1 USB/PCIe/PXIe-5321A/5322A Specification

Family of Simultaneous Data Acquisition Modules

Overview

USB/PCIe/PXIe-5321A/5322A is a high-precision, high-resolution, up to 32-channel simultaneous voltage measurement module. It can be used to measure the DC voltage, the AC voltage as well as many dynamic signals. Depending on the model number, a JY-5321A/5322A series provide different AI channels, sampling rate.

Please download JYTEK <JYPEDIA>, you can quickly inquire the product prices, the key features and available accessories.

1.1 Main Features

- High accuracy: 210 ppm
- 32 (JY-5321A） / 16 (JY-5322A) simultaneous voltage measurement channels
- 1 MS/s per channel(JY-5321A JY-5322A)
- 18 bits resolution
- Up to 220 KHz bandwidth
- Better than -90dB THD
- System noise as low as 9μVrms
- Up to 2μV resolution
- Voltage ranges:±0.25V/±0.5V/±1V/±2V/±5V/±10V (JY-5321A,JY-5322A)
- 240M samples of FIFO buffer
- Analog and Digital Trigger

1.2 Analog Input Specifications

Analog Input Number of Input channels	5321A/5322A
ADC resolution (Bits)	18
Sampling Rate	1 M Sample/s
Clock	Internal/External
Input range(V)	±10/±5/±2/±1/±0.5/±0.25
Maximum Working Voltage(V)	±10 V (ref. AIGND)
Input Terminal Type	Differential
Input impedance(AI+ vs AIGND)	>1 GΩ || 0.5 pF
Input impedance(AI- vs AIGND)	>1 GΩ || 0.5 pF
Input coupling	DC
CMRR(@60 Hz)*1	88 dB (25 KHz Bandwidth) 91 dB (220 KHz Bandwidth)
Crosstalk(@10 KHz)*2	-110 dB (25 KHz Bandwidth) -109 dB (220 KHz Bandwidth)
Input FIFO	240M Samples
Trigger Type	Analog/Digital/Software
Analog Trigger Voltage Range	Software Programmable Between -10 V ~ +10 V
Trigger Mode	StartTrigger, ReferenceTrigger, ReTrigger
Interval of retrigger	5 Samples
Bandwidth (-3 dB)*2	25 kHz (25 kHz Bandwidth) 220 kHz (220 kHz Bandwidth)
Overvoltage protection	-27 V ~ +27 V
Input current during overvoltage protection	≤6 mA
*1 ±1 V Range Typical Test *2 ±10 V Range Typical Test	

Table 1 Analog Input Specifications

1.3 Digital IO Specifications

Number of channels	32(JY-5321A)/16(JY-5322A)
Ground reference	D GND
Directional control	Independent control of each port
Initial state	Input
Digital Input	Logic Low: VIL Min : 0 V / Max : 0.8 V Logic High: VIH Min : 2.0V / Max : 5.0V
Digital Output	Logic Low : 0 V, IOL Max: 12 mA Logic High : 2.0 V ~ 3.3 V, IOH: -12 mA ~ 0 mA
Overvoltage Protection	Continuous 12 mA 0.0 V ~ 5.0 V; Instantaneous 50 mA -2.2 V ~ 5.0 V; Duty cycle of instantaneous current pulse does not exceed 15%

Table 2 DIO Specifications

1.4 PFI Specifications

Number of channels	32(JY-5321A)/16(JY-5322A)
External digital trigger interface	Trigger voltage 3.3 V LVTTL; trigger edge: Rising/Falling
Initial state	Input

Table 3 PFI Specifications

1.5 DC Accuracy

An instrument's DC accuracy is defined by the gain and offset errors as follows:

Accuracy = Gain Error (% of reading) + Offset Error (% of range)

Equation 1 Gain and Offset Errors

It should be noted when the reading is close to zero, the gain error is very small and negligible, the offset error is dominant; when the reading is getting close to the full range, the gain error becomes more significant.

1.5.1 Basic JY-5321A/5322A DC Accuracy

The Basic DC Accuracy of JY-5321A/JY-5322A .

JY-5321A/5322A Basic Accuracy = ±(% Reading+% Range)											
Nominal Range (V)	24 Hour Tcal ±1C°			90 Days Tcal ± 5°			Temperature Coefficients(/°C)			24 Hr Full Scale Accuracy	90 Days Full Scale Accuracy	Full Scale Accuracy(%)
0.25	0.003	+	0.014	0.008	+	0.016	0.0004	+	0.0006	0.05 mV	0.06 mV	0.024
0.5	0.004	+	0.012	0.010	+	0.014	0.0004	+	0.0005	0.08 mV	0.12 mV	0.024
1	0.004	+	0.010	0.010	+	0.011	0.0004	+	0.0005	0.14 mV	0.21 mV	0.021
2	0.006	+	0.011	0.017	+	0.013	0.0005	+	0.0005	0.34 mV	0.60 mV	0.030
5	0.005	+	0.010	0.012	+	0.011	0.0006	+	0.0005	0.75 mV	1.15 mV	0.023
10	0.005	+	0.008	0.012	+	0.010	0.0006	+	0.0005	1.30 mV	2.20 mV	0.022
1. Accuracy valid to 97.5% of full range. 2.The 90-day data is estimated by 24-hour data*2.5												

Table 4 AI Accuracy of JY-5321A/JY-5322A

1.5.2 System Noise

Range (V)	SystemNoise(uVrms)*
0.25	9
0.5	12
1	20
2	39
5	95
10	190
*Results under 25 kHz	

Table 5 System Noise of JY-5321A/JY-5322A

1.6 Dynamic Performance

Table 6 shows JY-5321A/JY-5322A Dynamic performance including AI Bandwidth, THD, SINAD, SFDR and Noise.

Nominal Range (V)	Max Sample Rate (S/s)	Resolution (μV)	Bandwidth -3dB (KHz)	THD (dB)	SINAD (dBc)	SFDR (dBc)	Noise (μVrms)
0.25	1 M	2	150	-94	82	94	17
0.5	1 M	4	220	-92	83	96	25
1	1 M	8	220	-97	89	98	33
2	1 M	16	220	-92	86	95	72
5	1 M	39	220	-99	89	101	161
10	1 M	76	220	-99	90	103	320
JY-5321A JY-5322A support 2 Bandwidth: 220 KHz and 25KHz, this table shows specification under 220 KHz							

Table 6 Dynamic Performance of JY-5321A/JY-5322A

Nominal Range (V)	Max Sample Rate (S/s)	Resolution (μV)	Bandwidth -3dB (KHz)	THD (dB)	SINAD (dBc)	SFDR (dBc)	Noise (μVrms)
0.25	1 M	2	25	-96	77	93	9
0.5	1 M	4	25	-93	77	96	10
1	1 M	8	25	-96	85	98	16
2	1 M	16	25	-92	82	96	38
5	1 M	39	25	-98	85	100	82
10	1 M	76	25	-98	86	100	162
JY-5321A JY-5322A support 2 Bandwidth: 220 KHz and 25 KHz, this table shows specification under 25 KHz							

Table 7 Dynamic Performance of JY-5321A/JY-5322A

1.6.1 AI Bandwidth

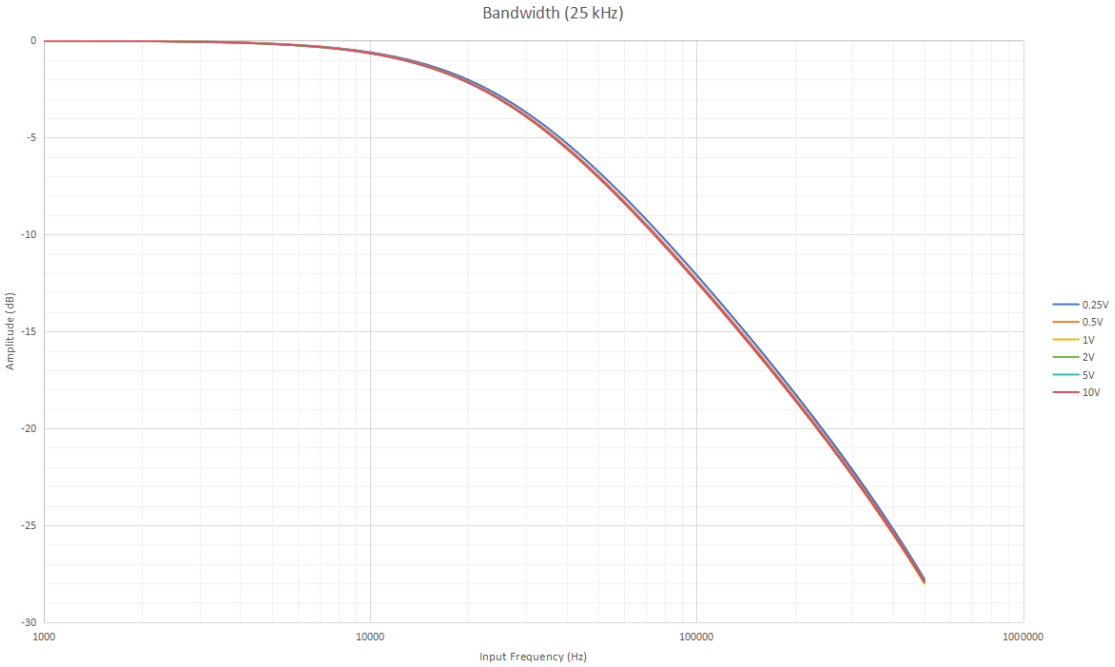

Figure 1 -3dB Bandwidth (25 KHz)

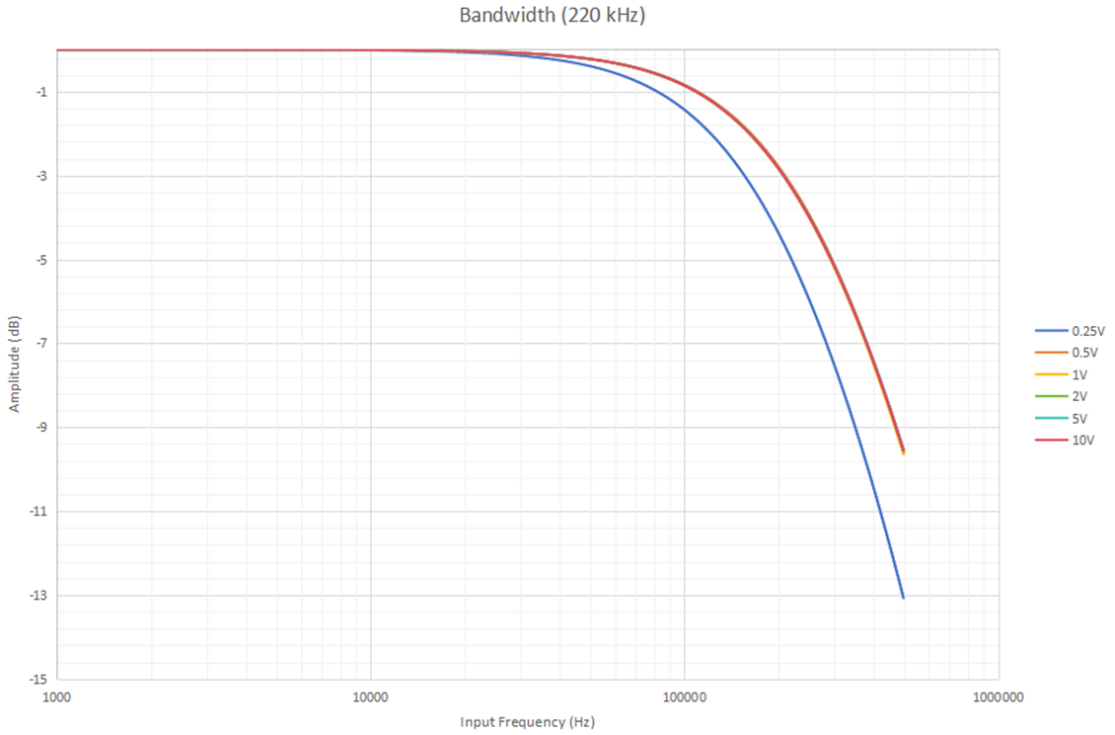

Figure 2 -3dB Bandwidth (220 KHz)

1.6.2 CMRR

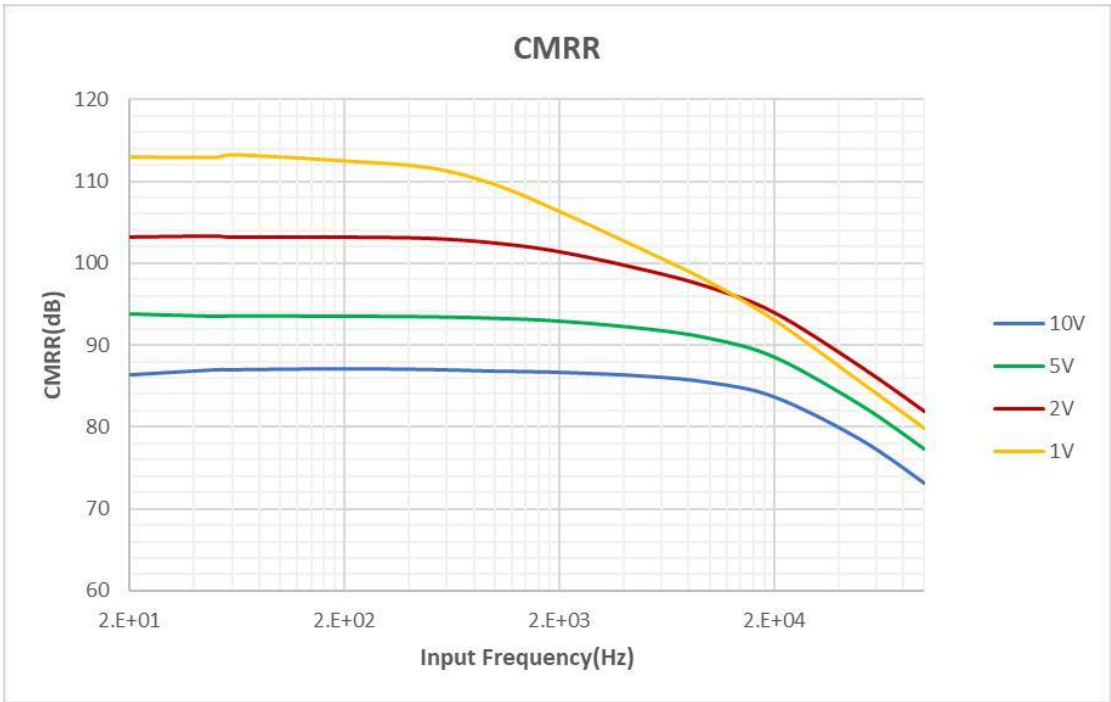

Figure 3 CMRR

Range (V)	CMRR (dB at 50 Hz)
10	86.9
5	93.6
2	103
1	113

Table 8 CMRR

1.6.3 Crosstalk

Adjacent channel

Crosstalk(Adjacent channel)

Range(V)	Crosstalk(dB at 10 kHz)
10V	-110
5V	-109
2V	-110
1V	-110
0.5V	-110
0.25V	-110

JY-5321A support 2 Bandwidth: 220 KHz and 25 KHz, this table shows specification under 25 KHz

Crosstalk(Adjacent channel)

Range(V)	Crosstalk(dB at 10 kHz)
10V	-110
5V	-109
2V	-109
1V	-109
0.5V	-109
0.25V	-110
JY-5321A support 2 Bandwidth: 220 KHz and 25 KHz, this table shows specification under 220 KHz	

Table 9 Crosstalk (Adjacent channel)

Non-adjacent channel

Crosstalk (Adjacent channel)	
Range(V)	Crosstalk(dB at 100 KHz)
10	-113
5	-113
2	-113
1	-113

Table 10 Crosstalk (Non-adjacent channel)

1.7 Group delay

Analog filter group delay(us)

Bandwidth(Hz) / Range(V)	220K	25K
10V	1.1	6.8
5V	1.1	6.8
2V	1.1	6.8
1V	1.1	6.8
0.5V	1.1	6.8
0.25V	1.5	6.8

Table 11 Crosstalk (Non-adjacent channel)

Digital filter group delay(Samples)

SampleRate	Digital filter group delay(Samples)	OSR
500 kS/s<fs≤1 MS/s	0.00	1
250 kS/s<fs≤500 kS/s	0.25	2
125 kS/s<fs≤250 kS/s	0.38	4
62.5 kS/s<fs≤125 kS/s	0.44	8
31.25 kS/s<fs≤62.5 kS/s	0.47	16
15.63 kS/s<fs≤31.25 kS/s	0.48	32
7.81 kS/s<fs≤15.63 kS/s	0.49	64
3.91 kS/s<fs≤7.81 kS/s	0.50	128
1.95 kS/s<fs≤3.91 kS/s	0.50	256

Table 12 Crosstalk (Non-adjacent channel)

1.8 CIO

CI/CO	5321A	5322A
Number of channels	2	1
Resolution	32	
CI	Edge count, Period measurement, Frequency measurement, Pulse width measurement, Two-edge interval measurement, Quadrature encoder, etc.	
CO	Single, Finite and Continuous pulse	
Clock	100 MHz	
FIFO	4M Samples	
Input	Gate, Source, Aux	
Output	OUT	

Table 13 CIO Specifications

1.9 Connector

1.9.1 JY-5321A provides 32 channels

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

Connector 0			
Pin	Signal Name	Pin	Signal Name
1	PFI 0	35	PFI 1
2	PFI 2	36	PFI 3
3	PFI 4	37	PFI 5
4	PFI 6	38	PFI 7
5	D_GND	39	D_GND
6	PFI 8	40	PFI 9
7	PFI 10	41	PFI 11
8	PFI 12	42	PFI 13
9	PFI 14	43	PFI 15
10	D_GND	44	D_GND
11	AI 0-	45	AI 0+
12	AI_GND	46	AI_GND
13	AI 1-	47	AI 1+
14	AI 2-	48	AI 2+
15	AI_GND	49	AI_GND
16	AI 3-	50	AI 3+
17	AI 4-	51	AI 4+
18	AI_GND	52	AI_GND
19	AI 5-	53	AI 5+
20	AI 6-	54	AI 6+
21	AI_GND	55	AI_GND
22	AI 7-	56	AI 7+
23	AI 8-	57	AI 8+
24	AI_GND	58	AI_GND
25	AI 9-	59	AI 9+
26	AI10-	60	AI10+
27	AI_GND	61	AI_GND
28	AI 11-	62	AI 11+
29	AI 12-	63	AI 12+
30	AI_GND	64	AI_GND
31	AI 13-	65	AI 13+
32	AI 14-	66	AI 14+
33	AI_GND	67	AI_GND
34	AI 15-	68	AI 15+

Connector 1		Connector 0	
Pin	Signal Name	Pin	Signal Name
1	CTR1_Source/A	1	CTR0_Source/A
35	CTR1_Gate/Z	35	CTR0_Gate/Z
2	CTR1_AUX/B	2	CTR0_AUX/B
3	CTR1_ OUT	3	CTR0_ OUT

Table 14 Pinout definition of JY-5321A

1.9.2 JY-5322A provides 16 channels

Connector 0			
Pin	Signal Name	Pin	Signal Name
1	PFI 0	35	PFI 1
2	PFI 2	36	PFI 3
3	PFI 4	37	PFI 5
4	PFI 6	38	PFI 7
5	D_GND	39	D_GND
6	PFI 8	40	PFI 9
7	PFI 10	41	PFI 11
8	PFI 12	42	PFI 13
9	PFI 14	43	PFI 15
10	D_GND	44	D_GND
11	AI 0-	45	AI 0+
12	AI_GND	46	AI_GND
13	AI 1-	47	AI 1+
14	AI 2-	48	AI 2+
15	AI_GND	49	AI_GND
16	AI 3-	50	AI 3+
17	AI 4-	51	AI 4+
18	AI_GND	52	AI_GND
19	AI 5-	53	AI 5+
20	AI 6-	54	AI 6+
21	AI_GND	55	AI_GND
22	AI 7-	56	AI 7+
23	AI 8-	57	AI 8+
24	AI_GND	58	AI_GND
25	AI 9-	59	AI 9+
26	AI10-	60	AI10+
27	AI_GND	61	AI_GND
28	AI 11-	62	AI 11+
29	AI 12-	63	AI 12+
30	AI_GND	64	AI_GND
31	AI 13-	65	AI 13+
32	AI 14-	66	AI 14+
33	AI_GND	67	AI_GND
34	AI 15-	68	AI 15+

Connector 0	
Pin	Signal Name
1	CTR0_Source/A
35	CTR0_Gate/Z
2	CTR0_AUX/B
3	CTR0_ OUT

Table 15 Pinout definition of JY-5322A

1.10 Front Panel connections and Pinout Definition

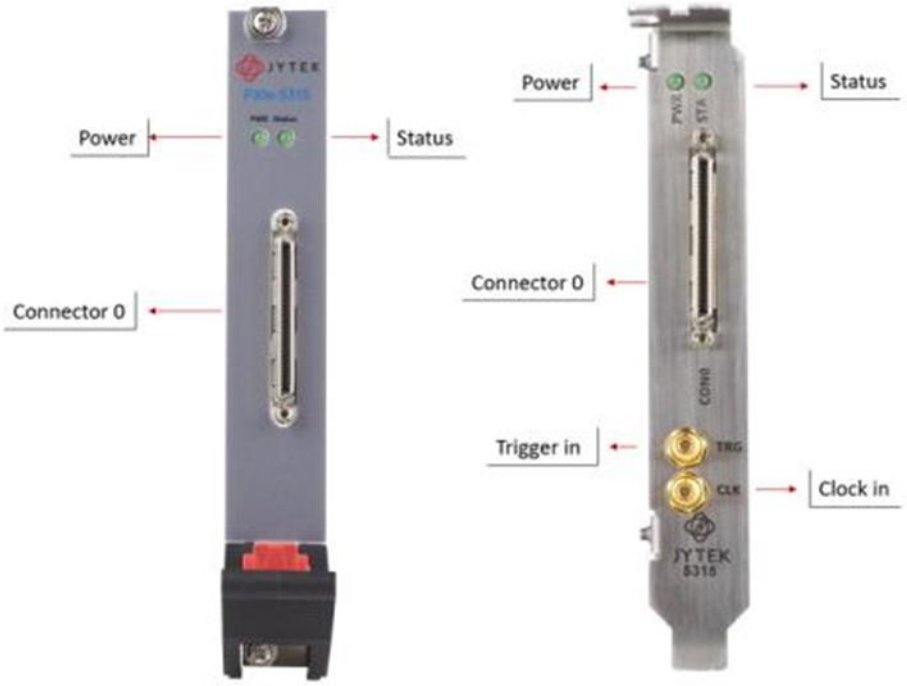

Figure 4 JY-5321A/5322A Front Panel

1.11 Physical and Environment

Bus

PXIe standard	x4 PXI Express peripheral module Specification V1.0 compliant
Slot supported	x1 and x4 PXI Express or PXI Express hybrid slots

Size

External physical size	3 U PXIe
Weight	190 g

Operating Temperature

Operating ambient temperature range	0-50°C

Storage Environment

Ambient temperature range	-20°C to 80°C
Relative humidity range	10% to 90%, None-condensing

Power

3.3V	0.5A
12V	0.5A

Table 16 Physical and Environment

1.12 Synchronization Accuracy

	5321A/5322A
Synchronization Accuracy(Single module)	30ns
Synchronization Accuracy(Two modules in one chassis)	50ns

Table 17 Synchronization Accuracy

Table of Contents

2 Software

2.1 System Requirements

JY-5321A/5322A/5323A/5324A modules can be used in a Windows or a Linux operating system.

Microsoft Windows: Windows 7 32/64 bit, Windows 10 32/64 bit. We highly recommend the user to use Windows 10 whenever possible.

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 wi ll 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 18 Supported Linux Versions

2.2 System Software

When using JY-5321A/5322A/5323A/5324A in the Window environment, you

need to install the following software from Microsoft:

Visual Studio Version 2015 or above,

.NET version is 4.0 or above.

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

Given the resources limitation, JYTEK only tested JY-5321A/5322A/5323A/5324 A modules with .NET 4.0 with Visual Studio 2015. JYTEK relies on Microsoft

to maintain the compatibility for the newer versions.

2.3 C# Programming Language

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

2.4 C ++ Programming Language

JYTEK provides QT C++ drivers for C++ programmers. We also provide many QT C++ examples. However, due to our limited resources, we do not support C++ based applications.

2.5 JY-5321A/5322A/5323A/5324A Hardware Driver

After installing the required application development environment as described above, you need to install the JY-5321A/5322A/5323A/5324A hardware driver.

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

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

Hardware Specific Driver: Each JYTEK hardware has a C# hardware specific driver. This driver provides rich and easy-to-use C# interfaces for users to operate various JY-5321A/5322A/5323A/5324A function. JYTEK has standardized the ways JYTEK and other vendor's DAQ cards 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 using the same methods.

2.6 Install the SeeSharpTools from JYTEK

To efficiently and effectively use JY-5321A/5322A/5323A/5324A boards, you ne ed to install a set of free C# utilities from JYTEK SeeSharp Test and Measure ment platform. The SeeSharp platform offers rich user interface functions you will find convenient in developing your applications. They are also needed t o run the examples come with JY-5321A/5322A/5323A/5324A hardware. Plea se register and down load the latest SeeSharpTools from our website www.jy tek.com.

2.7 Running C# Programs in Linux

Most C# written programs in Windows can be run by Microsoft Mono develo pment system in a Linux environment. You would develop your C# applicatio ns in Windows using Visual Studio. Once it is done, run this application in th e Mono environment. This is JYTEK recommended way to run your C# progr ams in a Linux environment.

If you want to use your own Linux development system other than Mono, y ou can do it using our Linux driver. However, JYTEK does not have the capa bility to support the Linux applications. JYTEK completely relies upon Microsof t to maintain the cross-platform compatibility between Windows and Linux us ing Mono.

3 Order Information

- PXIe-5321A (PN: JY8149280-01)

32-ch, 16-bit, 1MS/s/ch , 16 DIO, PXIe Simultaneous sampling Multifunction I/O Module

- PXIe-5322A (PN: JY6410588-01)

16-ch, 16-bit, 1MS/s/ch , 8 DIO, PXIe Simultaneous sampling Multifunction I/O Module

- PXIe-5323A (PN: JY1816101-01)

32-ch, 16-bit, 200kS/s/ch , 16 DIO, PXIe Simultaneous sampling Multifunction I/O Module

- PXIe-5324A (PN: JY8920241-01)

16-ch, 16-bit, 200kS/s/ch , 8 DIO, PXIe Simultaneous sampling Multifunction I/O Module

- PCIe-5321A (PN: JY7961198-01)

32-ch, 16-bit, 1MS/s/ch , 16 DIO, PCIe Simultaneous sampling Multifunction I/O Module

- PCIe-5322A (PN: JY2511025-01)

16-ch, 16-bit, 1MS/s/ch , 8 DIO, PCIe Simultaneous sampling Multifunction I/O Module

- PCIe-5323A (PN: JY7037274-01)

32-ch, 16-bit, 200kS/s/ch , 16 DIO, PCIe Simultaneous sampling Multifunction I/O Module

- PCIe-5324A (PN: JY3315663-01)

16-ch, 16-bit, 200kS/s/ch , 8 DIO, PCIe Simultaneous sampling Multifunction I/O Module

- USB-5321A (PN: JY9188254-01)

32-ch, 16-bit, 1MS/s/ch , 16 DIO, USB Simultaneous sampling Multifunction I/O Module

- USB-5322A (PN: JY1290213-01)

16-ch, 16-bit, 1MS/s/ch , 8 DIO, USB Simultaneous sampling Multifunction I/O Module

- USB-5323A (PN: JY1743328-01)

32-ch, 16-bit, 200kS/s/ch , 16 DIO, USB Simultaneous sampling Multifunction I/O Module

- USB-5324A (PN: JY9890308-01)

16-ch, 16-bit, 200kS/s/ch , 8 DIO, USB Simultaneous sampling Multifunction I/O Module

4 JYPEDIA

JYPEDIA is an excel file. It contains JYTEK product information, pricing, inventory information, drivers, software, technical support, knowledge base etc. You can register and download a JYPEDIA excel file from our web www.jytek.com. JYTEK highly recommends you use this file to obtain information from JYTEK.

5 Additional Hardware Information

5.1 DC Accuracy

DC voltage measurement refers to the measurement of a slowly changing voltage. The accuracy of the DC measurement is affected by gain error and offset error. An instrument's DC accuracy is defined by the gain and offset errors as follows:

Accuracy = Gain Error (% of reading) + Offset Error (% of range)

Equation 2 Gain and Offset Errors

It should be noted when the reading is close to zero, the gain error is very small and negligible, the offset error is dominant; when the reading is getting close to the full range, the gain error becomes more significant.

The AI DC Accuracy of JY-5321A/5322A is shown in and Table 4.

5.2 Dynamic Performance

JY-5321A/5322A/5323A/5324A offers excellent dynamic performances as shown in Table 6, where THD stands for the total harmonic distortion; SINAD stands for Signal-to-Noise And Distortion; SFDR stands for Spurious-Free Dynamic Range.

6 Additional Software Information

6.1 System Requirements

JY-5321A/5322A/5323A/5324A series modules can be used in a Windows or a Linux operating system.

Microsoft Windows: Windows 7 32/64 bit, Windows 10 32/64 bit. We highly recommend the user to use Windows 10 whenever possible.

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 19 Supported Linux Versions

6.2 System Software

When using JY-5321A/5322A series in the Window environment, you need to install the following software from Microsoft:

Visual Studio Version 2015 or above,

.NET version is 4.0 or above.

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

Given the resources limitation, JYTEK only tested JY9817 modules with .NET 4. 0 with Visual Studio 2015. JYTEK relies on Microsoft to maintain the compati bility for the newer versions.

6.3 C# Programming Language

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

6.4 C ++ Programming Language

JYTEK provides QT C++ drivers for C++ programmers. We also provide many QT C++ examples. However, due to our limited resources, we do not support C++ based applications.

6.5 Python

JYTEK provides and supports a native python driver for JY-5321A/5322A/5323A /5324A cards. There are many different versions of Python. JYTEK has only te sted in CPython version 3.5. There is no guarantee that JYTEK python drivers will work correctly with other versions of Python. If you want to be our part ner to support different Python platforms, please contact us.

7 Operating JY-5321A/5322A/5323A/5324A Module

This chapter provides the operation guides for JY-5321A/5322A/5323A/5324A, i ncluding AI and programmable I/O interface, etc.

JYTEK provides extensive examples, on-line help and documentation to assist y ou to operate the JY-5321A/5322A/5323A/5324A board. JYTEK strongly recom mends you go through these examples before writing his own application. In many cases, an example can also be a good starting point for a user applicat ion.

7.1 Quick Start

After you have installed the driver software and the SeeSharpTools, you are r eady to use Visual Studio and our C# measurement and control platform tool s (SeeSharp Platform) to operate the JY-5321A/5322A/5323A/5324A board.

If you are already familiar with Microsoft Visual Studio and C# programming, the quickest way to use JY-5321A/5322A/5323A/5324A cards is to go through our extensive examples. We provide source code of our examples. In many c ases, you can modify the source code and start to write your applications.

7.2 Data Acquisition Methods

JY-5321A/5322A/5323A/5324A uses a simultaneous method to acquire analog data, meaning there is sixteen ADC chip on the device and each input chann el uses individual ADC to capture data. In the simultaneous acquisition mode, you need to configure AI channels and set up some parameters through JY-5 321A/5322A/5323A/5324A driver software. The most important parameters are *Data Acquisition mode*, *Sample Rate*, *SamplesToAcquire*, *Channel*, and *Analog Input Terminal Type*.

AI Acquisition mode **(***AIMode***)**: JY-5321A/5322A/5323A/5324A provides 4 acquis

ition modes, **Continuous**, **Finite**, **Single Point**, **Record** described in details in S ection 7.2.1 to 7.2.4.

SampleRate: How fast multi-channel data are collected per second per channe l. Example, if your 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 acquis ition modes. In the continuous acquisition mode, *SamplesToAcquire* is the buff er size used in the AI acquisition task; in the finite acquisition mode, it is to specify the number of samples to capture.

Channels: how many channels do you want to collect data. You can set up t he channels in different orders, for instance 2,3,1,0. The acquired data will be arranged in the way you specified. In this particular case, Channel_Count is 4.

Once the Sample Rate is set, the actual hardware sampling rate is achieved by internal clock or external clock source. The sample clock source is also co nfigured by software. Data acquisition sequence is shown in Figure 5.

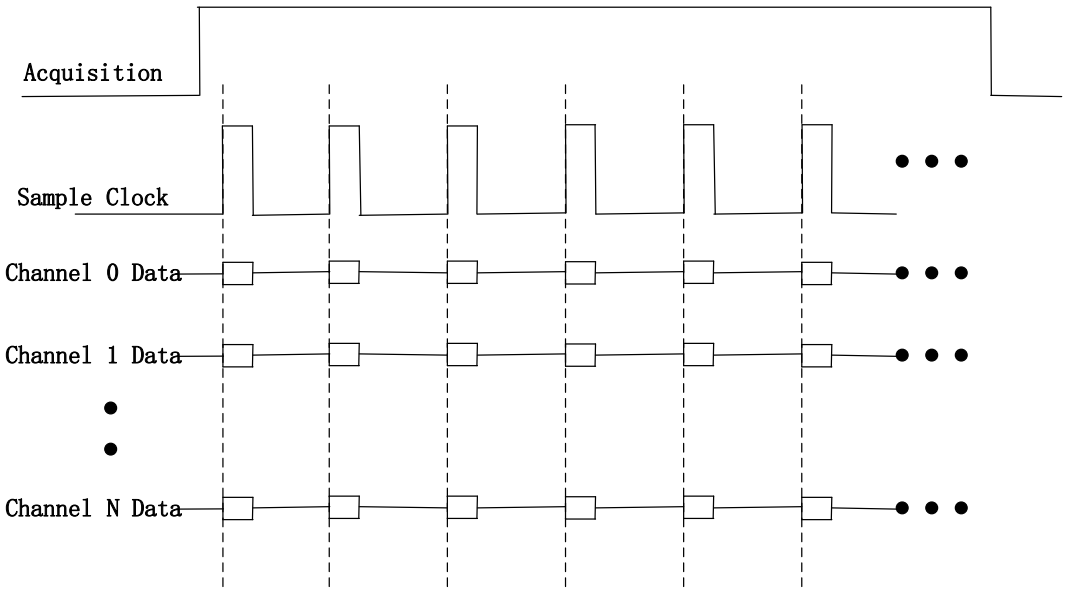

Figure 5 AI Simultaneous Mode Acquisition

7.2.1 Continuous Acquisition

An AI acquisition task will acquire the data continuously until the task is stop ped when you start the task in the Continuous acquisition mode. You will ne ed to configure the appropriate buffer size by the parameter, *SamplesToAcquir e*, to achieve the best performance of capture and display.

You use the sample program **Analog Input --> Winform AI Continuous** to lear n more about Continuous Acquisition

Learn by Example 8.2.1

You use the sample program Analog Input --> Winform AI Continuous to lear n more about Continuous Acquisition.

- Connect the one signal source's positive outputs to JY-5323A AI Ch2 (AI2 +, Pin #48), one negative terminals to AI Ch2 negative (AI2-, Pin#14) .
- Set other numbers as shown and click start.

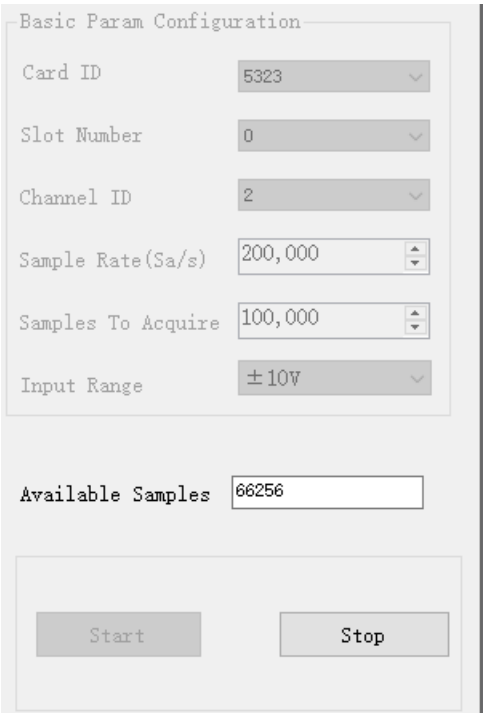

Figure 6 Set other numbers

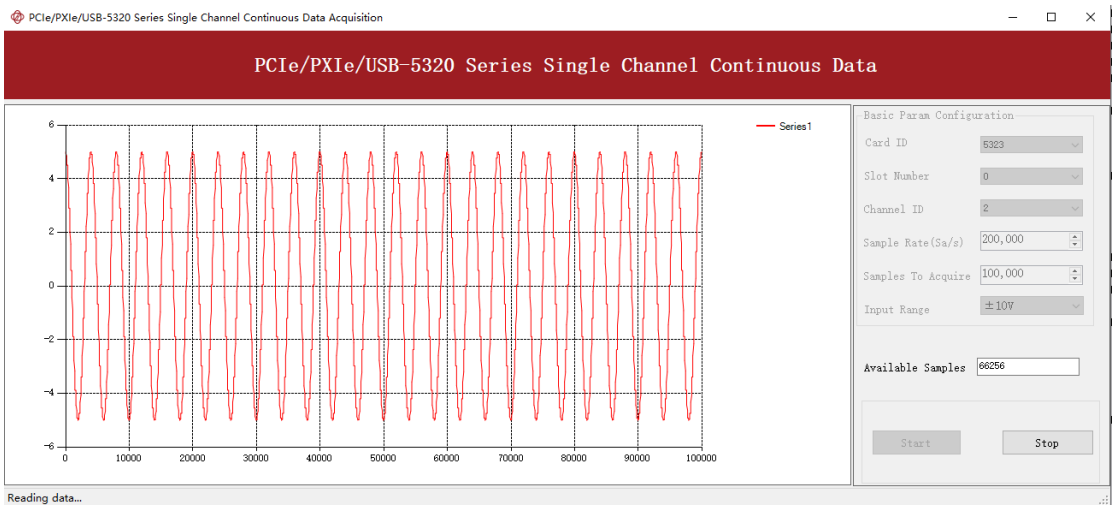

Figure 7 AI Continuous Data

7.2.2 Finite Acquisition

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

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

7.2.3 Single Acquisition

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

7.2.4 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 rando mly when the capturing process is available. The mode is particularly useful f or high-speed acquisition and recording applications.

You can use sample program: **Record -->Winform AI Finite Streaming** to learn more about the single point Acquisition.

Learn by Example 8.2.4

- Connect two signal source's positive outputs to JY-5323A AI Ch1(AI1+,Pin# 47) and AI Ch2(AI2+,Pin#48),two negative terminals to AI Ch1 negative (AI 1-,Pin#13),and AI Ch2 negative (AI2-mPin#14) .
- Set two sinewave signals(f=50 Hz,Vpp=8 V;f=50 Hz,vpp=5 V).

■ Click **Strat** to run the task.

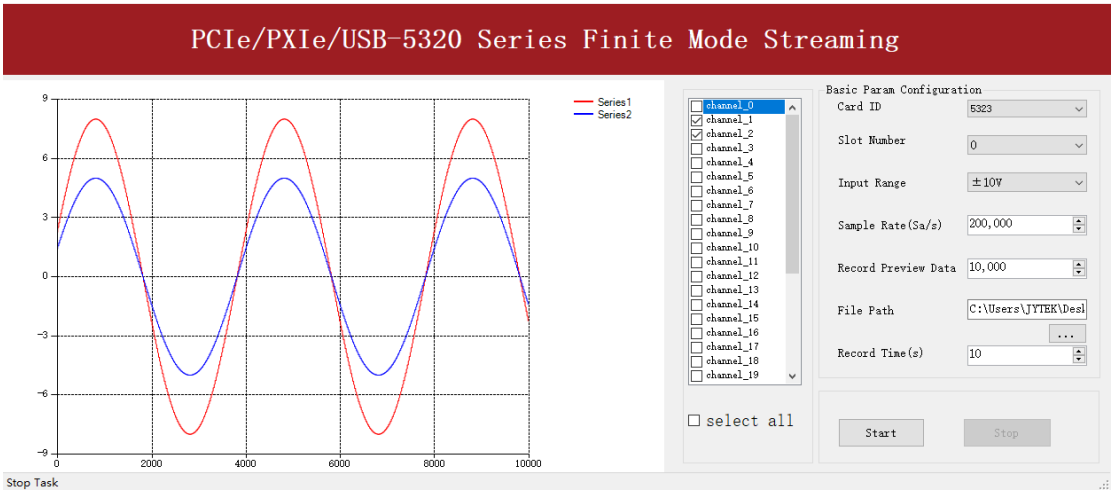

Figure 8 Streaming

7.3 Analog Input Terminal Type

The JY-5321A/5322A/5323A/5324A only provides one analog input terminal type: Differential (DIFF). In the differential mode you can connect ground-referenced signal to the JY-5321A/5322A/5323A/5324A. Connect positive side of signal to AI+ terminal, negative side of signal to AI- terminal as shown in Figure 9. Please see the provided software examples for more information.

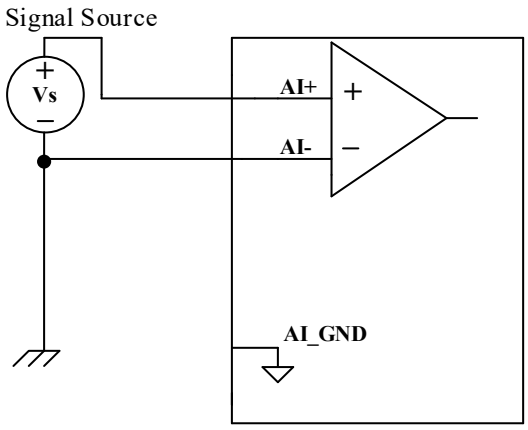

Figure 9 Differential Mode for Grounding Signal

If 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 in Figure 10. 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 10 KΩ to 100 KΩ. At

this level, R has very little impact on the measurement.

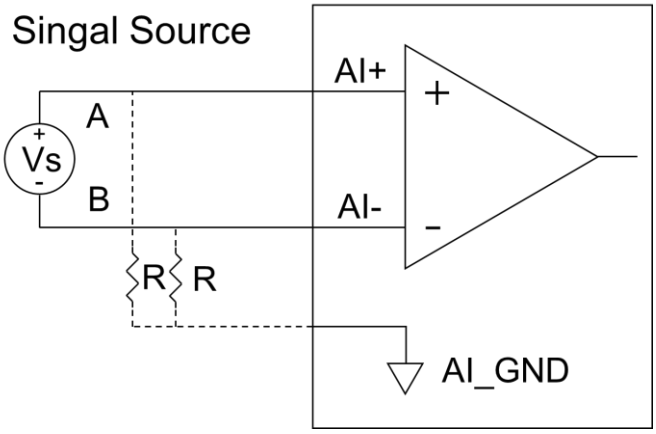

Figure 10 Differential Mode for Floating Signal

7.4 Trigger Source

There are 4 trigger types:

- Immediate trigger
- Software trigger
- analog trigger
- digital trigger

The trigger type is a property and set by driver software. Please find the pro vided software examples for more information.

7.4.1 Immediate trigger

The module will acquire the signal immediately after executing the AI Task wi thout any trigger condition setting by default.

Learn by Example 8.4.1

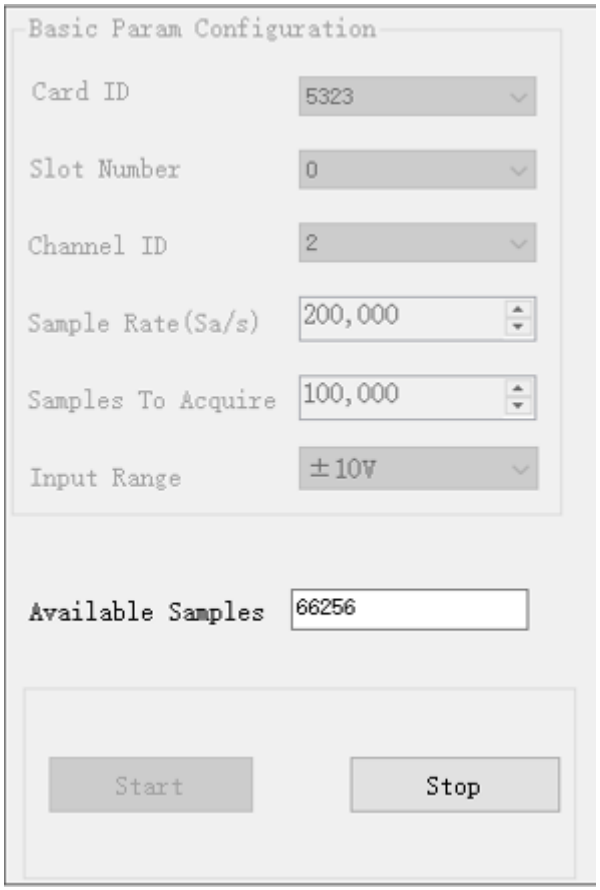

Figure 11 Immediate trigger Paraments

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

7.4.2 Software Trigger

The analog acquisition task will wait on the software trigger signal in the sof tware trigger mode until receiving a software trigger signal from driver, then AI task will start to acquire the data

Learn by Example 8.4.2

- Connect the signal source's positive outputs to JY-5323A AI Ch1(AI1+,Pin#4 7),the negative terminals to AI Ch1 negative (AI1-,Pin#13).
- Set the sinewave signal(f=5 Hz,Vpp=5 V).
- Click **Strat** to run the task.

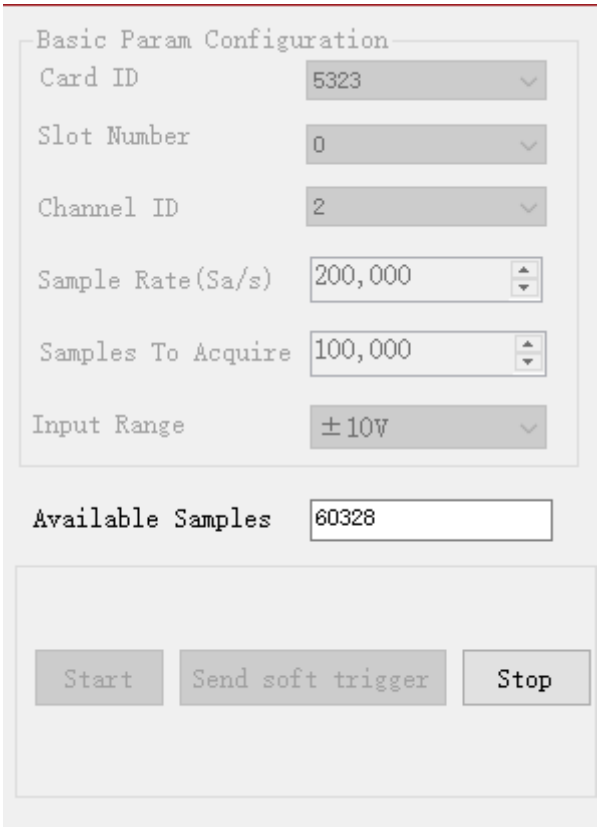

Figure 12 Software trigger Paraments

- Data will not be acquired until there is a positive signal from Software Tr igger when **Send Soft Trigger** is clicked.

- After sending the trigger signal, the result will be like this:

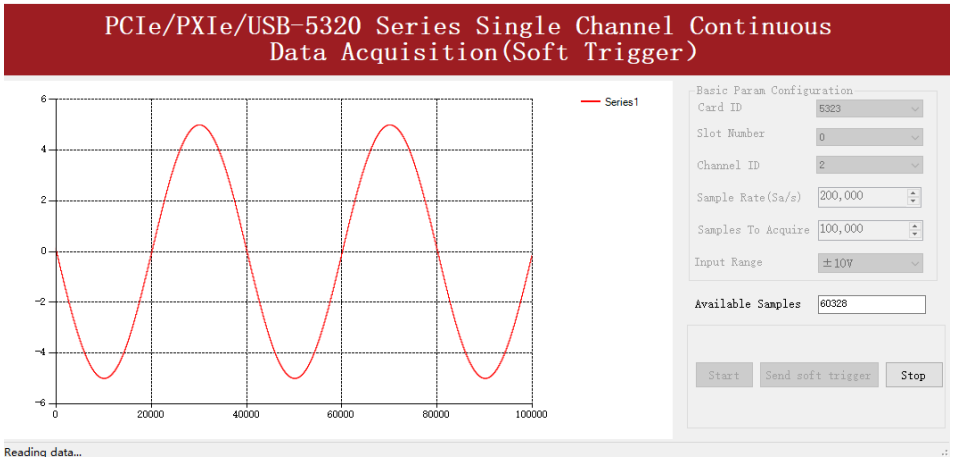

Figure 13 Software trigger Acquisition

7.4.3 External Analog Trigger

User can assign one of measurement channels as the analog trigger source. T

his module provides three kinds of analog trigger: Edge comparator, hysteresis comparator and window comparator. User can select the different analog trig ger comparator by setting the parameter "comparator".

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

Edge comparator

In the Edge comparator, there are two trigger conditions. One is measured va lue above the specified threshold which is called "Rising Slope Trigger". Anoth er is measured value below the specified threshold which is called "Falling Sl ope Trigger".

"Rising Slope Trigger": The Edge comparator will output high when the signal go above the threshold. And it will change to low when the signal goes belo w the threshold as shown in Figure 14.

"Falling Slope Trigger": The Edge comparator will output high when the signal go below the threshold. And it will change to low when the signal goes abo ve the threshold as shown in Figure 15.

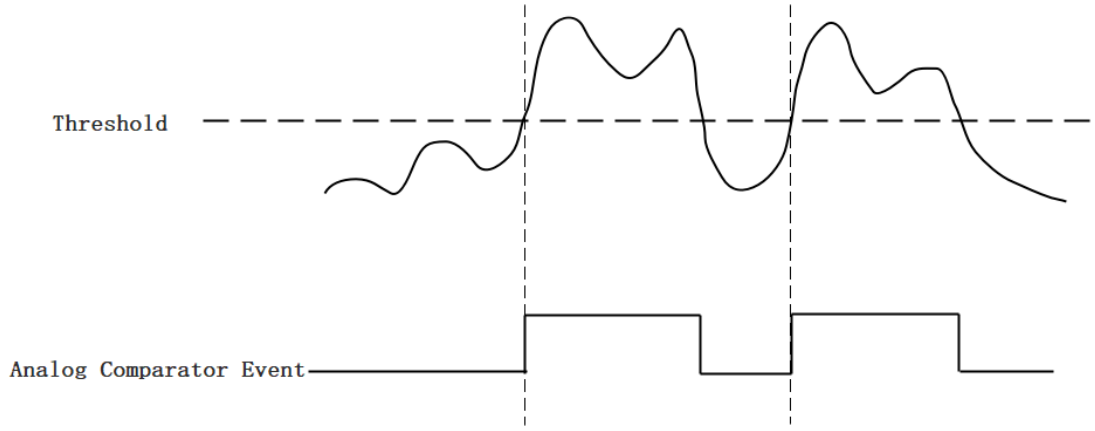

Figure 14 Rising Slope Trigger

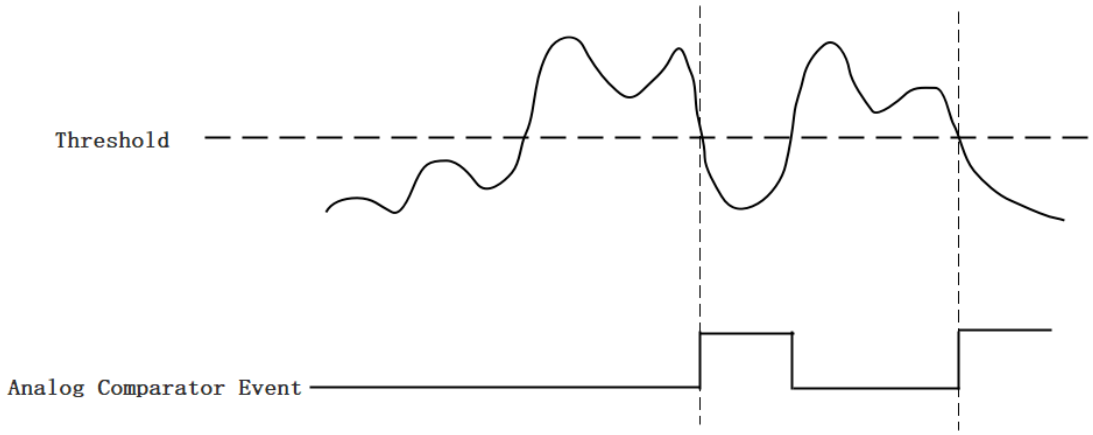

Figure 15 Falling Slope Trigger

Hysteresis comparator

The hysteresis comparator is designed for preventing spurious triggering. User can set hysteresis region by setting high threshold and low threshold. There are two trigger conditions. One is signal passing the hysteresis region from low threshold to high threshold which is called "Hysteresis with Rising Slope Trigger". Another is signal passing the hysteresis region from high threshold to low threshold which is called "Hysteresis with Falling Slope Trigger".

"Hysteresis with Rising Slope Trigger”: The Hysteresis comparator will output high when the signal must first be below the low threshold, then go above the high threshold. It will change to low when the signal goes below the low threshold as shown in Figure 16.

"Hysteresis with Falling Slope Trigger”: The Hysteresis comparator will output high when the signal must first be above the high threshold, then go below the low threshold. It will change to low when the signal goes above the high threshold as shown in Figure 17.

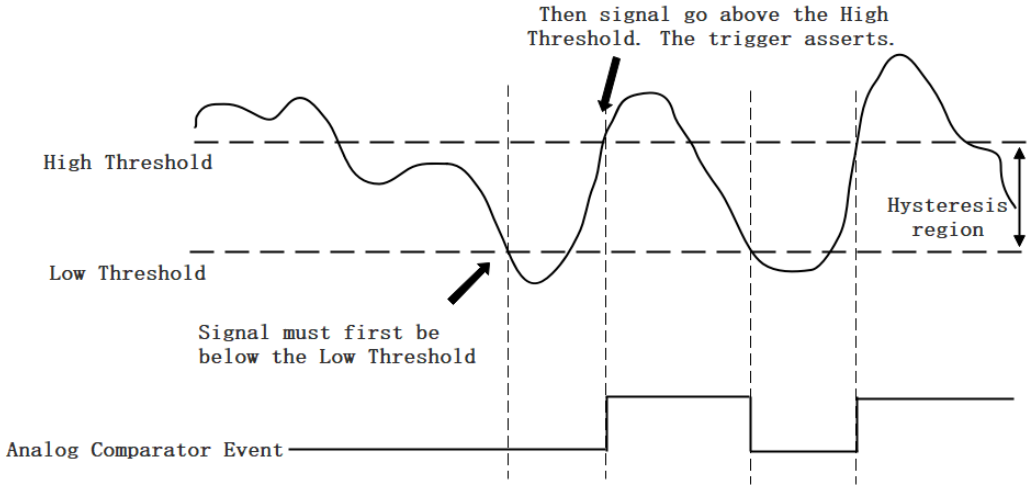

Figure 16 Hysteresis with Rising Slope Trigger

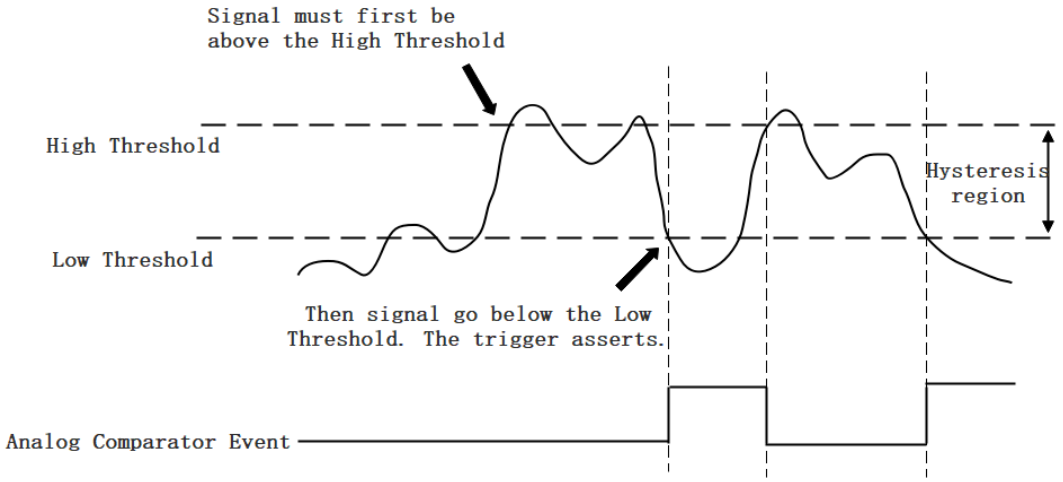

Figure 17 Hysteresis with Falling Slope Trigger

Window comparator

The window comparator is designed to acquire signal from interesting window by setting High Threshold and Low Threshold. There are two trigger conditio ns. One is acquired the signal inside of the window which is call ed "Enterin g Window Trigger". Another is acquired the signal outside of the window whi ch is called "Leaving Window Trigger".

"Entering Window Trigger": The window comparator will output high when th e signal enters the window defined by the Low Threshold and High Threshol d. And it will output low when the signal leaves the window as shown in Fi gure 18.

"Leaving Window Trigger": The window comparator will output high when the

signal leaves the window defined by the Low Threshold and High Threshold. And it will output low when the signal enters the window as shown in Figur e 19.

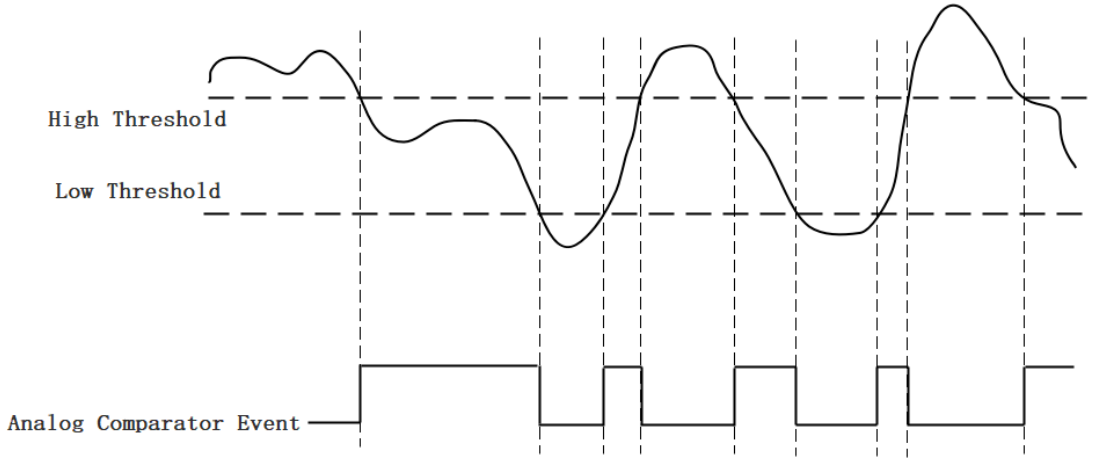

Figure 18 Entering Window Trigger

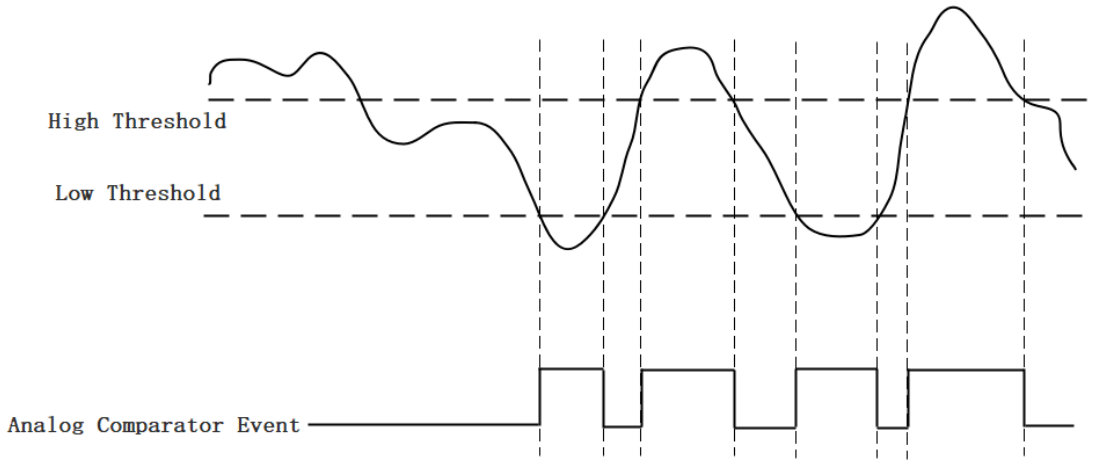

Figure 19 Leaving Window Trigger

Learn by Example 8.4.3

- Connect the signal source's positive outputs to JY-5323A AI Ch0(AI0+,Pin#4 5),the negative terminals to AI Ch0 negative (AI0-,Pin#11).
- Set the sinewave signal(f=50 Hz,Vpp=5 V).
- Click **Strat** to run the task.
- Open **Analog Input-->Winform AI Continuous Analog Trigger**, set the follo wing numbers as shown.

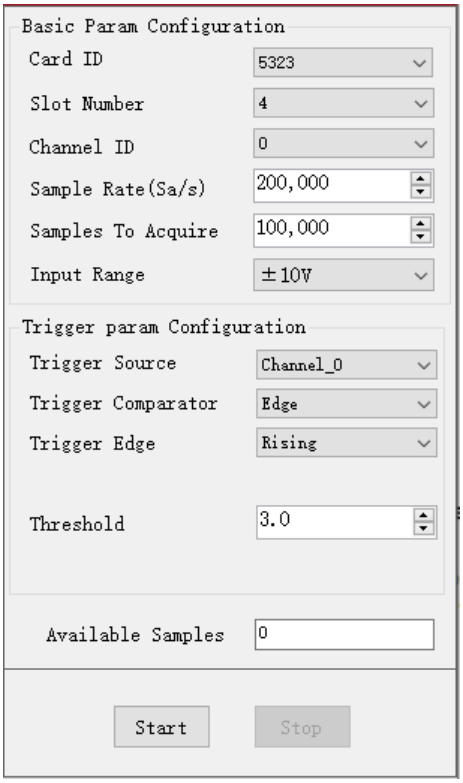

Figure 20 Analog Trigger Paraments

- 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 3.0 V, which is set by **Threshold** above.

■ Click Start, a message will appear in the lower left corner:

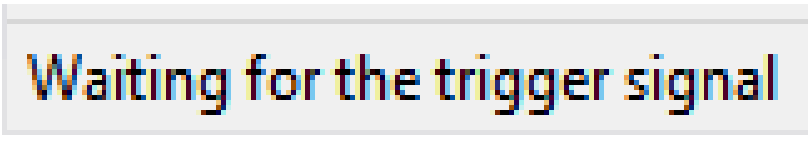

Figure 21 Waiting For Trigger

■ The result is shown below:

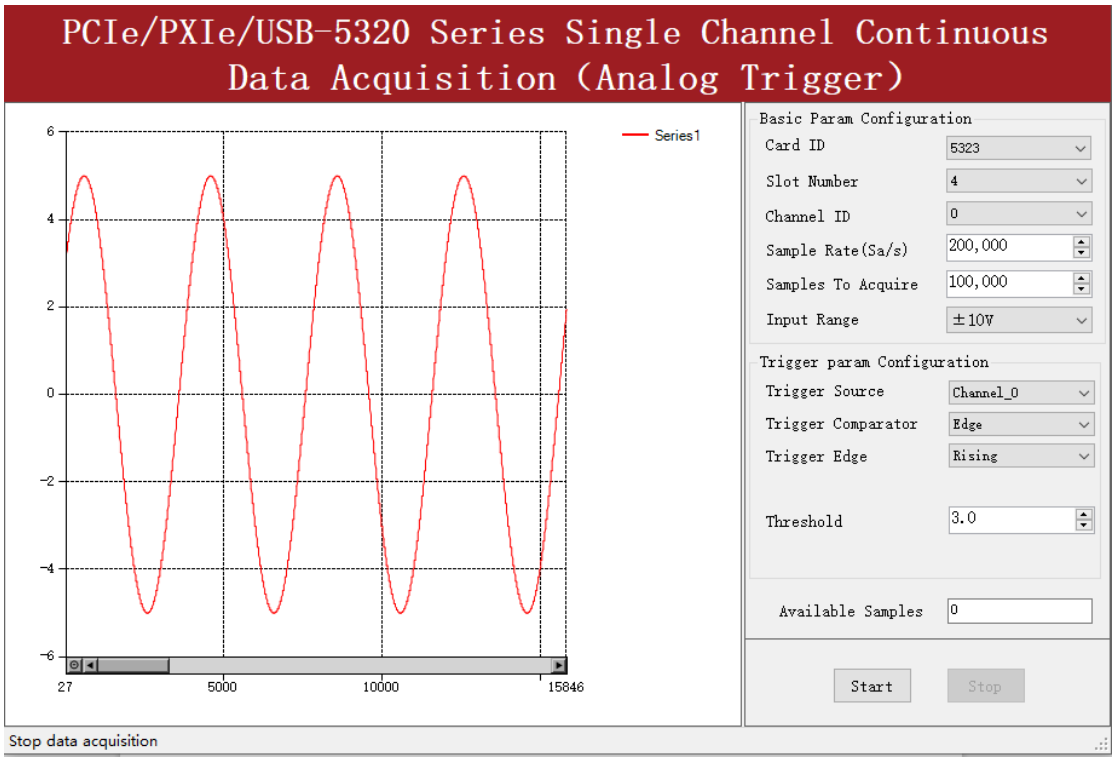

Figure 22 Analog Trigger Acquisition

7.4.4 External Digital Trigger

The module 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 t rigger. The module will monitor the signal on digital trigger source and wait f or the rising edge or falling edge of digital signal which depending on the se t trigger condition, then cause the module to acquire the data as shown in F igure 23.

Figure 23 External Digital Trigger

Learn by Example 8.4.4

■ Connect the signal source two positive terminals to JY-5323A AI Ch1, (AI1

+, Pin #47) and digital trigger source (PFI 0, Pin#1), two negative terminals to the ground of analog input (AI1-, Pin#13) and the ground of digital input/output (DGND, Pin#5). (AI1+, AI_GND) consists of a RSE input. (PFI0, DGND) provides the trigger signal.

- Set a sinewave signal (f=20Hz, Vpp=5V) and a squarewave signal (f=8Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous Digital Trigger**.
- **Trigger Source** must match the pin on 5323A.
- There are two **Trigger Edge**: **Rising** and **Falling**.
- Click **Start** and the result shows below:

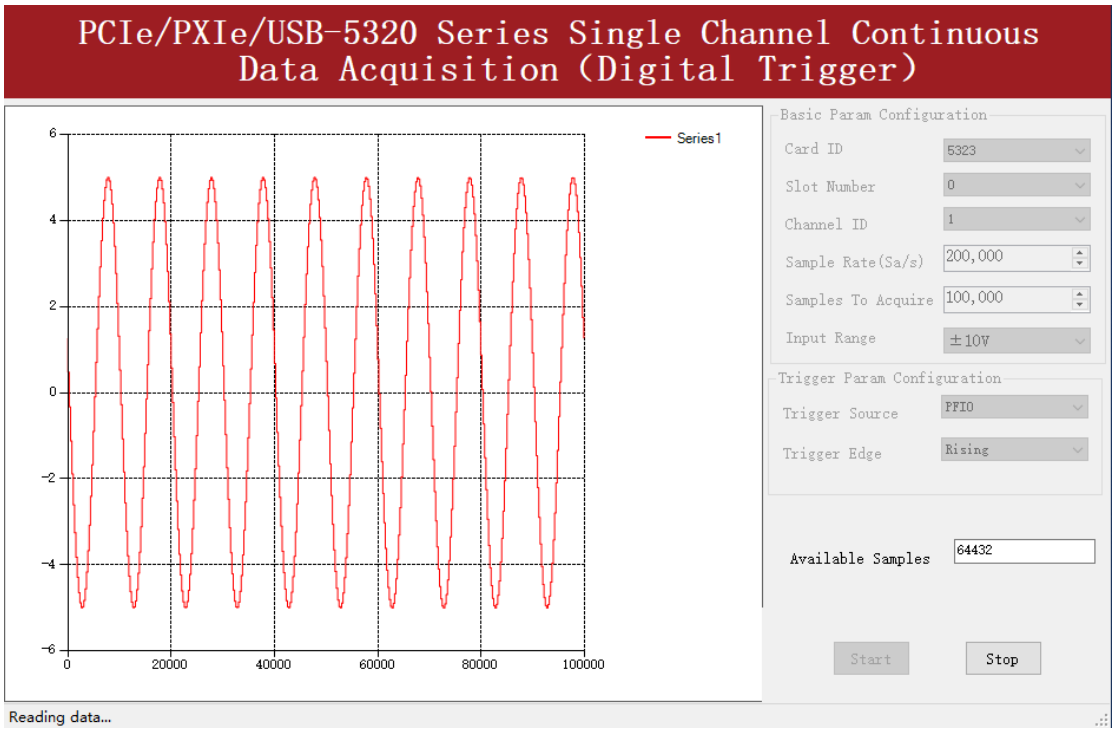

Figure 24 Digital Trigger Acquisition

- 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.

7.5 Trigger Mode

JY-5321A/5322A/5323A/5324A supports several trigger modes: Start Trigger, Reference Trigger, ReTrigger and Sample Clock.

7.5.1 Start Trigger

In this mode, data acquisition begins immediately after the trigger. This trigge r mode is suitable for continuous acquisition and finite acquisition. As shown in Figure 25. Please see the provided software examples for more information.

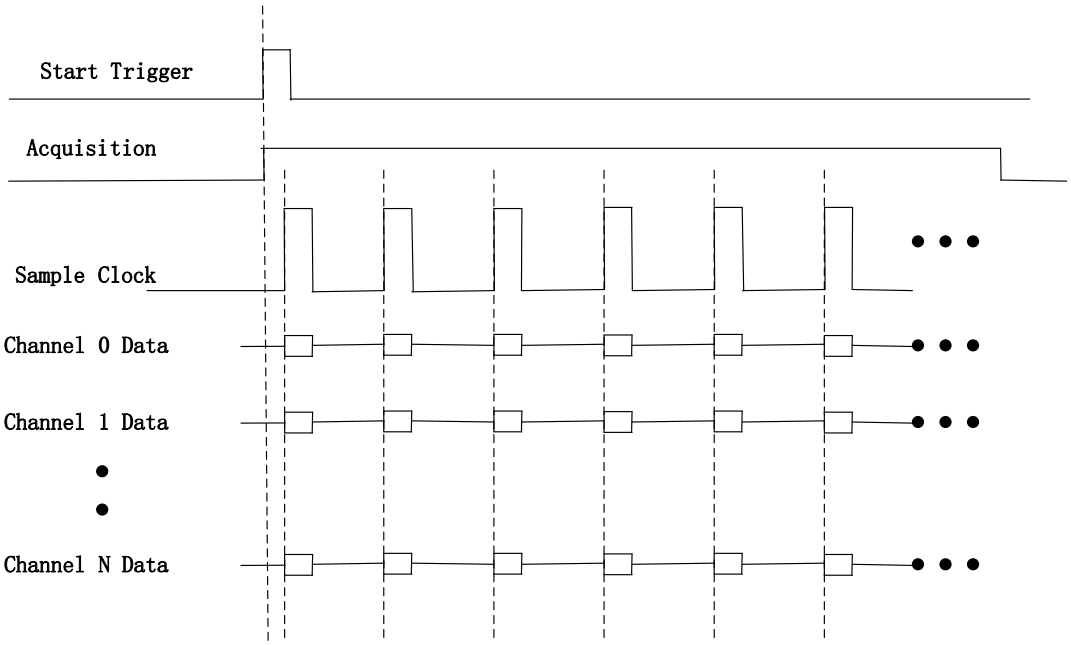

Figure 25 Start Trigger

7.5.2 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 i s 0. First, you need to start the acquisition task and then the acquisition tas k will return the acquired data when the trigger condition is met. An exampl e is shown below.

Example

- Total samples: 1000;
- Channel Count: 1
- Pre-trigger samples: 10;
- It returns total 1000 samples, 10 samples to be acquired before trigger, 9 90 samples after trigger.

The acquisition timing is shown in Figure 26. Please see the provided softwar e examples for more information.

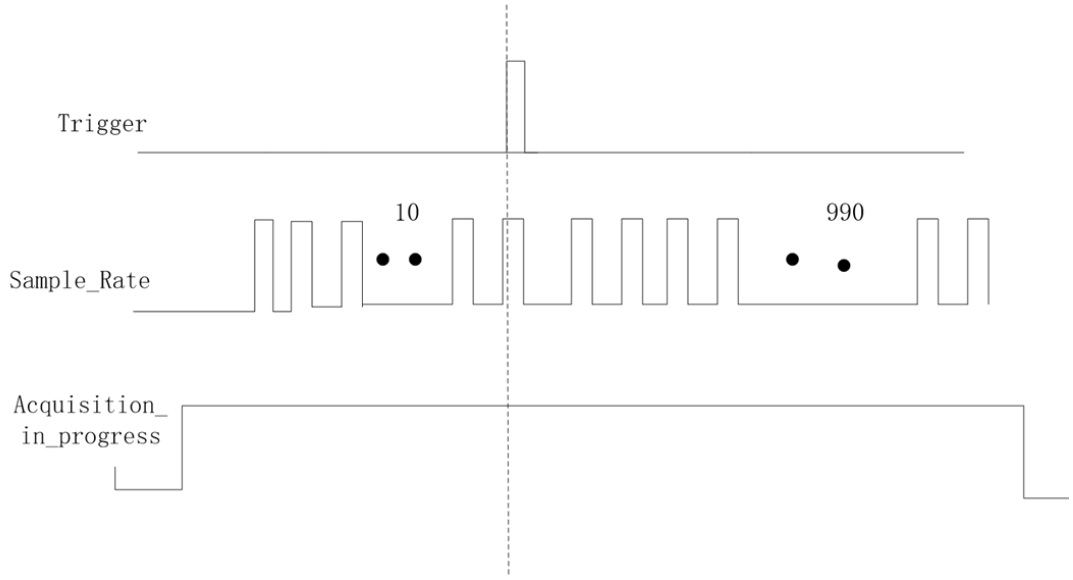

Figure 26 Reference Trigger

7.5.3 ReTrigger

JY-5321A/5322A/5323A/5324A series modules support retrigger mode. In the r etrigger mode, you can set the number of trigger times and the length of ea ch acquisition. Assuming that the number of trigger times is N and the lengt h of each trigger acquisition is M, the length of all acquisition data is N * M * channel counts as shown in Figure 27.

When the number of trigger times is - 1, it will wait on trigger infinitely.

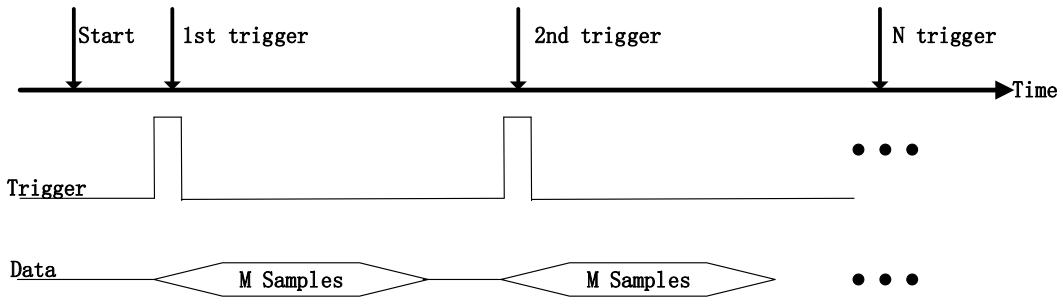

Figure 27 Re-Trigger

Learn by Example 8.5.3

- Connect the signal source's positive terminal to JY-5323A AI Ch0 (AI0+, Pi n#45), the negative terminal to the ground (AI0-, Pin#11) consists of an RSE input.
- Set a sinewave signal (f=50Hz, Vpp=5V).

■ Open **Analog Input-->Winform AI Finite Analog Trigger,** set the following numbers as shown.

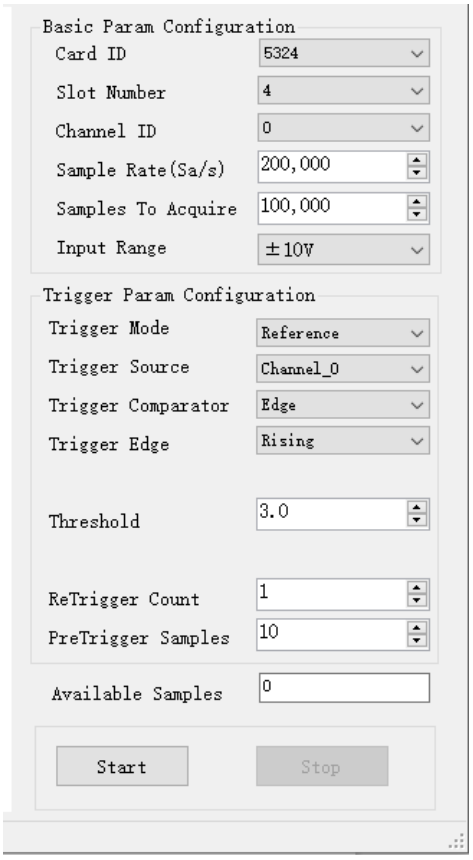

Figure 28 Retrigger Paraments

- You can use three different kinds of triggers in this program as mentioned in 错误!未找到引用源。. *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:

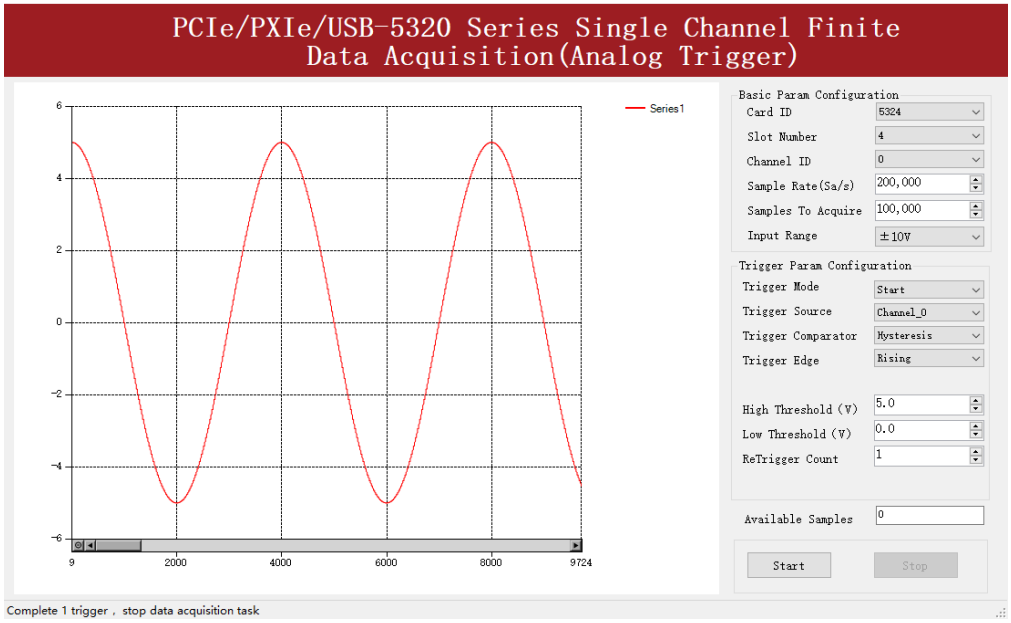

Figure 29 Retrigger In Start Trigger Mode

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

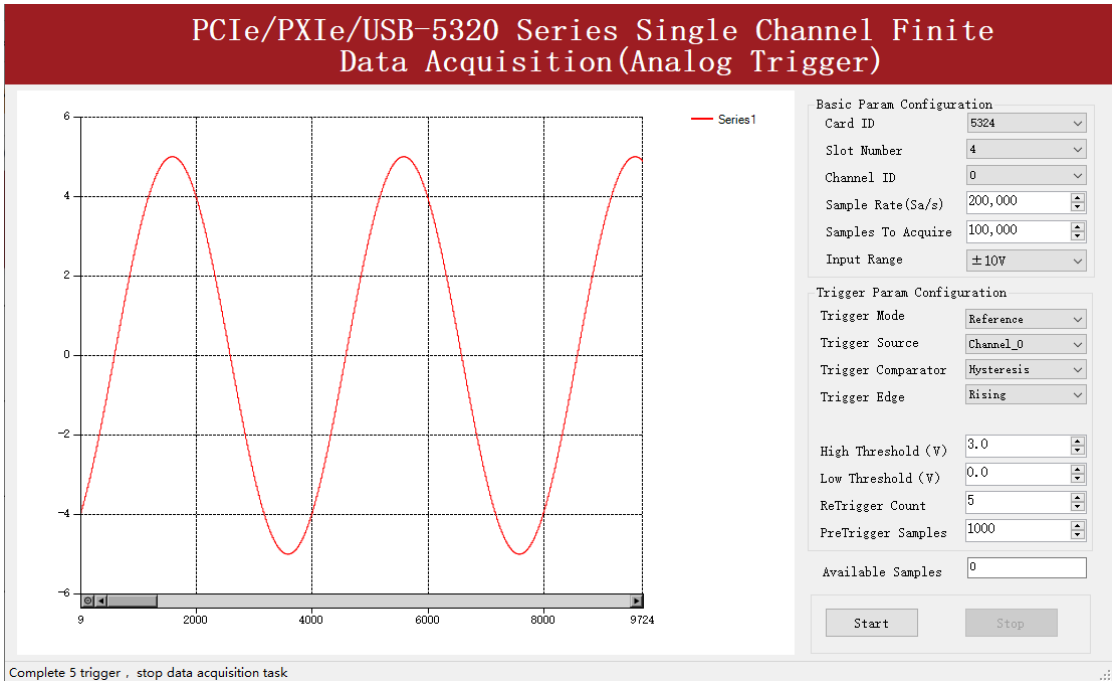

Figure 30 Retrigger In Reference Trigger Mode

- 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”.

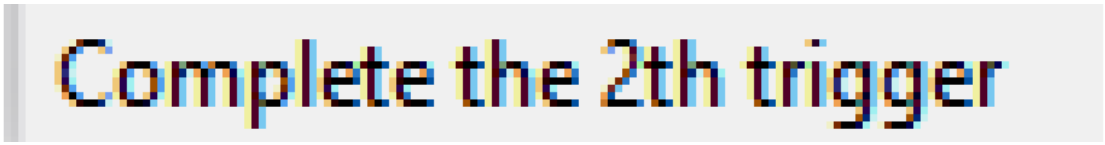

Figure 31 Complete Retrigger Count

➢ It shows the acquisition process through every trigger signal.

7.5.4 Sample Clock

For all counter measurement applications with buffered measurement, JY-5321A/5322A/5323A/5324A provides 3 sample clock options as follows:

- Internal

The internal sample clock is generated by dividing down the 200MHz base clock, and can be set independently for each counter.

To use the internal sample clock, configure as follows:

1. Set JY5320CITask.SampleClock.Source to CISampleClockSource.Internal
2. Set JY5320CITask.SampleClock.Internal.Rate to specify sample rate

- External

External sample clock refers to an external signal input from a terminal as the sample clock

To use the external sample clock, configure as follows:

1. Set JY5320CITask.SampleClock.Source to CISampleClockSource.External
2. Set JY5320CITask.SampleClock.External.ExpectedRate to the rate of the external sample rate.

   This property's value helps the driver determine a more suitable DDR writing frequency.

   Can be set to an approximate value (sometimes the external sample clock may not have a fixed frequency). But cannot be less than the actual sampling rate, otherwise it will cause DDR write exception.

   The default value of this property is -1, which means that the driver will be set with the safest value.

- Implicit

Using an implicit sampling clock means the counter will send data to the buf fer whenever there is a new measurement or count value.

To use the implicit sample clock, configure as follows:

1. Set JY5320CITask.SampleClock.Source to CISampleClockSource.Implicit.
2. Set JY5320CITask.SampleClock.Implicit.ExpectedRate to desired sample rate. This property has the same effect as JY5320CITask.SampleClock.External.Exp ectedRate.

7.6 Digital I/O Operations

JY-5321A/5322A/5323A/5324A provides powerful programmable digital I/O func tions. Please see the provided software examples for more information.

7.6.1 Static DI/DO

Programmable I/O supports static TTL, 8 ports<0..7> with 2 lines which is tot al 16 channels digital I/O. User can read these I/O information through softw are polling.

Learn by Example 8.6.1

- In this example JY-5323A outputs a digital signal by its DO function and reads it back by its DI function.
- Connect Connector0 of JY-5323A to the TB-68 terminal block.
- Connect Port 0/Line 0~1 (P0.0~0.1) to Port 1/Line 0~1 (P1.0~1.1). JY-5323 A sends a digital signal through Port 0 and reads the signal back from P ort 1.
- Open the first program Digital Output-->Winform DO SinglePoint.
- Select port 1 for Digital Output, Set Line 0 in High-Level positions, make sure all other lines are in Low-Level positions. Click Start to generate the High-Levels as shown.

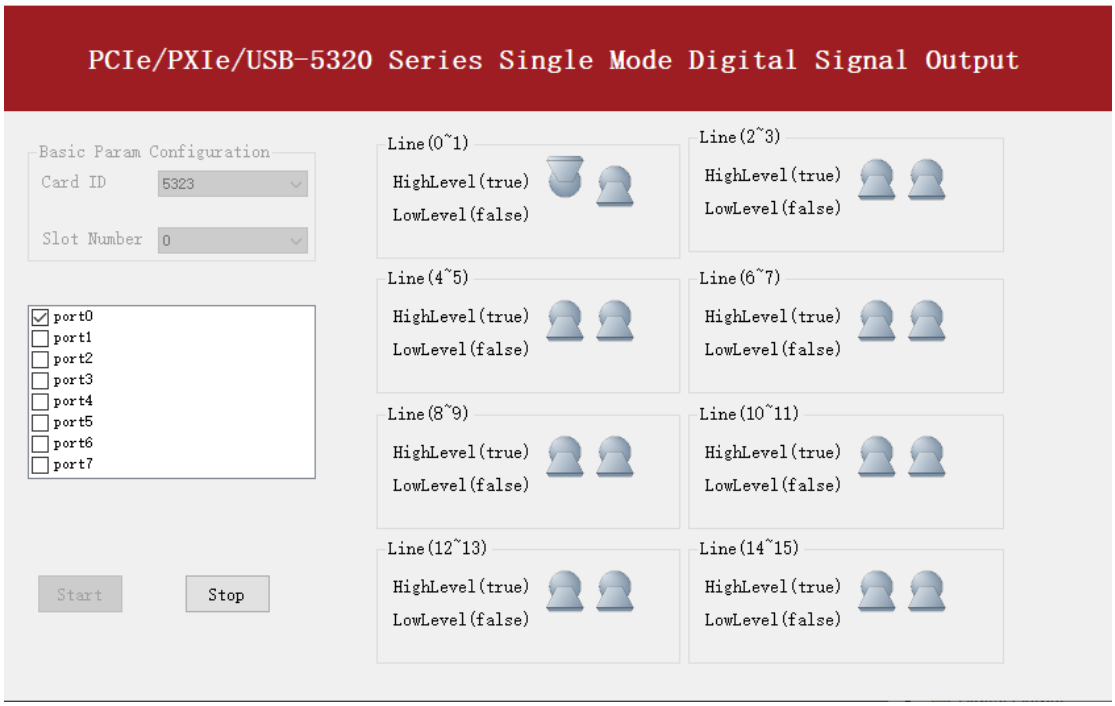

Figure 32 Single Digital Output

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

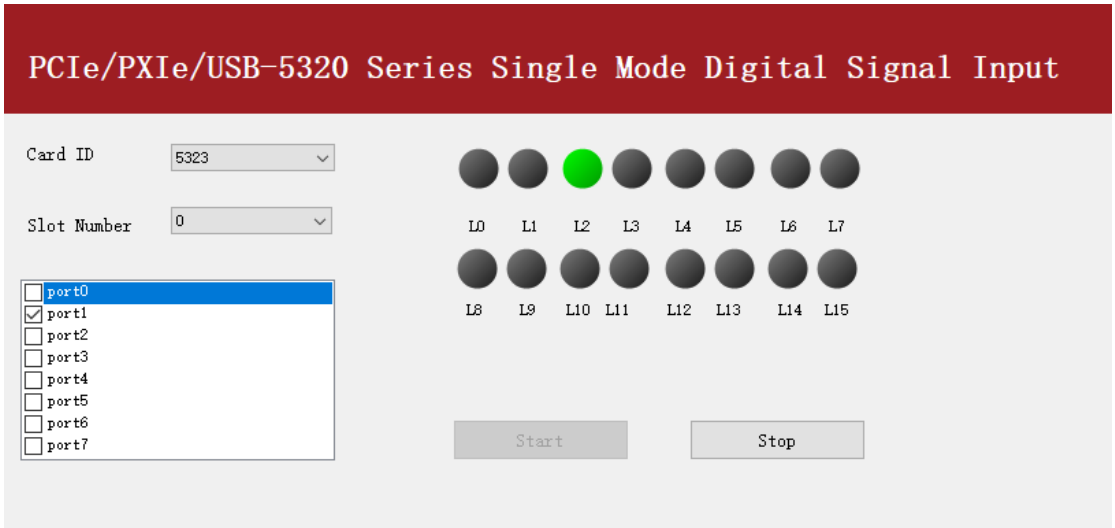

Figure 33 Single Digital Intput

7.7 Counter Measurement Operations

The JY-5321A/5322A/5323A/5324A has two or one identical 32-bit channels of timer/counter as shown in Figure 34.

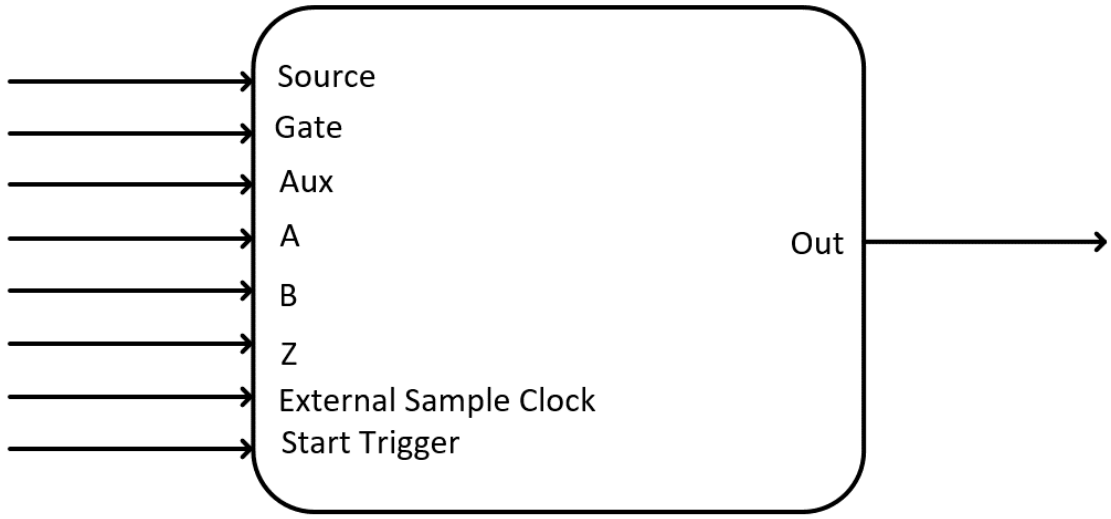

Figure 34 Counter Input Diagram (TBD)

Each counter has eight input terminals and one output terminal, and these te rminals have different functions in different counter measurement application described below:

- 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.

7.7.1 Edge Counting

The counter counts the number of active edges of input signal. Default, the i nput signal must be connected to Counter Source terminal.

Set JY5320CITask.Type to CIType.EdgeCounting to use this function.

Timing

1) Single Mode

The counting value is written to the register on each rising edge or falling ed

ge of the measured signal as shown in Figure 35.

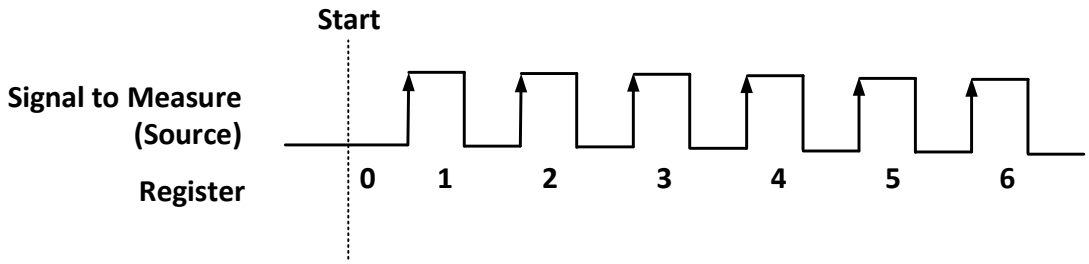

Figure 35 Simple Edge Counting in Single Mode

To configure the counter to work in this mode, set JY5320CITask.Mode to CI Mode.Single.

2) Finite/Continuous Mode with Explicit Sample Clock

The counting value is stored into the buffer on each rising edge of the samp le clock as shown in Figure 36.

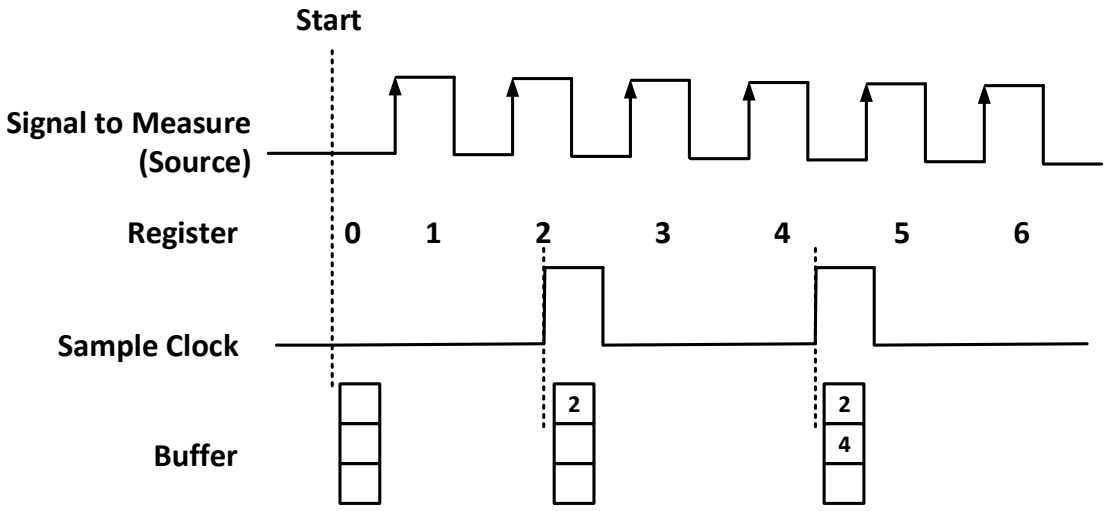

Figure 36 Buffered Edge Counting with Explicit Sample Clock

To configure the counter to work in this mode, set JY5320CITask.Mode to CI Mode.Finite or CIMode.Continuous, and set JY5320CITask.SampleClock.Source to CISampleClockSource.Internal or CISampleClockSource.External.

3) Finite/Continuous Mode with Implicit Sample Clock

In implicit mode, the signal active edge as the implicit sample clock edge. Th e counting value is stored into the buffer on each rising edge of the measur ed signal as shown in Figure 37.

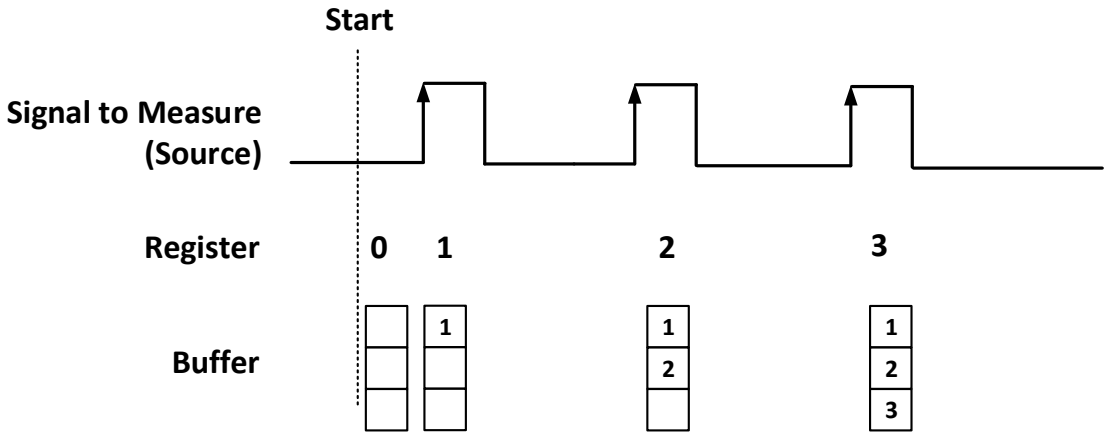

Figure 37 Simple Edge Counting with Implicit Sample Clock

To configure the counter to work in this mode, set JY5320CITask.Mode to CI Mode.Finite or CIMode.Continuous, and set JY5320CITask.SampleClock.Source to CISampleClockSource.Implicit.

Pause Trigger

Pause trigger is used to pause counting when the input signal is active depe nding on active polarity configuration as shown in Figure 38. Default, the Pau se Trigger signal must be connected to Counter Gate terminal.

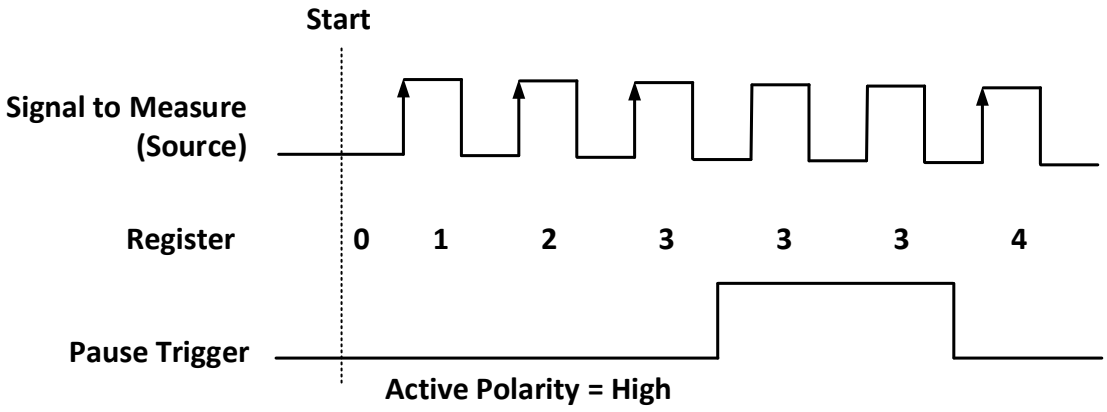

Figure 38 Pause Trigger

To cofigure the pause trigger, use the properties as below:

- JY5320CITask.EdgeCounting.Pause.ActivePolarity – To set active level (high o r low) to pause counting.

Count Direction

User can control the counting direction through software configuration or by an external input signal. Default, the external control direction signal must be

connected to Counter 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 39.

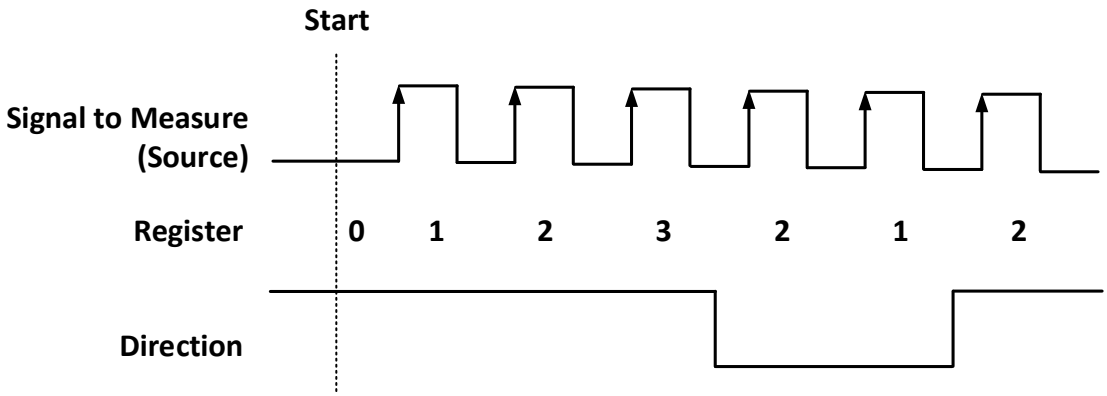

Figure 39 Count Direction

To cofigure the count direction, use the properties as belows:

- JY5320CITask.EdgeCounting.Direction – To specify count up, count down, or controlled by an external signal.

Exporting Count Event

When the counting value reaches the specified threshold, the counter will generate a pulse. To change the threshold, using following property:

- JY5320CITask.EdgeCounting.OutEvent.Threshold

Terminals

To change the terminal of signals instead of using its default value as shown in chapter 7.3, using following properties:

- JY5320CITask.EdgeCounting.InputTerminal – Signal-to-measure input terminal.
- JY5320CITask.EdgeCounting.Pause.Terminal – Pause signal input terminal.
- JY5320CITask.EdgeCounting.DirTerminal – External direction control signal input terminal.
- JY5320CITask.EdgeCounting.OutEvent.Terminal – Count event output terminal.

Learn by Examples8.7.1

Connect the signal source's positive terminal of a signal source to PCIe-5323A counter0's edge counting source (CTR0_Source/A, Pin#1), negative terminal to the ground (DGND, Pin#5) . (CTR0_Source, DGND) consists of an edge counting counter input and they share the same ground.

Set a squarewave signal (f=100Hz, Vpp=8V).

Single Mode

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

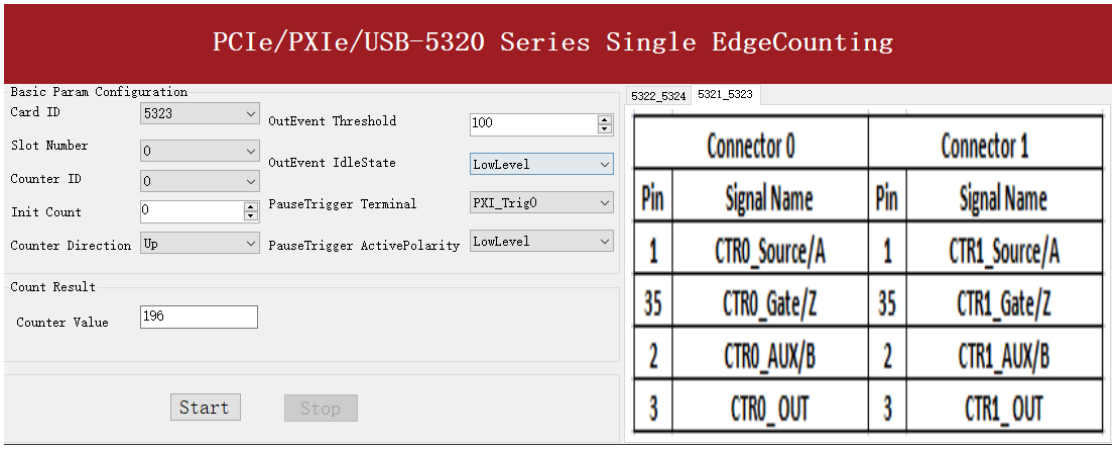

Figure 40 EdgeCounting For Single Mode

- 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 100 every second for a 100Hz sinewave.

Finite/Continuous Mode

- Open **Counter Input-->Winform CI Finite/Continuous EdgeCounting**, set the following numbers as shown:
- Click Start to start counting by rising edge. The result is shown below

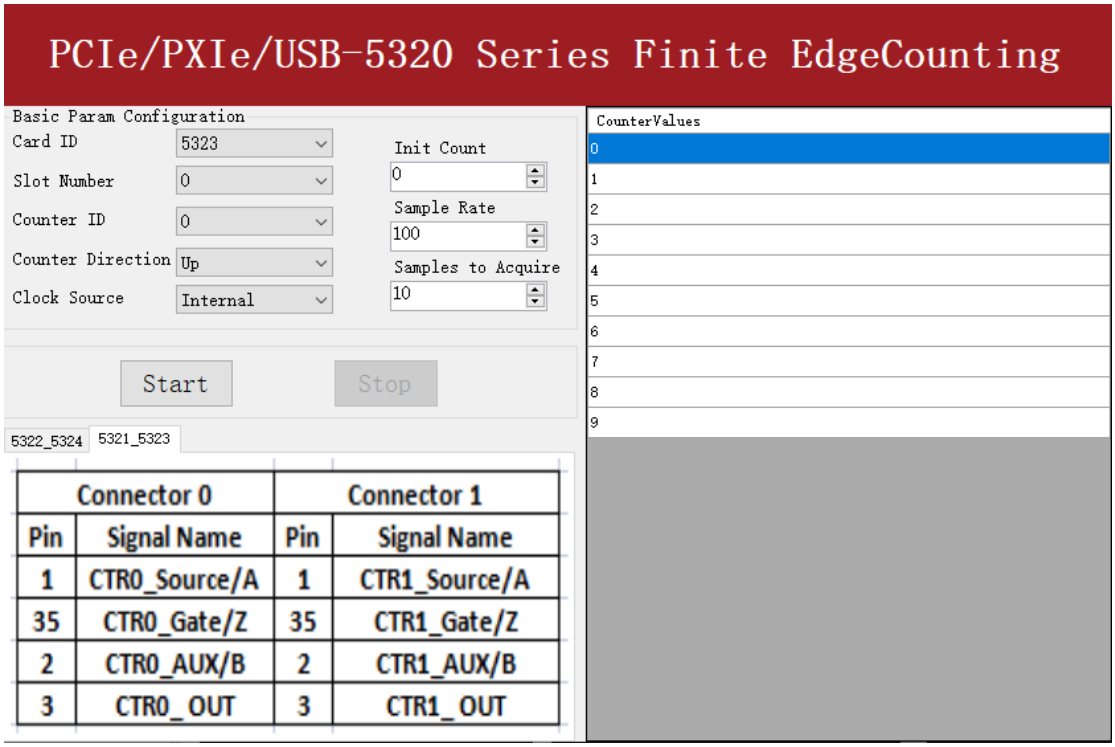

Figure 41 EdgeCounting For Finite Mode

- The table in the sample program is a connection diagram for your convenience.
- Counter Direction is set by **Counter Direction**.
- There are one clock sources in JY-5323A Internal: This example uses **Internal** mode set by **Clock Source**.

7.7.2 Pulse Measurement

The counter measures the high-level and low-level duration of a pulse on a signal. Default, the input signal must be connected to Counter Gate terminal.

Set JY5320CITask.Type to CIType.Pulse to use this function.

Timing

1) Single Mode

The counting 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 42.

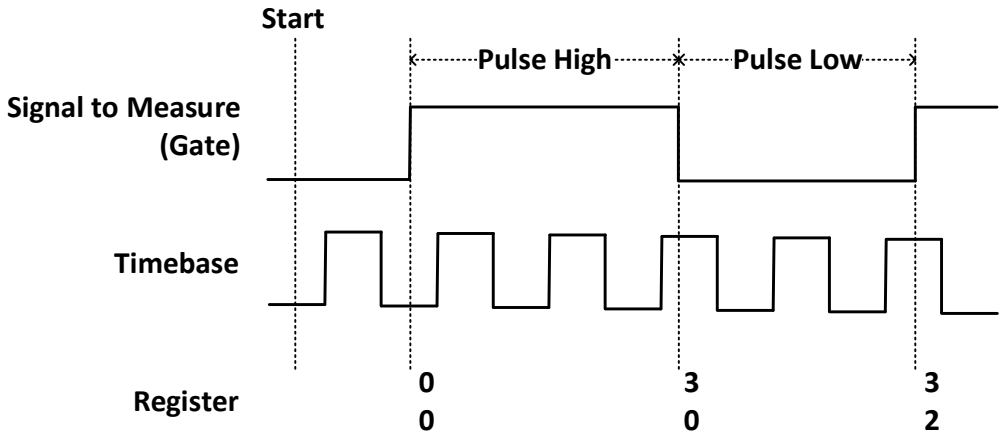

Figure 42 Pulse Measurement in Single Mode

To configure the counter to work in this mode, set JY5320CITask.Mode to CI Mode.Single.

2) Finite/Continuous Mode with Explicit Sample Clock

The counting value of the duration of the high-level or low-level is stored int o the buffer on each rising edge of the sample clock, as shown in Figure 43

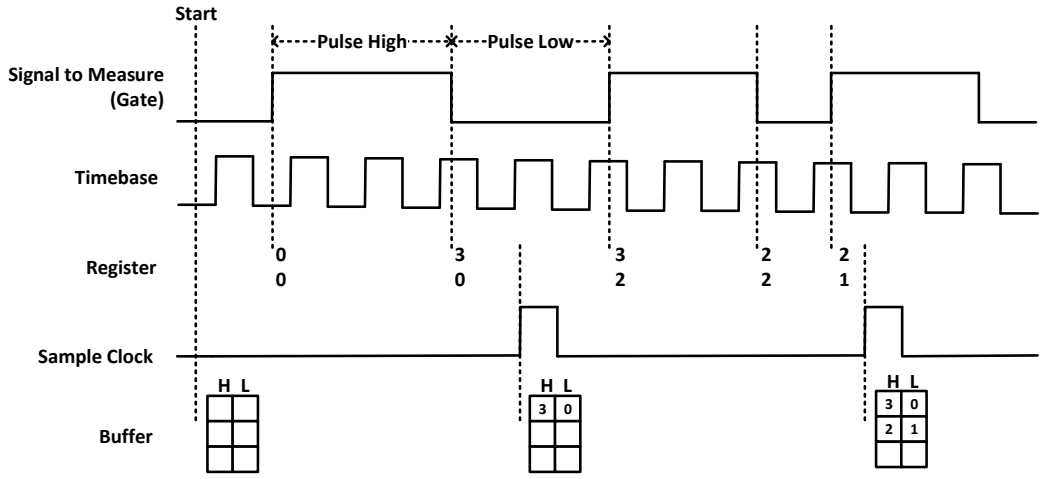

Figure 43 Pulse Measurement with Explicit Sample Clock

To configure the counter to work in this mode, set JY5320 CITask.Mode to CI Mode.Finite or CIMode.Continuous, and set JY5320 CITask.SampleClock.Source t o CISampleClockSource.Internal or CISampleClockSource.External.

3) Finite/Continuous Mode with Implicit Sample Clock

In implicit mode, the signal active edge as the implicit sample clock edge. Th e counting value of the duration of the high-level or low-level is stored i

nto the buffer on each rising edge of the measured pulse, as shown in F igure 44.

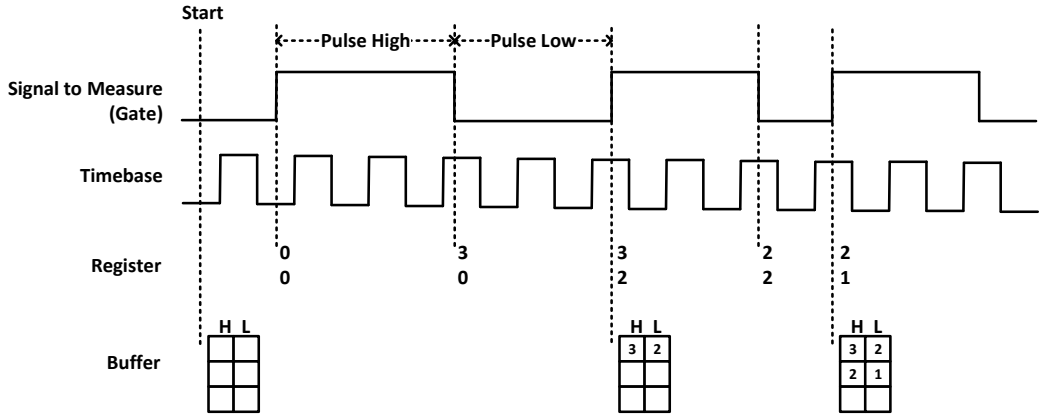

Figure 44 Pulse Measurement with Implicit Sample Clock

To configure the counter to work in this mode, set JY5320CITask.Mode to CI Mode.Finite or CIMode.Continuous, and set JY5320CITask.SampleClock.Source to CISampleClockSource.Implicit.

Timebase

JY-5321A/5322A/5323A/5324A provides four options for timebase source as fol lows:

- Internal 200MHz: - Same signal as the 200MHz base clock generated by PLL.
- Internal 5MHz – Generated by dividing down the 200MHz timebase.
- Internal 100kHz – Generated by dividing down the 200MHz timebase.
- External - Use a signal on a terminal as the timebase

By default, the counter uses the onboard 200MHz timebase to measure pulse s. Use the property JY5320CITask.PulseMeas.Timebase to configure the timebas e.

Terminals

To change the terminal of signals instead of using its default value shown in chapter 2.6, use following properties:

- JY5320CITask.PulseMeas.InputTerminal – Signal-to-measure input terminal.

- JY5320CITask.PulseMeas.Timebase.External.Terminal – External timebase input terminal.

Learn by Examples 8.7.2

- Connect the signal source's positive terminal to PCIe-5323A counter0's pulse measure source (CTR0_Gate/Z, Pin#35), negative terminal to the ground (DGND, Pin#39) . (CTR0_Gate/Z, DGND) consists of a pulse 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 PulseMeasure**
- 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):

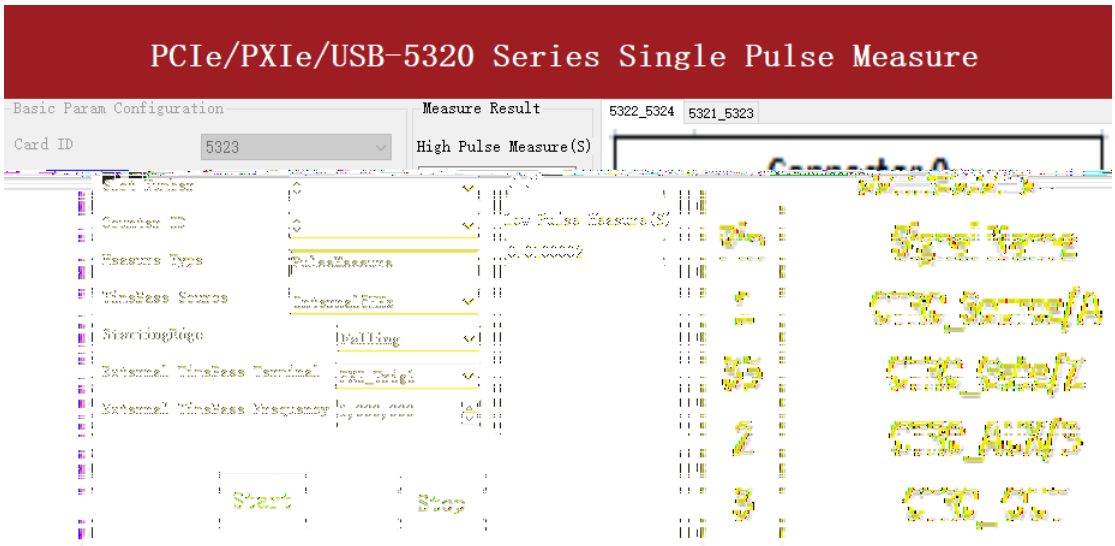

Figure 45 Pulse Measure Value For Single Mode

- 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.**
- Click Start to begin the finite/continuous pulse measurement. The result is shown below:
- The numbers show the duration of **High/Low Pulse** in one signal period

and match the duty cycle set before.

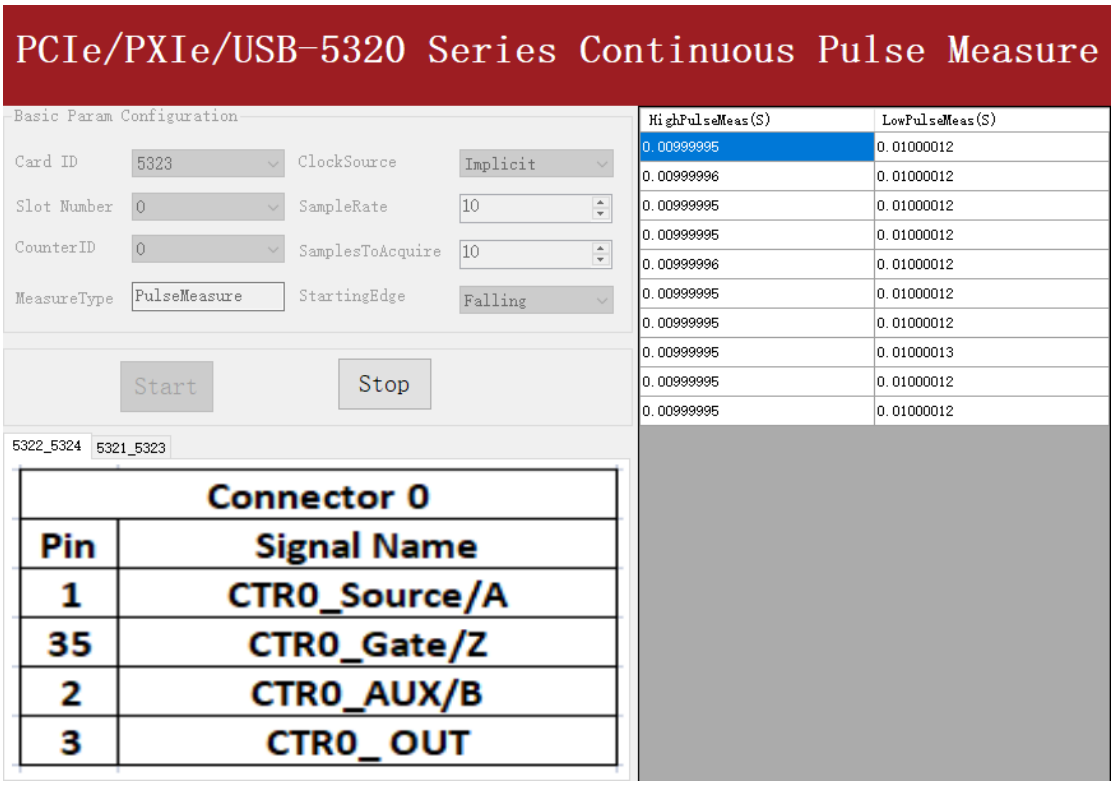

Figure 46 Pulse Measure Values For Continuous Mode

7.7.3 Frequency Measurement

The counter measures the frequency of the signal. Default, the measured sign al must be connected to Counter Gate terminal.

Set JY5320CITask.Type to CIType.Frequency to use this function.

Timing

1) Single Mode

Frequency Measurement without sample clock is actually using Pulse Width M easuement internally, refer to chapter 7.7.2 for more information.

Every time the user reads the data, driver will automatically calculate the fre quency (f_x) according to the HighTick ($tick_h$), LowTick ($tick_l$) values and kno wn frequency of the timebase (f_{base}) according to the fomular and return th e signal frequency to the user.

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

To configure the counter to work in this mode, set JY5320CITask.Mode to CI

Mode.Single.

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

Between every two rising edges of the sample clock, the counter counts the number of full periods ($T1$) of the signal, 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 47.

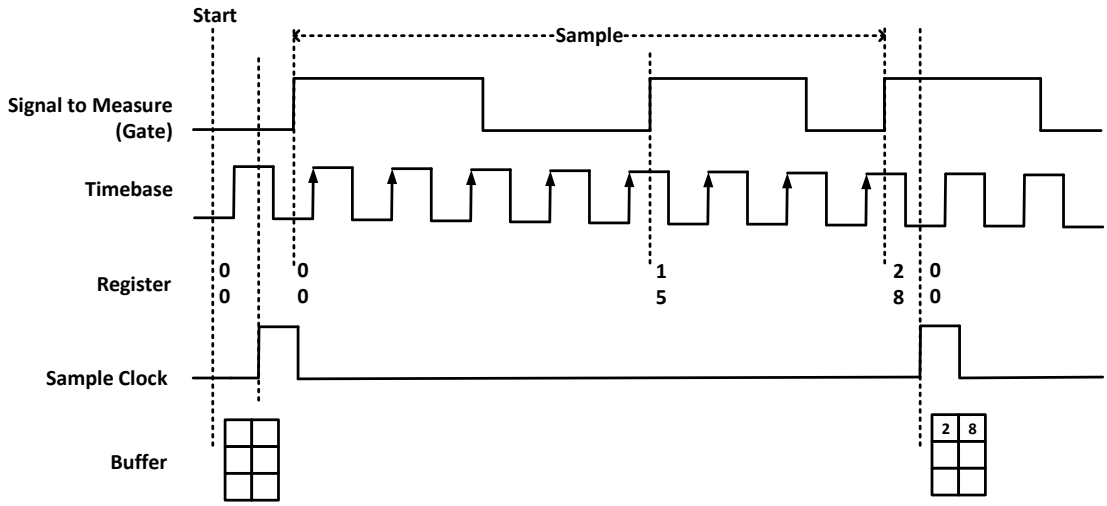

Figure 47 Frequency Measurement with Explicit Sample Clock

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

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

To configure the counter to work in this mode, set JY5320CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5320CITask.SampleClock.Source to CISampleClockSource.Internal or CISampleClockSource.External.

3) Finite/Continuous Mode with Implicit Sample Clock

Frequency Measurement with implicit sample clock is actually using Pulse Measuement internally. Refer to chapter 7.7.2 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 fomular and return the result to the user.

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

To configure the counter to work in this mode, set JY5320CITask.Mode to CI Mode.Finite or CIMode.Continuous, and set JY5320CITask.SampleClock.Source to CISampleClockSource.Implicit.

Timebase

JY-5321A/5322A/5323A/5324A provides four options for timebase source as fol lows:

- Internal 200MHz: - Same signal as the 200MHz base clock generated by PLL.
- Internal 5MHz – Generated by dividing down the 200MHz timebase.
- Internal 100kHz – Generated by dividing down the 200MHz timebase.
- External - Use a signal on a terminal as the timebase

By default, the counter uses the onboard 200MHz timebase to measure pulse s. Use the property JY5320CITask.FrequencyMeas.Timebase to configure the ti mebase.

Terminals

To change the terminal of signals instead of using its default value shown in chapter 7.3, using following properties:

- JY5320CITask.FrequencyMeas.InputTerminal – Signal-to-measure input termin al.
- JY5320CITask. FrequencyMeas.Timebase.External.Terminal – External timebas e input terminal.

Learn by Examples 8.7.3

- Connect the signal source's positive terminal to PCIe-5323A counter0's fre quency measure source (CTR0_Gate/Z, Pin#35), negative terminal to the g round (DGND, Pin#39) . (CTR0_Gate/Z, DGND) consists of a pulse measur e counter input and they share the same ground.
- Set a squarewave signal (f=50Hz, Duty Cycle=50%, Vpp=5V).

Fintie/Continuous Mode

- Open **Counter Input-->Winform CI Finite/Continuous Frequency Measure.**
  - The table in the sample program is a connection diagram for your conve

nience.

- 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.

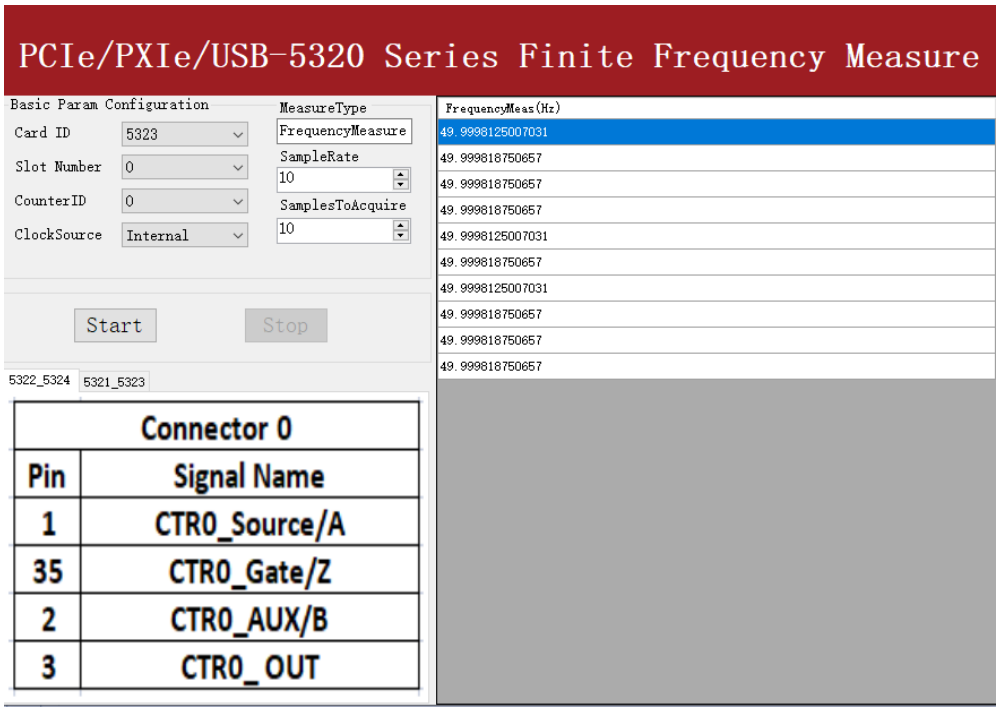

Figure 48 Frequency Measure For Continuous Mode

7.7.4 Period Measurement

The counter measures the period of the signal. Default, the signal must be connected to Counter Gate terminal.

Set JY5320CITask.Type to CIType.Period to use this function.

Period Measurements is using Frequency Measurement internally and returns the reciprocal of Frequency Measuremnt. Refer to chapter for more information.

Learn by Example 8.7.4

- Connect the signal source's positive terminal to PCIe-5323A counter0's frequency measure source (CTR0_Gate/Z, Pin#35), negative terminal to the ground (DGND, Pin#39) . (CTR0_Gate/Z, DGND) consists of a pulse 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 Period Measure** and click Start. The result is shown below by **Period Measure(S):**

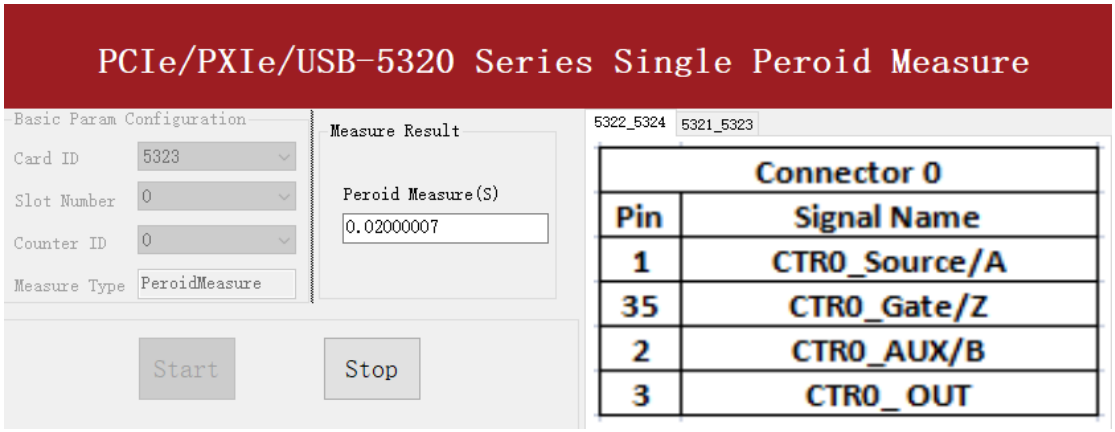

Figure 49 Period Measure For Single Mode

- 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)**.

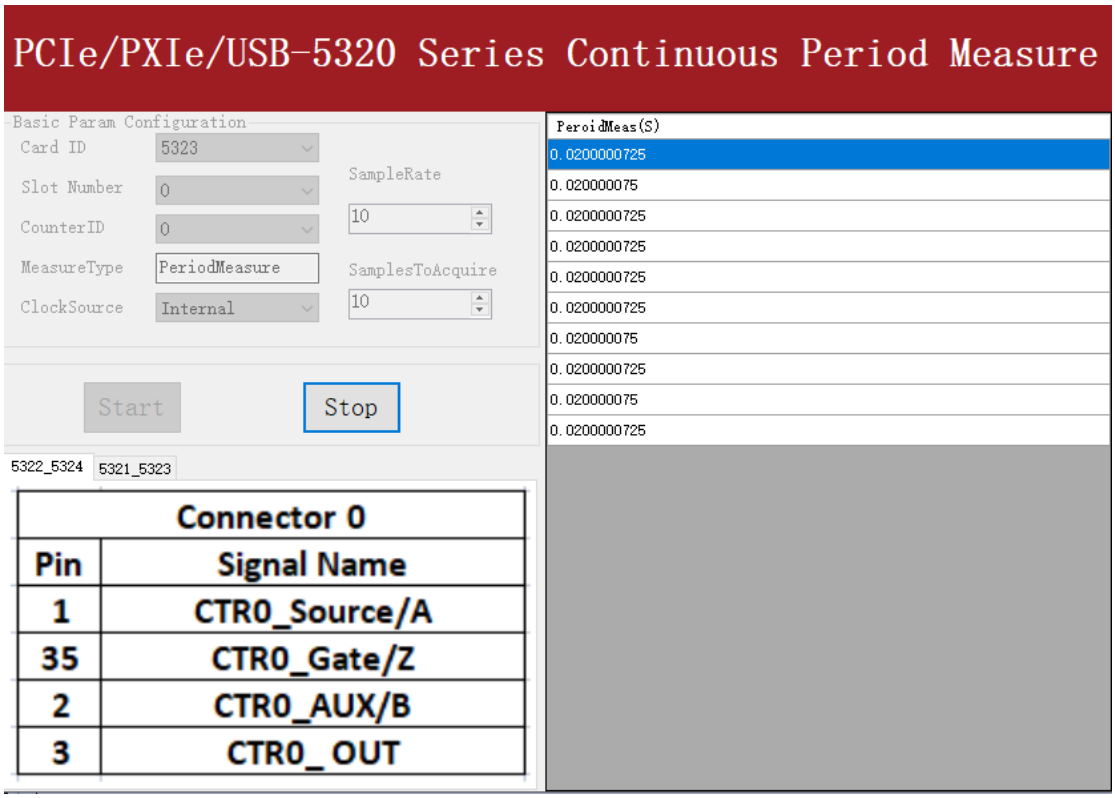

Figure 50 Period Measure For Continuous Mode

The table in the sample program is a connection diagram for your convenienc e.

The result of **Period Measure(S)** shows the correspond to the frequency set before.

7.7.5 Two-Edge Separation

The counter measures the speration (interval between each rising edges of tw o signals) between the rising edges of two signals. Default, the first signal m ust be connected to Counter Gate terminal and the second signal must be co nnect to Counter Aux terminal.

Set JY5320CITask.Type to CIType.TwoEdgeSeparation to use this function.

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 scecond 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 resg iter on each rising edge of the first signal.

As shown in Figure 51.

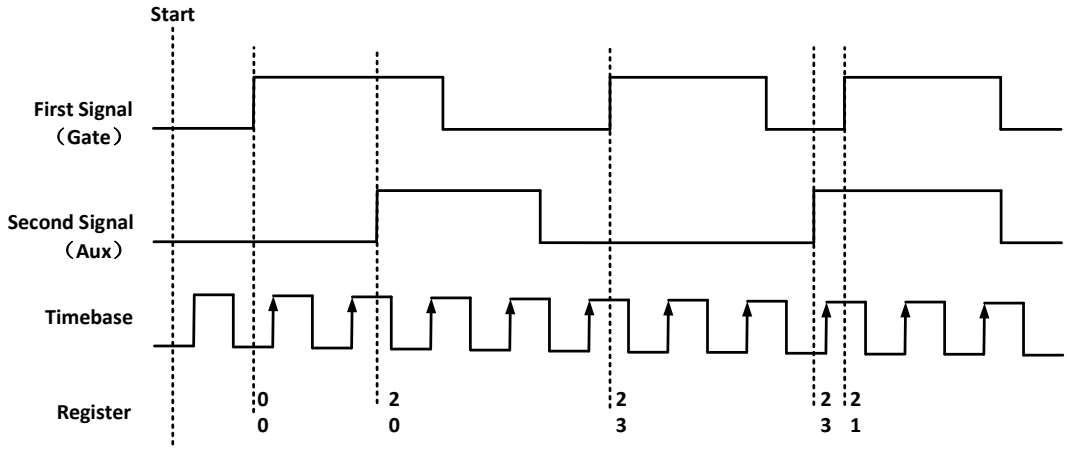

Figure 51 Two-Edge Separation in Single Mode

To configure the counter to work in this mode, set JY5320CITask.Mode to CI Mode.Single.

2) Finite/Continuous Mode with Explicit Sample Clock:

The counting values of rising edges of timebase between first signal and seco nd signal are stored into buffer on each rising edge of the sample clock, as shown in Figure 52.

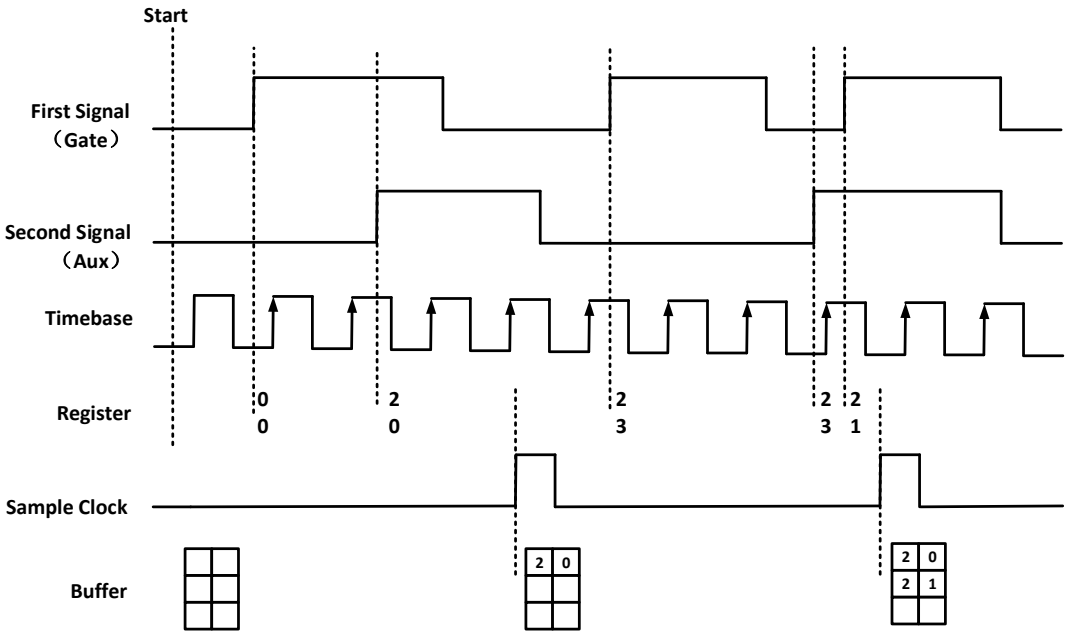

Figure 52 Two-Edge Seperation with Explicit Sample Clock

To configure the counter to work in this mode, set JY5320CITask.Mode to CI Mode.Finite or CIMode.Continuous, and set JY5320CITask.SampleClock.Source to CISampleClockSource.Internal or CISampleClockSource.External.

3) Finite/Continuous Mode with Implicit Sample Clock

In implicit mode, the signal active edge as the implicit sample clock edge. The counting 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 53.

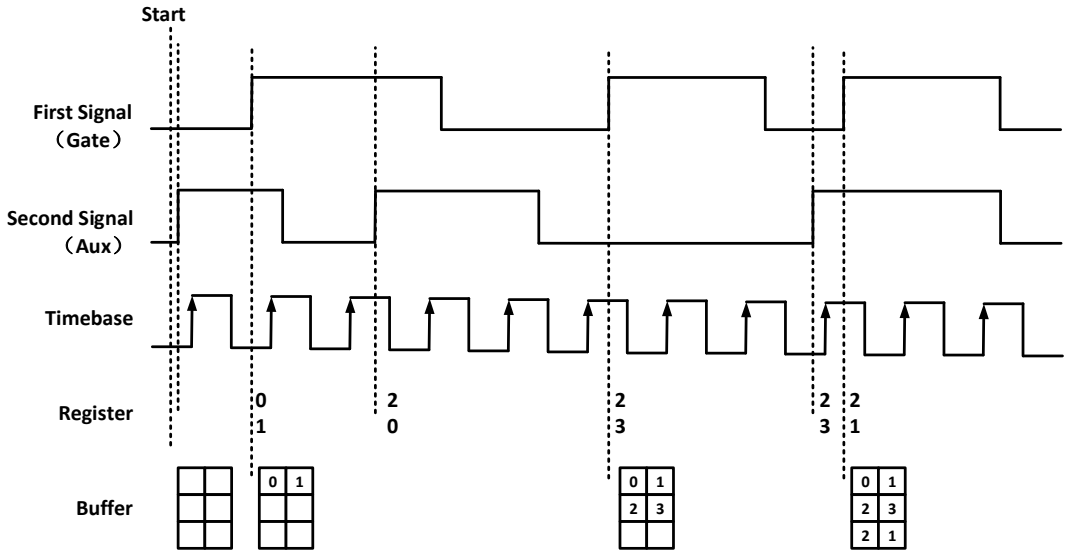

Figure 53 Two-Edge Seperation with Implicit Sample Clock

To configure the counter to work in this mode, set JY5320CITask.Mode to CI Mode.Finite or CIMode.Continuous, and set JY5320CITask.SampleClock.Source to CISampleClockSource.Implicit.

Timebase

JY-5321A/5322A/5323A/5324A provides four options for timebase source as follows:

- Internal 200MHz: - Same signal as the 200MHz base clock generated by PLL.
- Internal 5MHz – Generated by dividing down the 200MHz timebase.
- Internal 100kHz – Generated by dividing down the 200MHz timebase.
- External - Use a signal on a terminal as the timebase

By default, the counter uses the onboard 200MHz timebase to measure pulses. Use the property JY5320CITask.TwoEdgeSeparation.Timebase to configure the timebase.

Terminals

To change the terminal of signals instead of using its default value shown in chapter 7.3, using following properties:

- JY5320CITask.TwoEdgeSeparation.FirstInputTerminal – First signal-to-measure input terminal.
- JY5320CITask.TwoEdgeSeparation.SecondInputTerminal – First signal-to-measure input terminal.
- JY5320CITask.TwoEdgeSeparation.Timebase.External.Terminal – External timebase input terminal.

Learn by Examples 8.7.5

- Connect the signal source's two positive terminal to PCIe-5323A counter0's frequency measure source (CTR0_Gate/Z, Pin#35)and second signal input (CTR0_AUX/B,Pin#2) negative terminal to the ground (DGND, Pin#39)and (DGND, Pin#6) . (CTR0_Gate/Z, DGND) consists of a pulse measure counter input and they share the same ground.
- Set a sinewave signal (f=50Hz, Phase=0°) and a sinewave signal (f=50Hz, Phase=120°).

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:

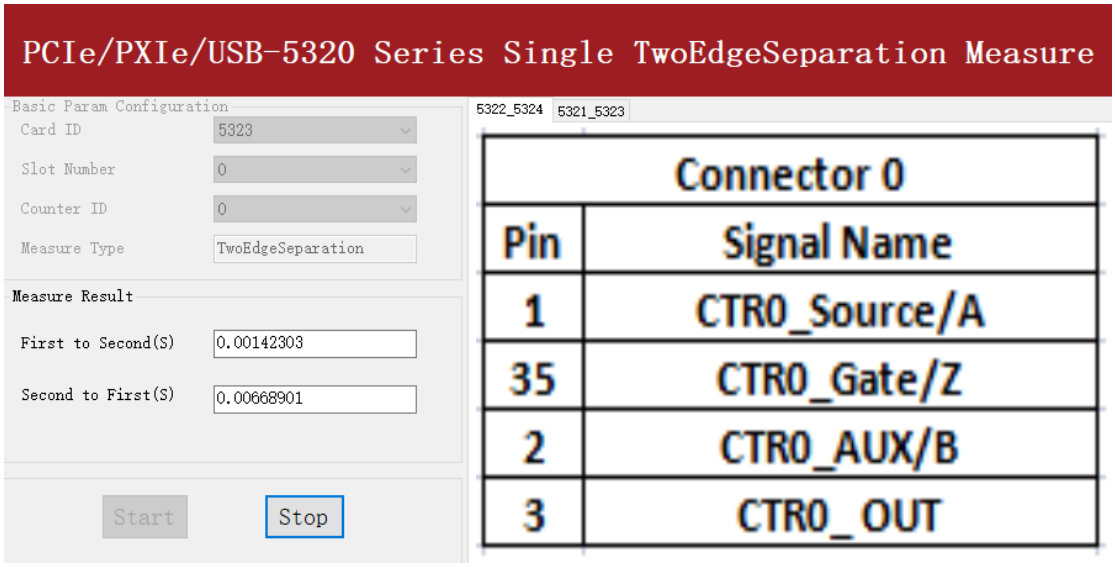

Figure 54 Two-EdgeSeparation Measure For Single Mode

- 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:

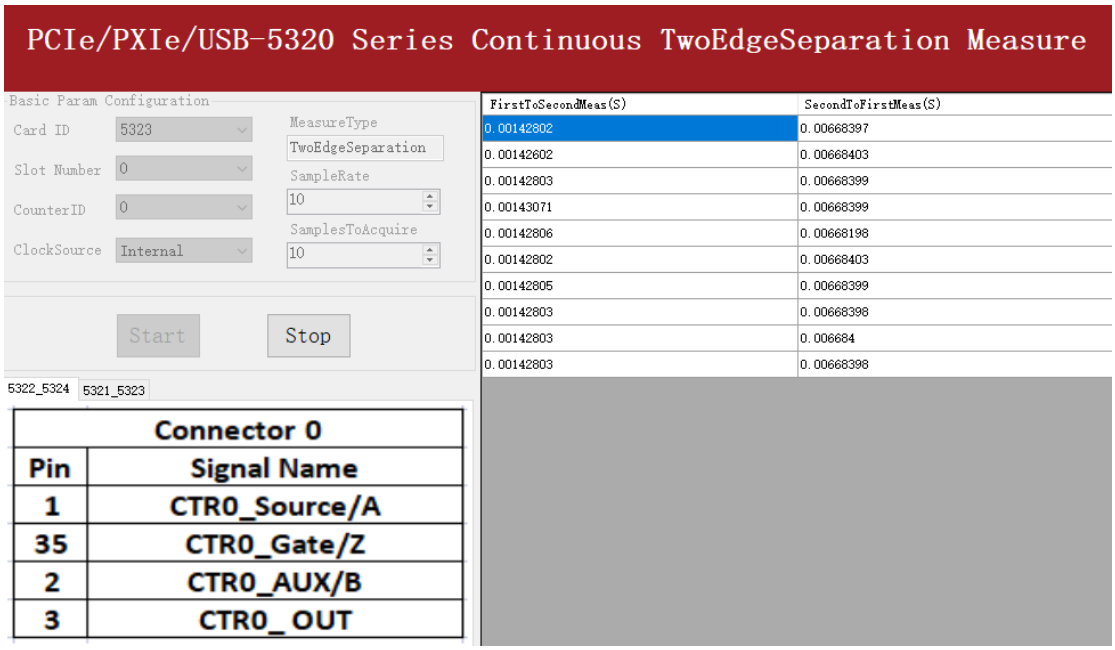

Figure 55 Two-EdgeSeparation Measure For Finite Mode

- 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.

7.7.6 Quadrature Encoder

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

Set JY5320CITask.Type to CIType. QuadEncoder to use this function, and use property JY5320CITask.QuadEncoder.EncodingType to change the type of encoding. Default, the A signal must be connected to Counter A terminal, the B signal mest be connected to Counter B terminal and the Z signal mest be connected to Counter Z terminal. For terminal Z, you can connect Z signal to it, or you can disable Z with property "ZReloadEnabled".

Encoding Type

1) x1 Encoding

When A signal leads B signal, the counter increases the count value on the rising edge of A signal; when B signal leads A signal, the counter decreases the count value on the falling edge of A signal as shown in Figure 56.

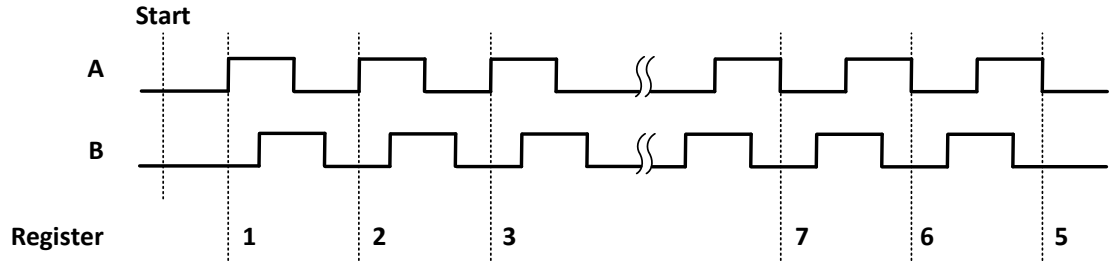

Figure 56 Quadrature Endcoder x1 Mode

2) x2 Encoding

When A signal leads B signal, the counter increases the count on the rising edge and the falling edge of A signal; when B signal leads A signal, the counter decreases the count value the rising edge and falling edge of A signal as shown in Figure 57.

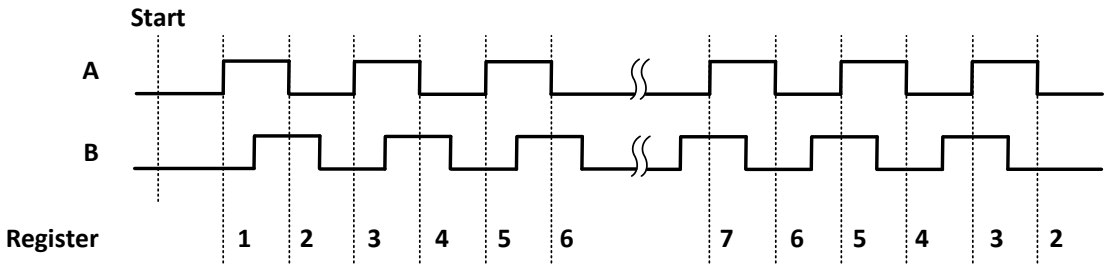

Figure 57 Quadrature Encoder x2 Mode

3) x4 Encoding

When A signal leads B signal, the counter increases the count value on the r ising and falling edges of A signal and B signal. When B signal leads A signal, the counter decreases the count value on the rising and falling edges of A signal and B signal. As shown in Figure 58.

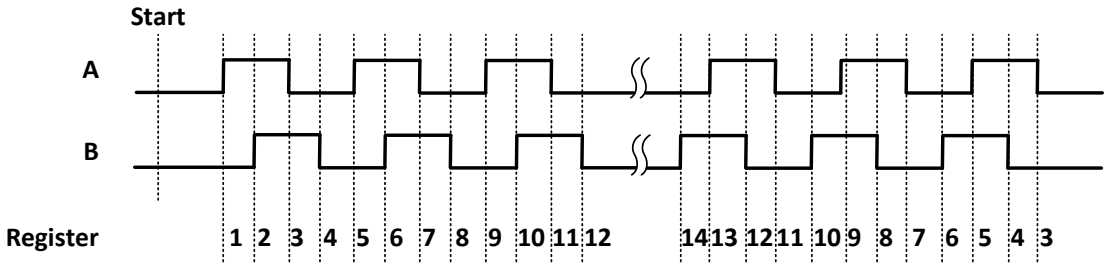

Figure 58 Quadrature Encoder x4 mode

Channel Z Behavior

The reload phase is when Z signal is high and A signal and B signal 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 A signa l, as shown in Figure 56.

To configure the counter to work in this mode, set JY5320CITask.Mode to CI Mode.Single.

2) Finite/Continuous Mode with Explicit Sample Clock

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

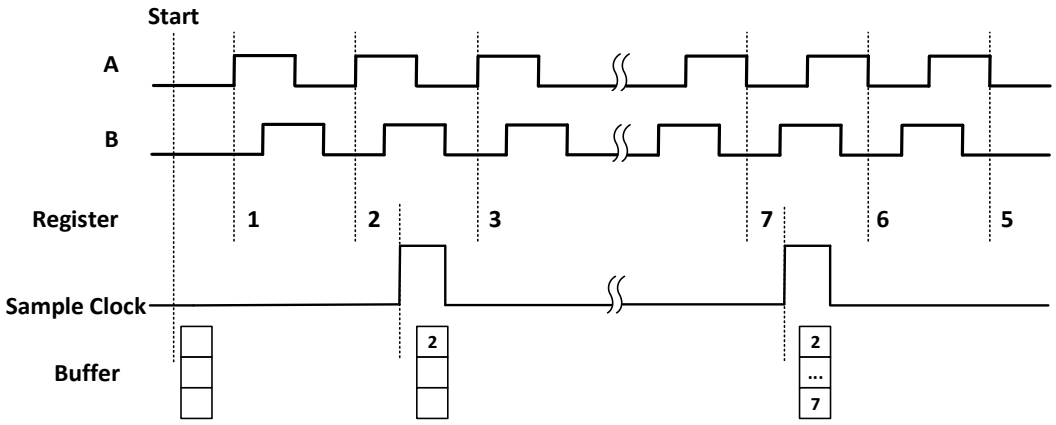

Figure 59 Quadrature Encoder x4 with Explicit Sample Clock

To configure the counter to work in this mode, set JY5320CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5320CITask.SampleClock.Source to CISampleClockSource.Internal or CISampleClockSource.External.

3) Finite/Continuous Mode with Implicit Sample Clock

In implicit mode, the signal active edge as the implicit sample clock edge. The count value is stored into the buffer on the transition of signal as shown in Figure 60.

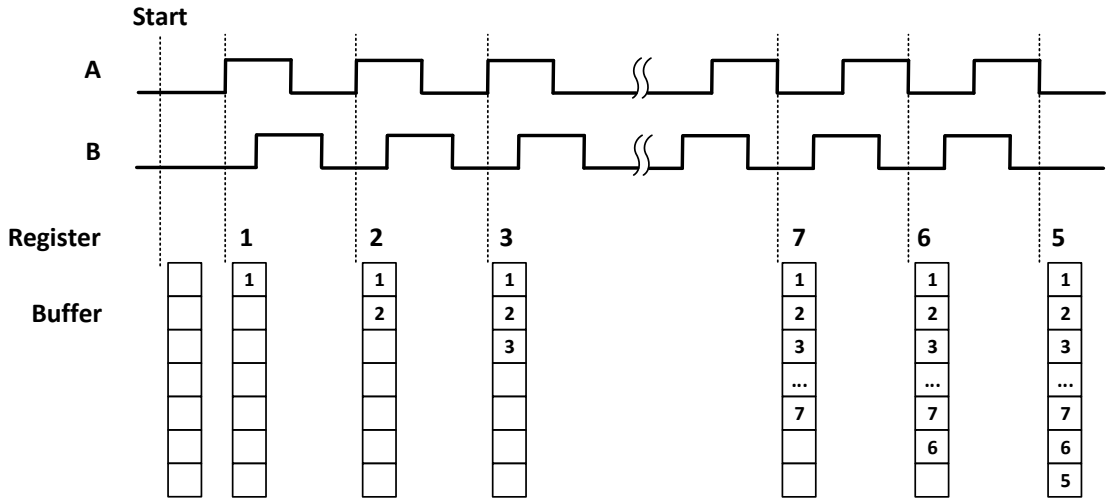

Figure 60 Quadrature Encoder x4 with Implicit Sample Clock

To configure the counter to work in this mode, set JY5320CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5320CITask.SampleClock.Source to CISampleClockSource.Implicit.

Terminals

To change the terminal of signals instead of using its default value as shown in chapter 7.3, use following properties:

- JY5320CITask.QuadEncoder.AInputTerminal – Signal A input terminal.
- JY5320CITask.QuadEncoder.ZInputTerminal – Signal Z input terminal.
- JY5320CITask.QuadEncoder.BInputTerminal – Signal B input terminal.

Learn by Examples 8.7.6

- Connect the signal source's two positive terminal to PCIe-5323A counter0' s frequency measure source (CTR0_Gate/Z, Pin#35)and second signal input (CTR0_AUX/B,Pin#2) negative terminal to the ground (DGND, Pin#39)and (DGND, Pin#6). (CTR0_Gate/Z, DGND) consists of a pulse measure counter input and they share the same ground.
- Set a sinewave signal (f=10Hz, Phase=90°) and a sinewave signal (f=10Hz, Phase=0°)

Single Mode

Open **Counter Input--> Winform CI Single QuadEncoder** and click Start. The r esult is shown：

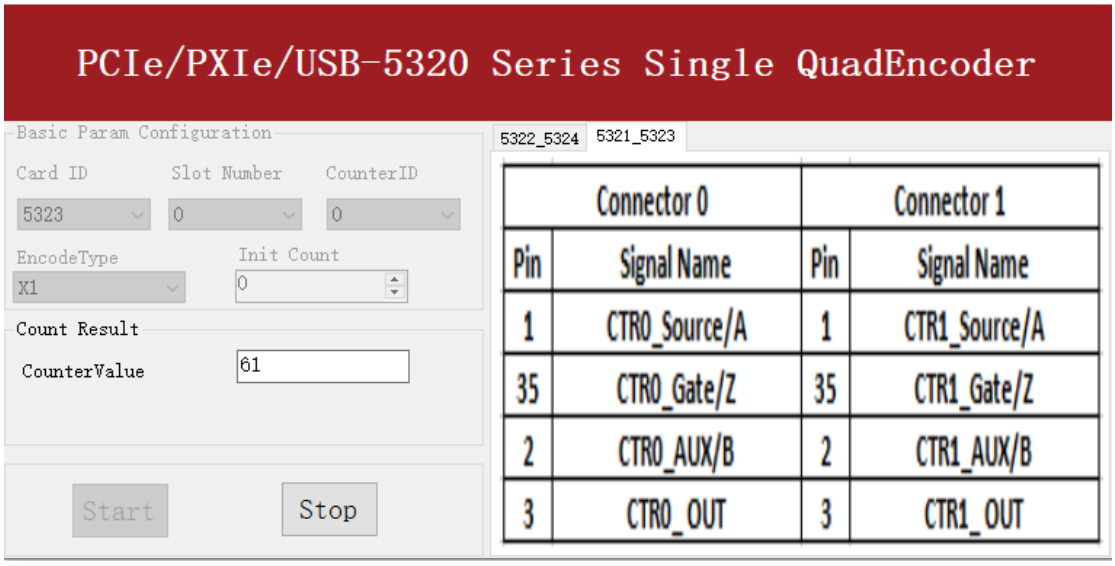

Figure 61 QuadEncoder For Single Mode

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

- 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 th e rising speed of **CounterValue** is twice and four times than x1Mode.

Continuous Mode

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

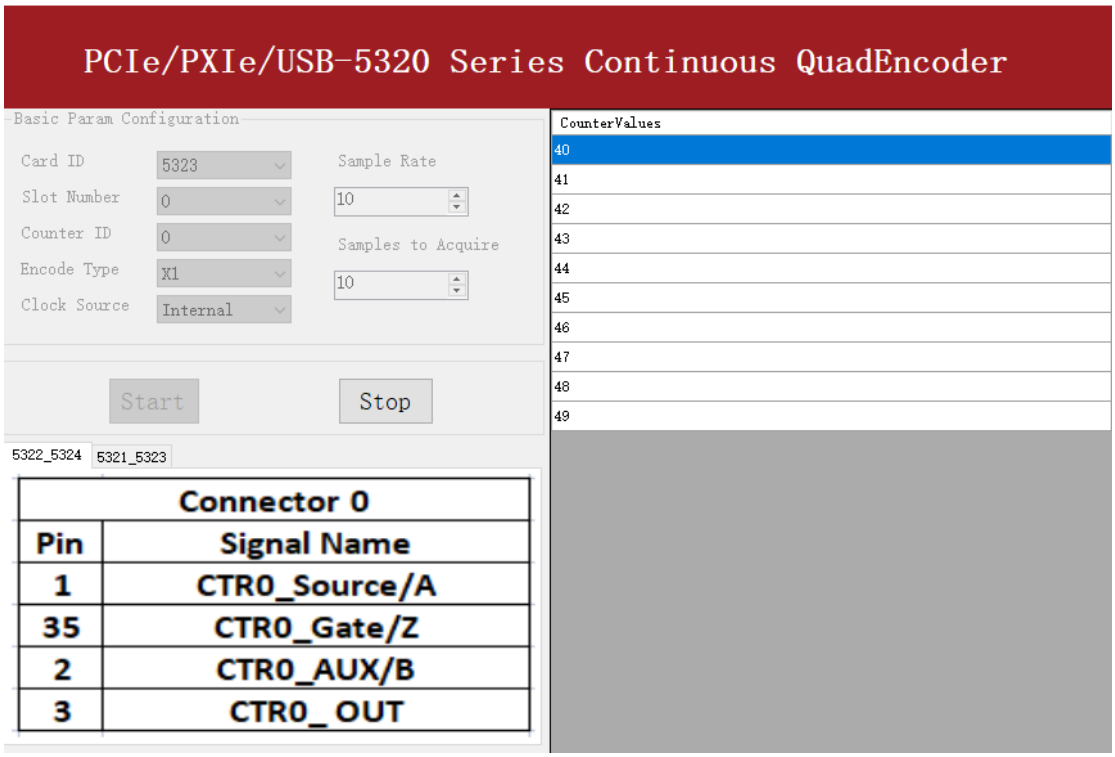

Figure 62 QuadEncoder For Continuous Mode

- The table in the sample program is a connection diagram for your conve nience.
- 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 th e rising speed of **CounterValue** is twice and four times than x1Mode.

7.7.7 Two-Pulse Encoder

The count value increases on the rising edge of A signal and decreases on th e rising edge of B signal. Default, the A signal must be connected to Counter A terminal, the B signal mest be connected to Counter B terminal.

Set JY5320CITask.Type to CIType.TwoPulseEncoder to use this function

Timing

1) Single Mode

The count value is written to the register on each rising edge of the A signa l, and B signel, as shown in Figure 63.

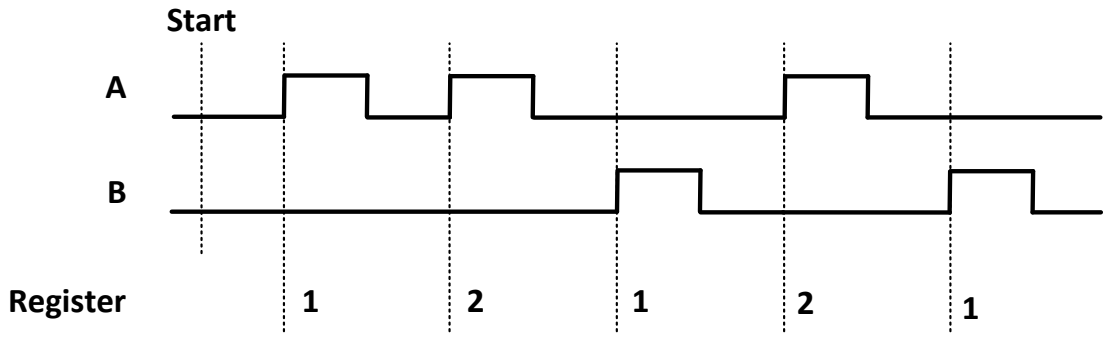

Figure 63 Two-Pulse Encoder in Single Mode

To configure the counter to work in this mode, set JY5320CITask.Mode to CI Mode.Single.

2) Finite/Continuous Mode with Explicit Sample Clock

The count value is stored into the buffer on each rising edge of the sample clock, as shown in Figure 64.

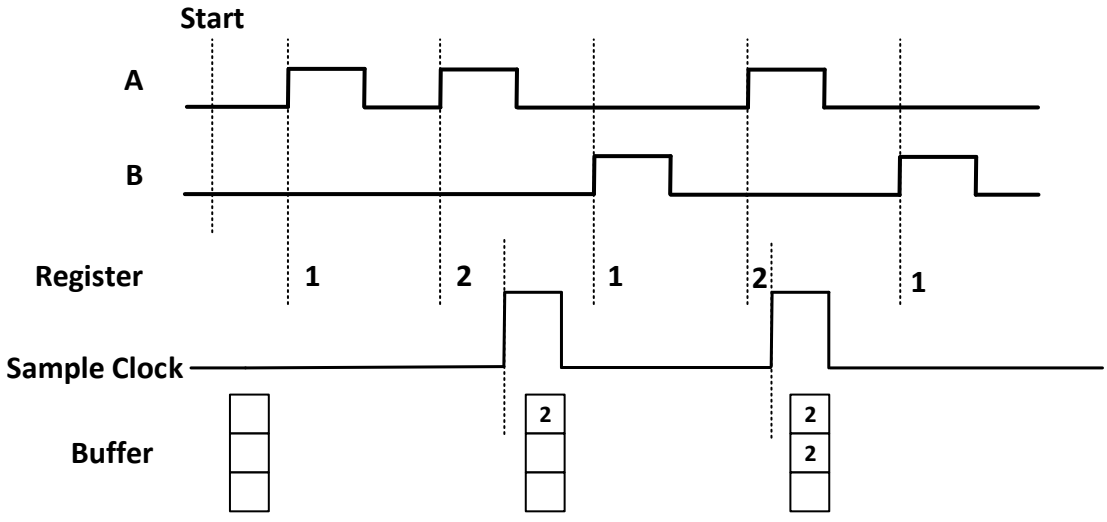

Figure 64 Two-Pulse Encoder with Explicit Sample Clock

To configure the counter to work in this mode, set JY5320CITask.Mode to CI Mode.Finite or CIMode.Continuous, and set JY5320CITask.SampleClock.Source to

CISampleClockSource.Internal or CISampleClockSource.External.

3) Finite/Continuous Mode with Implicit Sample Clock

In implicit mode, the signal active edge as the implicit sample clock edge. Th e count value is stored into the buffer on the rising edge of A signal and B signal, as shown in Figure 65.

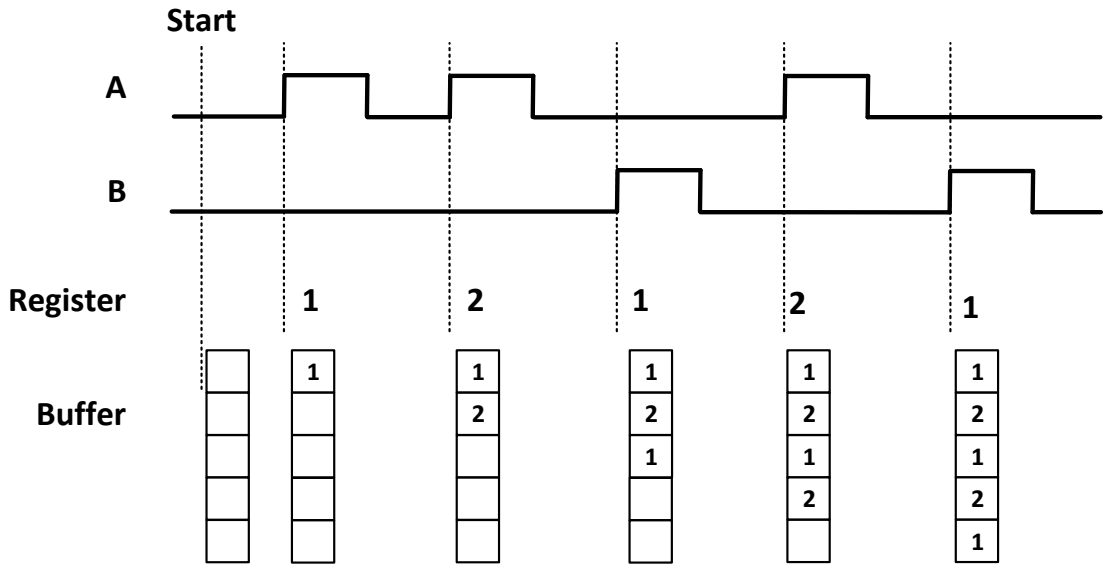

Figure 65 Two-Pulse Encoder with Implicit Sample Clock

To configure the counter to work in this mode, set JY5320CITask.Mode to CI Mode.Finite or CIMode.Continuous, and set JY5320CITask.SampleClock.Source to CISampleClockSource.Implicit.

Terminals

To change the terminal of signals instead of using its default value as shown in chapter 7.3, use following properties:

- JY5320CITask.TwoPulseEncoder.AInputTerminal – Signal A input terminal.
- JY5320CITask. TwoPulseEncoder.BInputTerminal – Signal B input terminal.

Learn by Examples 8.7.7

- Connect the signal source's two positive terminal to PCIe-5323A counter0' s frequency measure source (CTR0_Gate/Z, Pin#35)and second signal input (CTR0_AUX/B,Pin#2) negative terminal to the ground (DGND, Pin#39)and (DGND, Pin#6) . (CTR0_Gate/Z, DGND) consists of a pulse measure count er input and they share the same ground.
- 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.

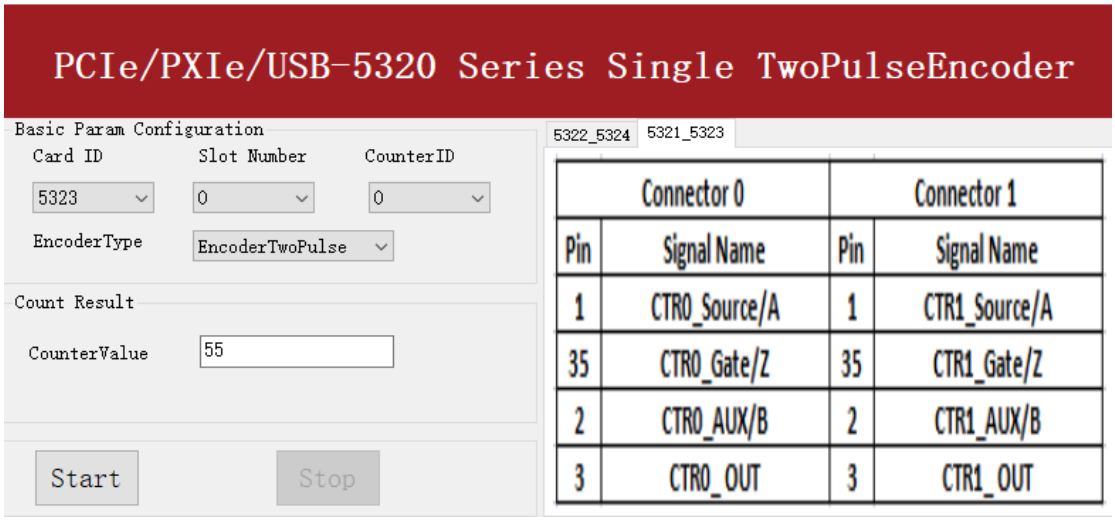

Figure 66 Two-PulseEncoder For Single Mode

- 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

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

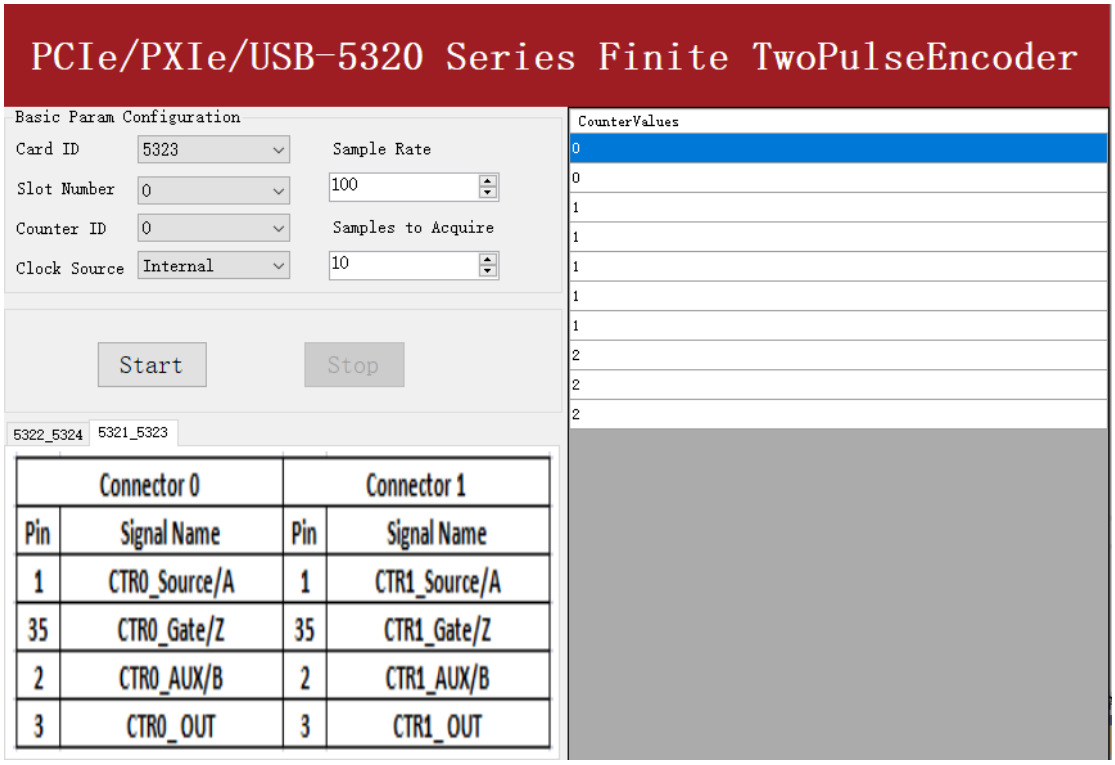

Figure 67 Two-PulseEncoder For Finite Mode

- 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

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

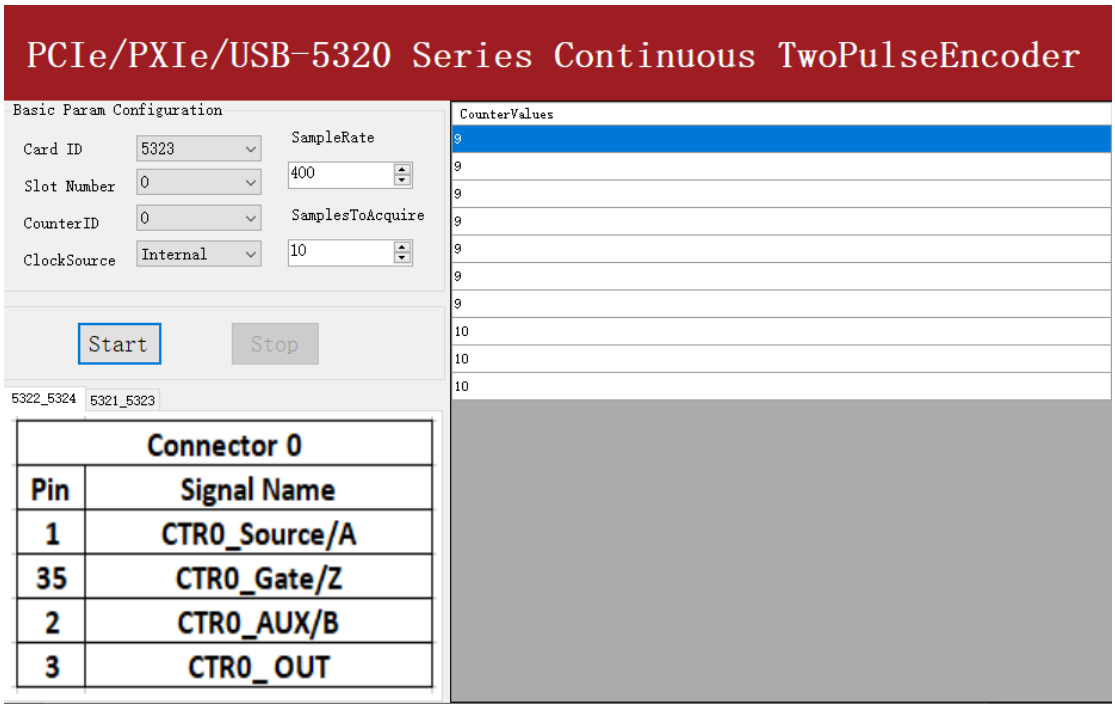

Figure 68 Two-PulseEncoder For Continuous Mode

- 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.

7.8 Counter Generation Operations

The JY-5321A/5322A/5323A/5324A can generate multiple forms of output signals according to different timing modes, including:

- Pulse generation with dynamic update
- Buffered pulse sequence generation

Timing

1) Single Mode

JY5320 can output a single pulse with a specified pulse configuration. The timing diagram of the pulse output is shown in Figure 69.

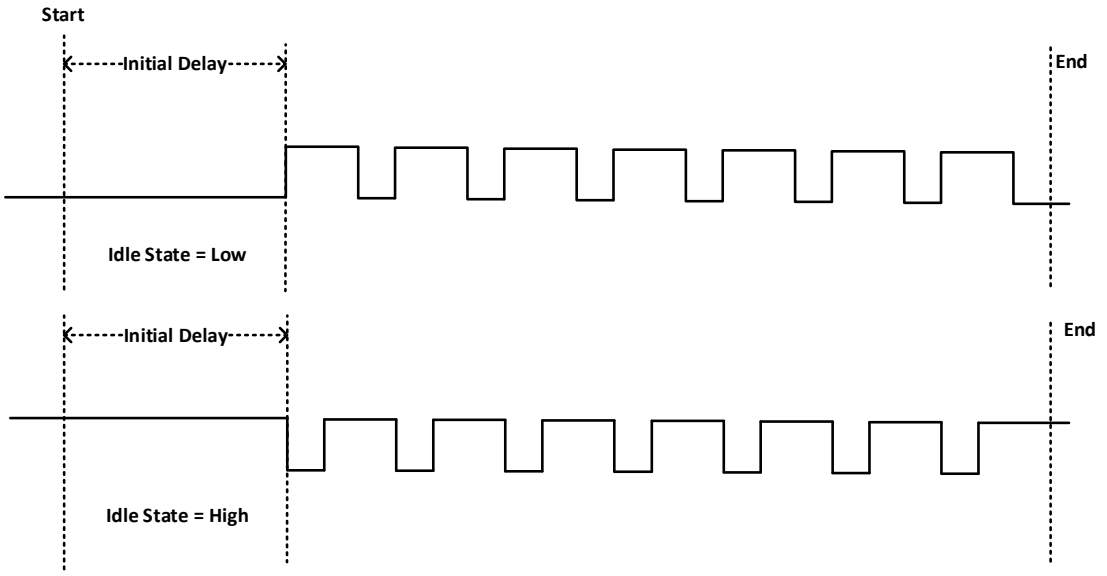

Figure 69 Pulse Output in Single Mode

Dynamic Update

If the number of pulses is set to -1, the pulses will be output continuously until requesting to stop. In this case, it is allowed to change the frequency a nd duty cycle on the fly. The timing diagram is shown as Figure 70.

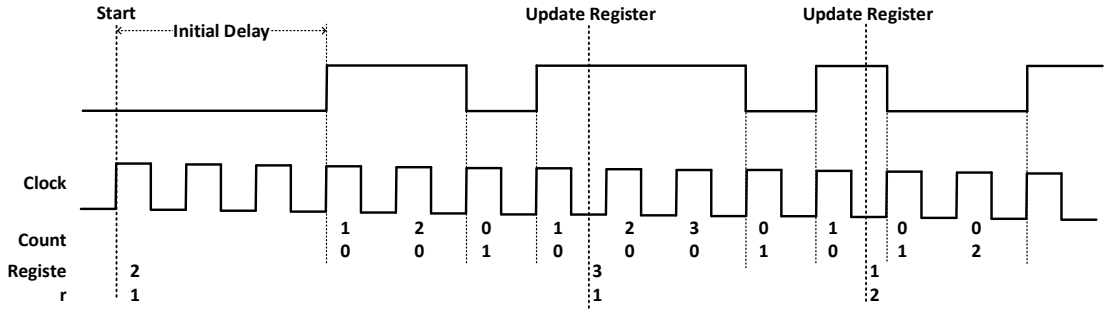

Figure 70 Pulse Output in Single Mode with Dynamic Update

Learn by Example 8.8.1

- To see the signal that JY-5323A Counter Output generates, it is recomme nded to connect JY-5323A Counter Output (CTR0_OUT, Pin#3) to JY5510 A I Ch0 input (AI0+, Pin#21). JY-5323A Counter GND (GND, Pin#7) to JY5510 AI Ch0 GND (Pin#55).
- Open **Counter Output-->Winform CO Single** and click Start and set the n umbers as follow:

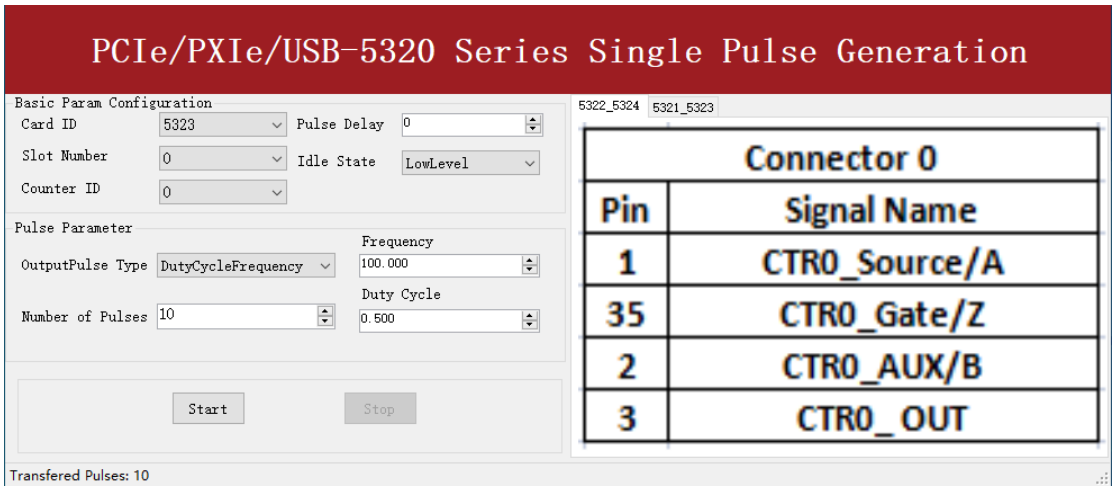

Figure 71 Single Pulse Generation

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

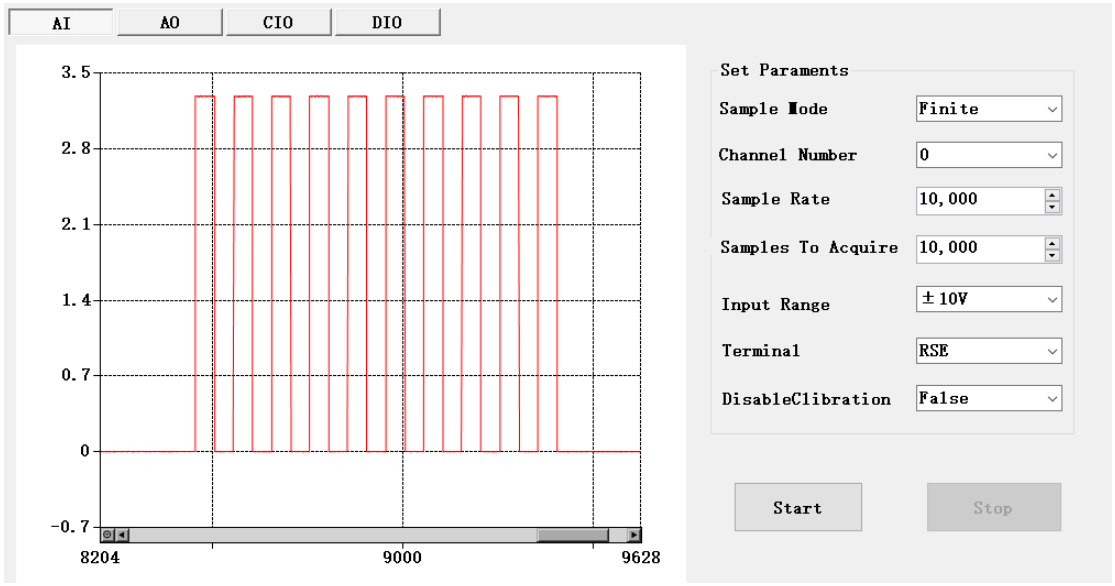

Figure 72 AI Acquisition Single Pulse

2) Finite/Continuous Mode

This mode allows the user to write all configurations of pulses to be output to the buffer in advance. After send the currently configured pulses, the coun

ter will automatically read the configuration of the next set of pulses to be s ent from the buffer and start output. The timing diagram is shown in Figure 73.

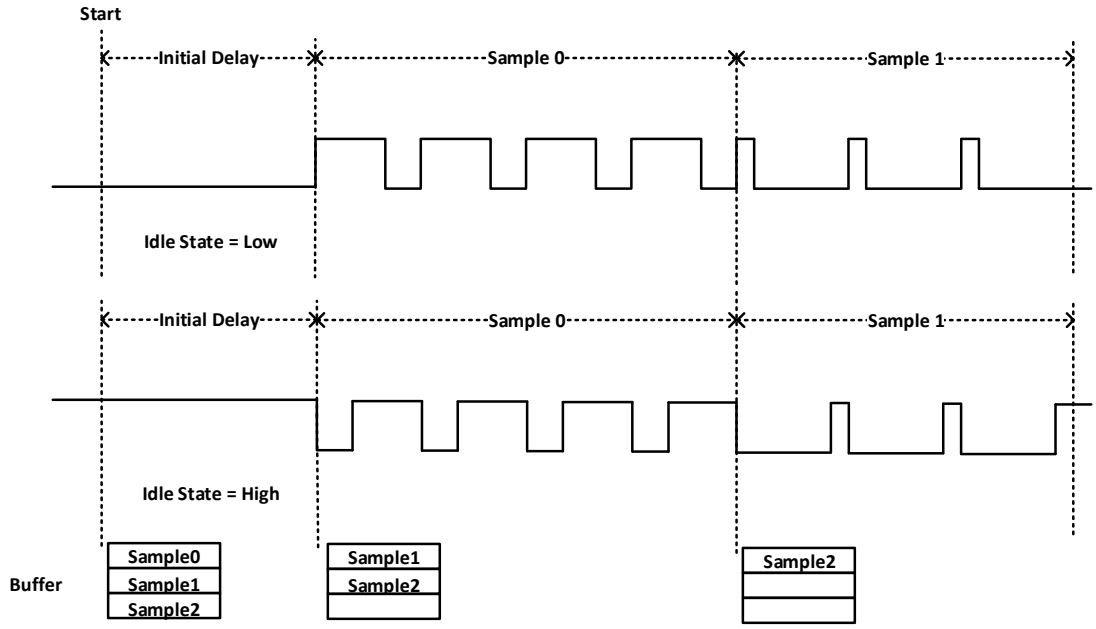

Figure 73 Buffered Pulse Sequence Generation

Each set of pulses can be configured in different ways:

- Frequency, Duty cycle and Number of pulses
- High time, Low time and Number of pulses
- High ticks, Low ticks and Number of pulses

Timebase

JY-5321A/5322A/5323A/5324A provides four options for timebase source as fol lows:

- Internal 200MHz: - Same signal as the 200MHz base clock generated by PLL.
- Internal 5MHz – Generated by dividing down the 200MHz timebase.
- Internal 100kHz – Generated by dividing down the 200MHz timebase.
- External - Use a signal on a terminal as the timebase

By default, the counter uses the onboard 200MHz timebase to generate pulse s. Use the property JY5320COTask.Timebase to configure the timebase.

Terminals

To change the terminal of signals instead of using its default value as shown in chapter 7.3, using following properties:

- JY5320COTask.OutputTerminal – Signal output terminal.

Learn by Example 8.8.2

- To see the signal that JY-5323A Counter Output generates, it is recommended to connect JY-5323A Counter Output (CTR0_OUT, Pin#3) to JY5510 AI Ch0 input (AI0+, Pin#21). JY-5323A Counter GND (GND, Pin#7) to JY5510 AI Ch0 GND (Pin#55).
- Open **Counter Output-->Winform CO Finite** and click Start and set the numbers as follow:

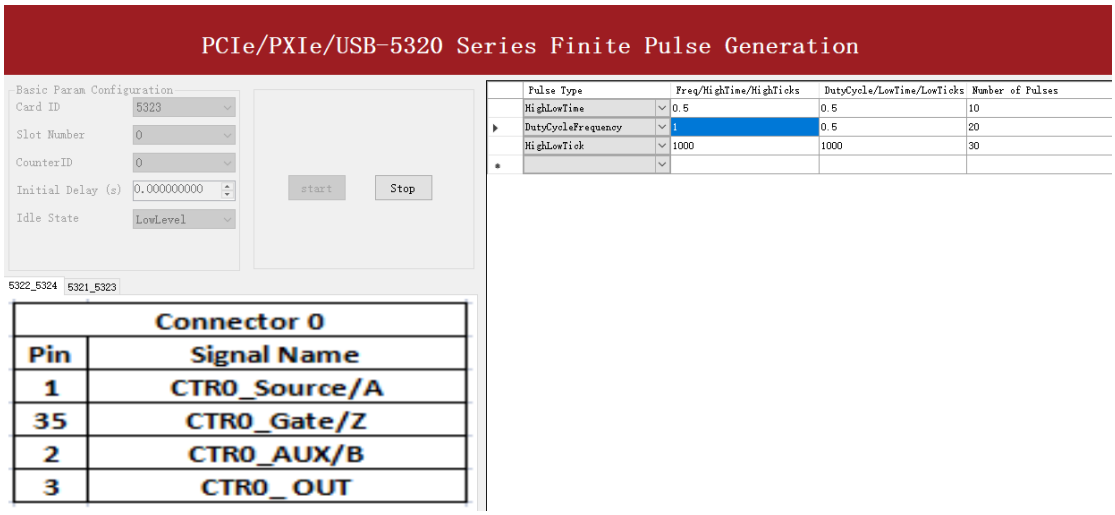

Figure 74 Finite Pulses Generation

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

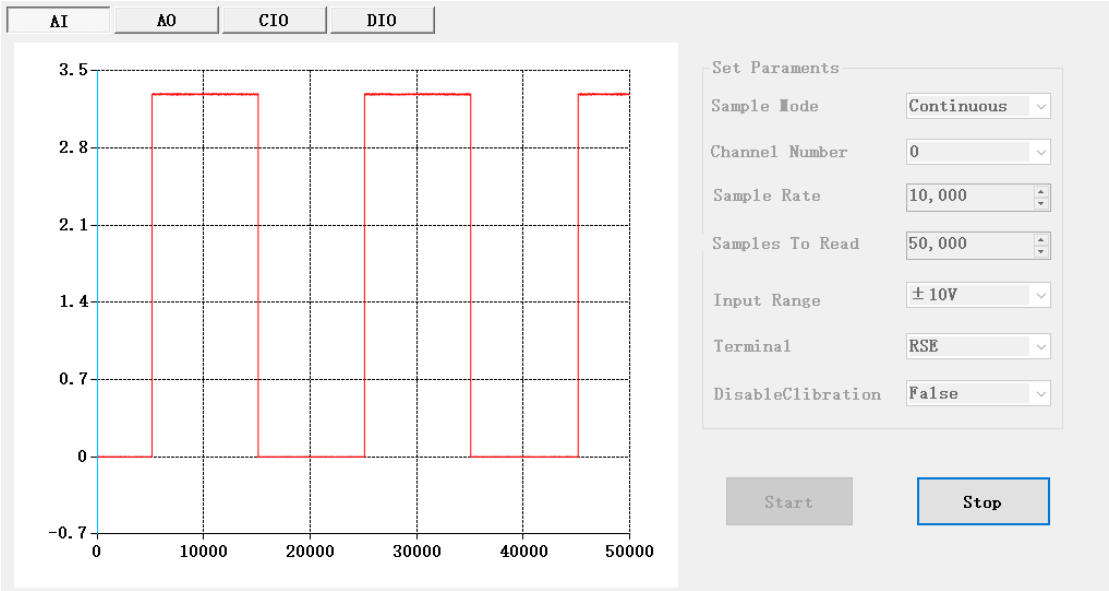

Figure 75 AI Acquisition Finite Pulse

Learn by Example 8.8.3

- To see the signal that JY-5323A Counter Output generates, it is recommended to connect JY-5323A Counter Output (CTR0_OUT, Pin#3) to JY5510 AI Ch0 input (AI0+, Pin#21). JY-5323A Counter GND (GND, Pin#7) to JY5510 AI Ch0 GND (Pin#55).
- Open **Counter Output-->Winform CO Continuous** and click Start and set the numbers as follow:

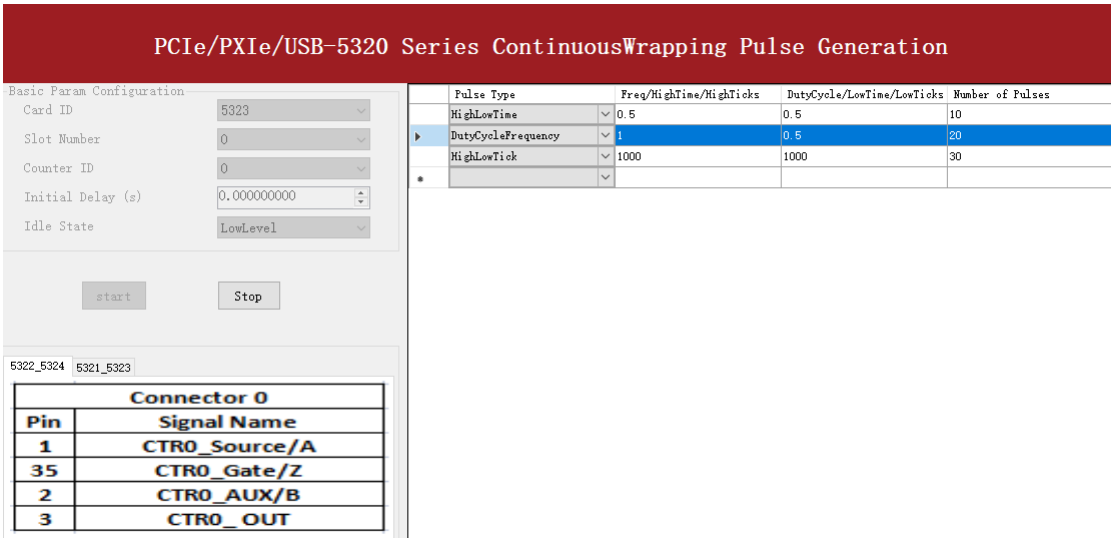

Figure 76 Continuous Pulse Generation

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

nience.

- The frequency and duty cycle of the pulse are set by **Frequency** and **Duty Cycle**.

- Please refer Learn by **Example 8.2** to configure an analog input to receive the signal from Counter Output.
- Click Start to generate a single pulse as shown.

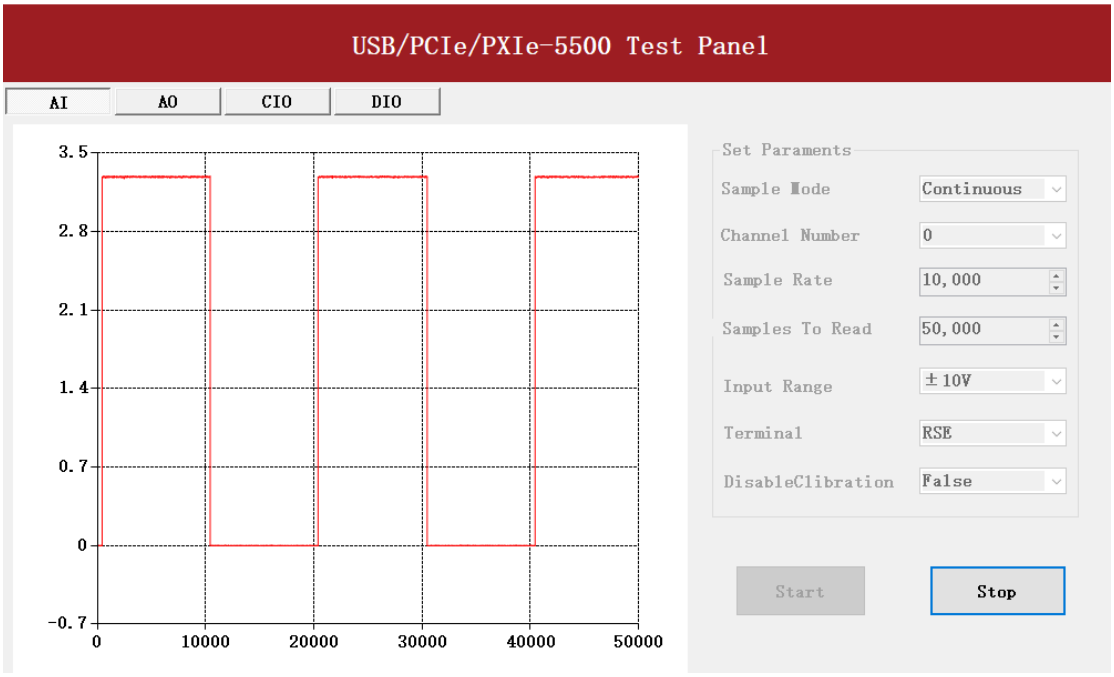

Figure 77 AI Acquisition Continuous Pulse

7.9 Multi-Devices Synchronization

The synchronization between PCIe modules is 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.

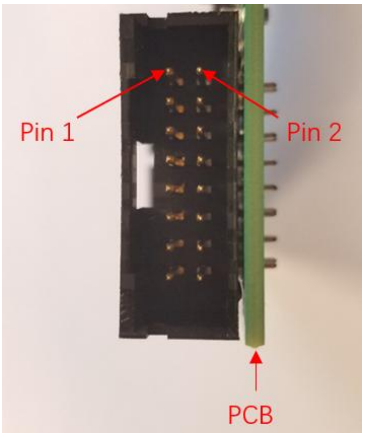

Figure 78 SSI Connector in PCIe-5320

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 20 SSI Connector Pin Assignment for PCIe-5320

7.10 DIP Switch in PCIe-5320

PCIe-5320 series board has 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.

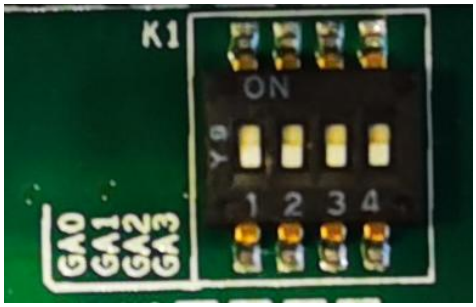

Figure 79 DIP Switch in PCIe-5310

	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 21 Relationship between switch position and slot number

8 Calibration

JYTEK 5320 boards are precalibrated before the shipment. We recommend you recalibrate JY-5321A/5322A/5323A/5324A boards periodically to ensure the measurement accuracy. A commonly accepted practice is one year. If you need to recalibrate your board, please contact JYTEK.

9 Appendix

9.1 Typical Measurement Error

Typical measurement error is a term used to describe the variation or uncertainty in a measurement that is repeated under the same conditions. It can be caused by random errors (chance differences between observed and true values) or systematic errors (consistent biases in measurement).

Typical measurement error can be expressed as a standard deviation (the typical error of measurement) or as a percentage of the mean (the coefficient of variation) .

9.2 System Noise

System noise refers to any unwanted and random fluctuations or disturbances in a physical or electronic system that can interfere with its normal operation. System noise can arise from various sources such as electrical interference, thermal noise, environmental factors, and inherent limitations of the system's components.

In electronic systems, system noise can affect the accuracy and reliability of signal processing and communication. For example, in audio systems, system noise can lead to hissing or humming sounds, and in wireless communication systems, it can cause interference and reduce the quality of the signal.

Reducing system noise is an important consideration in the design and operation of many types of systems, and engineers use various techniques to mitigate its effects, including shielding, filtering, and signal processing algorithms.

9.3 Temperature Drift

Temperature drift refers to the phenomenon where the performance or behavior of a physical or electronic system changes as the temperature changes. Temperature drift can affect various parameters such as frequency, voltage, resistance, and sensitivity, and it can cause errors or inaccuracies in the system's operation.

In electronic systems, temperature drift can arise due to the temperature dependence of the properties of the system's components, such as resistors, capacitors, and transistors. For example, the resistance of a resistor increases with temperature, and this can affect the accuracy of voltage measurements in a circuit. Similarly, the frequency of an oscillator can drift due to the temperature dependence of its resonant circuit components.

Temperature drift is an important consideration in the design and operation of many types of systems, particularly those that require high accuracy and stability over a wide range of temperatures. Engineers use various techniques to compensate for temperature drift, including using temperature sensors to monitor and control the temperature, selecting components with low temperature coefficients, and implementing temperature compensation algorithms in software or firmware.

10 About JYTEK

10.1 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 has evolved from re-branding and reselling PXI(e) and DAQ products to a fully-fledged product company. The company offers complete lines of PXI, DAQ, USB products. More importantly, JYTEK has been promoting open-sourced based ecosystem and offers complete software products. Presently, JYTEK is focused on the Chinese market. Our Shanghai headquarters and production service center have regular stocks to ensure timely supply; we also have R&D centers in Xi'an and Chongqing. We also have highly trained direct technical sales representatives in Shanghai, Beijing, Tianjin, Xi'an, Chengdu, Nanjing, Wuhan, Guangdong, Haerbin, and Changchun. We also have many patners who provide system level support in various cities.

10.2 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.

10.3 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.

11 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 JY-5320 family of temperature sensor data acquisition cards. 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