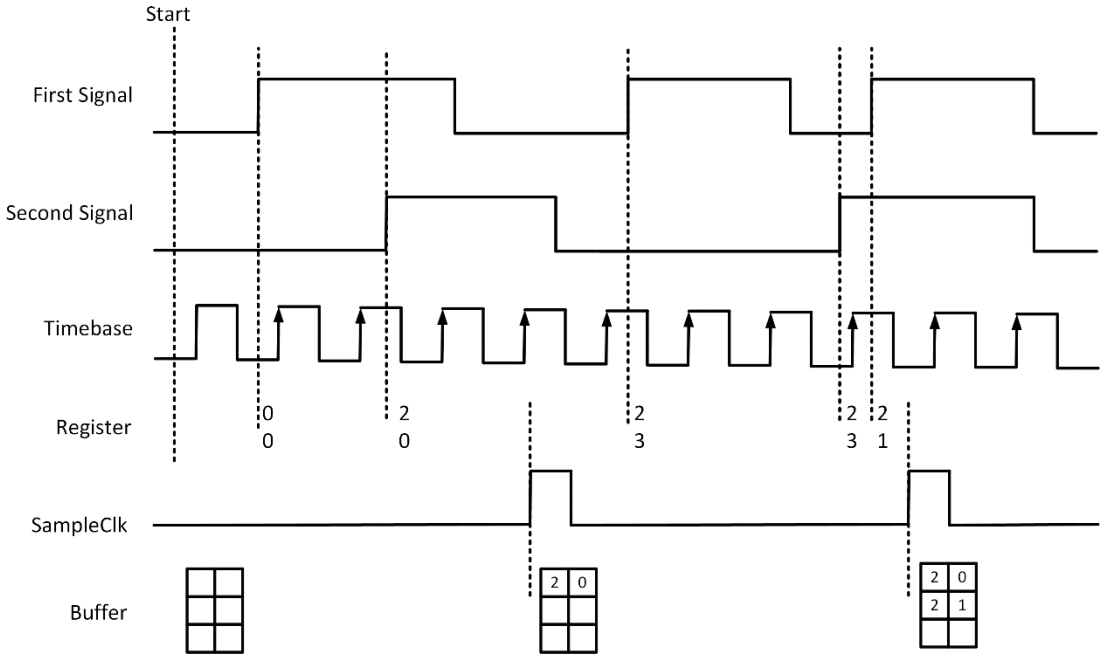


Figure 79Two-Edge Separation with Internal Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

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

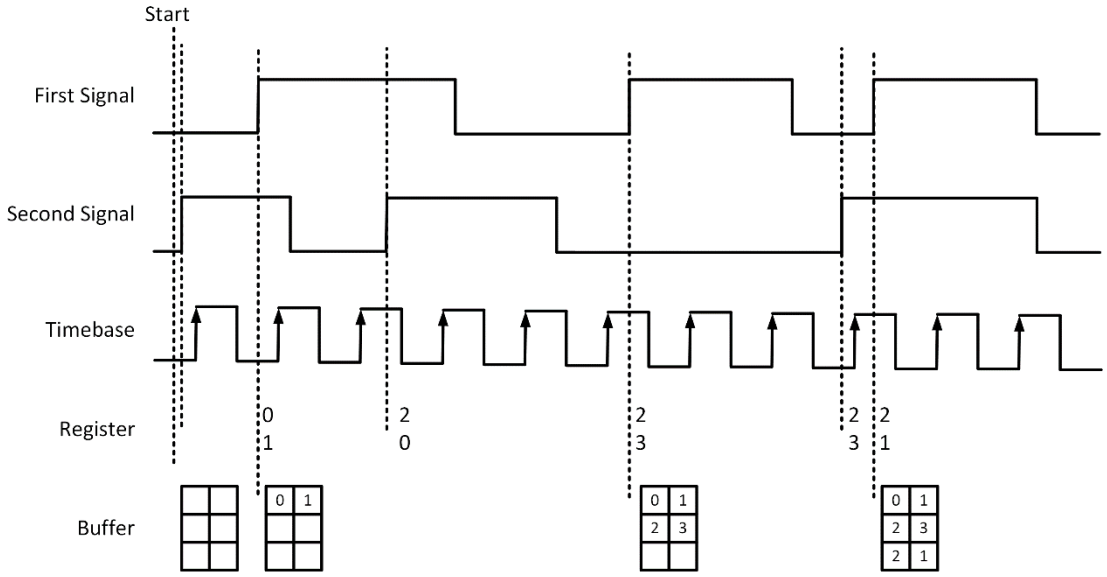


Figure 80 Two-Edge Separation with Implicit Sample Clock

Learn by Examples 4.8.5

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

Single Mode

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

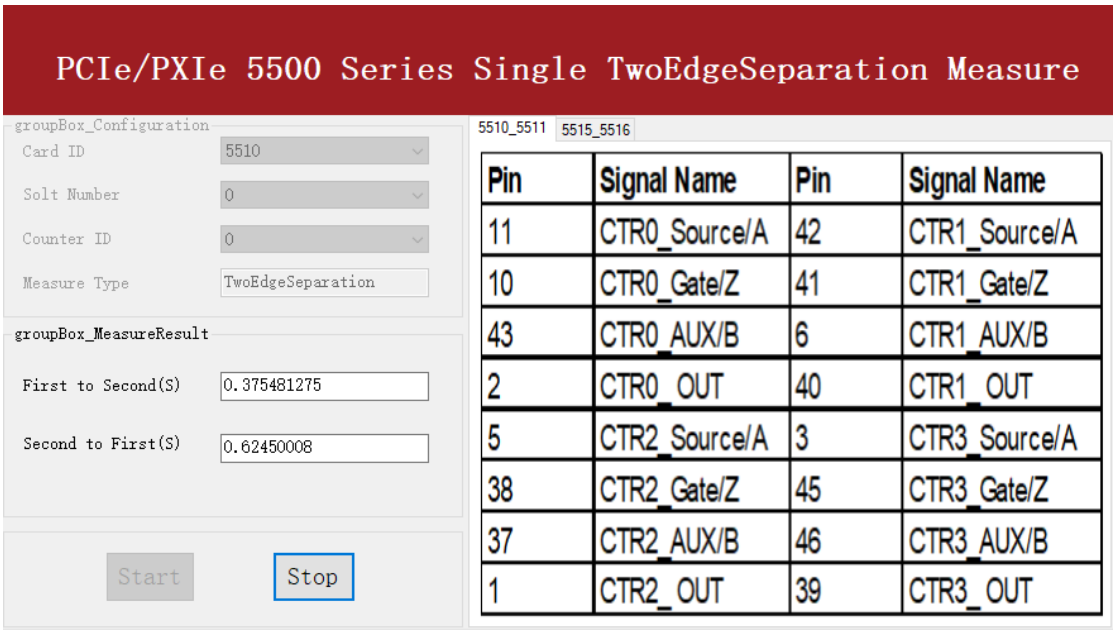


Figure 81 Two-EdgeSeparation Measure For Single Mode

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

Finite/Continuous Mode

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

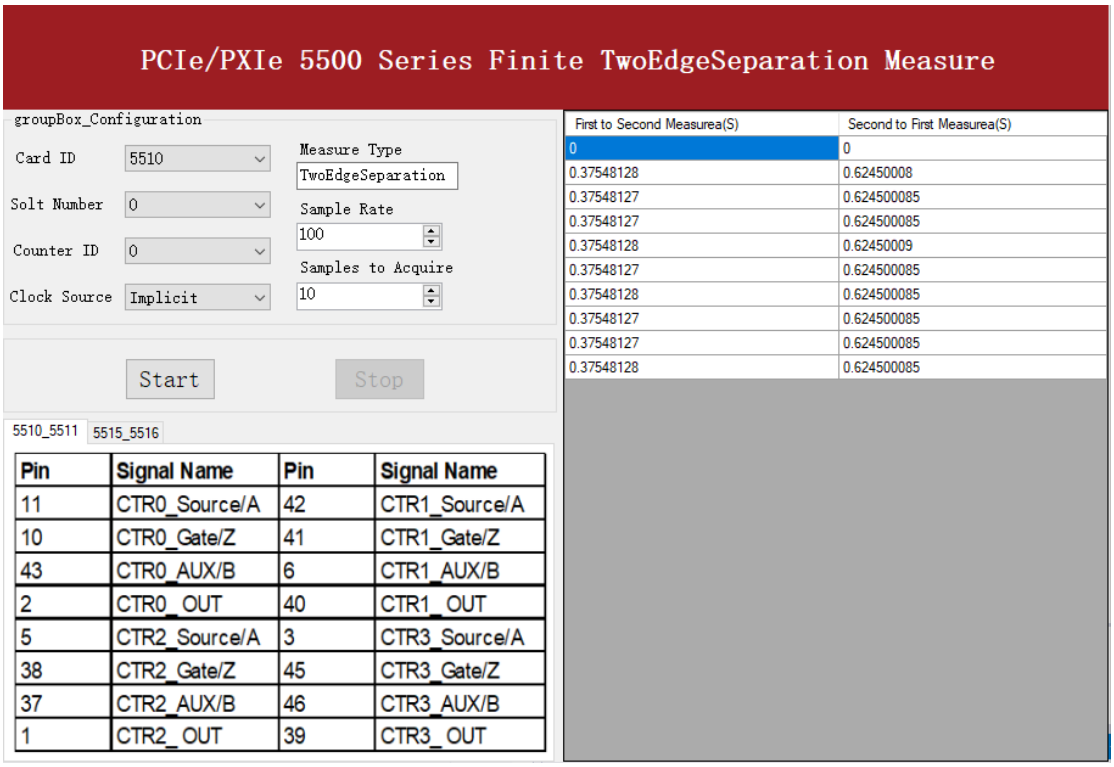


Figure 82 Two-EdgeSeparation Measure For Finite Mode

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

4.8.6 Quadrature Encoder

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

Encoding Type

1) x1 Encoding

When A leads B, the count increase occurs on the rising edge of A; when B leads A, the count decrease occurs on the falling edge of A as shown in Figure 83.

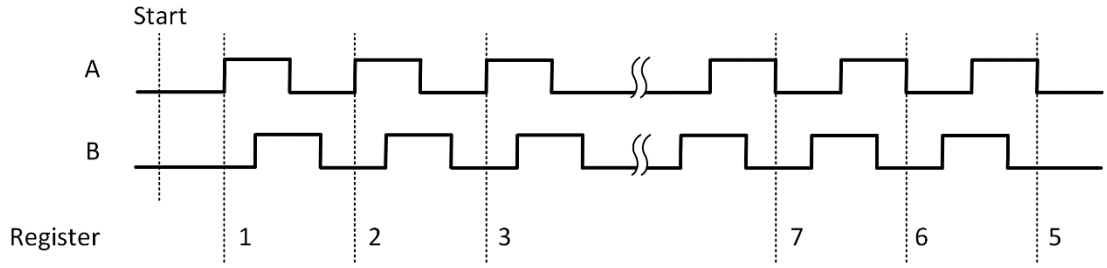


Figure 83 Quadrature Encoder x1 Mode

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 84.

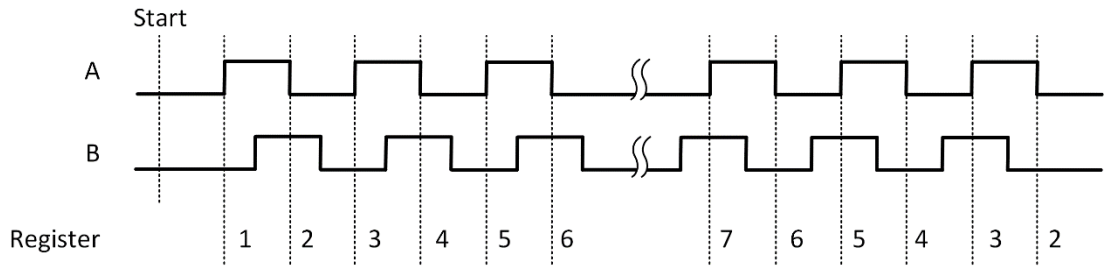


Figure 84 Quadrature Encoder x2 Mode

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 85.

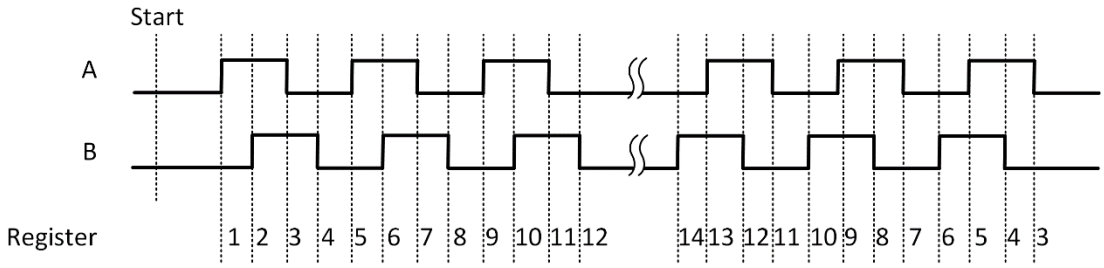


Figure 85 Quadrature Encoder x4 mode

Channel Z Behavior

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

Timing

Take Encoding x1 mode as an example.

1) Single Mode

The count value is written to the register on each rising edge of the signal A, as shown in Figure 57.

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

2) Finite/Continuous Mode with Internal Sample Clock

The count value is stored into the buffer on each rising edge of the sample clock, as shown in Figure 86.

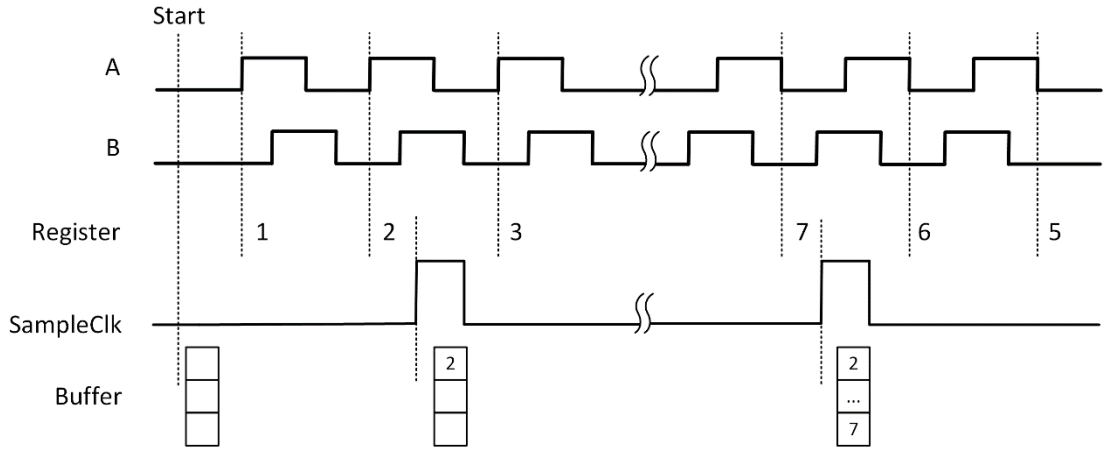


Figure 86 Quadrature Encoder x1 with Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

The count value is stored into the buffer every time it changes, as shown in Figure 87.

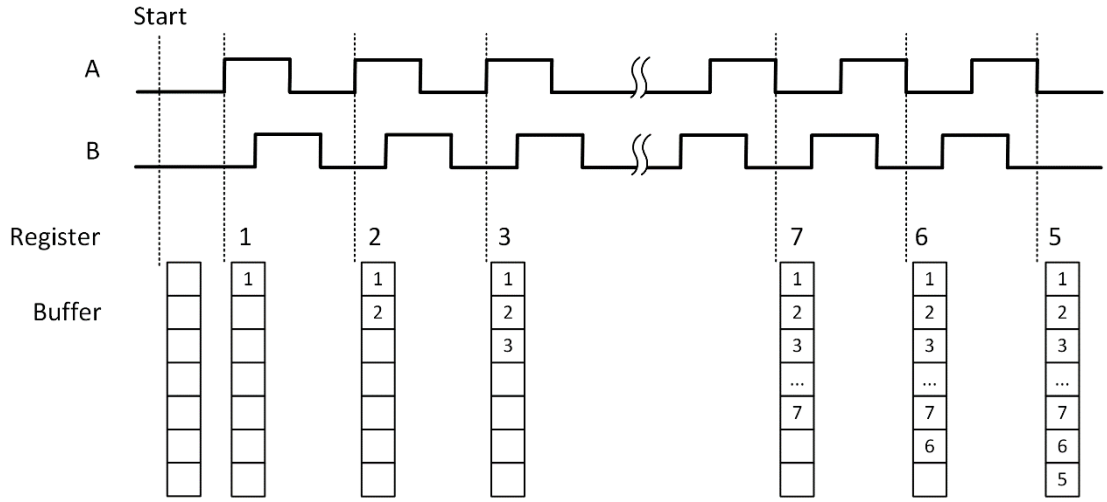


Figure 87 Quadrature Encoder x4 with Implicit Sample Clock

Learn by Examples 4.8.6

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

Single Mode

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

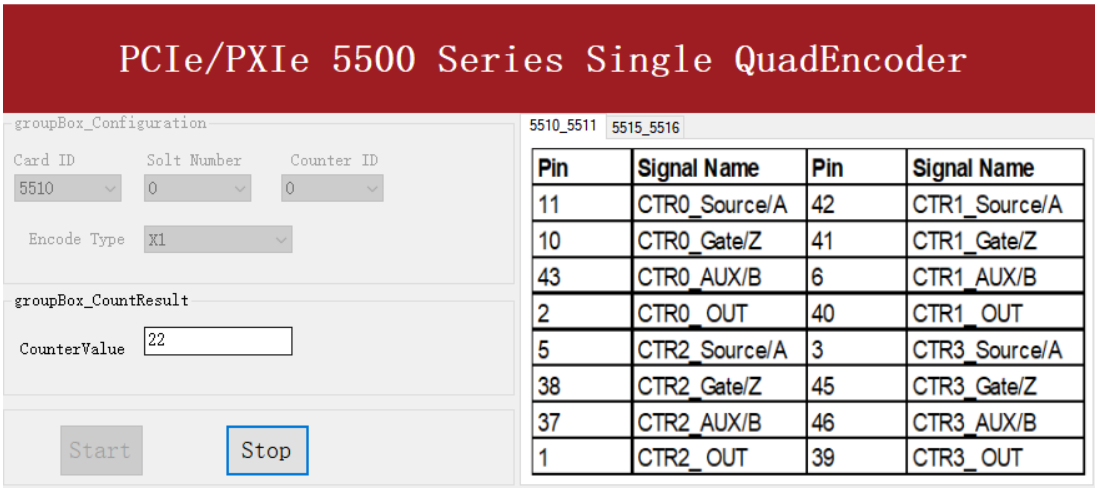


Figure 88 QuadEncoder For Single Mode

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

Continuous Mode

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

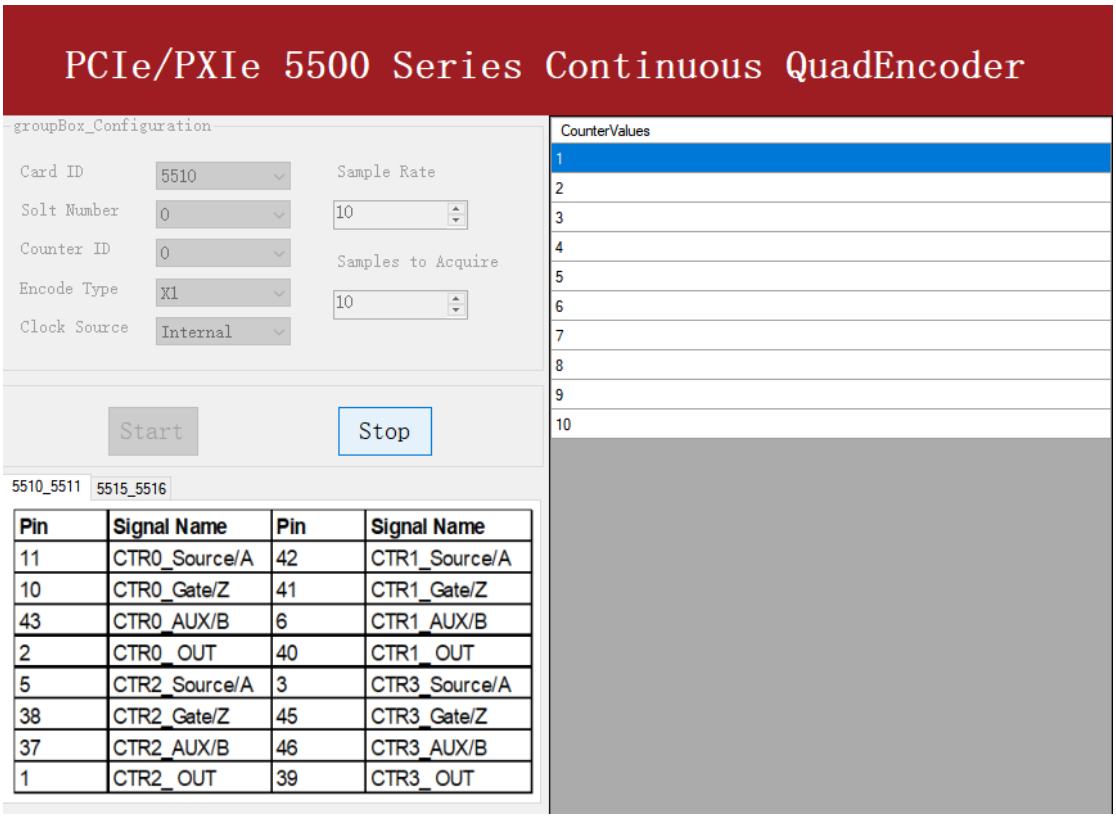


Figure 89 QuadEncoder For Continuous Mode

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

4.8.7 Two-Pulse Encoder

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

Timing

1) Single Mode

The count value is written to the register on each rising edge of the signal A, and signal B, as shown in Figure 90.

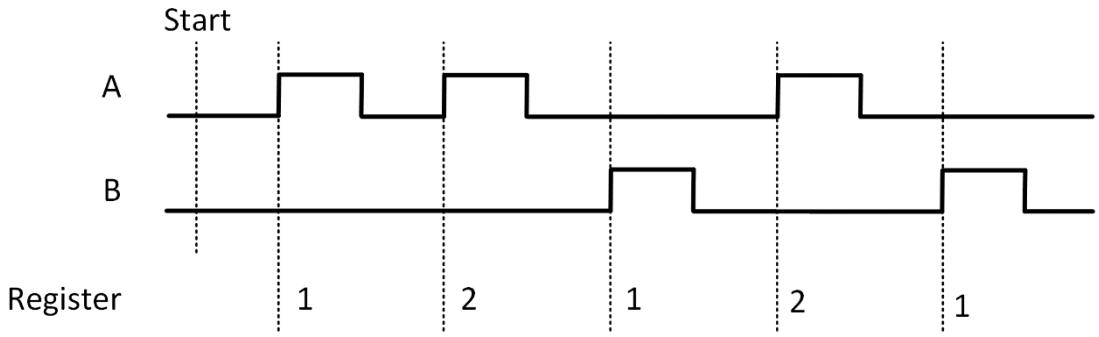


Figure 90 Two-Pulse Encoder in Single Mode

2) Finite/Continuous Mode with Internal Sample Clock

The count value is stored into the buffer on each rising edge of the sample clock, as shown in Figure 91.

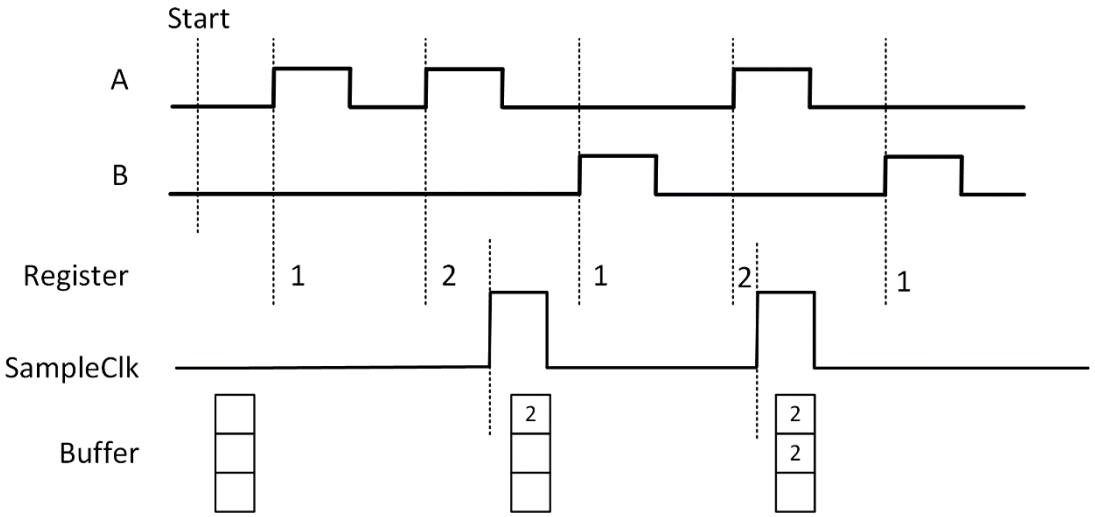


Figure 91 Two-Pulse Encoder with Internal Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

The count value is stored into the buffer every time it changed, as shown in Figure 92.

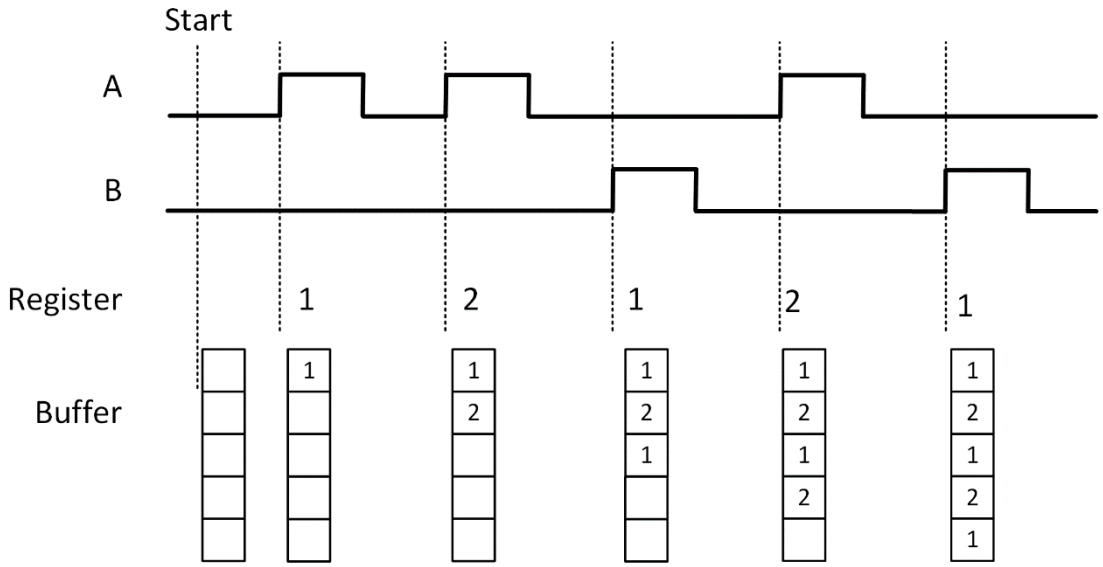


Figure 92 Two-Pulse Encoder with Implicit Sample Clock

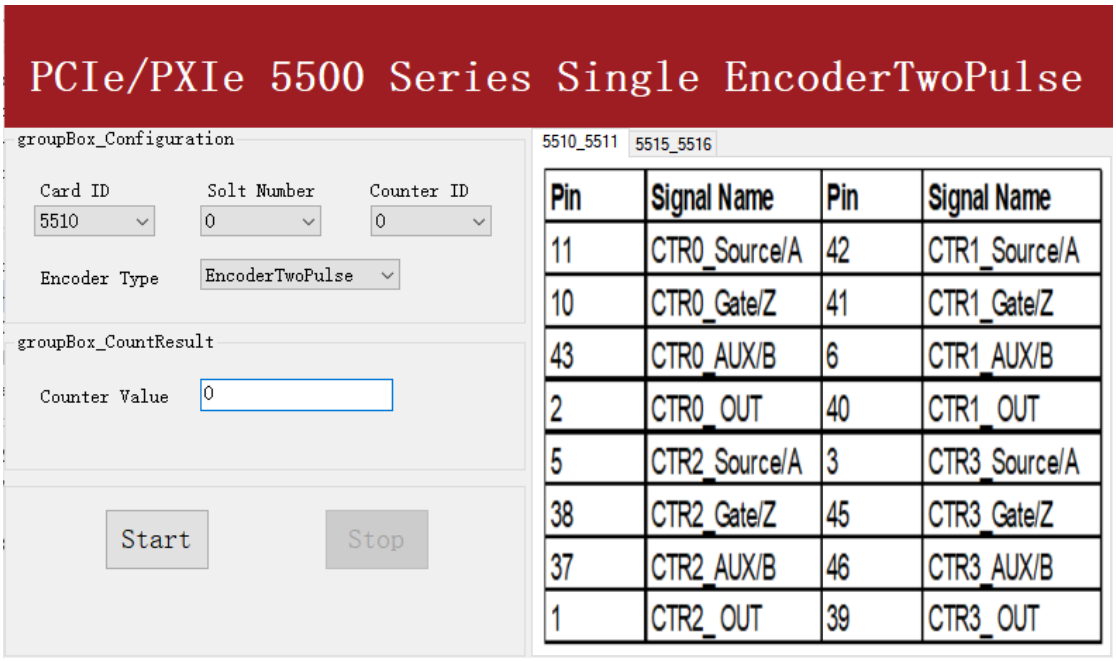
Learn by Examples 4.8.7

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

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

Single Mode

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



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

Figure 93 Two-PulseEncoder For Single Mode

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

- Click **Start** to start counting. You can see a continuously rising of the **Counter Value,** which follows the counting rules explained in this chapter.

Finite Mode

- Change the frequency of the second channel squarewave from 50Hz to 120Hz.

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

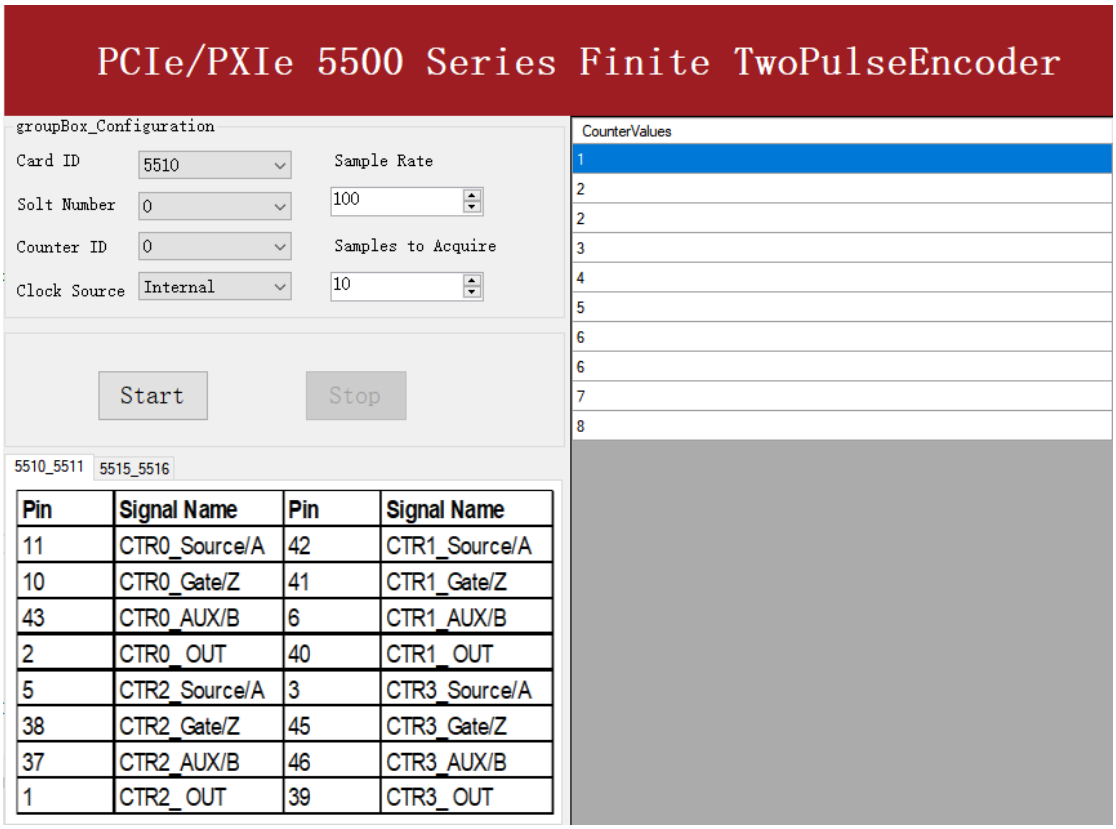


Figure 94 Two-PulseEncoder For Finite Mode

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

■ Click **Start** and the result is shown above by **Counter Value** in the right list, which follows the counting rules explained in this chapter.

Continuous Mode

■ Change the frequency of the second squarewave back to 50 Hz.

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

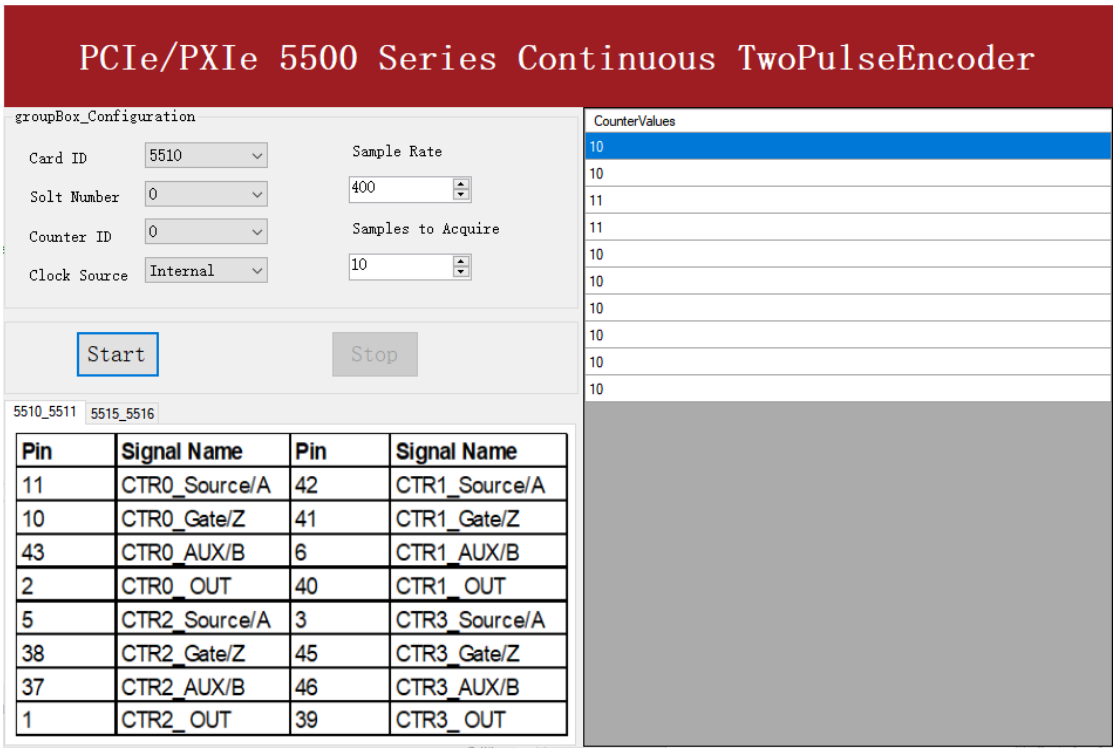


Figure 95 Two-PulseEncoder For Continuous Mode

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

- Click **Start** and you can see a group of rising numbers in **CounterValues**, which follows the counting rules explained in this chapter.

4.9 Counter Output Operations

4.9.1 Single Pulse Output

The JY5500 timer/counter can output a single pulse with a specified pulse width. The timing diagram of the pulse output is shown in Figure 96.

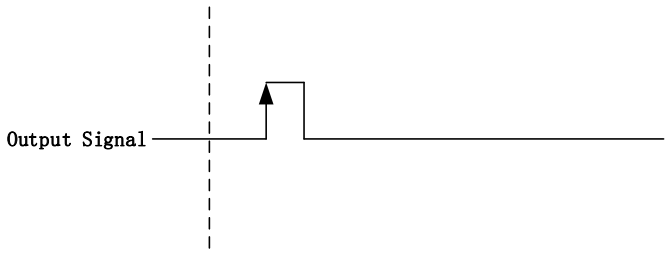


Figure 96 Single Pulse Output

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:

Frequency = 1 / (1ms / 0.5) = 500Hz

Learn by Example 4.9.1

- To see the signal that JY5500 Counter Output generates, it is recommended to connect JY5500 Counter Output (CTR0_OUT, Pin#2) to JY5500 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:

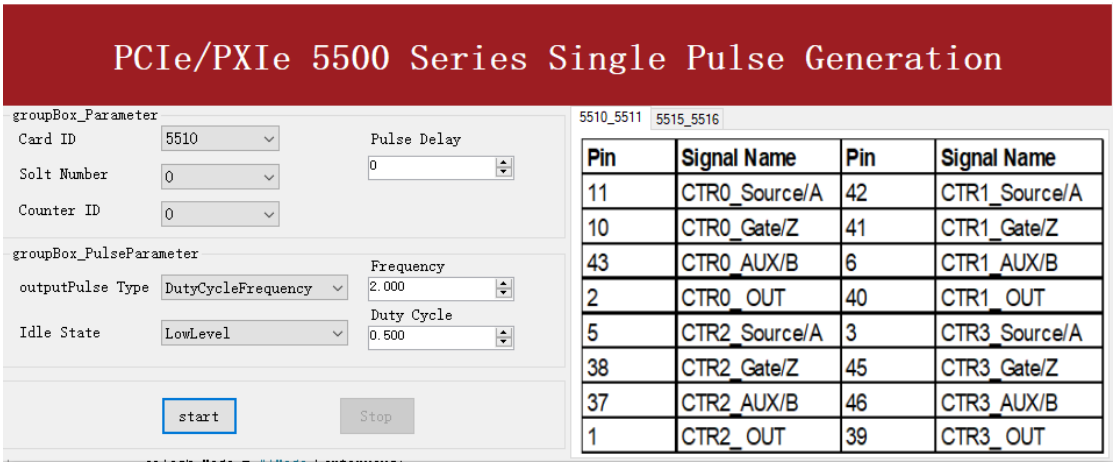


Figure 97 Single Pulse Generation

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

- Please refer Learn by Example 4.2 to configure an analog input to receive the signal from Counter Output.
- Click **Start** to generate a single pulse as shown.

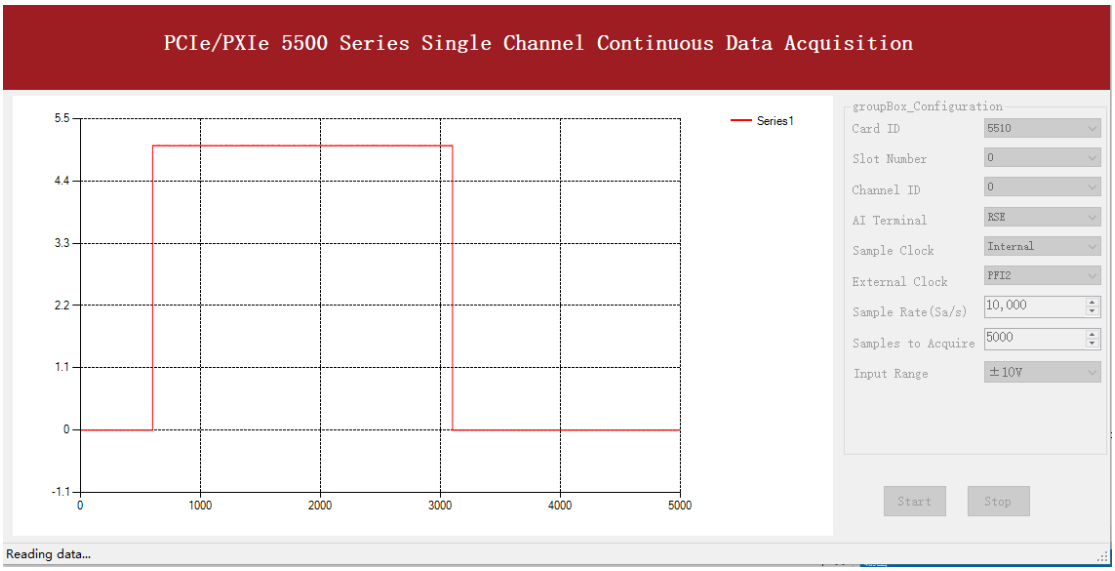


Figure 98 AI Acquisition Single Pulse

4.9.2 Finite Pulse Output

The pulse output timing is as shown in Figure 99.

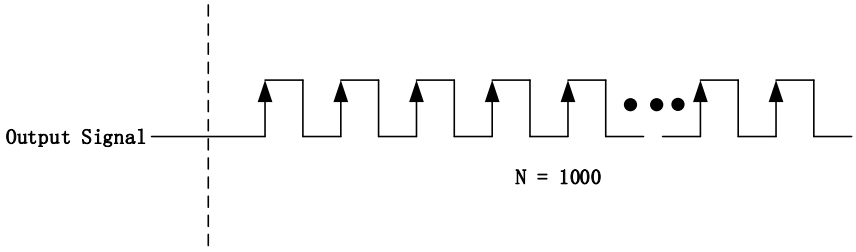


Figure 99 Finite Pulse Output

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:

Set frequency = 1 / (1ms / 0.5) = 500Hz

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

Learn by Example 4.9.2

- To see the signal that JY5500 Counter Output generates, it is recommended to connect JY5500 Counter Output (CTR0_OUT, Pin#2) to JY5500 AI Ch0 input (AI0+, Pin#68). Please note Counter Output and AI Ch0 input share the same ground so only one connection is needed.
- Open **Counter Output-->Winform CO Finite** and click **Start** and set the numbers as follow:

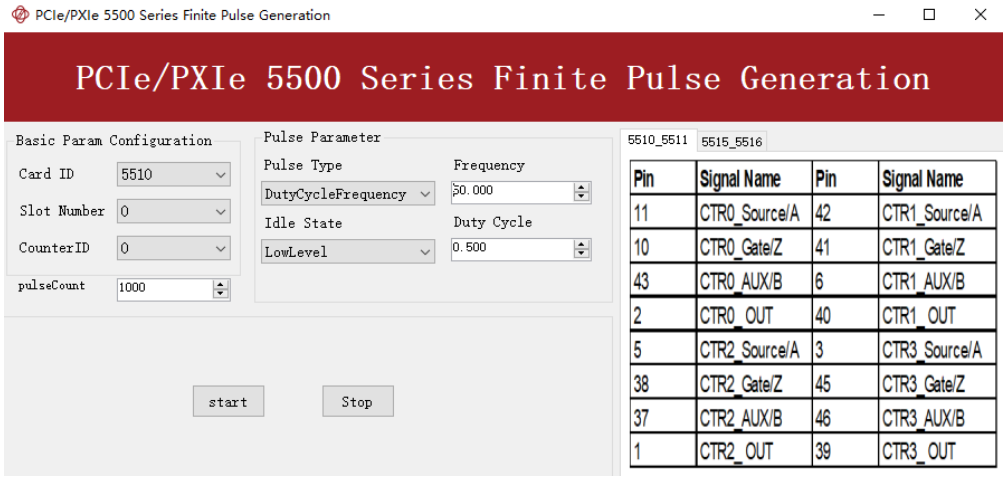


Figure 100 Finite Pulses Generation

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

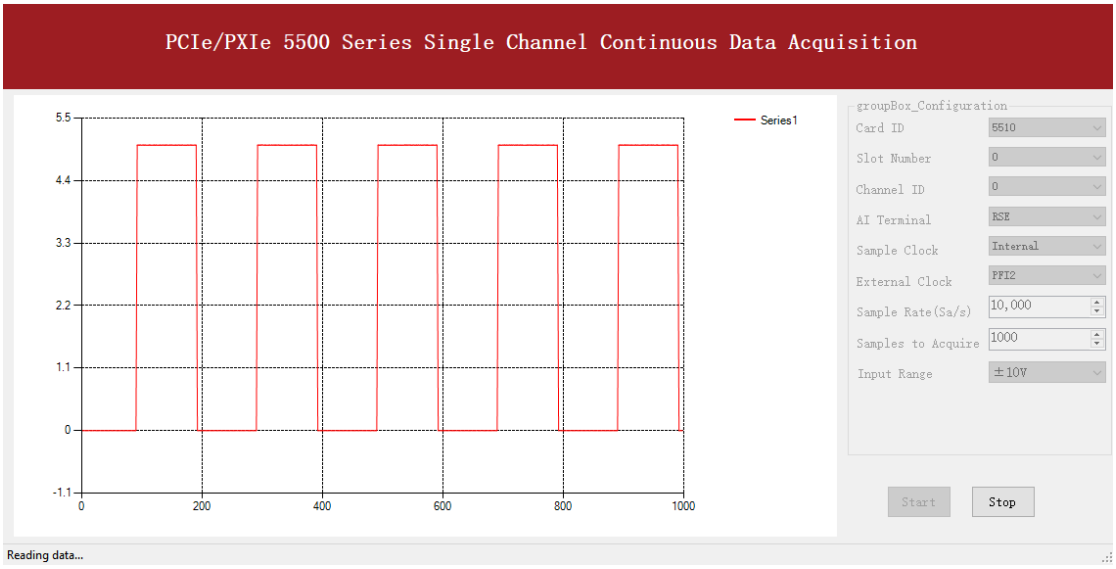


Figure 101 AI Acquisition Finite Pulse

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

4.9.3 Continuous Pulse Output

The pulse output timing is shown in Figure 102 below.

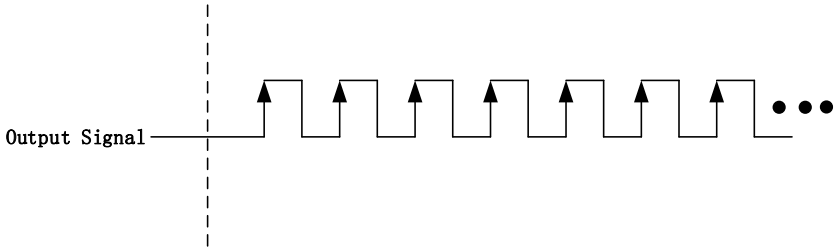
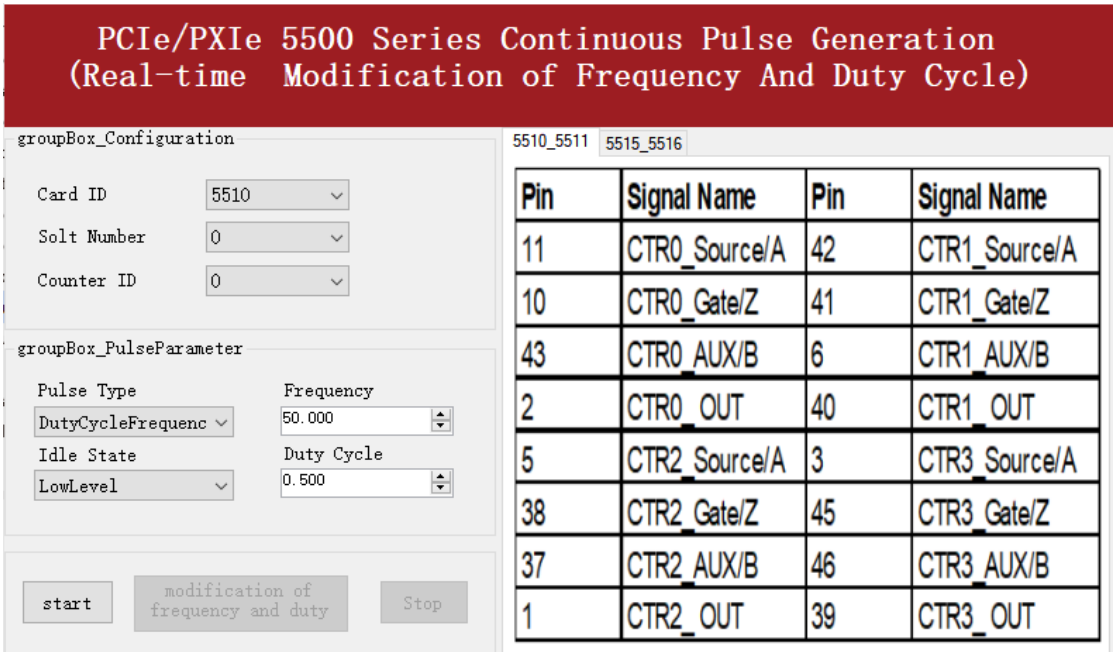


Figure 102 Continuous Pulse Output

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

Learn by Example 4.9.3

- To see the signal that JY5500 Counter Output generates, it is recommended to connect JY5500 Counter Output (CTR0_OUT, Pin#2) to JY5500 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:



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

Figure 103 Continuous Pulse Generation

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

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

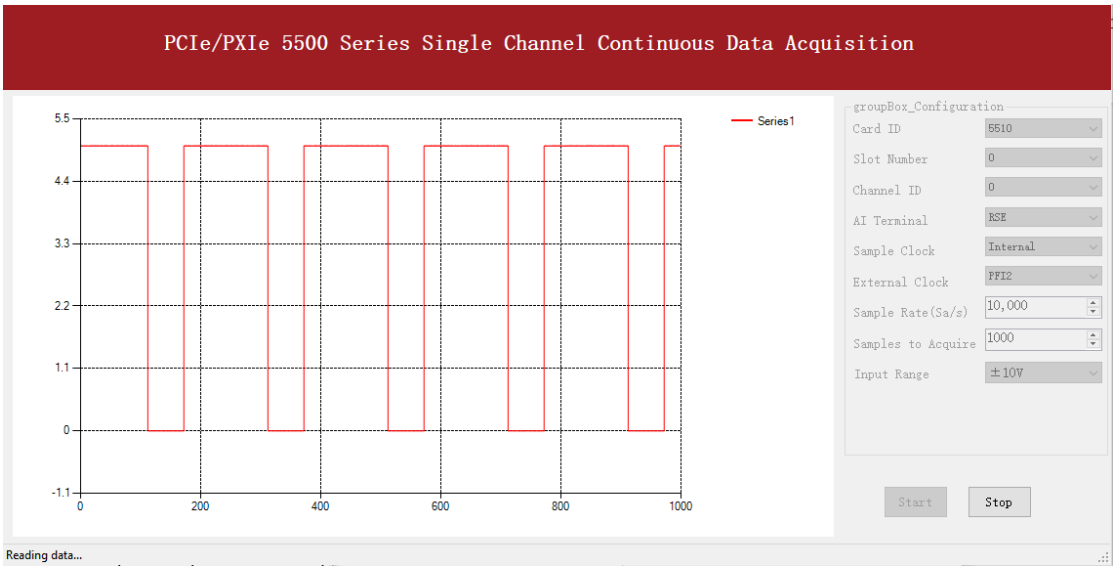


Figure 104 AI Acquisition Continuous Pulse

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

4.10 System Synchronization Interface（SSI）for PCIe Modules

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

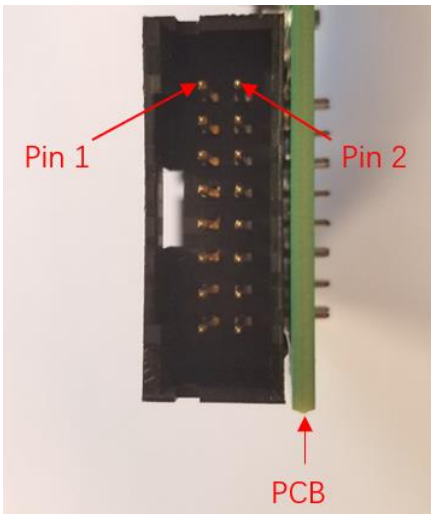


Figure 105 SSI Connector in PCIe-5500

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 26 SSI Connector Pin Assignment for PCIe-5500

4.11 DIP Switch in PCIe-5500

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

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

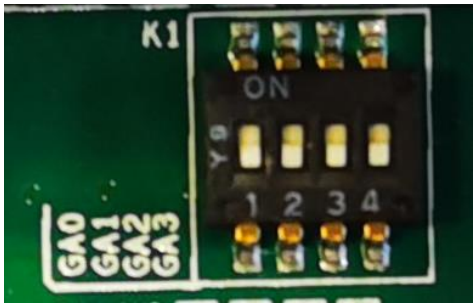


Figure 106 DIP Switch in PCIe-5500

	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 27 Relationship between switch position and slot number

5. Calibration

JY5500 Series boards are precalibrated before the shipment. We recommend you recalibrate JY5500 board periodically to ensure the measurement accuracy. A commonly accepted practice is one year. If for any reason, you need to recalibrate your board, please contact JYTEK.

6. Using JY5500 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 JY5500 DAQ card using one of this software.

6.1 Python

JYTEK provides and supports a native Python driver for JY5500 boards. There are many different versions of Python. JYTEK has only tested in CPython version 3.5.4. There is no guarantee that JYTEK python drivers will work correctly with other versions of Python.

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

6.2 C++

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

6.3 LabVIEW

LabVIEW is a software product from National Instruments. JYTEK does not support LabVIEW and will no longer provide LabVIEW interface to JY5500 boards. Our third-party partners may have LabVIEW support to JY5500 boards. We can recommend you if you want to convert your LabVIEW applications to C# based applications.

7. About JYTEK

7.1 JYTEK China

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

7.2 JYTEK Hardware Products

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

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

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

7.3 JYTEK Software Platform

JYTEK has developed a complete software platform, SeeSharp Platform, for the test and measurement applications. We leverage the open sources communities to

provide the software tools. Our platform software is also open sourced and is free, thus lowering the cost of tests for our customers. We are the only domestic vendor to offer complete commercial software and hardware tools.

7.4 JYTEK Warranty and Support Services

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

8. Appendix 1 Common Analog Measurement Issues

8.1 Floating Signals and Ground Referenced Signals

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

8.2 Differential, NRSE, RSE Modes

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

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

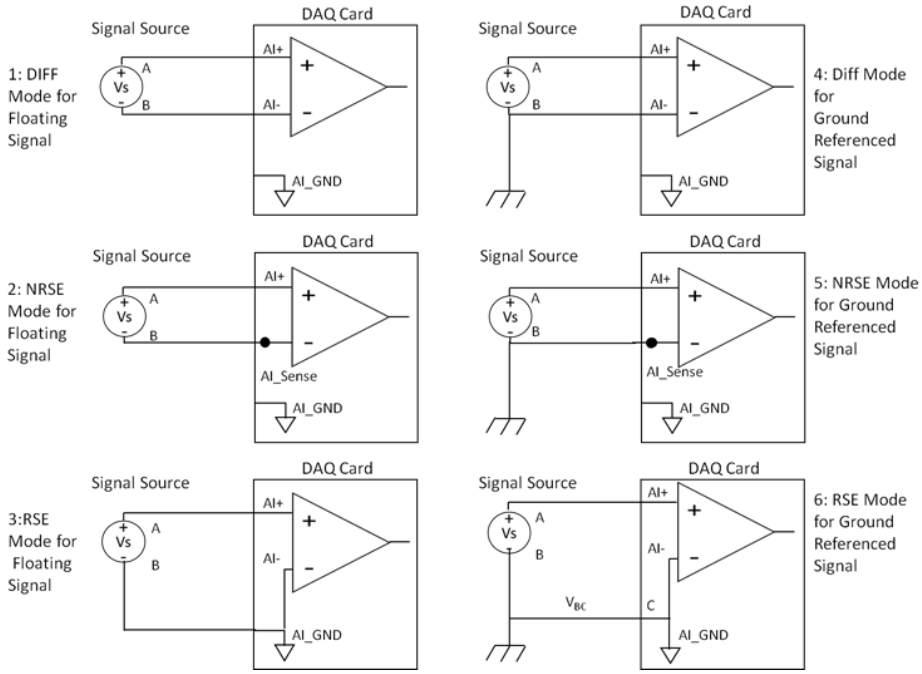


Figure 107 Six Measurement Scenarios

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

8.3 Reducing the Common Mode Voltage Effect

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

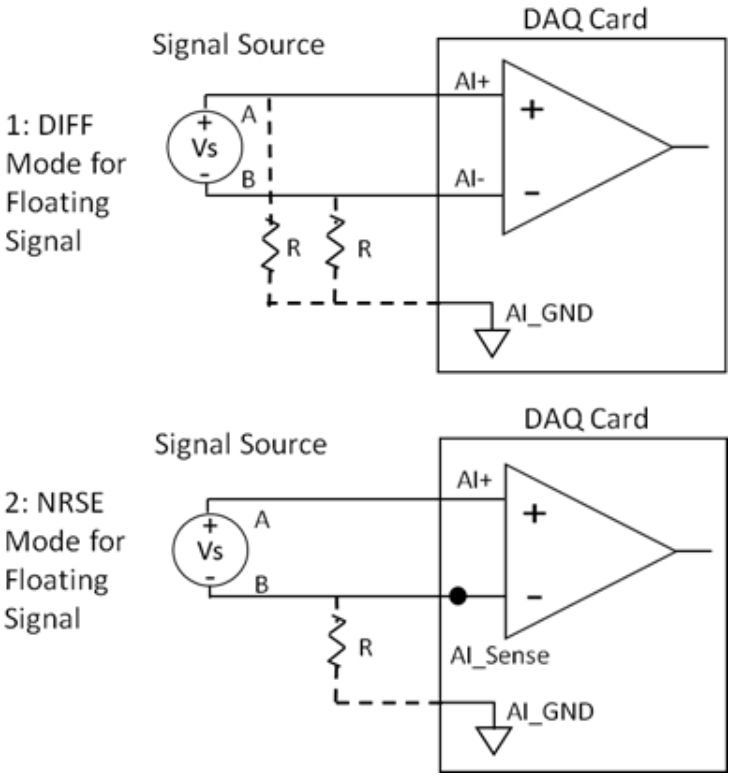


Figure 108 Using Resister to Reduce Common Mode Voltage Effect

8.4 DC, AC and DSA Mode

Figure 109 shows three different measurement modes: DC, AC and DSA. It is important to know what type of the measurement you are making. Table 28 shows differences and features in these three modes.

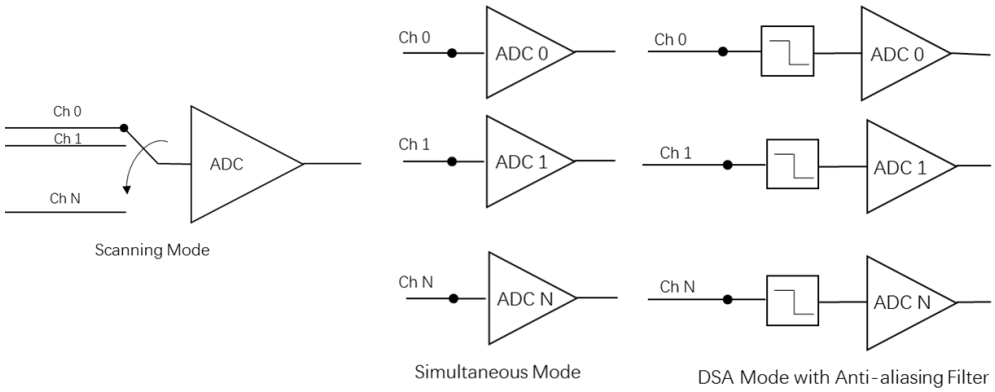


Figure 109 DC, AC and DSA Mode

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

Table 28 DC, AC, DSA Measurements

8.4.1 DC Mode

In a DC mode, the signal frequency f should be zero or very small. Many times, engineers use averaging to reduce the noise effect. But inappropriate use of averaging will not reduce the noise effect but introduce the error. Given the **Total Accuracy** Ɛ, from Sections 0, 0, 0, the maximum source signal frequency f should be bounded by:

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

where fs is the sample rate. This formula can be used in both the DAQ and the DS Mode. This formula suggests that a faster sampling device such as JY5500 can allow bigger signal changes and still achieve excellent accuracy.

8.4.2 AC Mode

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

If you use the 1M/N maximum channel sample rate of JY5500 in the multi-channel mode, the channel switching error cannot be ignored, as seen in Table 6. If you need better accuracy, you should consider using the simultaneous DAQ devices such as JYTEK PXIe-5315. These devices do not use scanning mode. Rather, each channel is serviced by a dedicated ADC to avoid the channel switching error. This ensures better AC accuracy as well as better synchronization.

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

8.4.3 DSA Mode

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

JY5500 boards are not designed for the DSA measurements. There is no anti-aliasing filter in a JY5500 board.

9. Statement

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

This manual provides the product review, quick start, some driver interface explanation for JYTEK JY5500 Series family of multi-function data acquisition boards. The manual is copyrighted by JYTEK.

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

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

Shanghai Jianyi Technology Co., Ltd.

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

Post Code: 201203

Tel: 021-5047 5899

Website: www.jytek.com