1. PCIe/PXIe-5113/5113s Specifications

Multi-functional Data Acquisition Boards

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

Overview

JYTEK JY-5113 board includes the PCIe-5113/5113s and PXIe-5113/5113s models, which belong to the JY-5110 series. The PCIe-5113 is a board based on the PCI Express interface，providing up to 64 channels of analog inputs of up to 1M samples per second, 4 channels of analog outputs, 16 channels of digital IO or 4 32-bit counters/timers; the PXIe-5113 is a board based on the PXI Express interface，providing up to 64 channels of analog inputs of up to 1M samples per second, 4 channels of analog outputs, 16 lines of Digital IO or 2 32-bit counters/timers.

1.1. Main Features

- High accuracy: 340 ppm
- Up to 64 single-ended/32 differential analog input channels;
- Sampling rates: 1MS/s; 8-ch 1MS/s/ch (5113s)
- 16 bits ADC
- 64M samples of analog input FIFO buffer
- 4 simultaneous 16-bit analog output channels
- 32M sample FIFO buffer for analog output
- 16 lines of Digital IO
- DIO supports hardware timing up to 10MHz
- 2 channels 32-bit timer/counter
- DMA for AI, AO and DIO
- Analog/Digital/Software Trigger

1.2. Analog Input

Analog Input	PCIe/PXIe-5113	PCIe/PXIe-5113s
Number of channels	64 SE/ 32 DIFF	64 SE/ 32 DIFF
ADC resolution (Bits)	16	16
Single channel maximum sample rate	1 MS/s	1 MS/s or 8-ch 1MS/s/ch
Multichannel maximum sample rate with same range(aggregate)	1 MS/s/n (n=1,2,3…64)	1 MS/s/n (n=1,2,3…8)
Clock	100 MHz	100 MHz
Input range(V)	±10/±5/±2.5	±10/±5/±2.5
Input mode	RSE / Differential	RSE / Differential
Input impedance	>1MΩ||330pF	>1MΩ||330pF
Input coupling	DC	DC
Overvoltage protection	±25 V	±25 V
CMRR	85 dB	85 dB
Crosstalk	-74 dB	-74 dB
DNL	No Missing Code	No Missing Code
INL	70 ppm of Range Typical	70 ppm of Range Typical
Input FIFO	128 M Samples	128 M Samples
Trigger type	Digital, Analog, Software	Digital, Analog, Software
Trigger mode	StartTrigger, ReferenceTrigger, ReTrigger	StartTrigger, ReferenceTrigger, ReTrigger
Analog trigger voltage range	±10 V Software Programmable	±10 V Software Programmable
Overvoltage Protection	Continuous : 20 mA, ±25 V	Continuous : 20 mA, ±25 V
	Instantaneous : 40 mA, ±25 V	Instantaneous : 40 mA, ±25 V

Table 1 Analog Input Specifications

1.3. Analog Output

Analog Output		PCIe/PXIe-5113	PCIe/PXIe-5113s
Number of channels		4	
DAC resolution		16 bits	
Maximum update rate (simultaneous)	1 channel	2 MS/s	
	2 channels (1 channel per bank)*	2 MS/s	
	4 channels	1 MS/s	
Clock		100 MHz	
Clock accuracy		Jitter <20 ps	
Output range(V)		±10	
Output mode		RSE	
Output impedance		0.2 Ω	
Output coupling		DC	
Output current drive		±10 mA	
Output FIFO		32 M Samples	
Trigger source		Digital, Software	
Trigger type		Start Trigger	
* Each bank consists of 2 AO channels. Any channels being used within a single bank will update simultaneously.			

Table 2 Analog Output Specifications

1.4. Counter Input/Output

CIO	PCIe/PXIe-5113	PCIe/PXIe-5113s
Number of channels	2	
Resolution	32 bits	
Counter Input	edge count, period measurement, frequency measurement, pulse width measurement, two-edge interval measurement, quadrature encoder, etc.	
Counter Output	Single, finite and continuous pulse	
Clock	100 MHz	
FIFO	4 M Samples	
Input	Gate，Source，Aux	
Output	OUT	

Table 3 Counter Input Operations Specifications

1.5. PFI Specifications

PFI	PCIe/PXIe-5113	PCIe/PXIe-5113s
Number of channels	16	
External digital trigger interface	Trigger voltage : 3.3 V TTL	
	Trigger edge: Rising/Falling	
Initial state	Input	

Table 4 PFI Specifications

1.6. Digital IO Specifications

DIO	PCIe/PXIe-5113	PCIe/PXIe-5113s
Number of channels	16 Lines	
Ground reference	D GND	
Directional control	Independent control of each port	
Max update rate	10 MHz	
Max	10 MHz	
DI FIFO	32M Samples	
DO FIFO	32M Samples	
Initial state	Input	
Digital Input	Logic Low: VIL Min : 0 / Max : 0.8 V	
	Logic High: VIH Min : 2 V / Max : 5.3 V	
Digital Output	Logic Low : 0 V, IOL Max: 12 mA	
	Logic High : 2.6 V～3.3 V, IOH: -12 mA～0 mA	

Table 5 Digital IO Specifications

1.7. Basic DC AI Accuracy

JY-5113/5113s Basic Accuracy = ±(% Reading+% Range)

Nominal Range (V)	24 Hour Tcal ±1C°			90 Day Tcal ±5C°			Temperature Coefficients(/°C)			24 Hr Full Scale Accuracy	90 Day Full Scale Accuracy	Full Scale Accuracy(%)
2.5	0.007	+	0.033	0.018	+	0.033	0.0004	+	0.0002	1000 uV	1300 uV	0.052
5	0.007	+	0.026	0.018	+	0.027	0.0004	+	0.0002	1650 uV	2300 uV	0.046
10	0.007	+	0.018	0.013	+	0.021	0.0004	+	0.0002	2500 uV	3400 uV	0.034

Accuracy valid to 95% of full range
Valid for one channel only. 95% of Confidence Interval
Max sampling rates for 5113: 1 MS/s
The 90-day accuracy is 2.5 times the 24-hour accuracy.

Table 6 Basic DC AI Accuracy

AI Bandwidth

Analog Input Bandwidth		
Nominal Range Full Scale (V)	**Bandwidth select**	**-3dB Bandwidth (kHz)**
All range	15KHz	15
All range	39KHz	31
All range	80KHz	73

Table 7 AI Bandwidth Specifications

1.8. Basic AO Accuracy

JY-5113/5113s Basic AO Accuracy = ±(% of Output+% of Range)

Nominal Range (V)	24 Hour Tcal ±1C°			90 Days Tcal ± 5°			Temperature Coefficients(/°C)			24 Hr Full Scale Accuracy	90 Day Full Scale Accuracy	Full Scale Accuracy(%)
10	0.009	+	0.016	0.0260	+	0.0270	0.0018	+	0.0013	2500 uV	5300 uV	0.053

Accuracy valid to 99% of full range.
Valid for all update rates.
Add accuracy adjustment if temperature is ouside calibration temperature range.
The 90-day accuracy is 2.5 times the 24-hour accuracy.

Table 8 Basic AO Accuracy

1.9. System Noise

Nominal Range (V)	Bandwidth select (KHz)	SystemNoise (μVrms)
2.5	15	56
5	15	91
10	15	170
2.5	39	73
5	39	110
10	39	190
2.5	80	230
5	80	300
10	80	410

Table 9 System Noise Specifications Physical and Environment

1.10. Physical and Environment

Operating Environment

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

Storage Environment

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

Table 10 Physical and Environment

1.11. Power Consumption (typical)

Voltage	Current	Power	Total Power
12V	0.75A	9W	9.2W
3.3V	0.06A	0.2W	

Table 11 5113 Power Consumption

1.12. Front Panel and Pin Definition

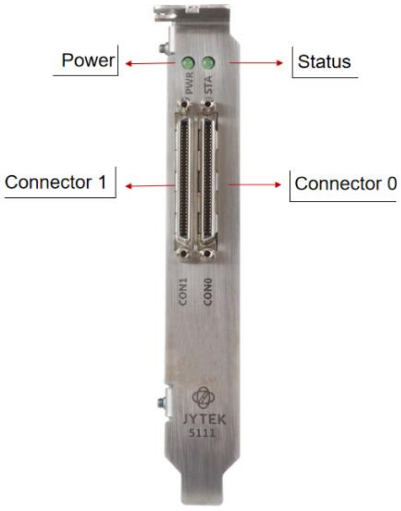


Figure 1 PCIe/PXIe 5113 Front Panel

Connector 1			
Pin	Identification	Pin	Identification
1	PFI 8 / DIO_8	35	D_GND
2	PFI 9 / DIO_9	36	D_GND
3	PFI 10 / DIO_10	37	D_GND
4	PFI 11 / DIO_11	38	D_GND
5	PFI 12 / DIO_12	39	D_GND
6	PFI 13 / DIO_13	40	D_GND
7	PFI 14 / DIO_14	41	D_GND
8	PFI 15 / DIO_15	42	D_GND
9	AO2	43	AO_GND
10	AO3	44	AO_GND
11	AI 32 (AI 16+)	45	AI 40 (AI 16-)
12	AI_GND	46	AI_GND
13	AI 33 (AI 17+)	47	AI 41 (AI 17-)
14	AI 34 (AI 18+)	48	AI 42 (AI 18-)
15	AI_GND	49	AI_GND
16	AI 35 (AI 19+)	50	AI 43 (AI 19-)
17	AI 36 (AI 20+)	51	AI 44 (AI 20-)
18	AI_GND	52	AI_GND
19	AI 37 (AI 21+)	53	AI 45 (AI 21-)
20	AI 38 (AI 22+)	54	AI 46 (AI 22-)
21	AI_GND	55	AI_GND
22	AI 39 (AI 23+)	56	AI 47 (AI 23-)
23	AI 48 (AI 24+)	57	AI 56 (AI 24-)
24	AI_GND	58	AI_GND
25	AI 49 (AI 25+)	59	AI 57 (AI 25-)
26	AI 50 (AI 26+)	60	AI 58 (AI 26-)
27	AI_GND	61	AI_GND
28	AI 51 (AI 27+)	62	AI 59 (AI 27-)
29	AI 52 (AI 28+)	63	AI 60 (AI 28-)
30	AI_GND	64	AI_GND
31	AI 53 (AI 29+)	65	AI 61 (AI 29-)
32	AI 54 (AI 30+)	66	AI 62 (AI 30-)
33	AI_GND	67	AI_GND
34	AI 55 (AI 31+)	68	AI 63 (AI 31-)

Connector 0			
Pin	Identification	Pin	Identification
1	PFI 0 / DIO_0	35	D_GND
2	PFI 1 / DIO_1	36	D_GND
3	PFI 2 / DIO_2	37	D_GND
4	PFI 3 / DIO_3	38	D_GND
5	PFI 4 / DIO_4	39	D_GND
6	PFI 5 / DIO_5	40	D_GND
7	PFI 6 / DIO_6	41	D_GND
8	PFI 7 / DIO_7	42	D_GND
9	AO0	43	AO_GND
10	AO1	44	AO_GND
11	AI 0 (AI 0+)	45	AI 8 (AI 0-)
12	AI_GND	46	AI_GND
13	AI 1 (AI 1+)	47	AI 9 (AI 1-)
14	AI 2 (AI 2+)	48	AI 10 (AI 2-)
15	AI_GND	49	AI_GND
16	AI 3 (AI 3+)	50	AI 11 (AI 3-)
17	AI 4 (AI 4+)	51	AI 12 (AI 4-)
18	AI_GND	52	AI_GND
19	AI 5 (AI 5+)	53	AI 13 (AI 5-)
20	AI 6 (AI 6+)	54	AI 14 (AI 6-)
21	AI_GND	55	AI_GND
22	AI 7 (AI 7+)	56	AI 15 (AI 7-)
23	AI 16 (AI 8+)	57	AI 24 (AI 8-)
24	AI_GND	58	AI_GND
25	AI 17 (AI 9+)	59	AI 25 (AI 9-)
26	AI 18 (AI 10+)	60	AI 26 (AI 10-)
27	AI_GND	61	AI_GND
28	AI 19 (AI 11+)	62	AI 27 (AI 11-)
29	AI 20 (AI 12+)	63	AI 28 (AI 12-)
30	AI_GND	64	AI_GND
31	AI 21 (AI 13+)	65	AI 29 (AI 13-)
32	AI 22 (AI 14+)	66	AI 30 (AI 14-)
33	AI_GND	67	AI_GND
34	AI 23 (AI 15+)	68	AI 31 (AI 15-)

Note:

- At JY-5113s AI channel synchronization, each bank can only select one;
  In RSE mode, the following is the distribution of the banks:

  Bank1 AI_0 - AI_7
  Bank2 AI_8 - AI_15
  Bank3 AI_16 - AI_23
  Bank4 AI_24 - AI_31
  Bank5 AI_32 - AI_39
  Bank6 AI_40 - AI_47
  Bank7 AI_48 - AI_55
  Bank8 AI_56 - AI_63

  In DIFF mode, the following is the distribution of the banks:

  Bank1 AI_0 - AI_7
  Bank2 AI_8 - AI_15
  Bank3 AI_16 - AI_23
  Bank4 AI_24 - AI_31

  Please pay attention to the wiring method.

Connector 0		Connector 0	
Pin	Identification	Pin	Identification
1	Ctr0_Source/A	5	Ctr1_Source/A
2	Ctr0_Gate/Z	6	Ctr1_Gate/Z
3	Ctr0_Aux/B	7	Ctr1_Aux/B
4	Ctr0_Out	8	Ctr1_Out

Table 12 5113 Pin Definition

AI_GND	Analog Input Reference Ground
AI<0..31>	Analog Input channel
AO_GND	Analog Output Reference Ground
AO<0..3>	Analog Output Channel
D_GND	Digital Signal Reference Ground
DIO<0..15>	Digital I/O Channel
PFI<0..15>	Programmable Function Interface
+5V_OUT	5V power supply

Table 13 Cable Specification

1.13. Accessories

- DIN-68 (PN: JY1717615-01)

  SCSI 68-pin Terminal board

- ACL-1016868-1 (PN: JY1016868-01)

  1M 68pin VHDCI68M-SCSI68M cable

- ACL-1016868-2 (PN: JY1016868-02)

  2M 68pin VHDCI68M-SCSI68M cable

2. Special Operation Note

In the operation of the JY-5113, it is crucial to note that the device **does not support the mixed use of single-ended and differential measurements.** All input channels must be configured consistently as either single-ended or differential for the entirety of the operation.

- **Single-ended mode:** In this mode, each signal is measured with respect to a common ground.
- **Differential mode:** In this mode, the measurement is taken between two input signals, ensuring noise rejection and higher signal integrity.

If your application requires switching between these two modes, it is recommended to reconfigure the channels uniformly before starting a new measurement session.

3. Table of Contents

Table of Contents

1. PCIe/PXIe-5113/5113s Specifications 1

1.1. Main Features 1

1.2. Analog Input 2

1.3. Analog Output 2

1.4. Counter Input/Output 3

1.5. PFI Specifications 3

1.6. Digital IO Specifications 4

1.7. Basic DC AI Accuracy 4

1.8. Basic AO Accuracy 4

1.9. System Noise 5

1.10. Physical and Environment 5

1.11. Power Consumption (typical) 5

1.12. Front Panel and Pin Definition 6

1.13. Accessories 8

2. Special Operation Note 9

3. Table of Contents 10

4. Software 16

4.1. Supported Operating System 16

4.2. Programming Languages 16

5. Order Information 16

6. JYPEDIA 16

7. Additional Hardware Information 17

7.1. Basic DC AI Accuracy 17

7.2. Basic AO Accuracy 17

8. Additional Software Information 19

8.1. System Requirements 19

8.2. System Software 19

8.3. C# Programming Language 20

8.4. PCIe/PXIe-5113 Hardware Driver 20
8.5. Install the SeeSharpTools from JYTEK 21
8.6. Running C# Programs in Linux 21
9. Operating JY5113 22
9.1. Quick Start 22
9.2. Data Acquisition Methods 22
9.2.1. Continuous Acquisition 26
9.2.2. Finite Acquisition 26
9.2.3. Single Point Acquisition 27
9.2.4. Record Acquisition 27
9.3. Analog Input Terminal Type 27
9.3.1. DIFF Mode 27
9.3.2. RSE Mode 30
9.4. Trigger Source 31
9.4.1. Immediate trigger 31
9.4.2. Software Trigger 32
9.4.3. External Analog Trigger 33
9.4.4. External Digital Trigger 38
9.5. Trigger Mode 40
9.5.1. Start Trigger 40
9.5.2. Reference Trigger 41
9.5.3. ReTrigger 42
Learn by Example 9.5 42
9.6. AO Operations 45
9.6.1. Finite Output 45
9.6.2. Continuous NoWrappping Output 48
9.6.3. Continuous Wrapping Output 51
9.7. Digital I/O Operations 54
9.7.1. Static DI/DO 54
9.7.2. Dynamic DI/DO 56
9.8. Counter Input Operations 58
9.8.1. Edge Counting 59

9.8.2. Pulse Measurement 64
9.8.3. Frequency Measurement 69
9.8.4. Period Measurement 73
9.8.5. Two-Edge Separation 74
9.8.6. Quadrature Encoder 78
9.8.7. Two-Pulse Encoder 83
9.9. Counter Output Operations 88
9.9.1. Single Pulse Output 88
9.9.2. Finite Pulse Output 90
9.9.3. Continuous Pulse Output 92
9.10. System Synchronization Interface（SSI）for PCIe Modules 94
9.11. DIP Switch in PCIe/PXIe-5113 95
10. Calibration **95**
11. Appendix(Measurement Issues) **96**
11.1. Performance and Tests 96
11.1.1. AI Accuracy 96
11.1.2. AI Bandwidth 97
11.1.3. System Noise 97
11.1.4. PCIe/PXIe-5113 CMRR 97
11.1.5. AO Accuracy 98
11.2. Floating Signals and Ground Referenced Signals 98
11.3. Differential, NRSE, RSE Modes 98
11.4. Reducing the Common Mode Voltage Effect 99
12. About JYTEK **101**
12.1. JYTEK China 101
12.2. JYTEK Software Platform 101
12.3. JYTEK Warranty and Support Services 101
13. Statement **102**

Table 1 Analog Input Specifications 2
Table 2 Analog Output Specifications 2
Table 3 Counter Input Operations Specifications 3
Table 4 PFI Specifications 3

Table 5 Digital IO Specifications 4
Table 6 Basic DC AI Accuracy....4
Table 7 AI Bandwidth Specifications 4
Table 8 Basic AO Accuracy....4
Table 9 System Noise Specifications Physical and Environment....5
Table 10 Physical and Environment....5
Table 11 5113 Power Consumption....5
Table 12 5113 Pin Definition 7
Table 13 Cable Specification....7
Table 14 Supported Linux Versions 19
Table 15 SSI Connector Pin Assignment for PCIe-5113 94
Table 16 Typical AI Accuracy 96
Table 17 AI Bandwidth 97
Table 18 Typical AO Accuracy....98

Figure 1 PCIe/PXIe 5113 Front Panel....6
Figure 2 Sample Rate and Internal AD Conversion 23
Figure 3 PCIe-5111 experiment 24
Figure 4 TB-68 Terminal Block....25
Figure 5 Continuous MultiChannel Paraments 25
Figure 6 MultiChannel Continuous Acquisition....26
Figure 7 Differential Mode for Grounding Signals 28
Figure 8 10V Range Diamond Plot 28
Figure 9 5V Range Diamond Plot 29
Figure 10 2.5V Range Diamond Plot 29
Figure 11 Choose Differential in AI Terminal 30
Figure 12 RSE Mode for Floating Signals....31
Figure 13 Immediate trigger Paraments 32
Figure 14 Software trigger Paraments 32
Figure 15 Software trigger Acquisition 33
Figure 16 Rising Slope Trigger 34
Figure 17 Falling Slope Trigger....34
Figure 18 Hysteresis with Rising Slope Trigger 35
Figure 19 Hysteresis with Falling Slope Trigger....35
Figure 20 Entering Window Trigger 36
Figure 21 Leaving Window Trigger 36
Figure 22 Analog Trigger Paraments 37
Figure 23 Waiting For Trigger 38
Figure 24 Analog Trigger Acquisition 38
Figure 25 External Digital Trigger 39
Figure 26 Digital Trigger Paraments 39
Figure 27 Digital Trigger Acquisition 40
Figure 28 Start Trigger....41
Figure 29 Reference Trigger....42

Figure 30 ReTrigger 42
Figure 31 Retrigger Paraments 43
Figure 32 Retrigger In Start Trigger Mode 44
Figure 33 Retrigger In Reference Trigger Mode 44
Figure 34 Complete Retrigger Count 45
Figure 35 AI Continuous Paraments 46
Figure 36 AO Finite Output Paraments 46
Figure 37 AO Finite Signal 47
Figure 38 AI Acquisition Signal 47
Figure 39 AI Continuous Paraments 48
Figure 40 AO ContinuousNoWrapping Output Paraments 49
Figure 41 AO ContinuousNoWrapping Signal 50
Figure 42 AI Acquisition AO Sin Signal 50
Figure 43 Update AO Square Signal 51
Figure 44 AI Acquisition AO Square Signal 51
Figure 45 AI Continuous Paraments 52
Figure 46 AO Continuous Wrapping Paraments 53
Figure 47 AO Continuous Wrapping Signal 53
Figure 48 AI Acquisition AO Signal 54
Figure 49 Single Digital Output 55
Figure 50 Single Digital Input 56
Figure 51 DI Continuous Paraments 57
Figure 52 DO ContinuousNoWrapping Output 57
Figure 53 DI Continuous Acquisition 58
Figure 54 Counter Terminal 58
Figure 55 Counter Signal Wiring Instruction 59
Figure 56 Simple Edge Counting in Single Mode 60
Figure 57 Buffered Edge Counting with Internal Sample Clock 60
Figure 58 Simple Edge Counting with Implicit SampleClk 61
Figure 59 Count Direction 61
Figure 60 EdgeCounting For Single Mode 62
Figure 61 EdgeCounting For Finite Mode 63
Figure 62 Counter Values For Internal Clock 63
Figure 63 Counter Values For Implicit Clock 64
Figure 64 Pulse Measurement in Single Mode 65
Figure 65 Pulse Measurement with Internal SampleClk 65
Figure 66 Pulse Measurement with Implicit SampleClk 66
Figure 67 Pulse Measure For Single Mode 66
Figure 68 Pulse Measure Value For Single Mode 67
Figure 69 Pulse Measure For Finite Mode 68
Figure 70 Pulse Measure Values For Finite Mode 68
Figure 71 Frequency Measurement with Internal Sample Clock 70
Figure 72 Frequency Measure For Single Mode 71
Figure 73 Frequency Measure For Continuous Mode 72

Figure 74 Frequency Measure Values 72
Figure 75 Peroid Measure For Single Mode 73
Figure 76 Peroid Measure For Continuous Mode 74
Figure 77 Two-Edge Separation in Single Mode 75
Figure 78 Two-Edge Separation with Internal Sample Clock 75
Figure 79 Two-Edge Separation with Implicit Sample Clock 76
Figure 80 Two-EdgeSeparation Measure For Single Mode 77
Figure 81 Two-EdgeSeparation Measure For Finite Mode 78
Figure 82 Quadrature Encoder x1 Mode 78
Figure 83 Quadrature Encoder x2 Mode 79
Figure 84 Quadrature Encoder x4 mode 79
Figure 85 Quadrature Encoder x1 with Sample Clock 80
Figure 86 Quadrature Encoder x4 with Implicit Sample Clock 81
Figure 87 QuadEncoder For Single Mode 82
Figure 88 QuadEncoder For Continuous Mode 83
Figure 89 Two-Pulse Encoder in Single Mode 84
Figure 90 Two-Pulse Encoder with Internal Sample Clock 84
Figure 91 Two-Pulse Encoder with Implicit Sample Clock 85
Figure 92 Two-PulseEncoder For Single Mode 86
Figure 93 Two-PulseEncoder For Finite Mode 87
Figure 94 Two-PulseEncoder For Continuous Mode 88
Figure 95 Single Pulse Output 88
Figure 96 Single Pulse Generation 89
Figure 97 AI Acquisition Single Pulse 90
Figure 98 Finite Pulse Output 90
Figure 99 Finite Pulses Generation 91
Figure 100 AI Acquisition Finite Pulse 92
Figure 101 Continuous Pulse Output 92
Figure 102 Continuous Pulse Generation 93
Figure 103 AI Acquisition Continuous Pulse 93
Figure 104 SSI Connector in PCIe-5113 94
Figure 105 AI Measurement Error 96
Figure 106 AI Error due to System Random Noise 97
Figure 107 Common Mode Rejection Ratio 97
Figure 108 Six Measurement Scenarios 99
Figure 109 Using Resister to Reduce Common Mode Voltage Effect ... 100

4. Software

4.1. Supported Operating System

Windows 7/10, x64/x86, Linux. See additional information.

4.2. Programming Languages

Microsoft C#. See additional software information for other languages.

5. Order Information

- PCIe-5113 (PN: JY9785803-01)

  64-ch AI (16-Bit, 1 MS/s), 4-ch AO (16-Bit, 2 MS/s), 16 DIO PCIe Multifunction I/O Card

- PXIe-5113 (PN: JY4726411-01)

  64-ch AI (16-Bit, 1 MS/s), 4-ch AO (16-Bit, 2 MS/s), 16 DIO PXIe Multifunction I/O Card

- PCIe-5113s (PN: JY9785803-02)

  64-ch AI.1 MS/s or 8-ch 1MS/s/ch (16-Bit), 4-ch AO (16-Bit, 2 MS/s), 16 DIO PCIe Multifunction I/O Card

- PXIe-5113s (PN: JY4726411-02)

  64-ch AI 1 MS/s or 8-ch 1MS/s/ch (16-Bit), 4-ch AO (16-Bit, 2 MS/s), 16 DIO PXIe Multifunction I/O Card

6. 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.

7. Additional Hardware Information

7.1. Basic DC AI Accuracy

The DAQ mode is the normal data acquisition mode commonly found in commercial DAQ hardware. The basic AI DC accuracy Table 6 of the DAQ mode provides accuracy entries when PCIe/PXIe-5113 operates in the single channel mode and within the indicated calibration temperature range. Please note that this accuracy is valid for every single point regardless how many sample points you acquire.

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

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

For example, at the 2.5 V range and 24 Hours column, if your measurement or reading is 1V, the accuracy of this measurement is:

$$\pm(0.008\% * 1 + 0.052\% * 2.5) = \pm 0.00138\ \text{V} = \pm 1380\ \mu\text{V}$$

The basic accuracy table also provides full-scale accuracy entries for a quick and convenient look-up. For example, the full-scale accuracy for the 2.5 V range and the 24-Hour calibration column is 1500 µV.

7.2. Basic AO Accuracy

The AO output accuracy of PCIe/PXIe-5113 when using the analog output function can be calculated according to the corresponding parameters in the Table 8

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

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

For example, at the 10V range and 24 Hours column, if your output is 4V, the accuracy of this measurement is:

$$\pm(0.007\% * 4 + 0.018\% * 10) = \pm 0.00208\text{V} = \pm 2080\ \mu\text{V}$$

The basic accuracy table also provides full-scale accuracy entries for a quick and convenient look-up. For example, the full-scale accuracy for the 5V range and the 24-Hour calibration column is 2500 μV.

8. Additional Software Information

8.1. System Requirements

The PCIe/PXIe-5113 can be used in a Windows or a Linux operating system.

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

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

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 14 Supported Linux Versions

8.2. System Software

When using the PCIe/PXIe-5113 in the Window environment, you need to install the following software from Microsoft website:

Microsoft Visual Studio Version 2015 or above,

.NET Framework version is 4.0 or above.

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

Given the resources limitation, JYTEK only tested PCIe/PXIe-5113 be with .NET Framework 4.0 with Microsoft Visual Studio 2015. JYTEK relies on Microsoft to maintain the compatibility for the newer versions.

8.3. C# Programming Language

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

8.4. PCIe/PXIe-5113 Hardware Driver

After installing the required application development environment as described above, you need to install the PCIe/PXIe-5113 hardware driver.

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

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

Specific Hardware Driver: Each JYTEK hardware has a C# specific hardware driver. This driver provides rich and easy-to-use C# interfaces for users to operate various PCIe/PXIe-5113 function. JYTEK has standardized the ways which JYTEK and other vendor's DAQ boards are used by providing a consistent user interface, using the methods, properties and enumerations in the object-oriented programming environment. Once you get yourself familiar with how one JYTEK DAQ card works, you should be able to know how to use all other DAQ hardware by using the same methods.

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

8.5. Install the SeeSharpTools from JYTEK

To efficiently and effectively use PCIe/PXIe-5113 board, you need to install a set of free C# utilities, SeeSharpTools from JYTEK. The SeeSharpTools offers rich user interface functions you will find convenient in developing your applications. They are also needed to run the examples come with PCIe/PXIe-5113 hardware. Please register and download the latest SeeSharpTools from our website, www.jytek.com.

8.6. Running C# Programs in Linux

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

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

9. Operating JY5113

This chapter provides the operation guides for PCIe/PXIe-5113, including AI, AO, DI, DO, Timer and programmable I/O interface, etc.

JYTEK provides extensive examples, on-line help and documentation to assist you to operate the PCIe/PXIe-5113 board. JYTEK strongly recommends you go through these examples before writing your own application. In many cases, an example can also be a good starting point for a user application.

9.1. Quick Start

After you have installed the driver software and the SeeSharpTools, you are ready to use Microsoft Visual Studio C# to operate the PCIe/PXIe-5113 products.

If you are already familiar with Microsoft Visual Studio C#, the quickest way to use PCIe/PXIe-5113 board is to go through our extensive examples. We provide source code of our examples. In many cases, you can modify the source code and start to write your applications.

We also provide **Learn by Example** in the following sections. These examples will help you navigate and learn how to use this PCIe/PXIe-5113.

9.2. Data Acquisition Methods

PCIe/PXIe-5113 uses a scanning method to acquire analog data, meaning there is only one ADC chip on the device and all input channels share this ADC. In the scan acquisition mode, you need to configure AI channels and set up some parameters through PCIe/PXIe-5113 driver software. The most important parameters are *Data Acquisition mode*, *Sample Rate*, *SamplesToAcquire*, *Channel Count*, *ChannelRange* and *Analog Input Terminal Type*.

AI Acquisition mode **(***AIMode***)**: PCIe/PXIe-5113 provides 4 acquisition modes, **Continuous**, **Finite**, **Single Point**, **Record,** which will be described in details in Section 9.2.1-9.2.4.

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

SamplesToAcquire: This parameter behaves differently in the different AI acquisition modes. In the continuous acquisition mode, *SamplesToAcuire* is the buffer size used in the AI acquisition task, please see Section9.2.1; in the finite acquisition mode, it is the total number of samples to capture, please see Section9.2.2.

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

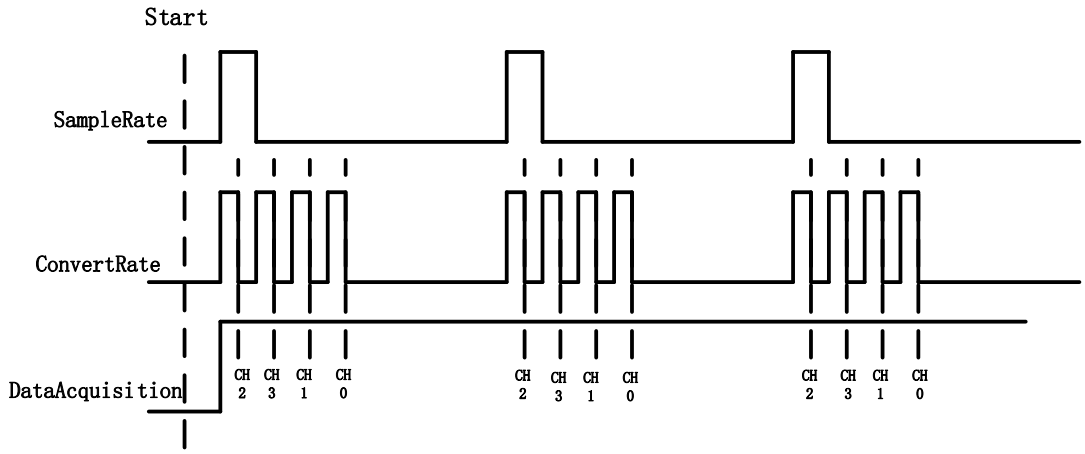


Figure 2 Sample Rate and Internal AD Conversion

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

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

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

Learn by Example 9.2

- Connect the two signal source's positive outputs to PCIe-5111 AI Ch0 (AI0+, Pin #68) and AI Ch1 (AI1+, Pin#33), two negative terminals to the ground (AI_GND, Pin#67) as shown in Figure 3 and Figure 4 (AI0+, AI_GND) and (AI1+, AI_GND) consist of two channels of RSE inputs and they share the same GND.

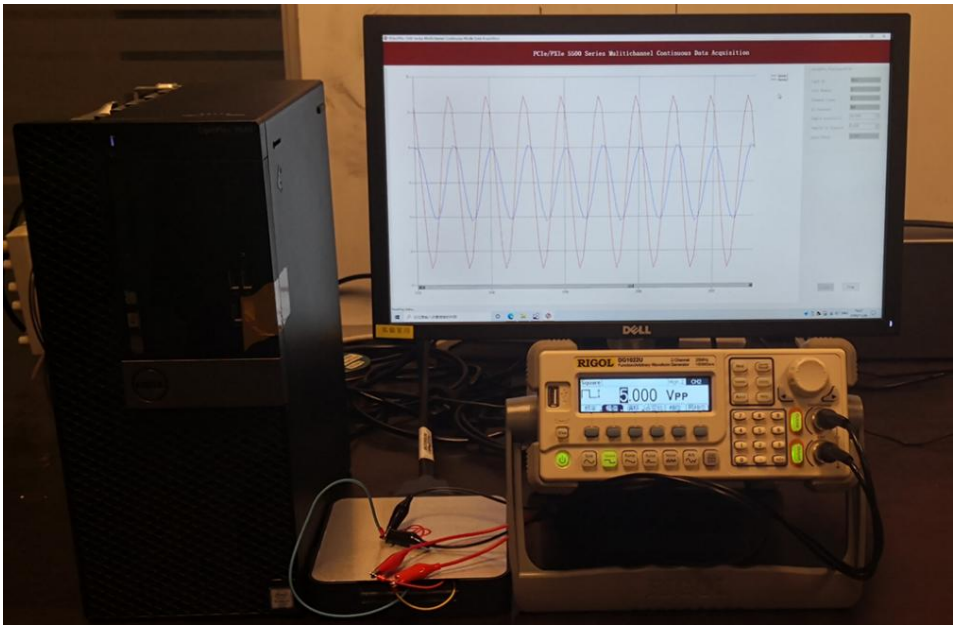


Figure 3 PCIe-5111 experiment

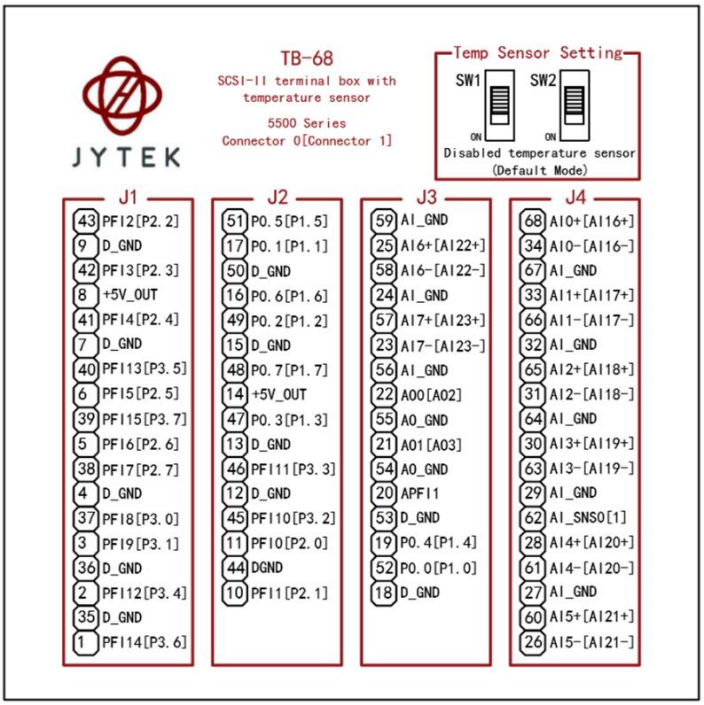


Figure 4 TB-68 Terminal Block

- Set a sinewave signal (f=4Hz, Vpp=5V) and a squarewave signal (f=4Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous MultiChannel**, set the following numbers as shown. This sample program will continuously acquire data from multiple channels.

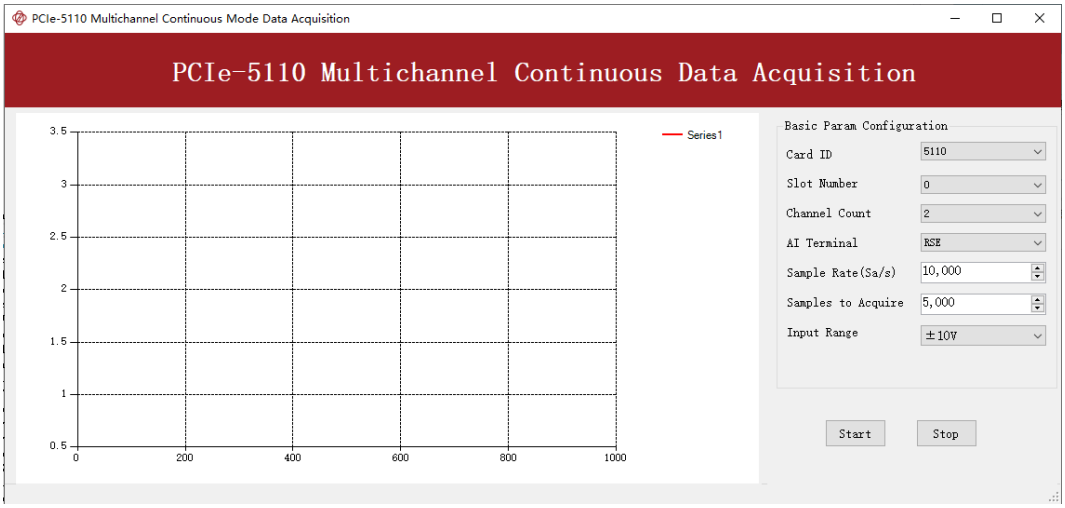


Figure 5 Continuous MultiChannel Paraments

➢ SampleRate is set by **Sample Rate**

- **Samples to Acquire** is the samples to be acquired for each channel in one block. The continuous mode will acquire blocks after blocks until **Stop** button is pressed.

- When start is clicked, it generates a software trigger, which starts the acquisition. The result is shown below.

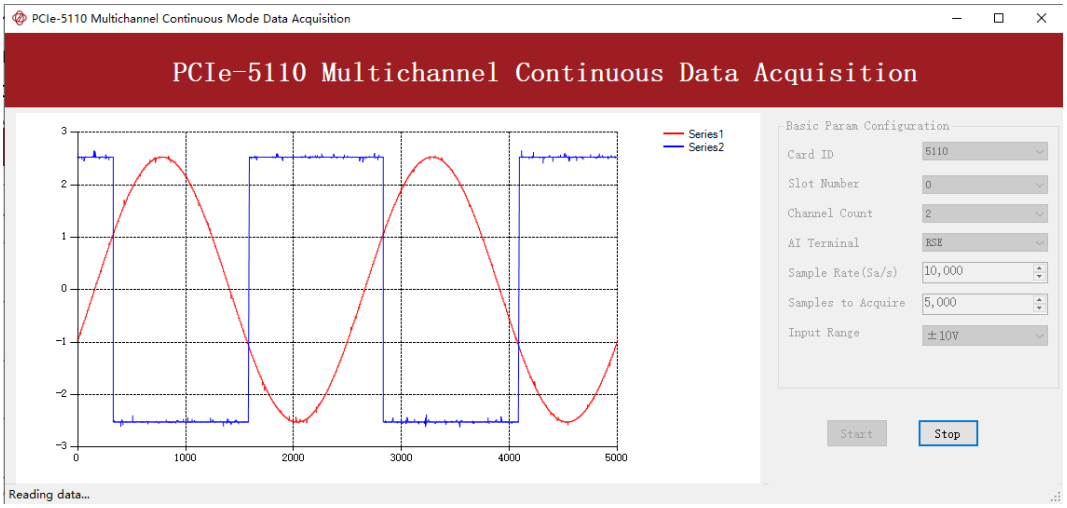


Figure 6 MultiChannel Continuous Acquisition

9.2.1. Continuous Acquisition

An AI acquisition task will acquire the data continuously until the task is stopped. The PCIe/PXIe-5113 device will continue acquiring data and save the data in a circular buffer. You specify how many samples to read back by the user buffer's length, if your program does not read the data fast enough, the circular buffer may overflow. In this case, the driver software will throw out an error message.

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

9.2.2. Finite Acquisition

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

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

9.2.3. Single Point Acquisition

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

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

9.2.4. Record Acquisition

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

9.3. Analog Input Terminal Type

The PCIe/PXIe-5113 provide 2 analog input terminal types:

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

The DIFF connection is recommended for ground-referenced signal sources and it is usually better in rejecting the common-mode noise. However, to acquire one input signal, two AI channels are required to form the differential pair. The RSE is recommended when the input signal sources are floating signals. In RSE mode, these floating signal sources all share the same ground reference (AI_GND). Because of it, the RSE mode can acquire twice as many channels than the DIFF mode. Appendix has more details on these 2 modes.

9.3.1. DIFF Mode

The DIFF mode connects signal's positive side to AI's positive input, signal's grounded negative side to AI's negative input as shown in Figure 7. The common noise appears on both positive and negative terminals of the differential amplifier; thus it will be

cancelled out. Therefore, the DIFF mode has better signal-to-noise ratio (SNR). Please see Appendix for more explanations.

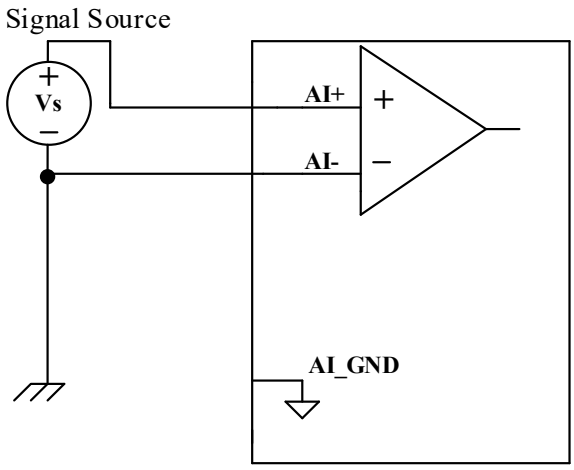


Figure 7 Differential Mode for Grounding Signals

The following figures show the actual collected voltage range diamond plots in DIFF mode for three ranges: ±10V(Figure 7), ±5V(Figure 7), and ±2.5V(Figure 7).The inputs at both ends Vin+ and Vin- must be within the range.

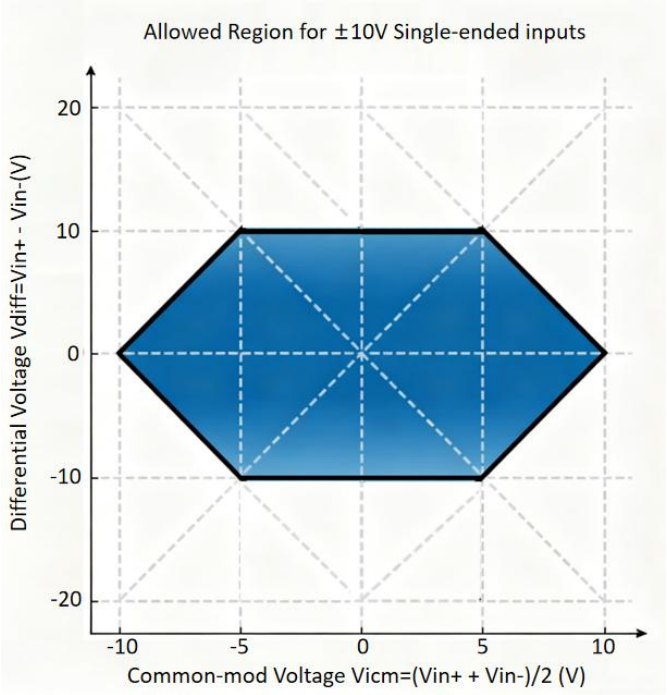


Figure 8 10V Range Diamond Plot

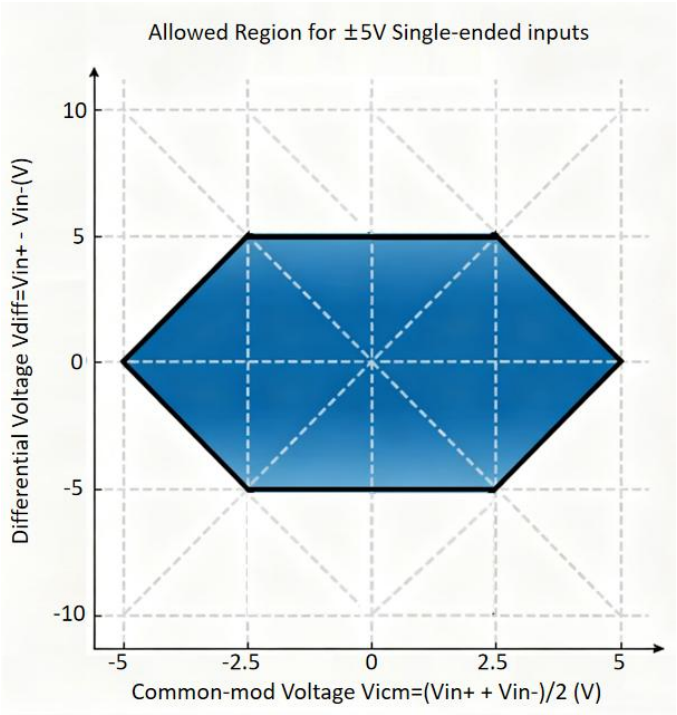


Figure 9 5V Range Diamond Plot

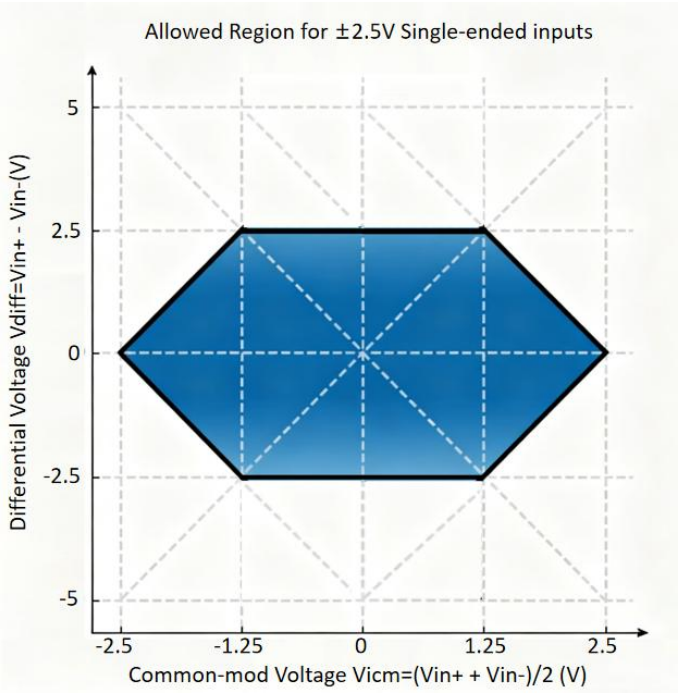


Figure 10 2.5V Range Diamond Plot

Learn by Example 9.3.1

- Open the program **Analog Input-->Winform AI Continuous MultiChannel**

- Connect the two signal source's positive outputs to PCIe-5111 AI Ch0 (AI0+, Pin #68) and AI Ch1 (AI1+, Pin#33), two negative terminals to AI Ch0 negative (AI0-, Pin#34) and AI Ch1 negative (AI1-, Pin#66) as shown in Figure 3 and Figure 4. (AI0+, AI0-) and (AI1+, AI1-) consist of two pairs of DIFF inputs;
- Choose Differential in **AI Terminal**;
- Set other numbers as shown and click **start**.

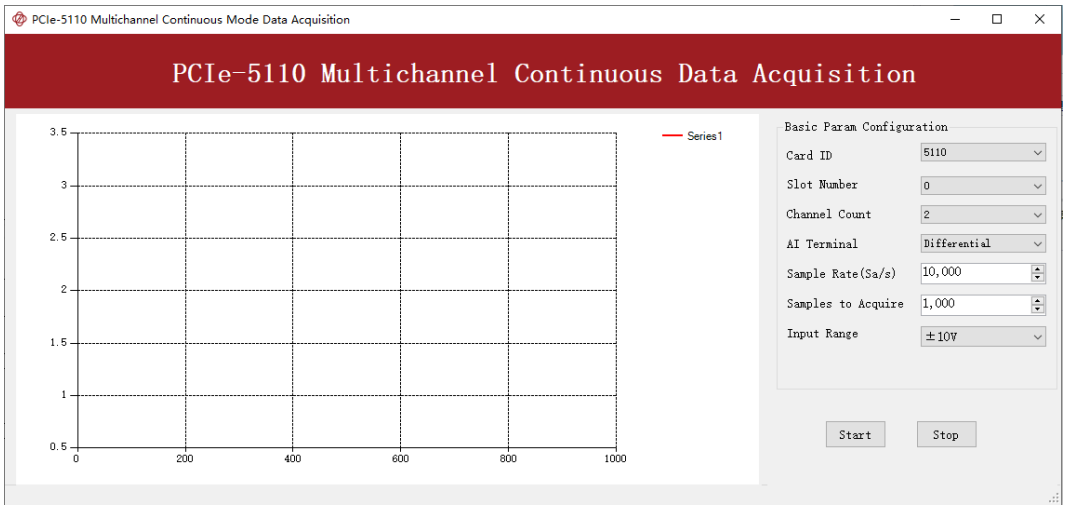


Figure 11 Choose Differential in AI Terminal

9.3.2. RSE Mode

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

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

The RSE mode offers twice as many measurement channels as the DIFF mode. Please see Appendix for more explanations.

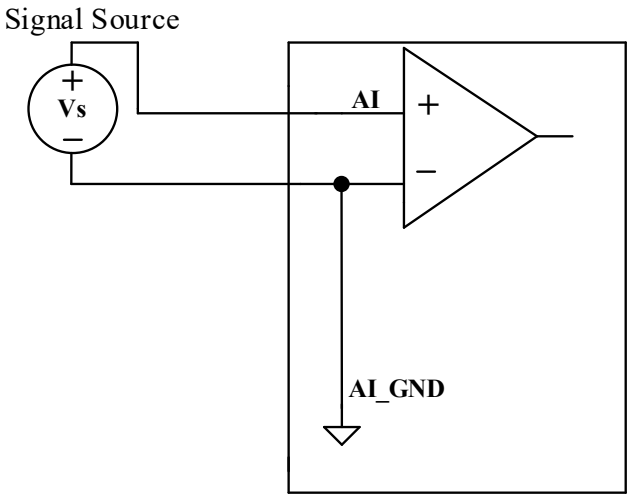


Figure 12 RSE Mode for Floating Signals

9.4. Trigger Source

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

9.4.1. Immediate trigger

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

Learn by Example 9.4.1

- Use the same program and connection as in **Learn by Example9.2.**

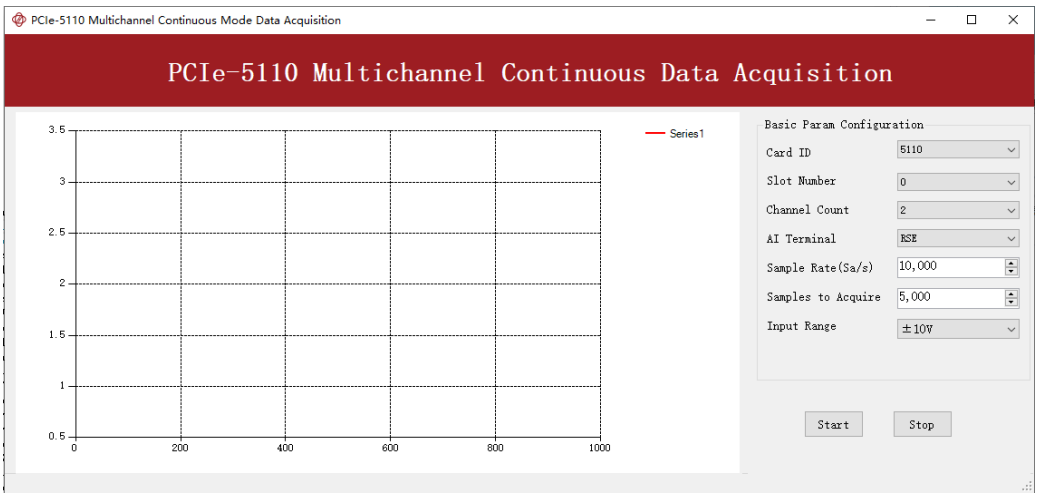


Figure 13 Immediate trigger Paraments

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

9.4.2. Software Trigger

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

Learn by Example 9.4.2

- Connect the signal source's positive terminal to PCIe-5111 AI Ch0 (AI0+, Pin#68), the negative terminal to the ground (AI_GND, Pin#67) as shown in Figure 3 and Figure 4. (AI0+, AI_GND) consists of a RSE input.
- Set a sinewave signal (f=4Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous Soft Trigger**, set the following numbers as shown.
- Click **Start** to run the task.

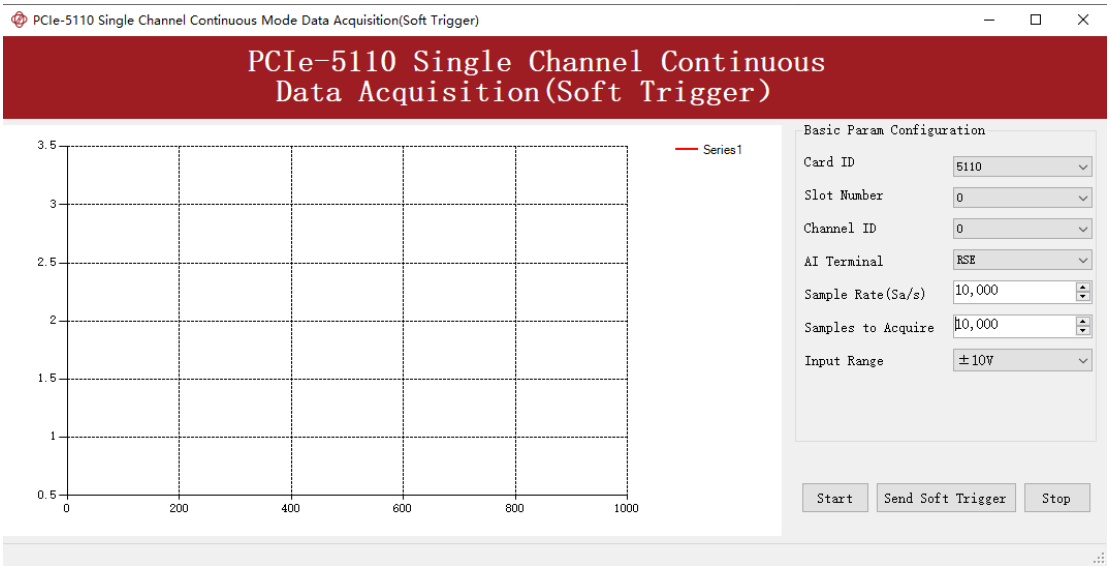


Figure 14 Software trigger Paraments

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

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

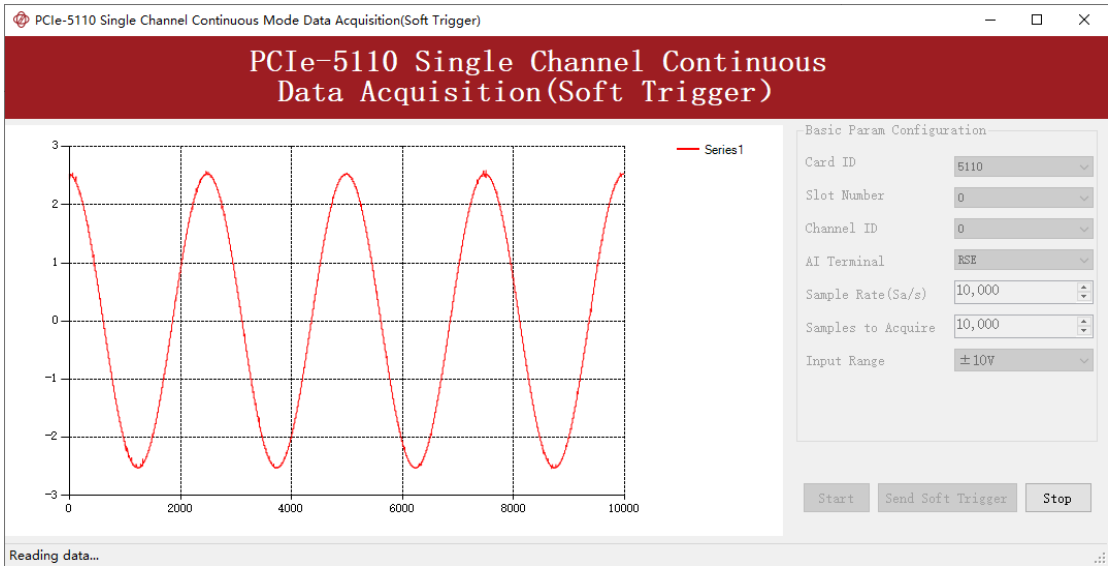


Figure 15 Software trigger Acquisition

9.4.3. External Analog Trigger

You can assign one of measurement channels as the analog trigger source. PCIe/PXIe-5113 provides three analog trigger modes:

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

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

Edge comparator

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

Rising Slope Trigger: The Edge comparator output is high when the signal goes above the threshold; the output is low when the signal goes below the threshold as shown in Figure 16.

Falling Slope Trigger: The Edge comparator output is high when the signal goes below the threshold; the output is low when the signal goes above the threshold as shown in Figure 17.

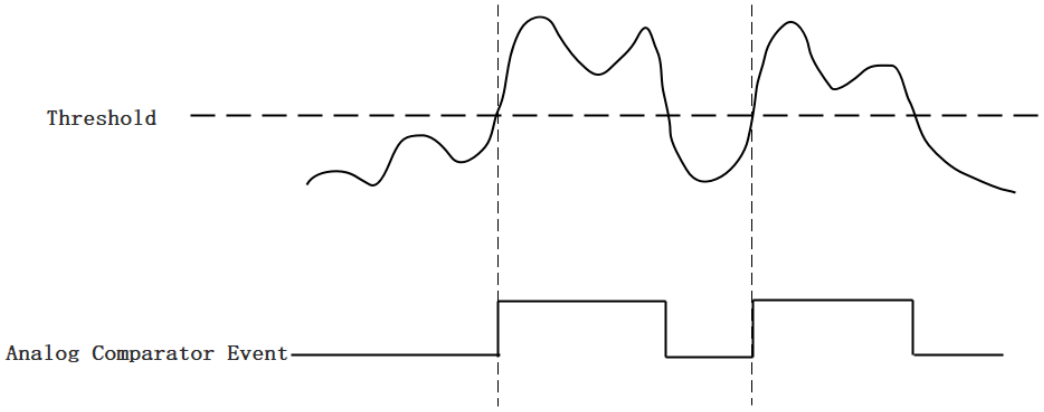


Figure 16 Rising Slope Trigger

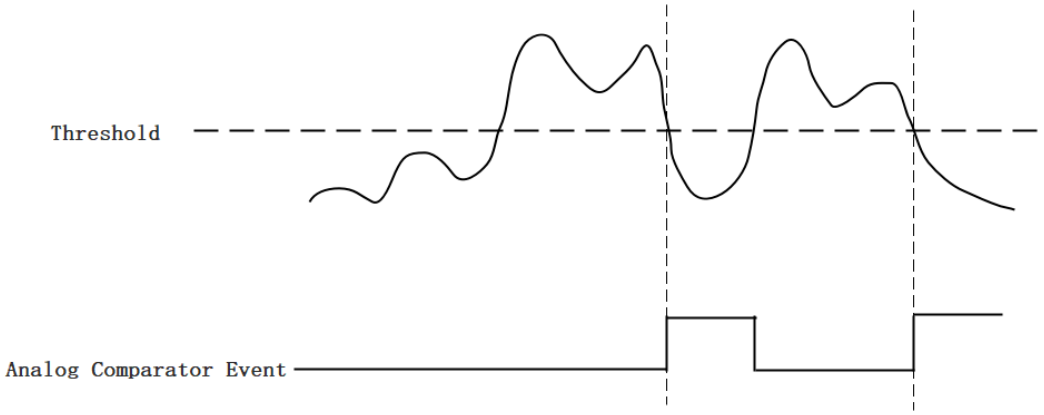


Figure 17 Falling Slope Trigger

Hysteresis Comparator

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

Hysteresis with Rising Slope Trigger: The Hysteresis comparator output is high when the signal must first be below the low threshold, then goes above the high threshold. The output will change to low when the signal goes below the low threshold as shown in Figure 18.

Hysteresis with Falling Slope Trigger: The Hysteresis comparator output is high when the signal must first be above the high threshold, then goes below the low threshold. The output will change to low when the signal goes above the high threshold as shown in Figure 19.

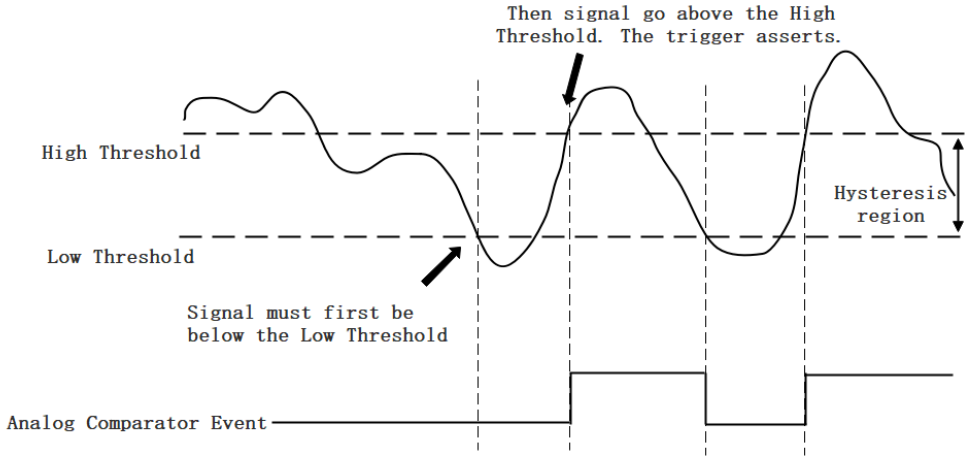


Figure 18 Hysteresis with Rising Slope Trigger

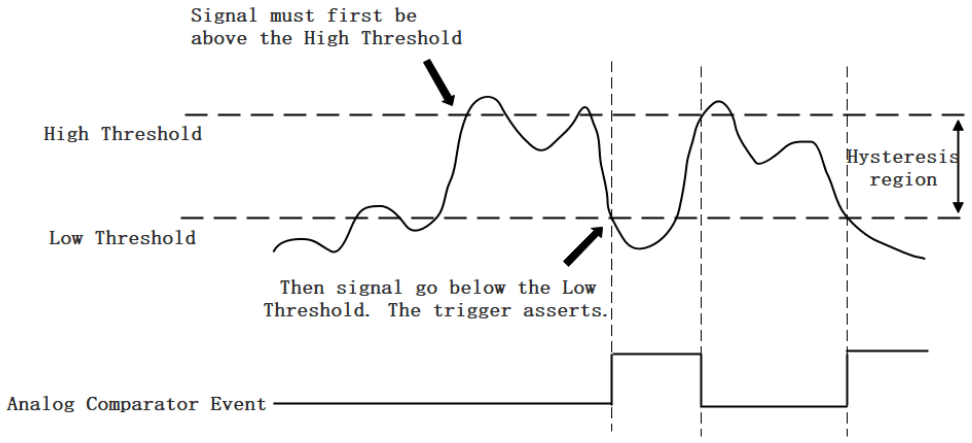


Figure 19 Hysteresis with Falling Slope Trigger

Window comparator

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

Entering Window Trigger: The window comparator output is high when the signal enters the window defined by the *Low Threshold* and *High Threshold*. The output will change to low when the signal leaves the window as shown in Figure 20.

Leaving Window Trigger: The window comparator output is high when the signal leaves the window defined by the *Low Threshold* and *High Threshold*. The output will change to low when the signal enters the window as shown in Figure 21 Leaving Window Trigger.

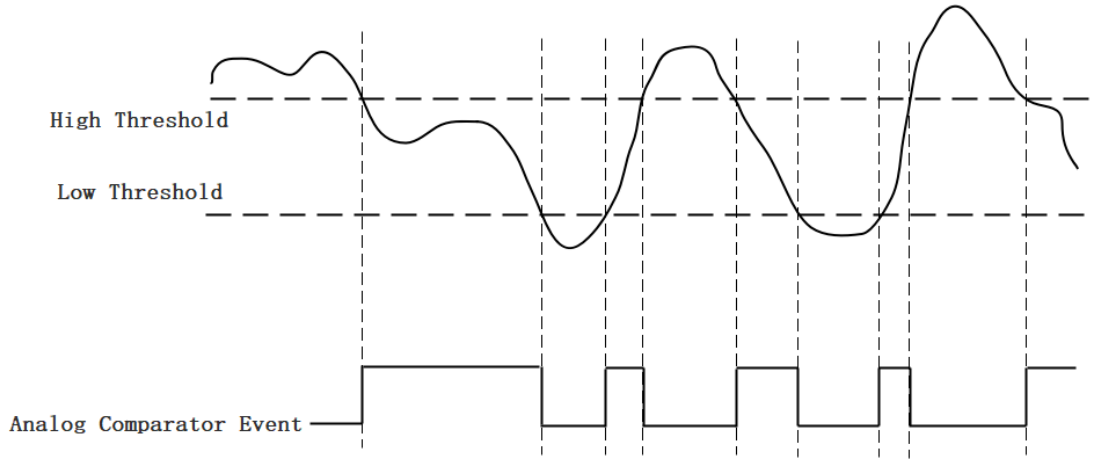


Figure 20 Entering Window Trigger

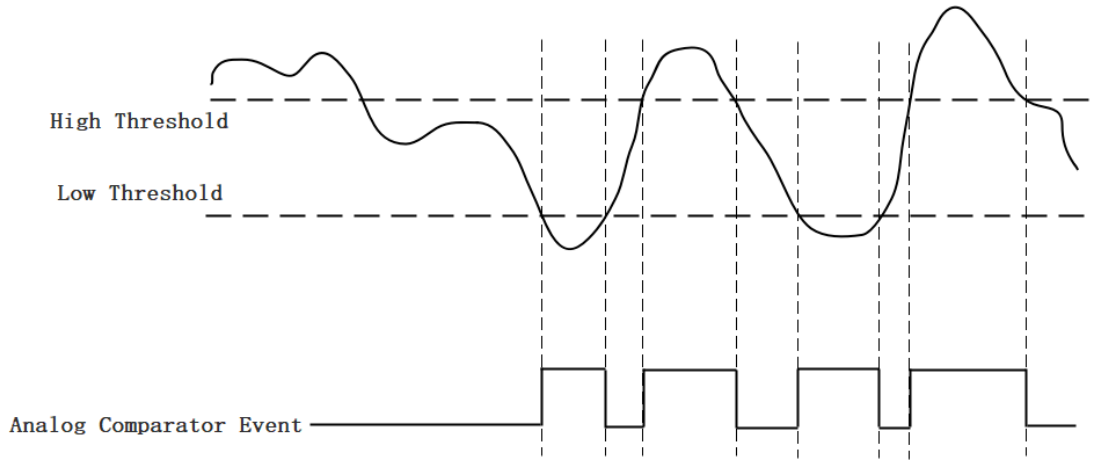


Figure 21 Leaving Window Trigger

Learn by Example 9.4.3

- Connect the signal source's positive terminal to PCIe-5111 AI Ch0 (AI0+, Pin#68), the negative terminal to the ground (AI_GND, Pin#67) as shown in Figure 3 and Figure 4. (AI0+, AI_GND) consists of a RSE input.
- Set a sinewave signal (f=4Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous Analog Trigger**, set the following numbers as shown.

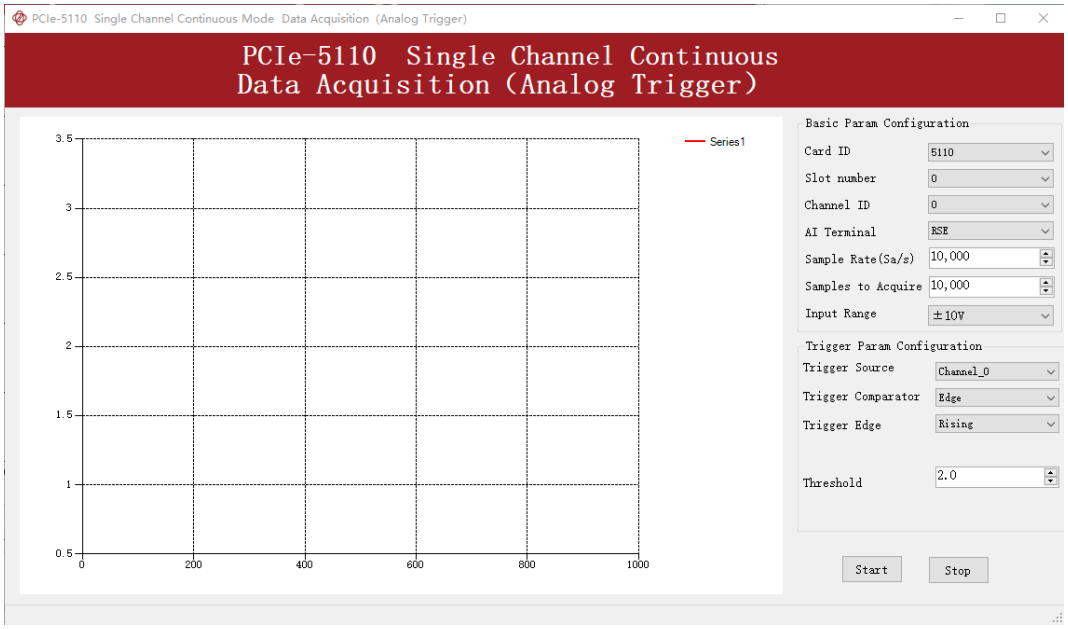


Figure 22 Analog Trigger Paraments

- 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 PCIe-5111 analog input. Set it to **Channel_0**.
- According to the rules of **Rising** mentioned above, the signal acquisition will not start until it raises to 2.0 V, which is set by **Threshold** above.

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

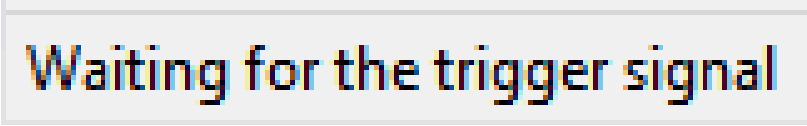


Figure 23 Waiting For Trigger

- This indicates the data acquisition will start only after a triggering event. In this example a trigger signal will occur when the *hysteresis comparator* meets the condition explained in **9.4.3**.

- The result is shown below:

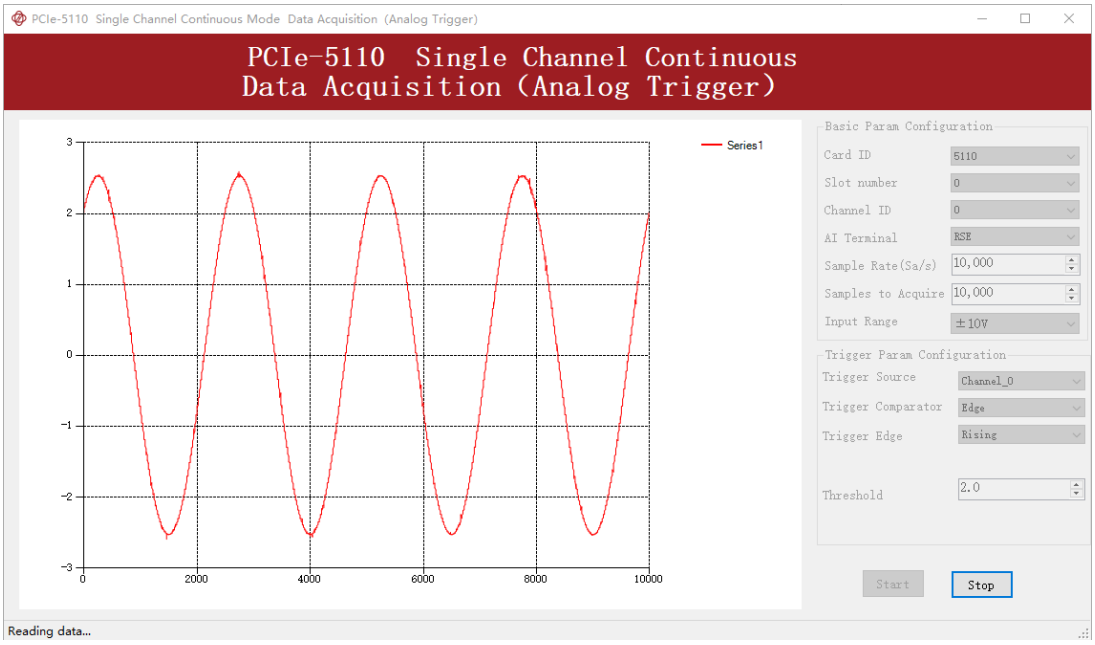


Figure 24 Analog Trigger Acquisition

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

9.4.4. External Digital Trigger

PCIe/PXIe-5113 supports different external digital trigger sources from PXI Trigger bus (PXI_TRIG<0..7>), PXI_STAR and connectors of front panel (PFI). The high pulse width of digital trigger signal must be longer than 20 ns for effective trigger. The module will monitor the signal on digital trigger source and wait for the rising edge or falling edge of digital signal which depending on the set trigger condition, then cause the module to acquire the data as shown in Figure 25

Figure 25 External Digital Trigger

Learn by Example 9.4.4

- Connect the signal source two positive terminals to PCIe-5111 AI Ch0, (AI0+, Pin #68) and digital trigger source (PFI 0, Pin#11), two negative terminals to the ground of analog input (AI_GND, Pin#67) and the ground of digital input/output (DGND, Pin#44) as shown in Figure 3 and Figure 4 (AI0+, AI_GND) consists of a RSE input. (PFI0, DGND) provides the trigger signal.
- Set a sinewave signal (f=4Hz, Vpp=5V) and a squarewave signal (f=4Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous Digital Trigger**, set the following numbers as shown.

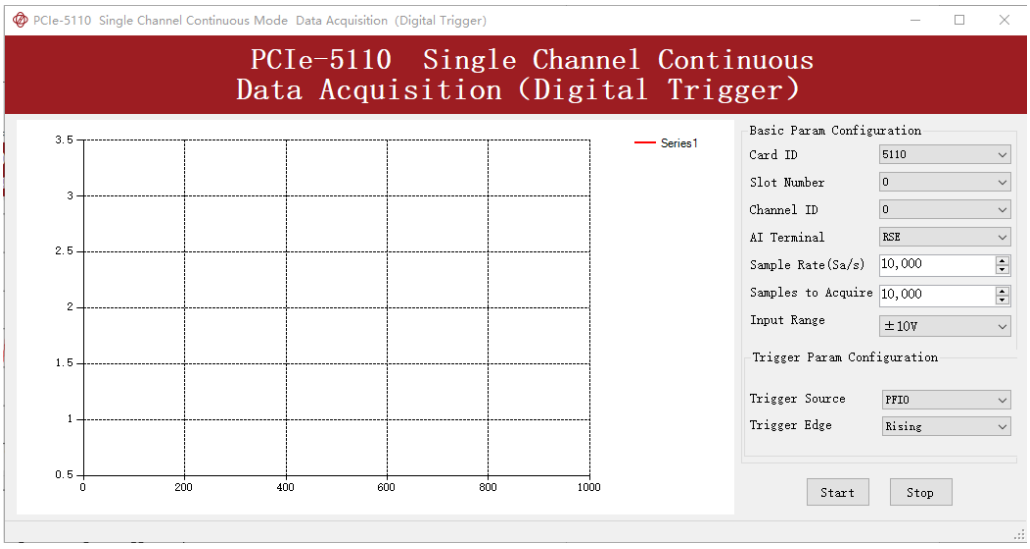


Figure 26 Digital Trigger Paraments

- **Trigger Source** must match the pin on 5110.
- There are two **Trigger Edge**: **Rising** and **Falling**.

■ Click **Start** and the result shows below:

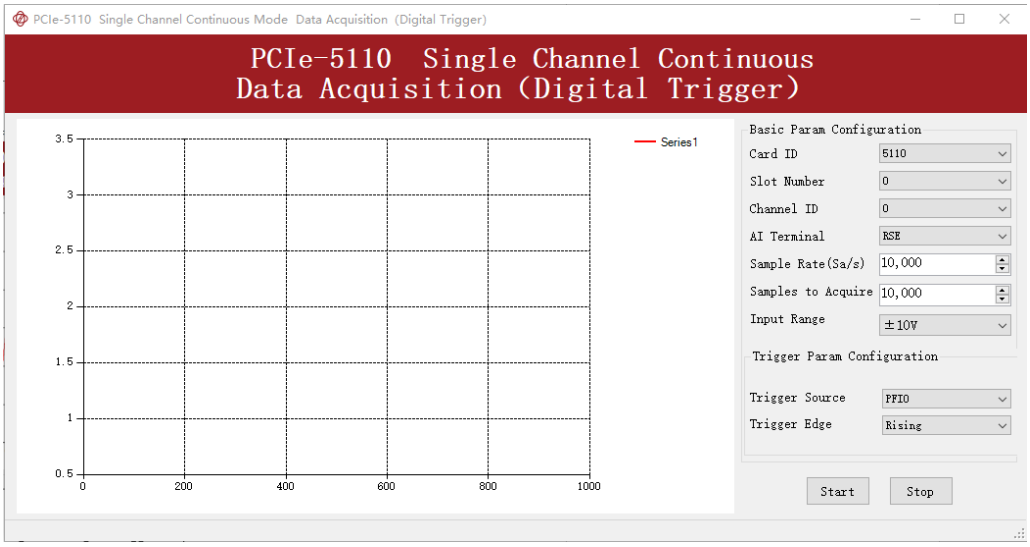


Figure 27 Digital Trigger Acquisition

➢ 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.

9.5. Trigger Mode

The PCIe/PXIe-5113's analog inputs support several trigger modes: start trigger, reference trigger, and re-trigger.

9.5.1. Start Trigger

In this mode, data acquisition begins immediately after the trigger. This trigger mode is suitable for continuous acquisition and finite acquisition. As shown in Figure 28.

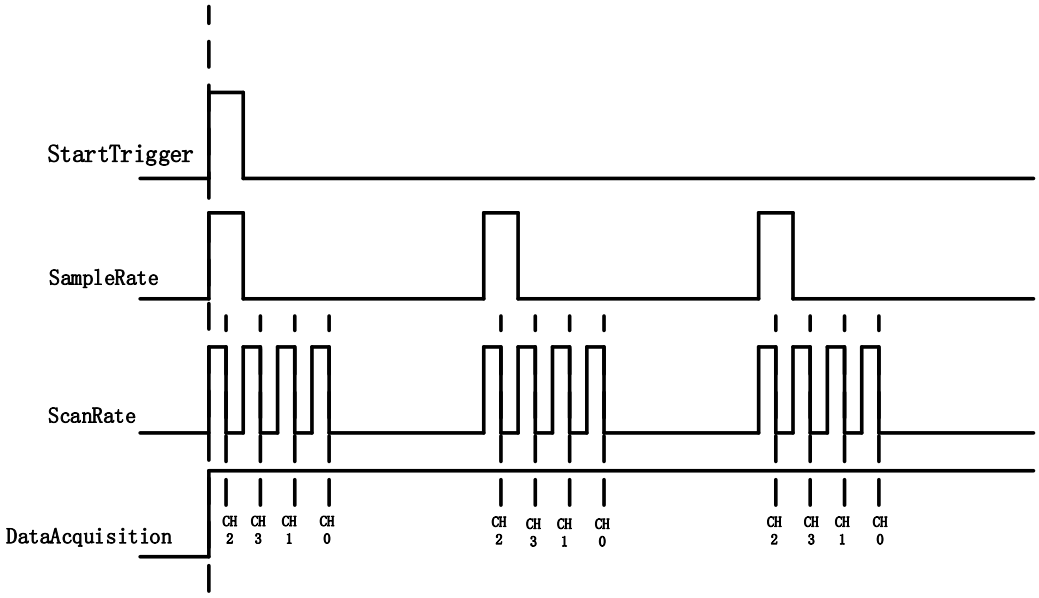


Figure 28 Start Trigger

9.5.2. Reference Trigger

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

Example

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

The principle is shown in Figure 29.

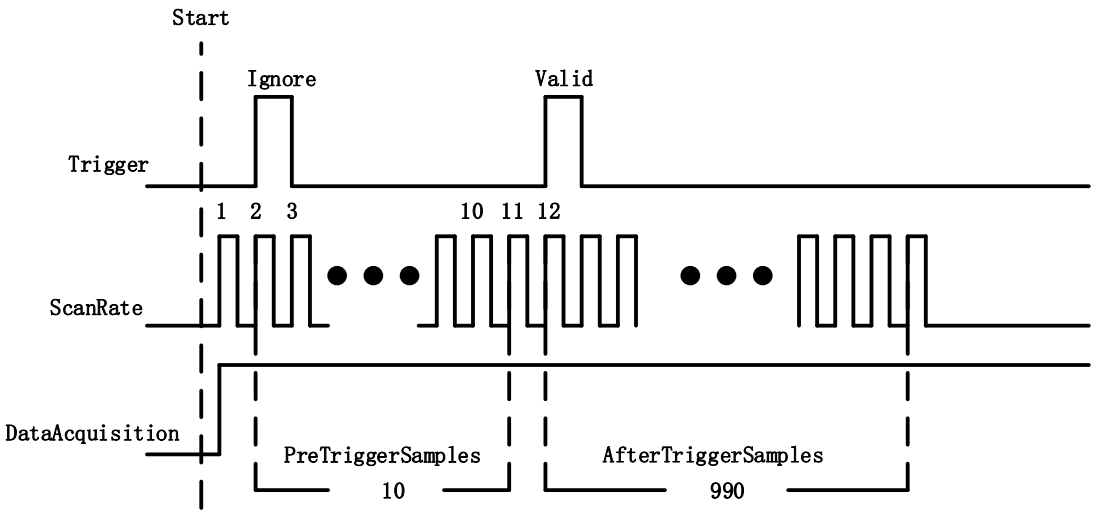


Figure 29 Reference Trigger

9.5.3. ReTrigger

PCIe/PXIe-5113 supports retrigger mode. In the retrigger mode, you can set the number of retrigger and the length of each acquisition. Assuming that the number of re triggers is n and the length of each trigger acquisition is m, the length of all acquisition data is n * m * channelcount. Show in Figure 30.

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

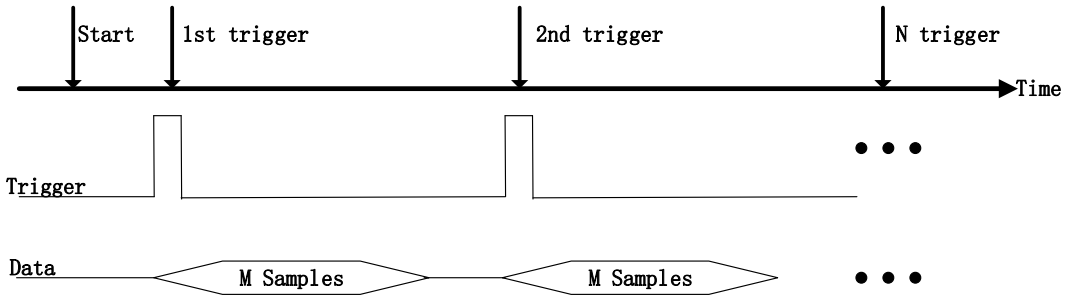


Figure 30 ReTrigger

Learn by Example 9.5

Connect the signal source's positive terminal to PCIe-5111 AI Ch0 (AI0+, Pin#68), the negative terminal to the ground (AI_GND, Pin#67) as shown in Figure 3 and Figure 4. (AI0+, AI_GND) consists of an RSE input.

- Set a sinewave signal (f=4Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Finite Analog Trigger**, set the following numbers as shown.

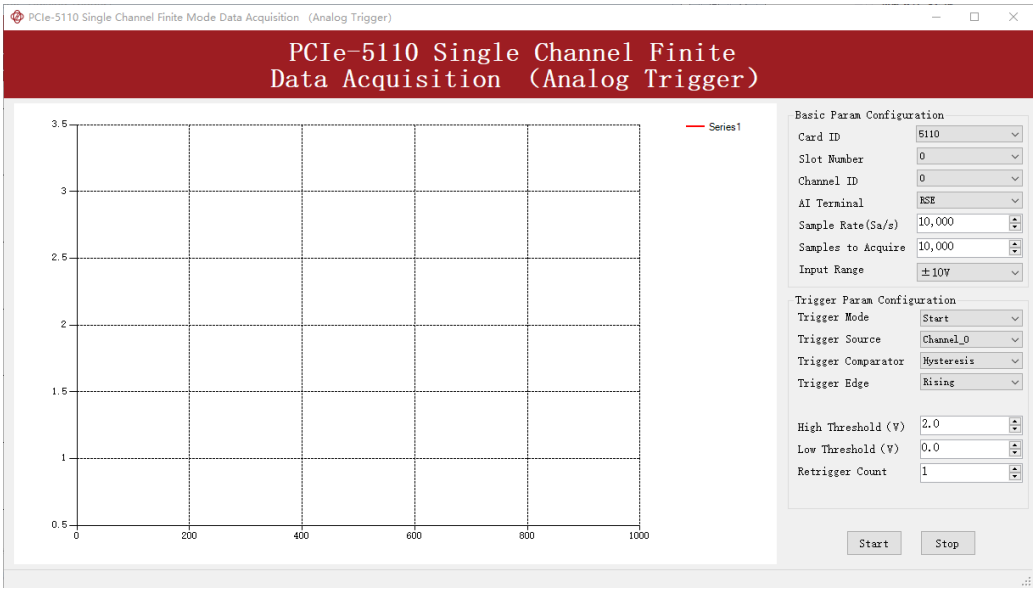


Figure 31 Retrigger Paraments

- You can use three different kinds of triggers in this program as mentioned in **9.5**. *Start Trigger* and *Reference Trigger* can be set by **Trigger Mode**. For *ReTrigger* can be used by changing the numbers in **Retrigger Count**.

- PretriggerSamples is set by **Pretrigger Samples**.

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

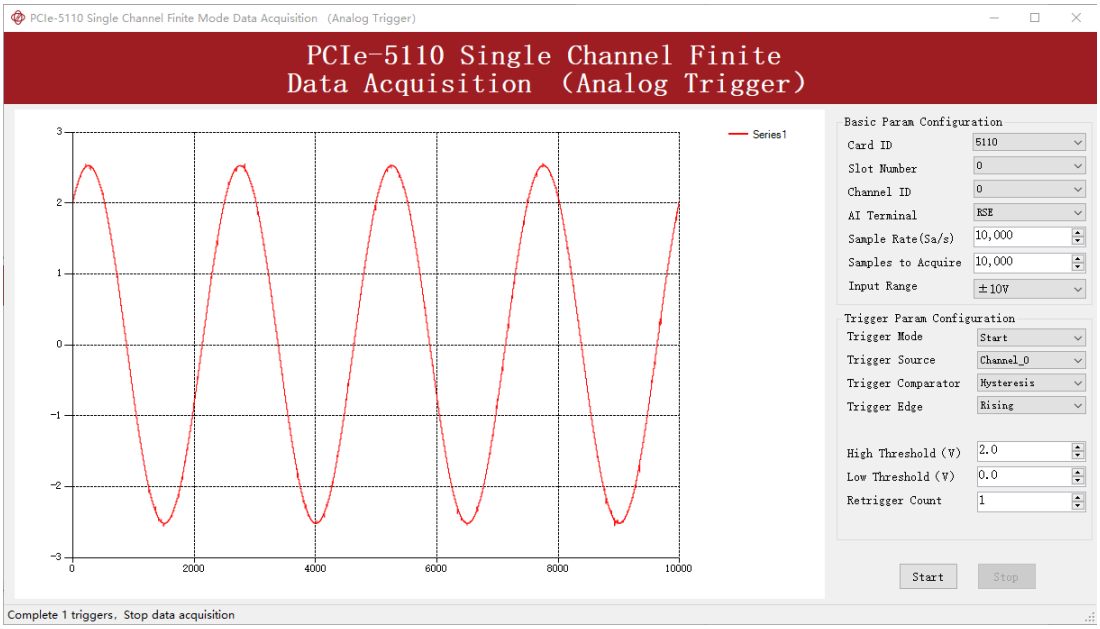


Figure 32 Retrigger In Start Trigger Mode

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

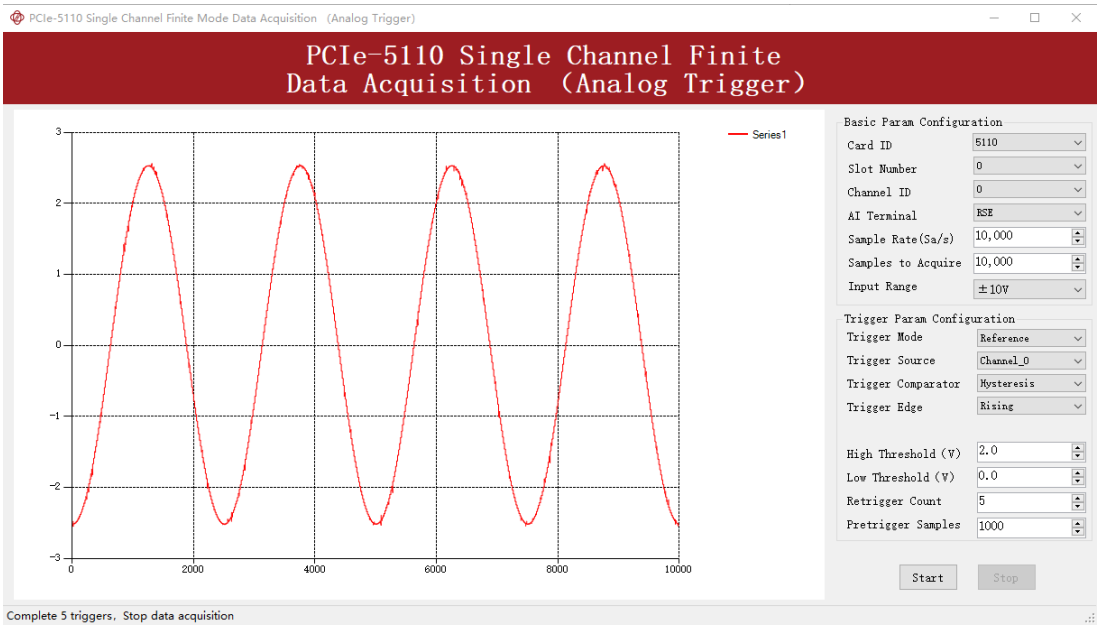


Figure 33 Retrigger In Reference Trigger Mode

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

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

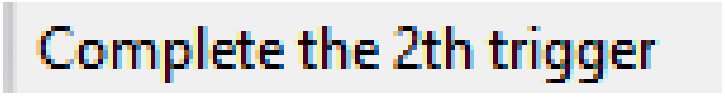


Figure 34 Complete Retrigger Count

  - It shows the acquisition process through every trigger signal.

9.6. AO Operations

The PCIe/PXIe-5113 AO provides 16-bit simultaneous outputs. The analog output has three modes of operation: Finite, ContinuousWrapping, and ContinuousNoWrapping.

9.6.1. Finite Output

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

Learn by Example 9.6.1

- Connect PCIe-5111 AO Ch0 (AO0, Pin #22) to AI Ch0 (AI0+, Pin#68), Ground of AO0 (AO_GND, Pin#55) to Ground of AI0 (AI_GND, Pin#67). (AI0+, AI_GND) consists of a RSE input; (AO0, AO_GND) consists of an output.
- PCIe-5111 sends an analog signal through (AO0, AO_GND) and reads back the signal from (AI0+, AI_GND).
- Open **Analog Input-->Winform AI Continuous**, set the following numbers as shown.

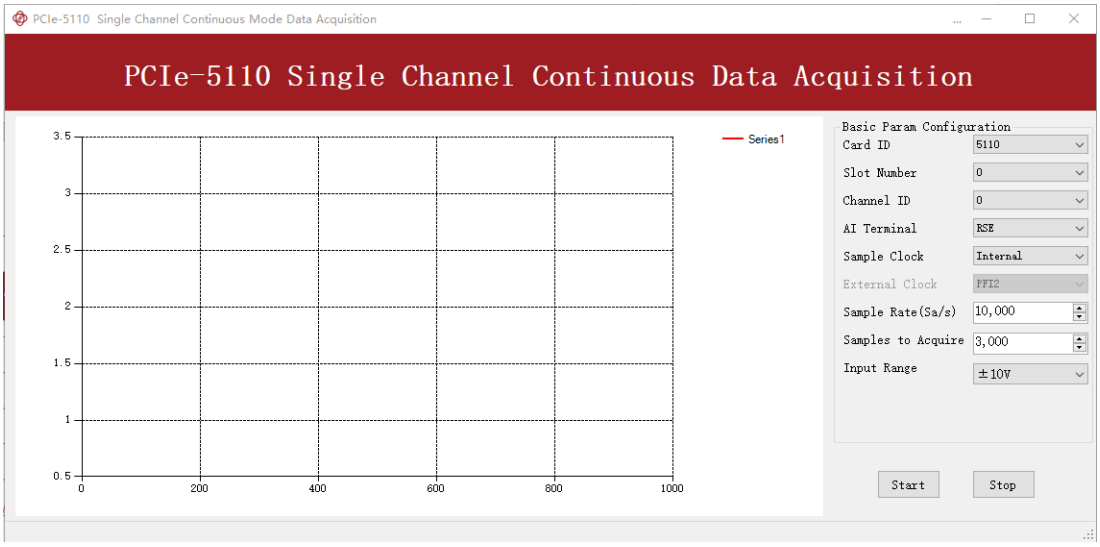


Figure 35 AI Continuous Paraments

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

- Open **Analog Output-->Winform AO Finite**, set the following numbers as shown:

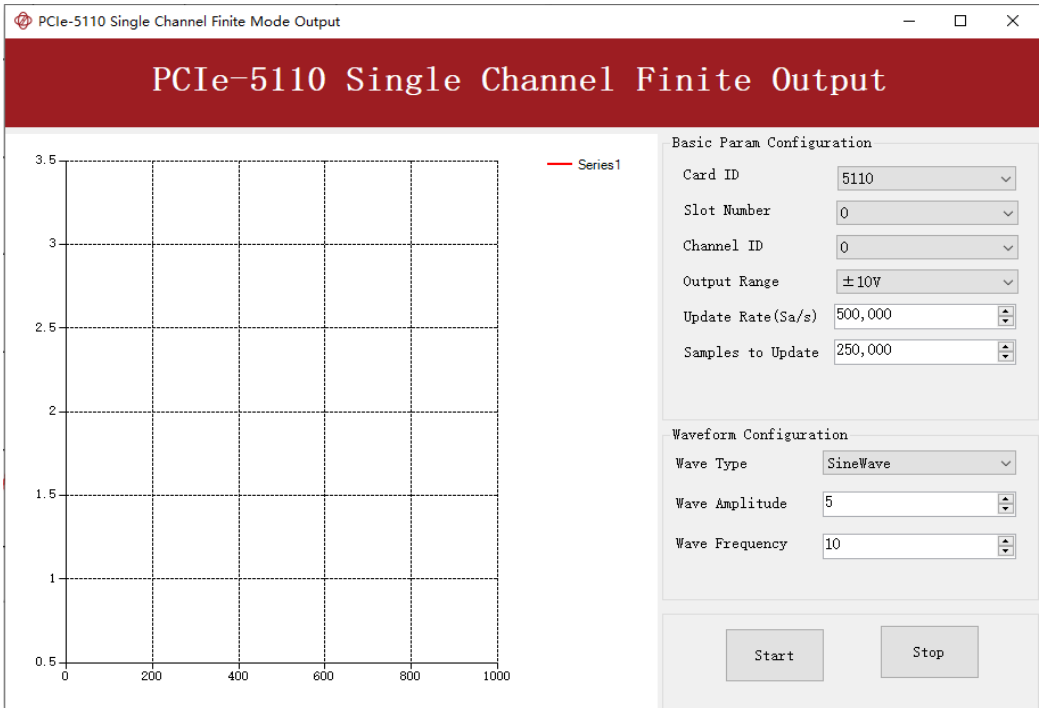


Figure 36 AO Finite Output Paraments

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

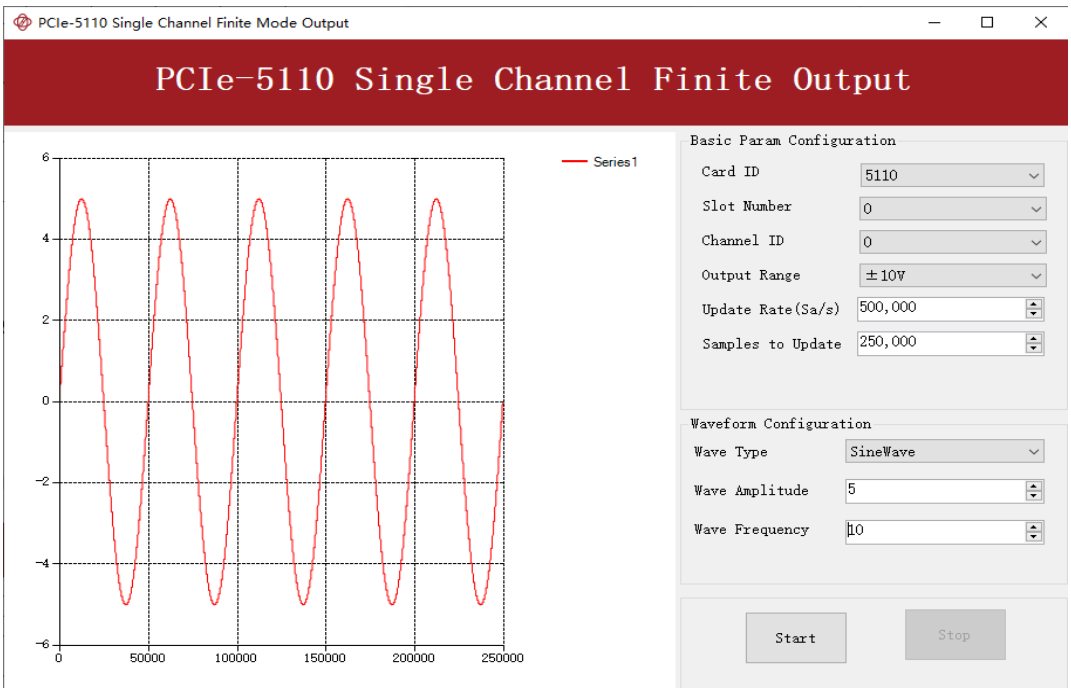


Figure 37 AO Finite Signal

- And the received signal is shown below.

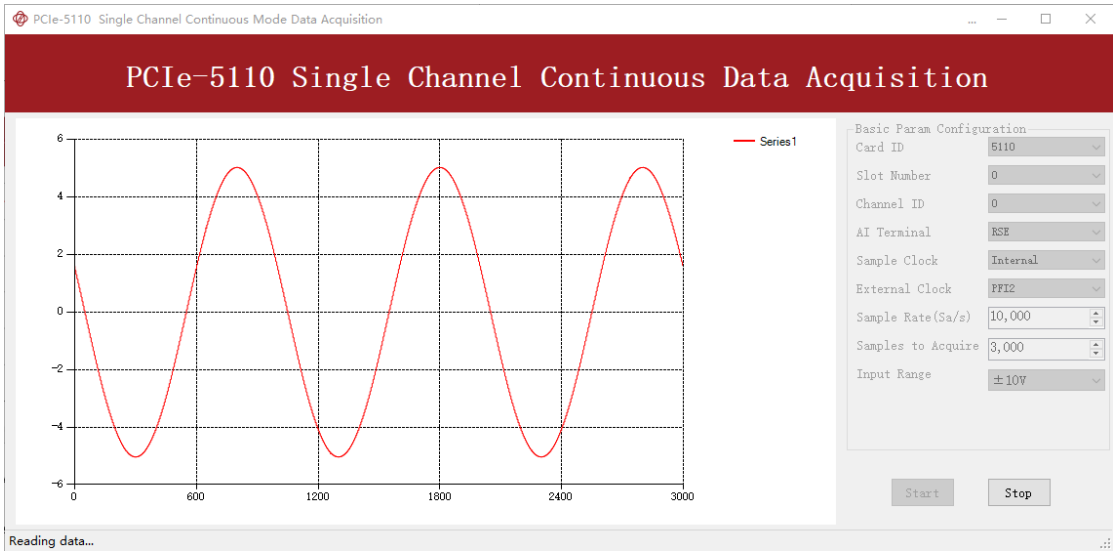


Figure 38 AI Acquisition Signal

- The analog signal is successfully generated and received by PCIe-5111.

9.6.2. Continuous NoWrappping Output

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

Learn by Example 9.6.2

- Connect PCIe-5111 AO Ch0 (AO0, Pin #22) to AI Ch0 (AI0+, Pin#68), Ground of AO0 (AO_GND, Pin#55) to Ground of AI0 (AI_GND, Pin#67). (AI0+, AI_GND) consists of a RSE input; (AO0, AO_GND) consists of an output.
- PCIe-5111 sends an analog signal through (AO0, AO_GND) and reads back the signal from (AI0+, AI_GND).
- Open **Analog Input-->Winform AI Continuous**, set the following numbers as shown.

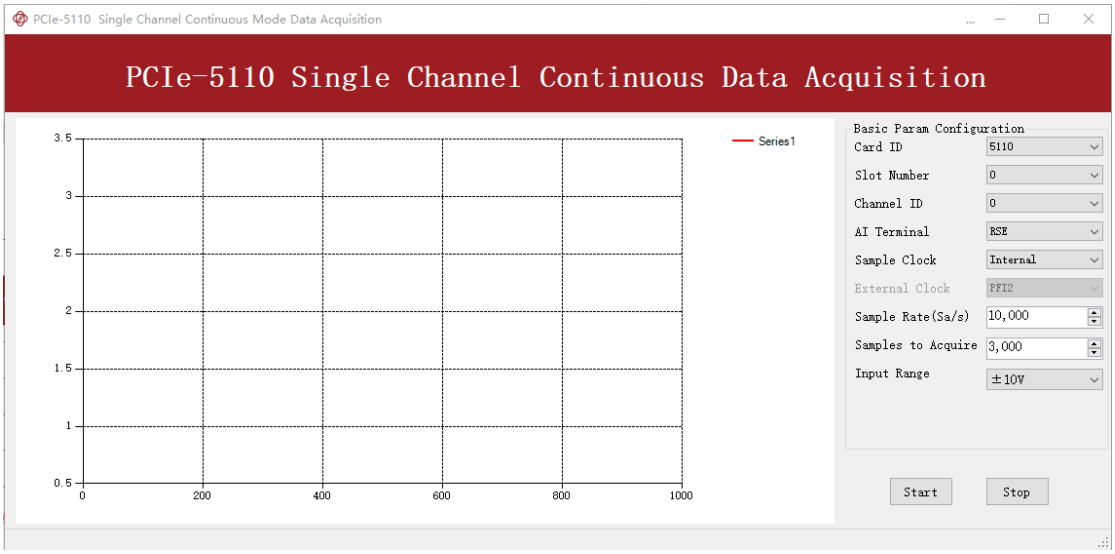


Figure 39 AI Continuous Paraments

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

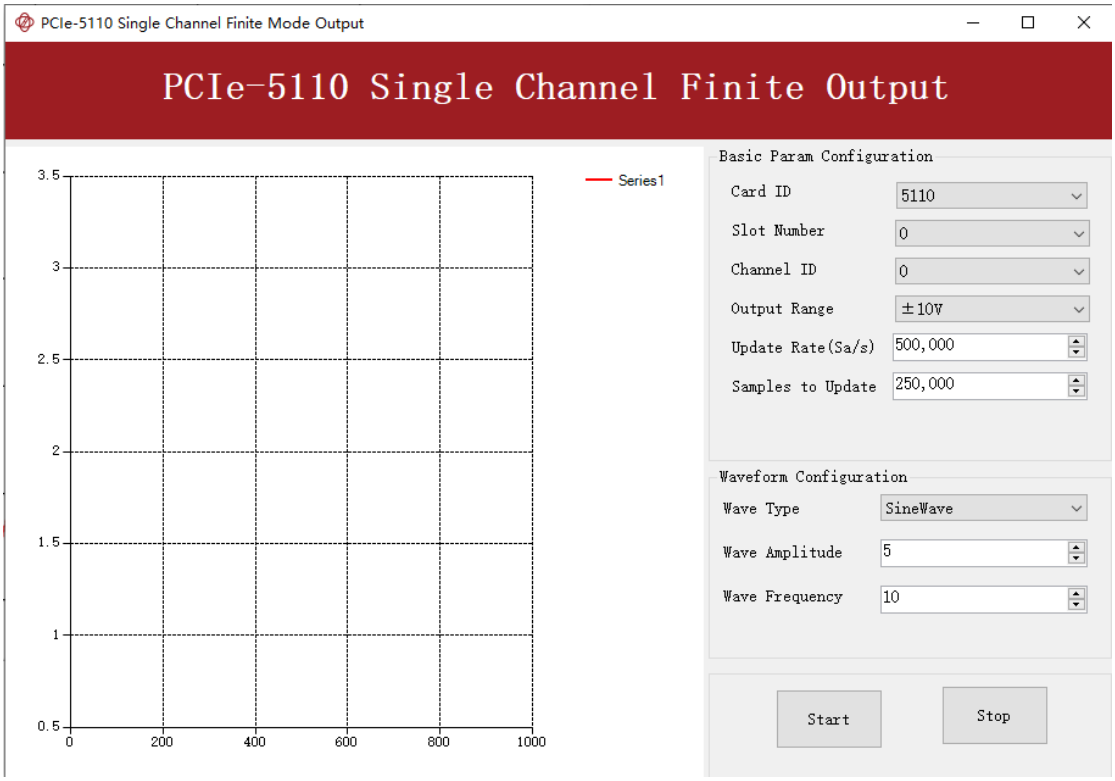


Figure 40 AO ContinuousNoWrapping Output Paraments

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

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

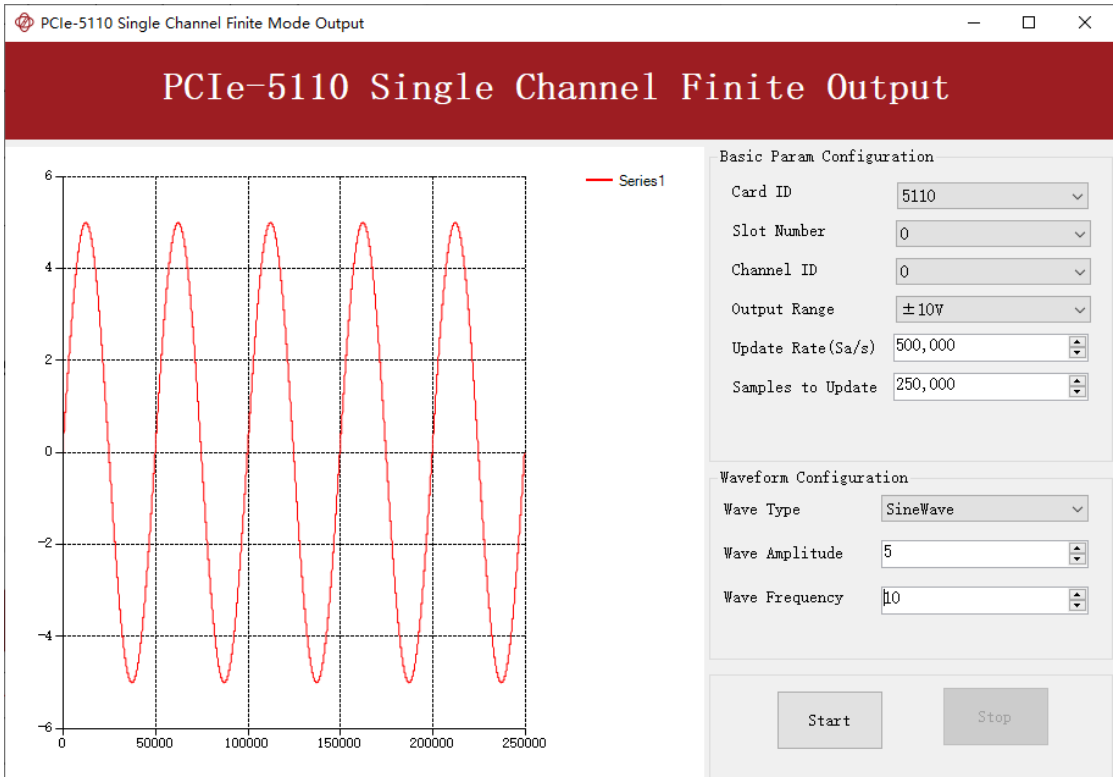


Figure 41 AO ContinuousNoWrapping Signal

- And the received signal is shown below.

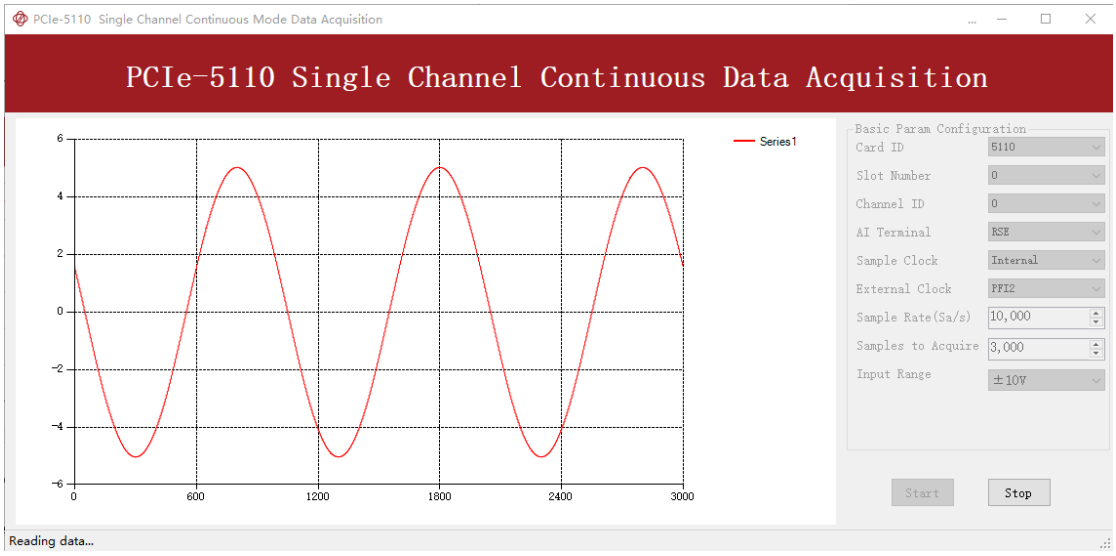


Figure 42 AI Acquisition AO Sin Signal

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

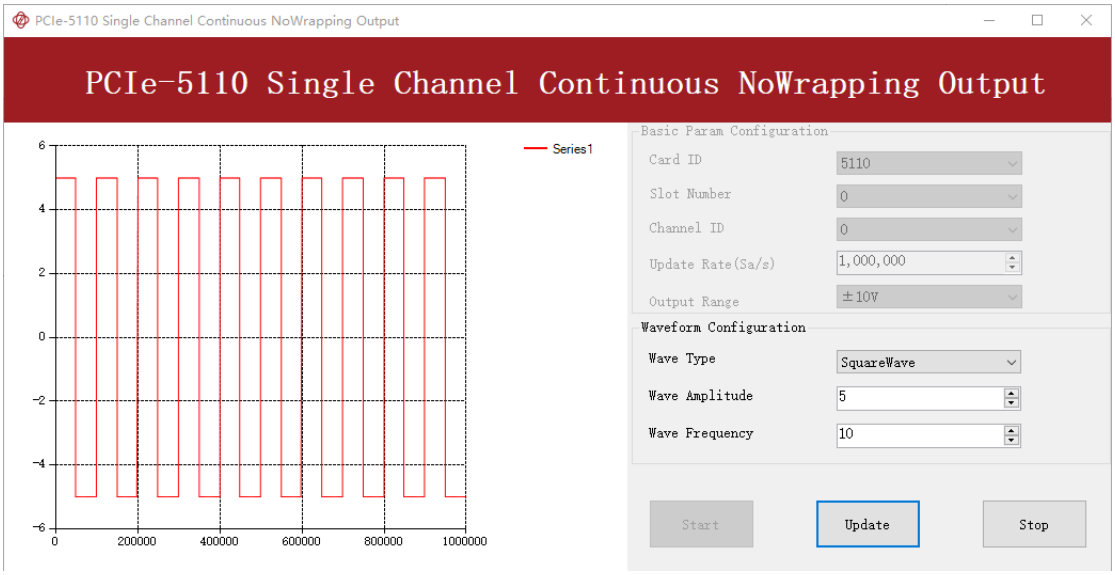


Figure 43 Update AO Square Signal

- And the received signal is shown below.

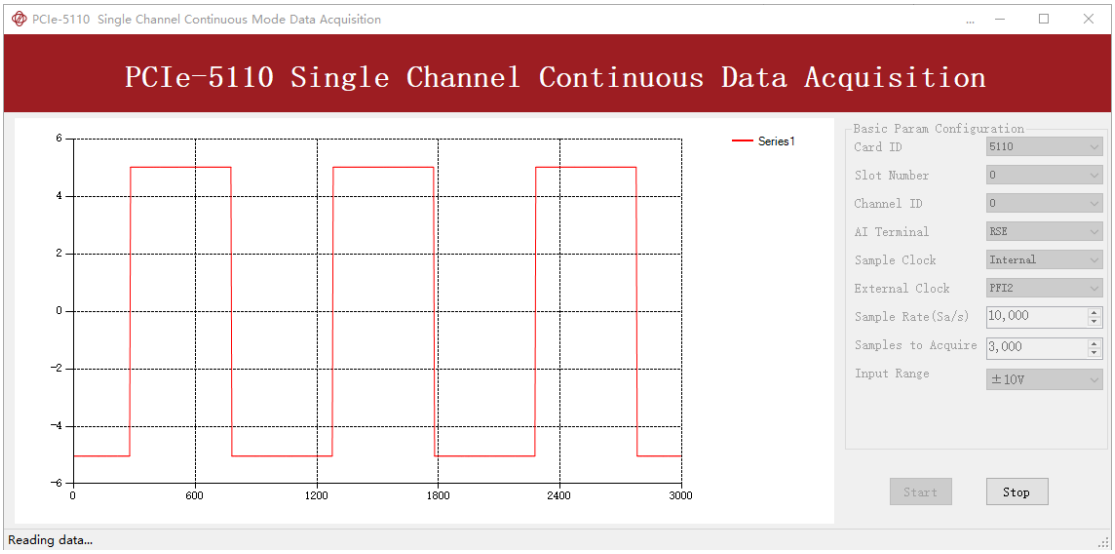


Figure 44 AI Acquisition AO Square Signal

- The analog signal is successfully generated and received by PCIe-5111.

9.6.3. Continuous Wrapping Output

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

Learn by Example 9.6.3

- Connect PCIe-5111 AO Ch0 (AO0, Pin #22) to AI Ch0 (AI0+, Pin#68), Ground of AO0 (AO_GND, Pin#55) to Ground of AI0 (AI_GND, Pin#67). (AI0+, AI_GND) consists of a RSE input; (AO0, AO_GND) consists of an output.
- PCIe-5111 sends an analog signal through (AO0, AO_GND) and reads back the signal from (AI0+, AI_GND).
- Open **Analog Input-->Winform AI Continuous**, set the following numbers as shown.

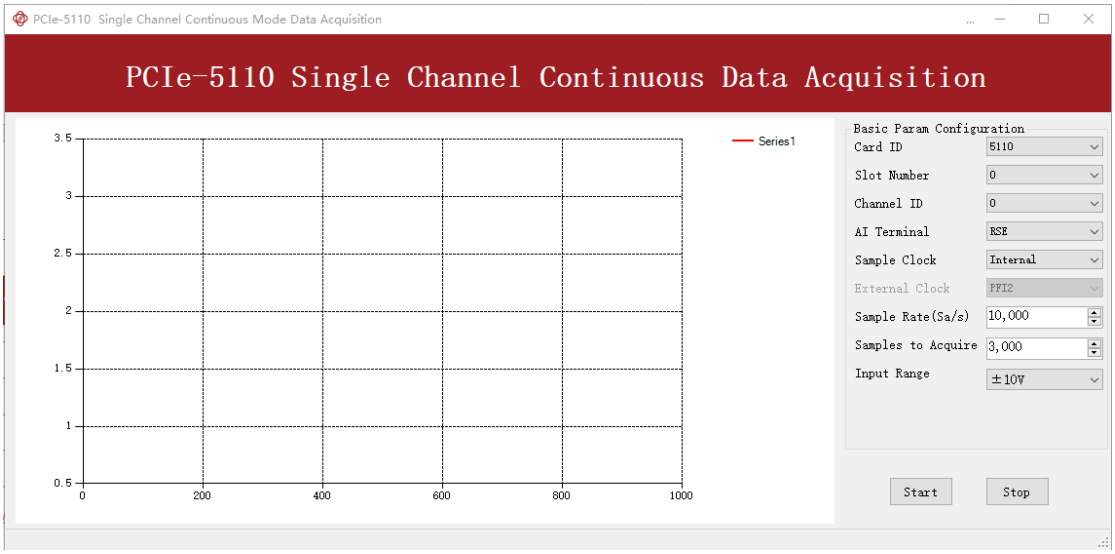


Figure 45 AI Continuous Paraments

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

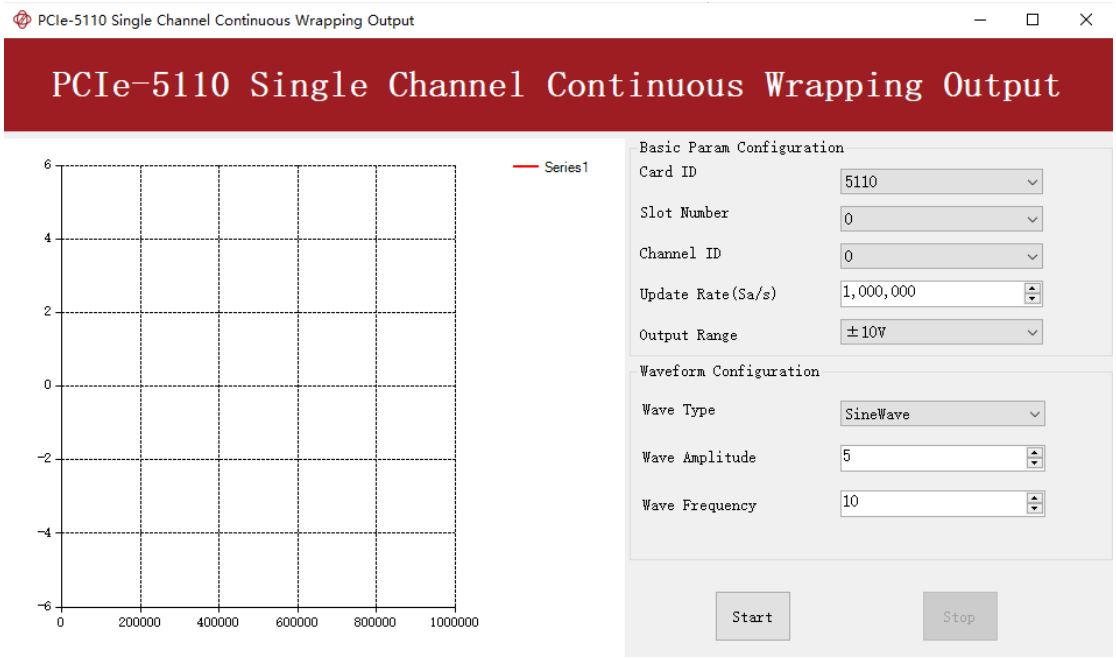


Figure 46 AO Continuous Wrapping Paraments

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

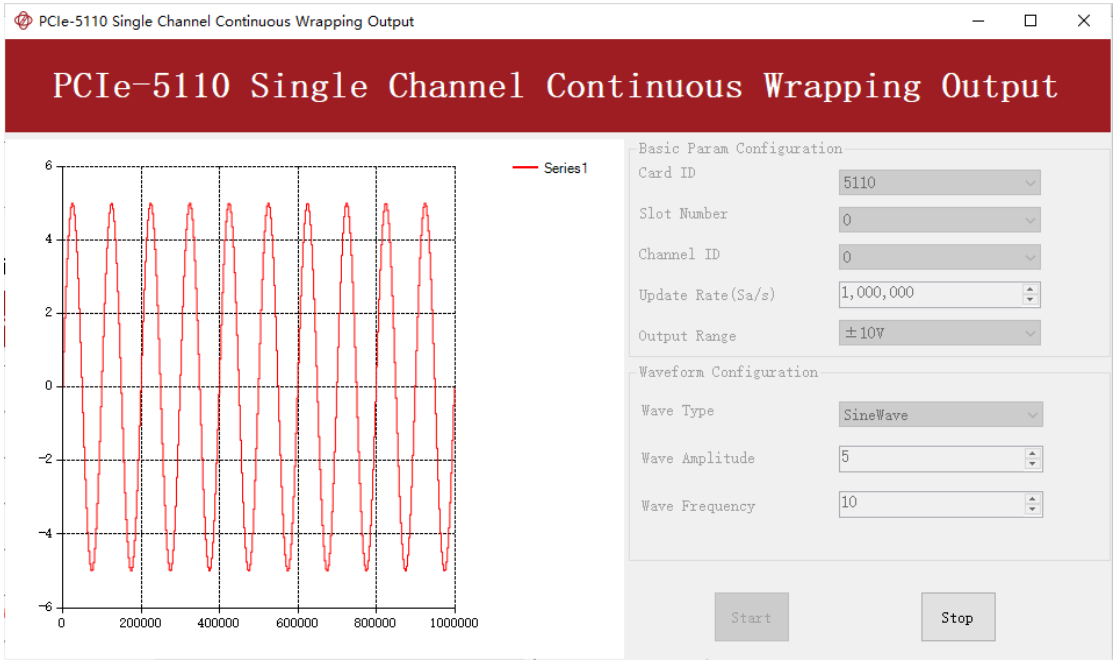


Figure 47 AO Continuous Wrapping Signal

- And the received signal is shown below.

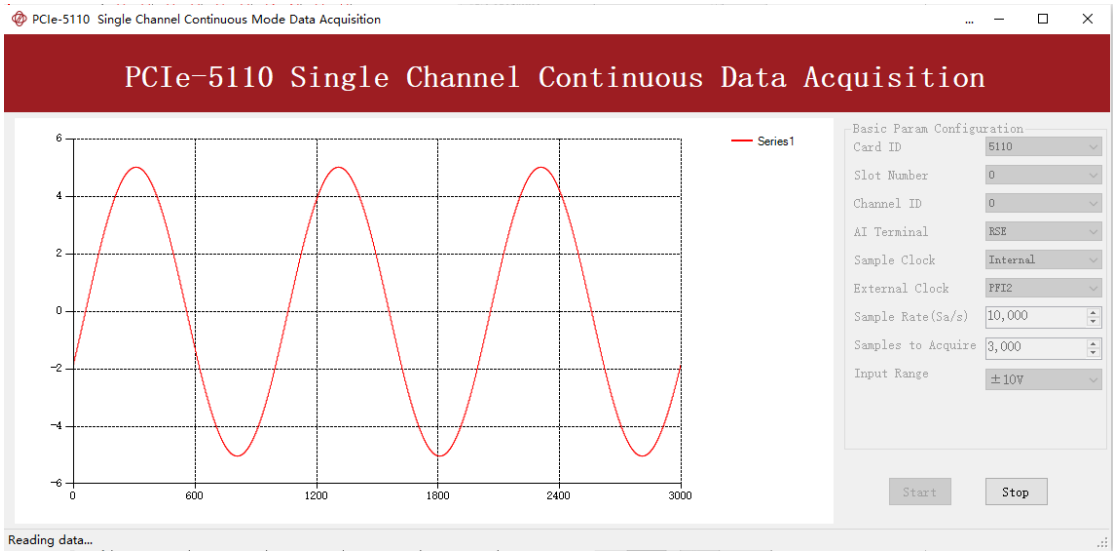


Figure 48 AI Acquisition AO Signal

➢ The analog signal is successfully generated and received by PCIe-5111.

9.7. Digital I/O Operations

The PCIe/PXIe-5113 provides powerful programmable digital I/O functions.

9.7.1. Static DI/DO

Programmable I/O supports static TTL, 16 digital I/O channels. User can access these I/O information through software polling.

Learn by Example 9.7.1

- In this example PCIe-5111 outputs a digital signal by its DO function and reads it back by its DI function.
- Connect Connector1 of PCIe-5111 to the TB-68 terminal block according to Figure 4.
- Connect Port 1/Line 0~7 (P1.0~P1.7) to Port 2/Line 0~7 (P2.1~2.7).PCIe-5111 sends a digital signal through Port 1 and reads the signal back from Port 2.
- Open the first program **Digital Output-->Winform DO SinglePoint.**

- Select **port 1** for **Digital Output**, Set Line 1,3,5 in High-Level positions, make sure all other lines are in Low-Level positions. Click **Start** to generate the High-Levels as shown.
- Open the second program **Digital Input-->Winform DI SinglePoint**.
- Select **port 2** for **Digital Input** as shown, and click **Check DI Status**. The result is shown below.

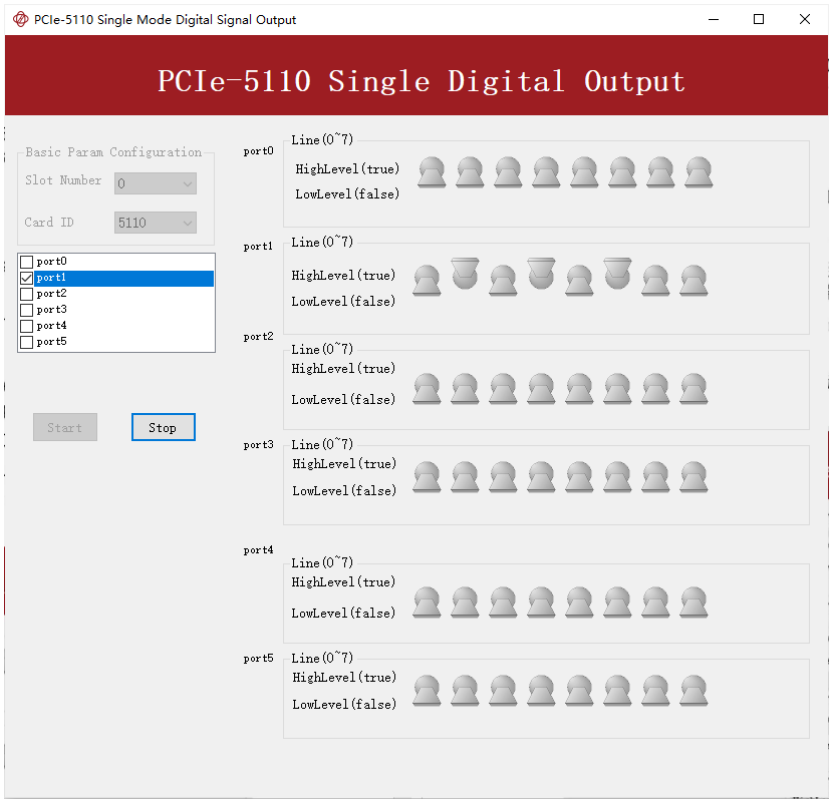


Figure 49 Single Digital Output

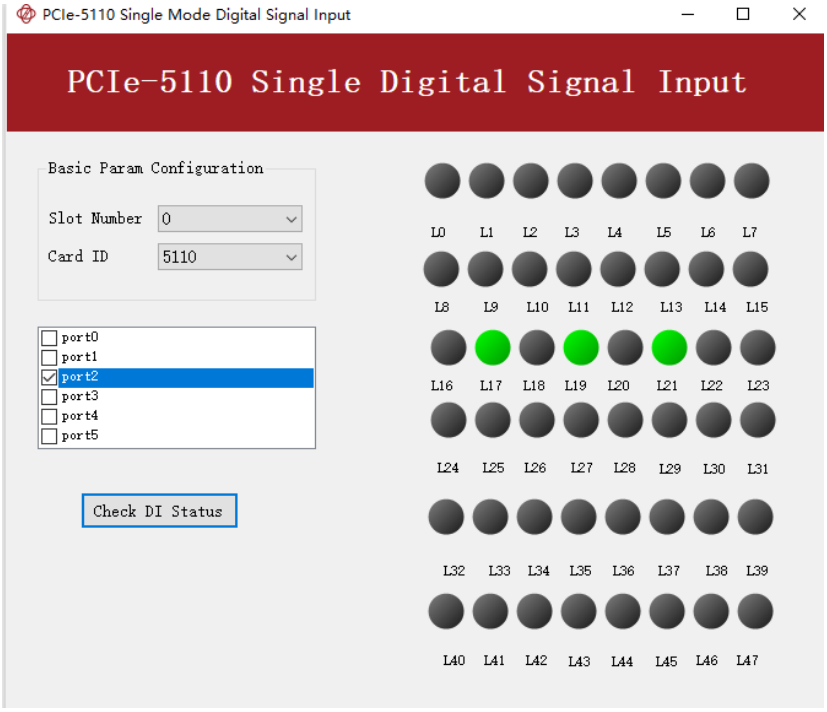


Figure 50 Single Digital Input

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

9.7.2. Dynamic DI/DO

The PCIe/PXIe-5113 supports both dynamic DI/DO operation with a maximum sample rate (update rate) of up to 10MHz. User can acquire or output digital waveforms in this way.

Learn by Example 9.7.2

- In this example PCIe-5111 outputs a squarewave by its DO function and reads it back by its DI function.
- Connect Connector1 of PCIe-5111
- Connect PCIe-5111 Port 1/Line 0 (P1.0,pin#52) to Port 2/Line 0(P2.0,pin #11)
- PCIe-5111 sends digital signals through Port 1/Port 0 and reads them back from Port 2/Port 1
- Open **Digital Input-->Winform DI Continuous** and set the numbers as shown. Select **port 2** or **port 1**.

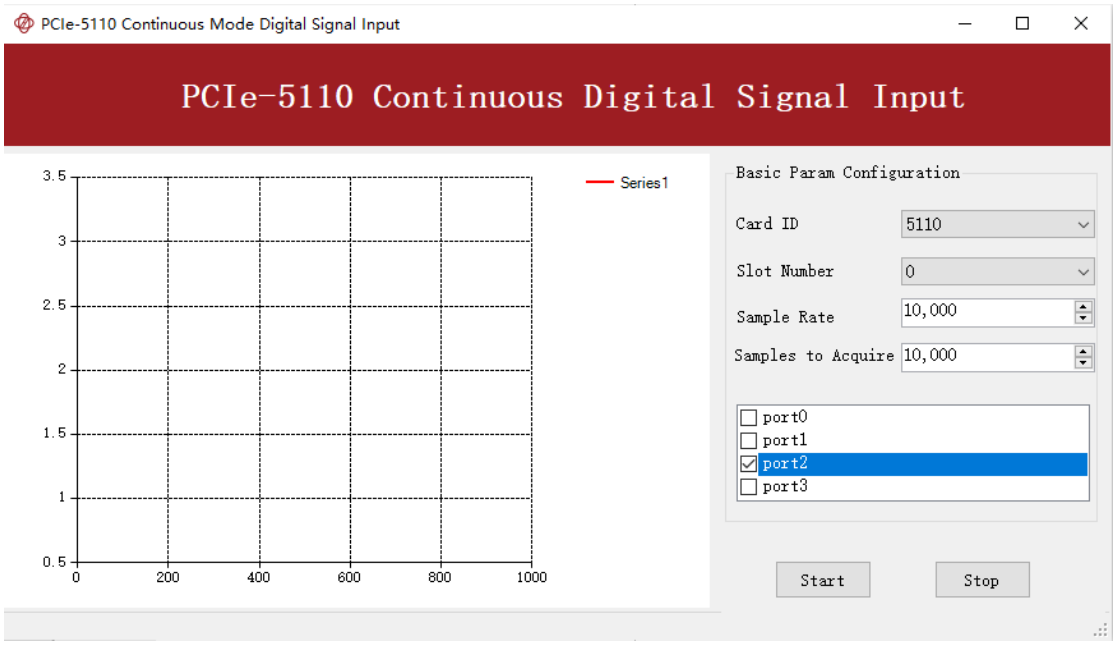


Figure 51 DI Continuous Paraments

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

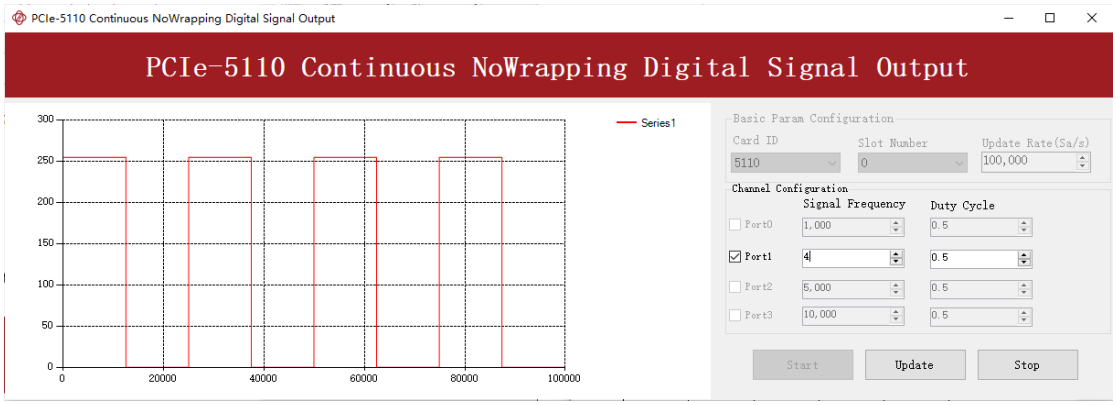


Figure 52 DO ContinuousNoWrapping Output

- In program **Winform DI Continuous**, you can see the acquired signal. Select **port 1** or **port 0**.

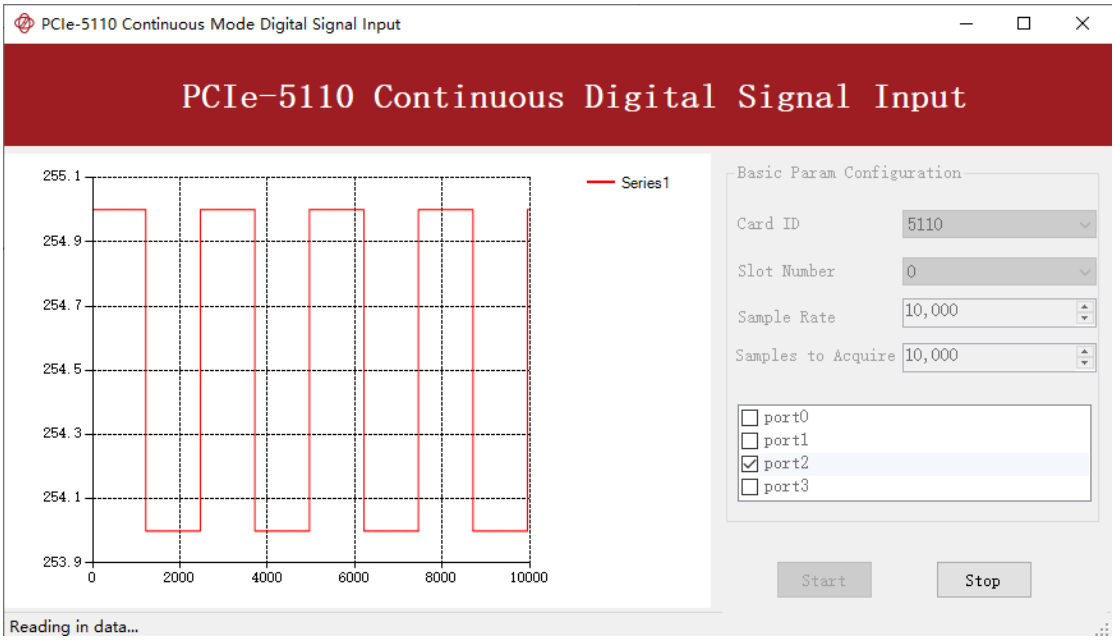


Figure 53 DI Continuous Acquisition

- The digital signal is successfully generated and acquired by PCIe-5111.

9.8. Counter Input Operations

The PCIe/PXIe-5113 has four or two identical 32 bits timers/counters.

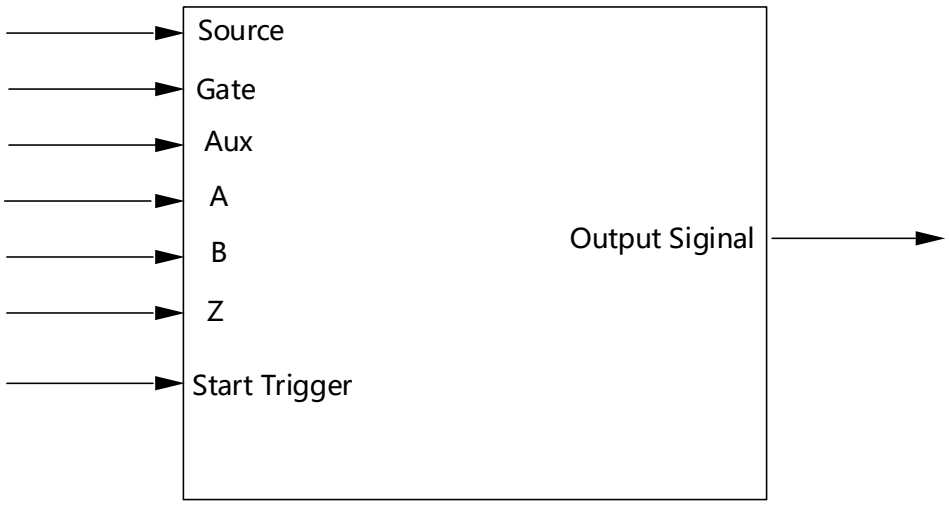


Figure 54 Counter Terminal

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

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

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

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

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

Figure 55 Counter Signal Wiring Instruction

9.8.1. Edge Counting

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

Timing

1) Single Mode

The count value is written to the register on each rising edge or falling edge of the signal to measure as shown in Figure 56.

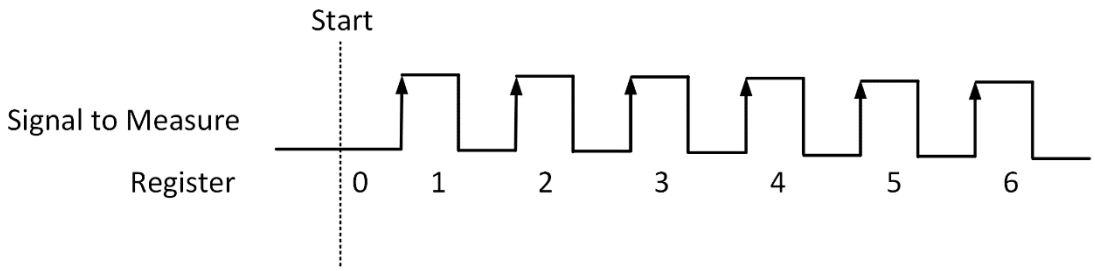


Figure 56 Simple Edge Counting in Single Mode

2) Finite/Continuous Mode with Internal Sample Clock

The count value is stored into the buffer on each rising edge or falling edge of the sample clock as shown in Figure 57.

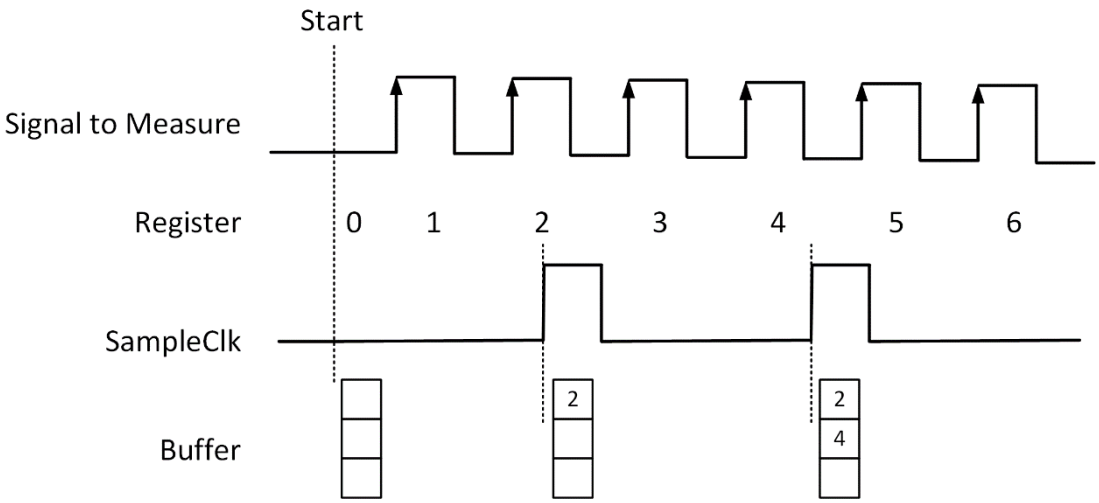


Figure 57 Buffered Edge Counting with Internal Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

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

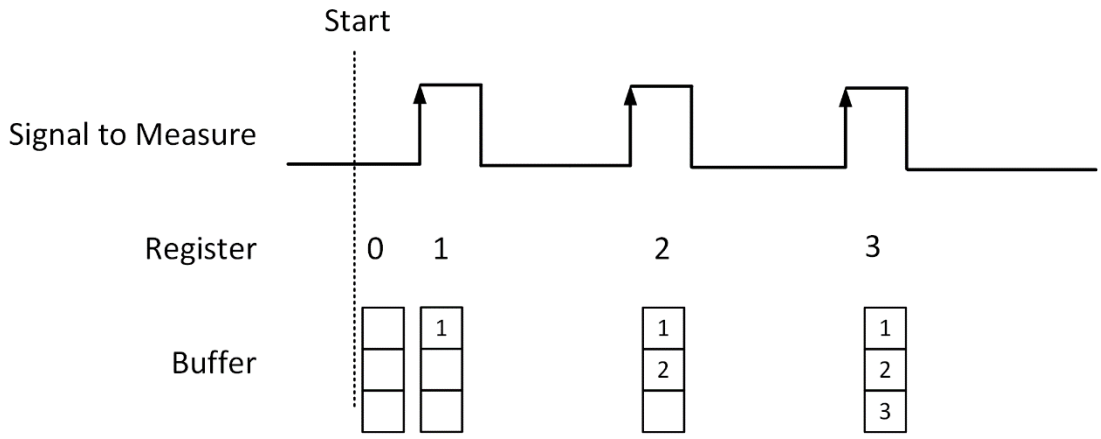


Figure 58 Simple Edge Counting with Implicit SampleClk

Counting Direction

User can control the counting direction through software configuration or by an input signal with Gate terminal. When using an input signal to control the counting direction, the counter counts up when the signal is high and counts down when the signal is low as shown in Figure 59.

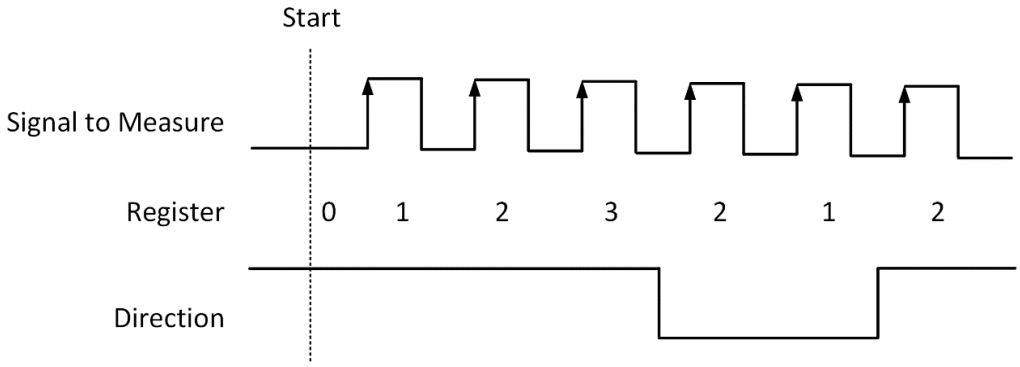


Figure 59 Count Direction

Learn by Examples9.8.1

- Connect the signal source's positive terminal of a signal source to PCIe-5111 counter0's edge counting source (CTR0_Source/A, Pin#11), negative terminal to the ground (DGND, Pin#44) as shown in Figure 3 and Figure 4. (CTR0_Source, DGND) consists of an edge counting counter input and they share the same ground.
- Set a squarewave signal (f=1Hz, Vpp=5V).

Single Mode

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

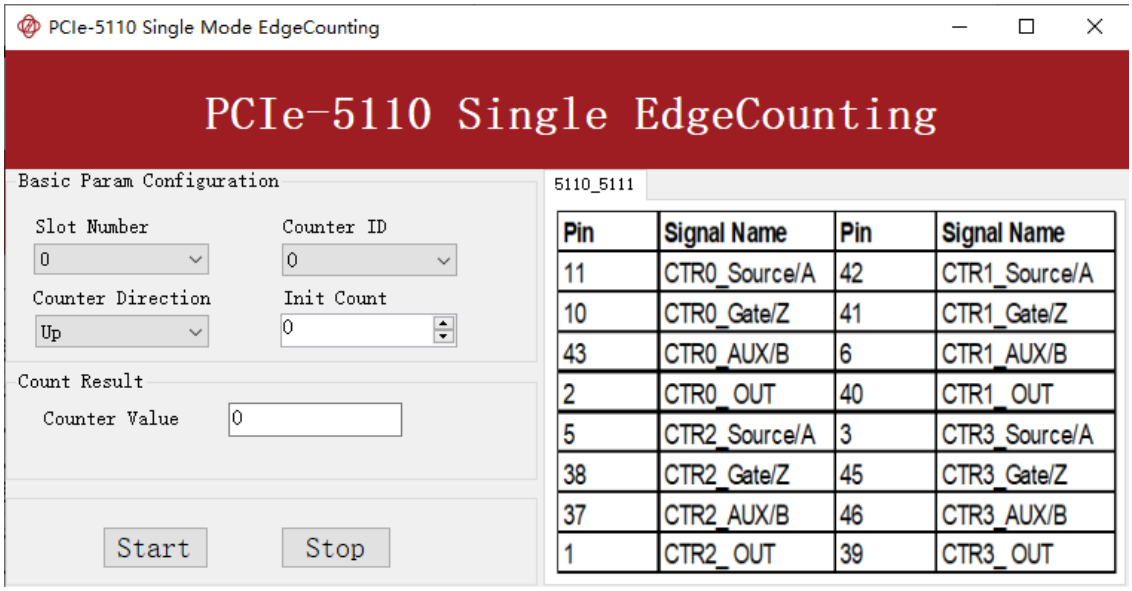


Figure 60 EdgeCounting For Single Mode

- Counter Direction is set by **Counter Direction**.
- The table in the sample program is a connection diagram for your convenience.

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

Finite/Continuous Mode

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

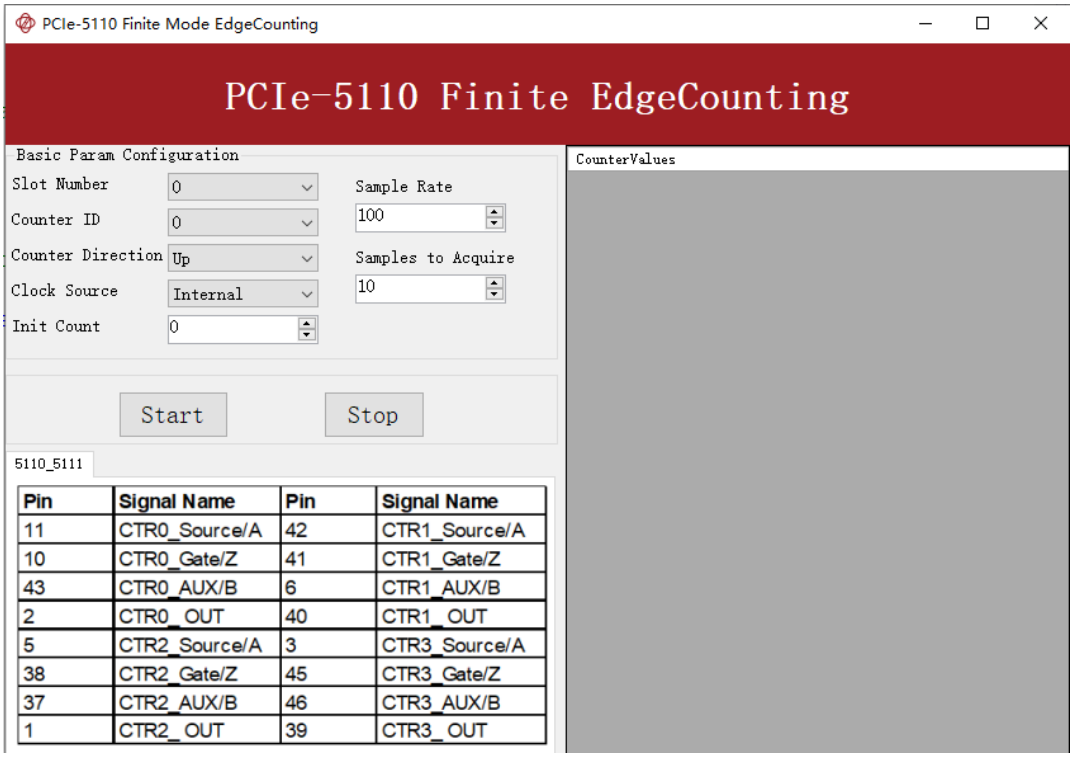


Figure 61 EdgeCounting For Finite Mode

- The table in the sample program is a connection diagram for your convenience.
- Counter Direction is set by **Counter Direction**.
- There are two clock sources in PCIe-5111 *Internal* and *Implicit*: This example uses **Internal** mode set by **Clock Source**.

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

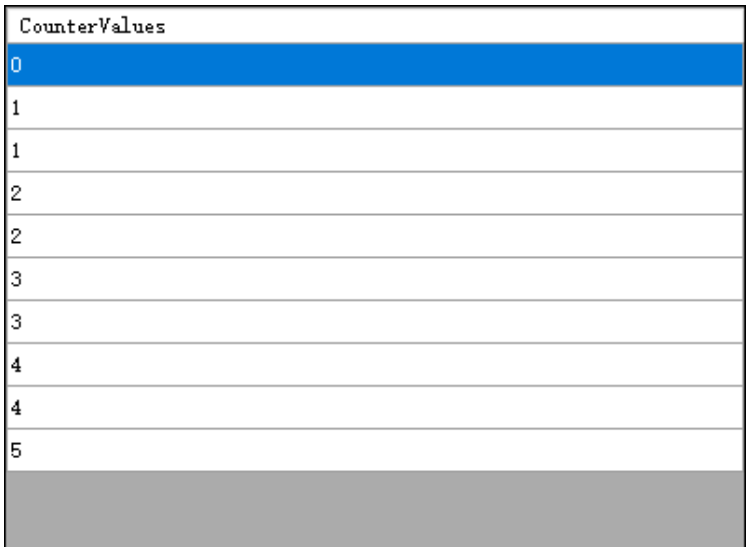


Figure 62 Counter Values For Internal Clock

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

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

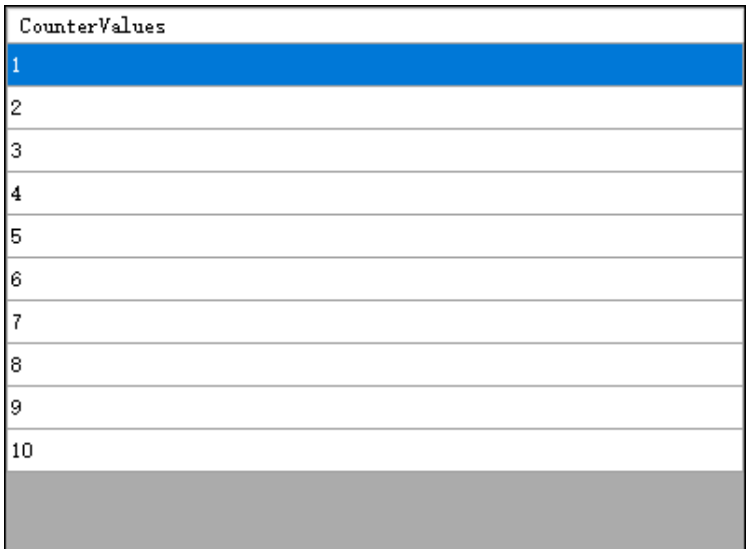


Figure 63 Counter Values For Implicit Clock

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

9.8.2. Pulse Measurement

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

Timing

1) Single Mode

The count value of the duration of the high-level or low-level is written to the register on each rising or falling edge of the pulse to measure, as shown in Figure 64.

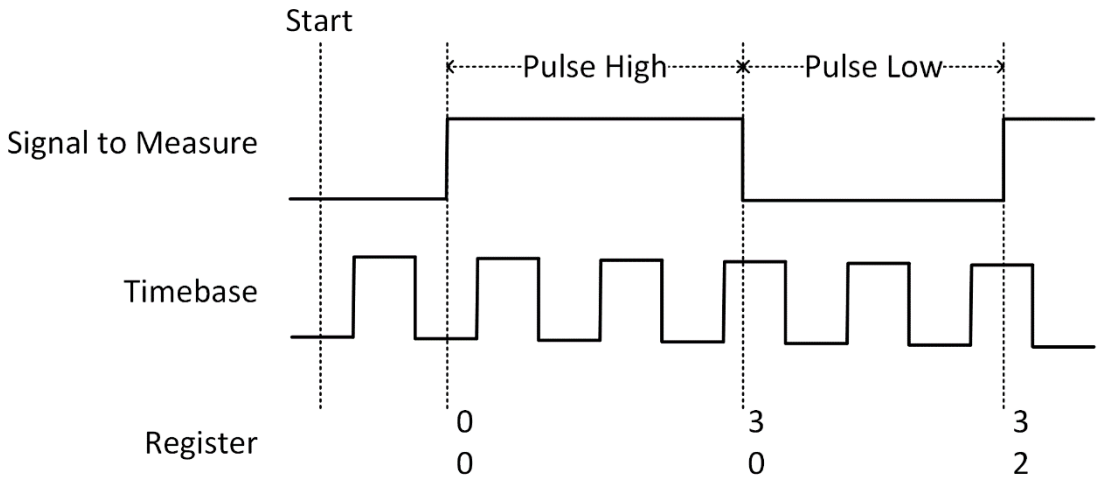


Figure 64 Pulse Measurement in Single Mode

2) Finite/Continuous Mode with Internal Sample Clock

The count value of the duration of the high or low level is stored into the buffer on each rising or falling edge of the sample clock, as shown in Figure 65.

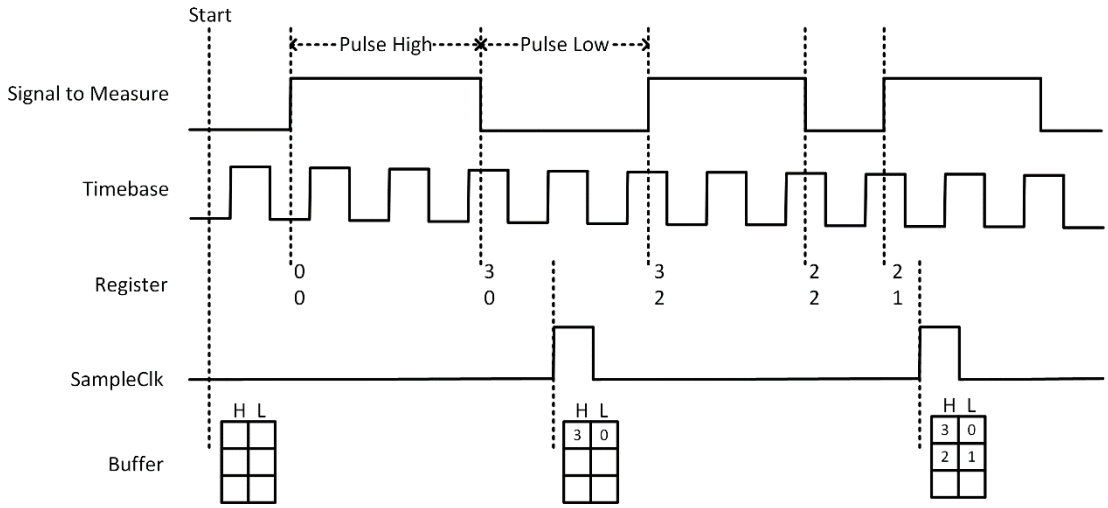


Figure 65 Pulse Measurement with Internal SampleClk

3) Finite/Continuous Mode with Implicit Sample Clock

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

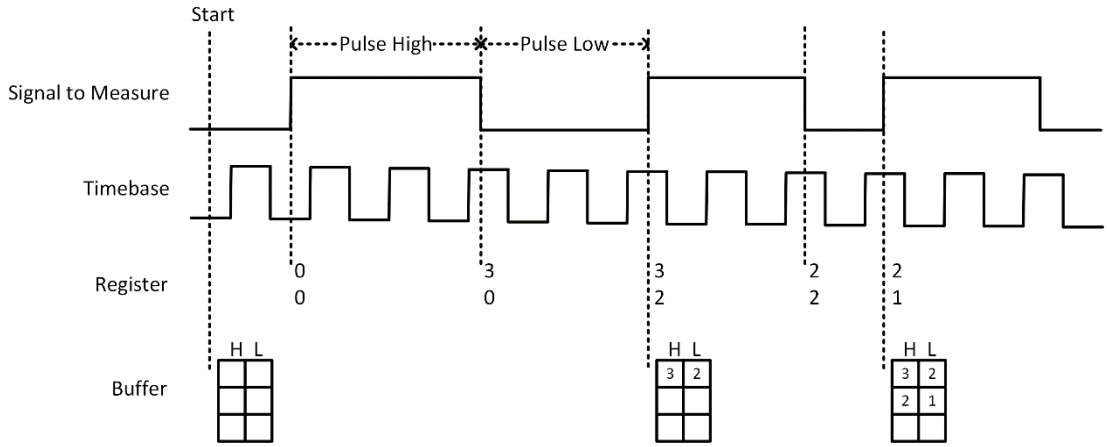


Figure 66 Pulse Measurement with Implicit SampleClk

Learn by Examples 9.8.2

- Connect the signal source's positive terminal to PCIe-5111 counter0's pulse measure source (CTR0_Gate/Z, Pin#10), negative terminal to the ground (DGND, Pin#44) as shown in Figure 3 and Figure 4. (CTR0_Gate/Z, DGND) consists of a pulse measure counter input and they share the same ground.

  - Set a squarewave signal (f=1Hz, Duty Cycle=50%, Vpp=5V).

Single Mode

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

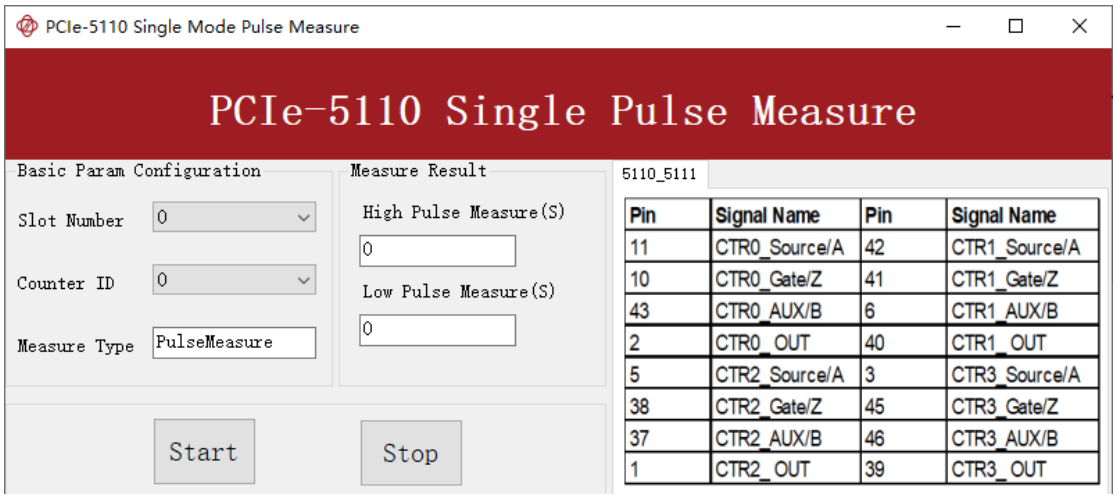


Figure 67 Pulse Measure For Single Mode

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

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

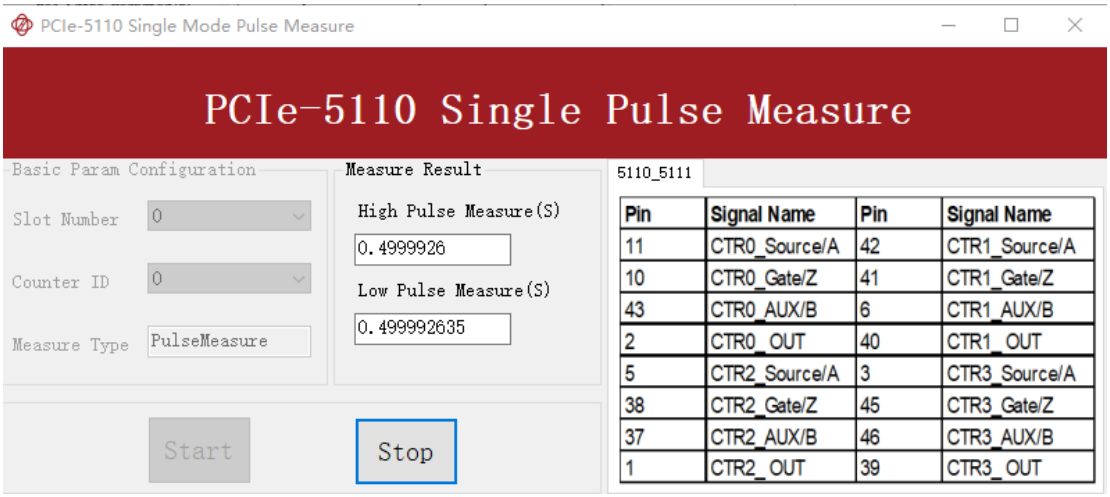


Figure 68 Pulse Measure Value For Single Mode

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

Finite/Continuous Mode

■ Change the frequency of Squarewave to 50 Hz.

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

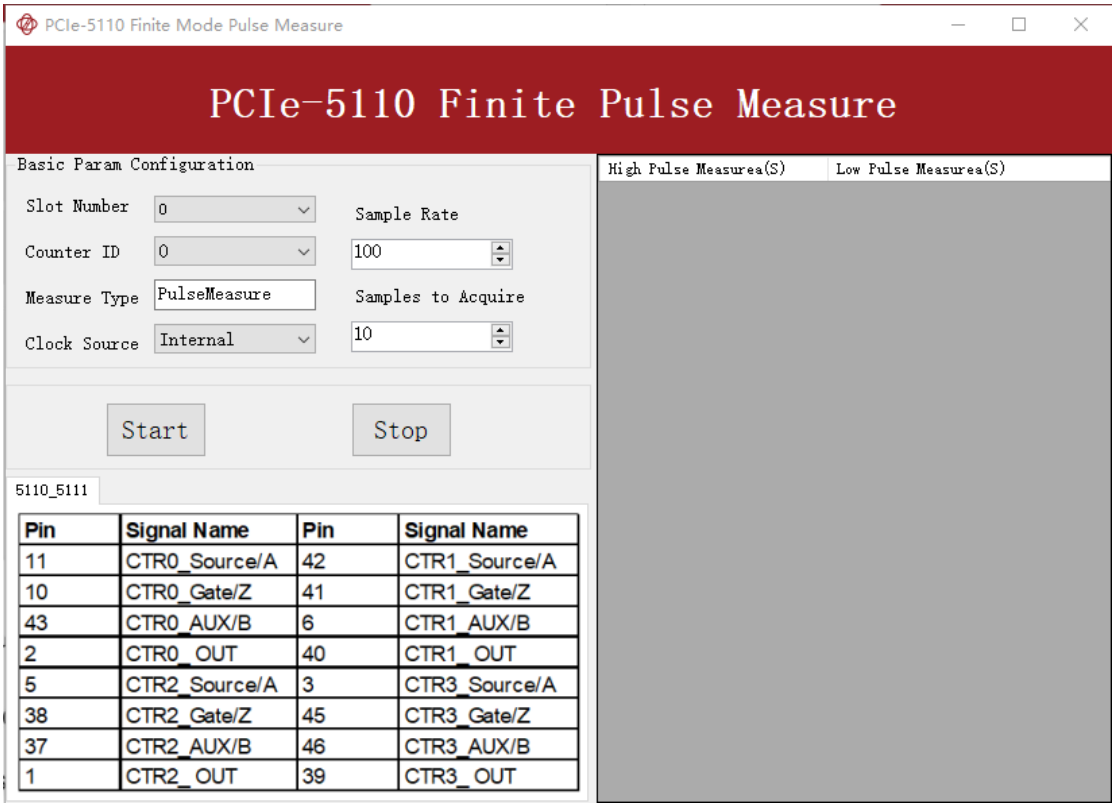


Figure 69 Pulse Measure For Finite Mode

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

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

High Pulse Measurea(S)	Low Pulse Measurea(S)
0	0
0.009999835	0
0.009999835	0.009999865
0.009999835	0.009999865
0.009999835	0.00999986
0.009999835	0.00999986
0.009999835	0.009999865
0.009999835	0.009999865
0.009999835	0.009999865
0.009999835	0.009999865

Figure 70 Pulse Measure Values For Finite Mode

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

- Please refer to **Learn by Examples9.8.1 Finite/Continuous** Mode about the difference between Internal and Implicit.

9.8.3. Frequency Measurement

The counter measures the frequency of signal to measure.

Timing

1) Single Mode

Frequency Measurement without sample clock is actually using Pulse Width Measurement internally, refer to chapter 9.8.2 for more information.

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

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

To configure the counter to work in this mode, set JY5113CITask.Mode to CIMode.Single.

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

Between every two rising edges of the sample clock, the counter counts the numbers of full periods ($T1$) of the signal to measure, and the number of rising edges of timebase ($T2$) during those full periods. These two values are stored into the buffer on each rising edge of the sample clock, as shown in Figure 71.

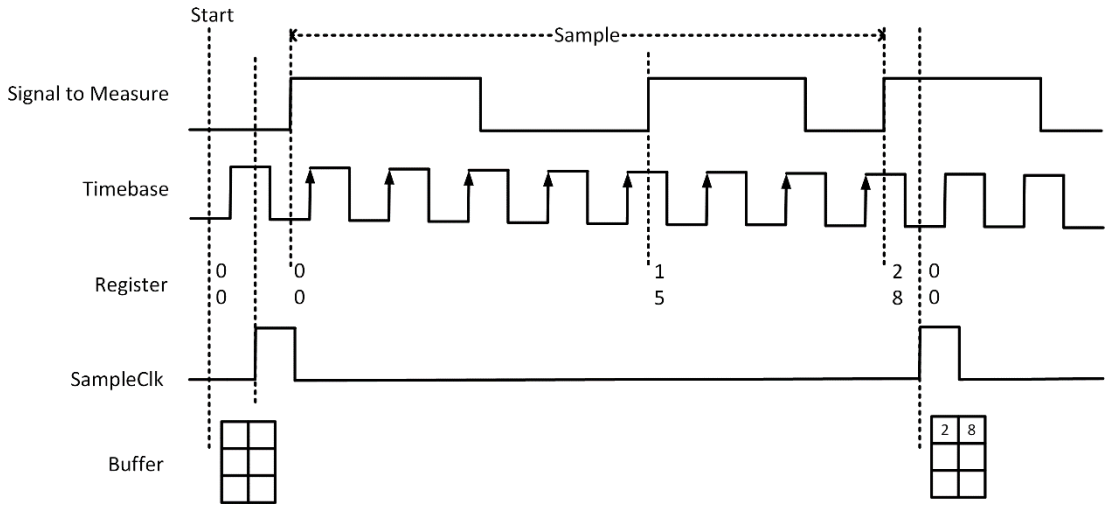


Figure 71 Frequency Measurement with Internal Sample Clock

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

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

3) Finite/Continuous Mode with Implicit Sample Clock

Frequency Measurement with implicit sample clock is actually using Pulse Measurement internally. refer to chapter 9.8.2 for more information.

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

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

Learn by Examples 9.8.3

- Connect the signal source's positive terminal to PCIe-5111 counter0's frequency measure source (CTR0_Gate/Z, Pin#10), negative terminal to the ground (DGND,

Pin#44) as shown in Figure 3 and Figure 4. (CTR0_Gate/Z, DGND) consists of a frequency measure counter input and they share the same ground.

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

Single Mode

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

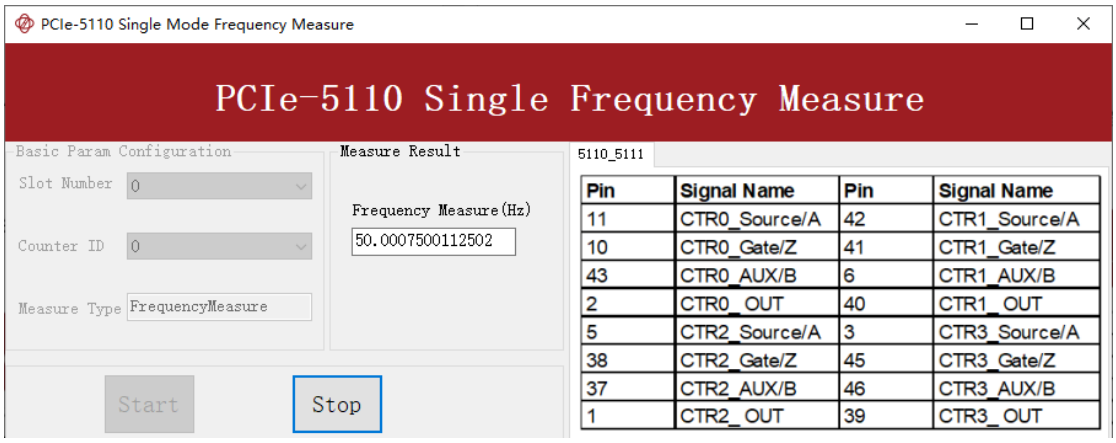


Figure 72 Frequency Measure For Single Mode

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

Finite/Continuous Mode

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

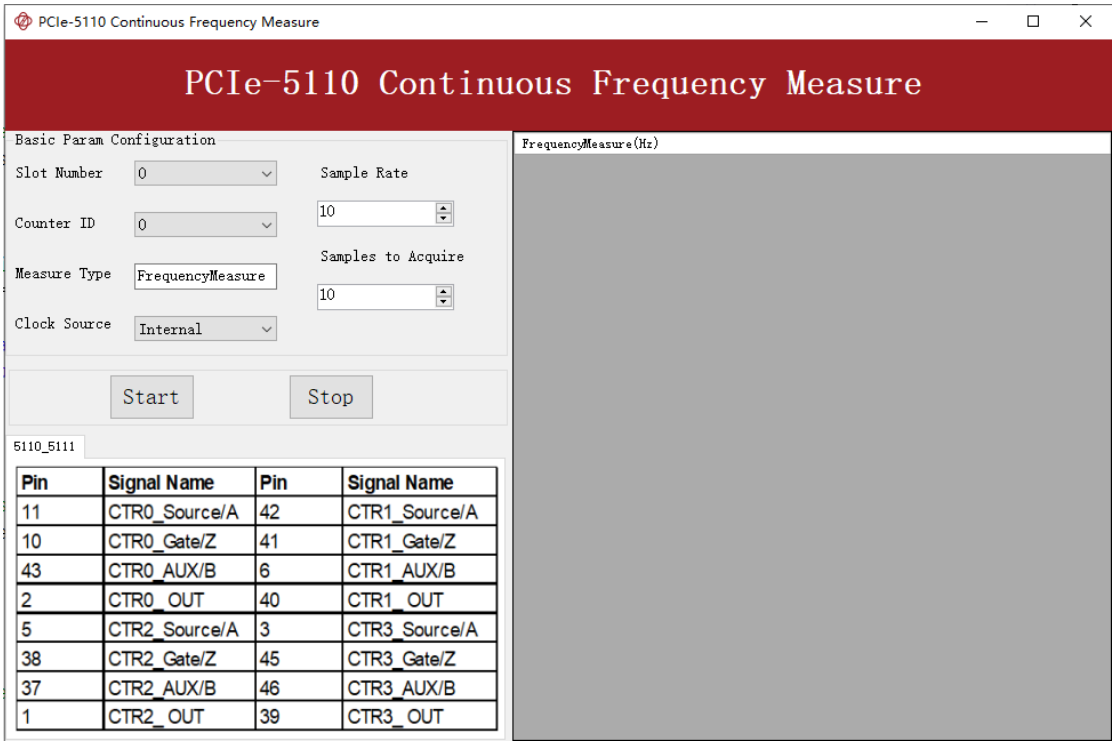


Figure 73 Frequency Measure For Continuous Mode

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

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

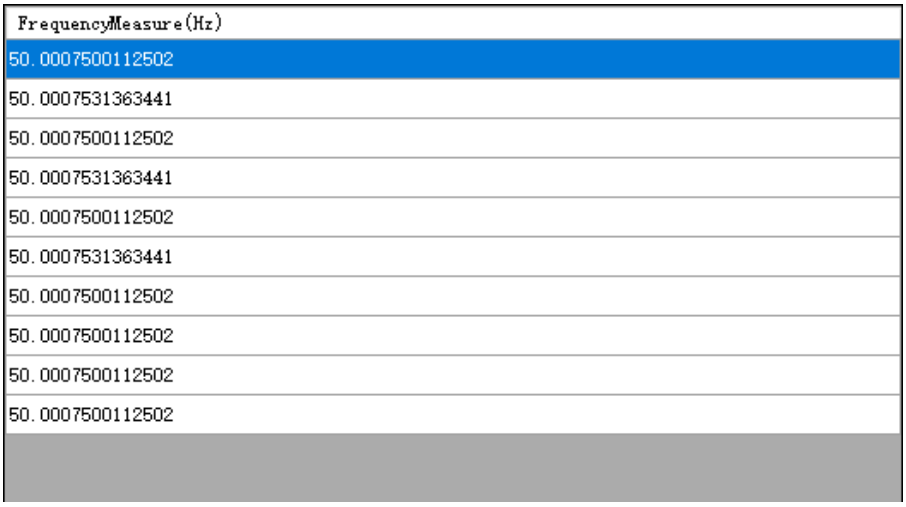


Figure 74 Frequency Measure Values

9.8.4. Period Measurement

The counter measures the period of signal to measure. Period Measurements is using Frequency Measurement internally and returns the inverse result of Frequency Measurement. Refer to chapter 9.8.3 for more information.

Learn by Examples 9.8.4

- Connect the signal source's positive terminal to PCIe-5111 counter0's period measure source (CTR0_Gate/Z, Pin#10), negative terminal to the ground (DGND, Pin#44) as shown in Figure 3 and Figure 4. (CTR0_Gate/Z, DGND) consists of a period measure counter input and share the same ground.
- Set a squarewave signal (f=200Hz, Duty Cycle=50%, Vpp=5V).

Single Mode

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

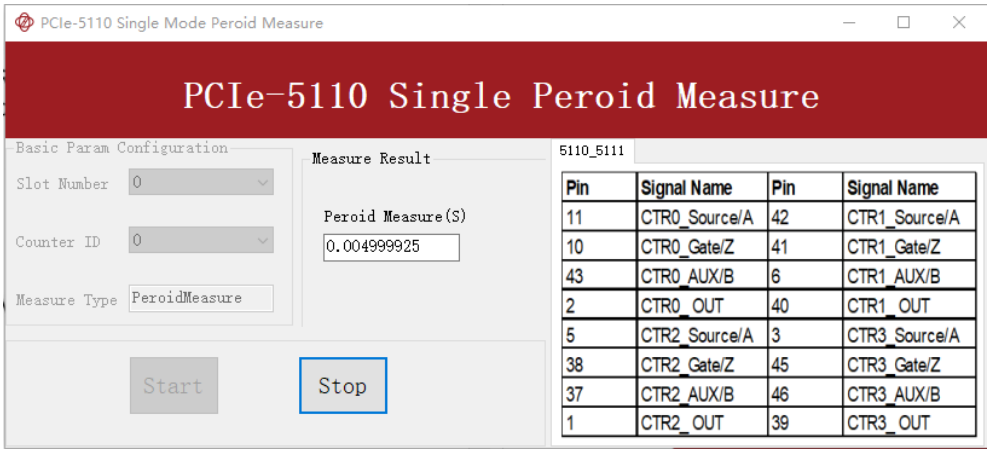


Figure 75 Peroid Measure For Single Mode

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

Finite/Continuous Mode

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

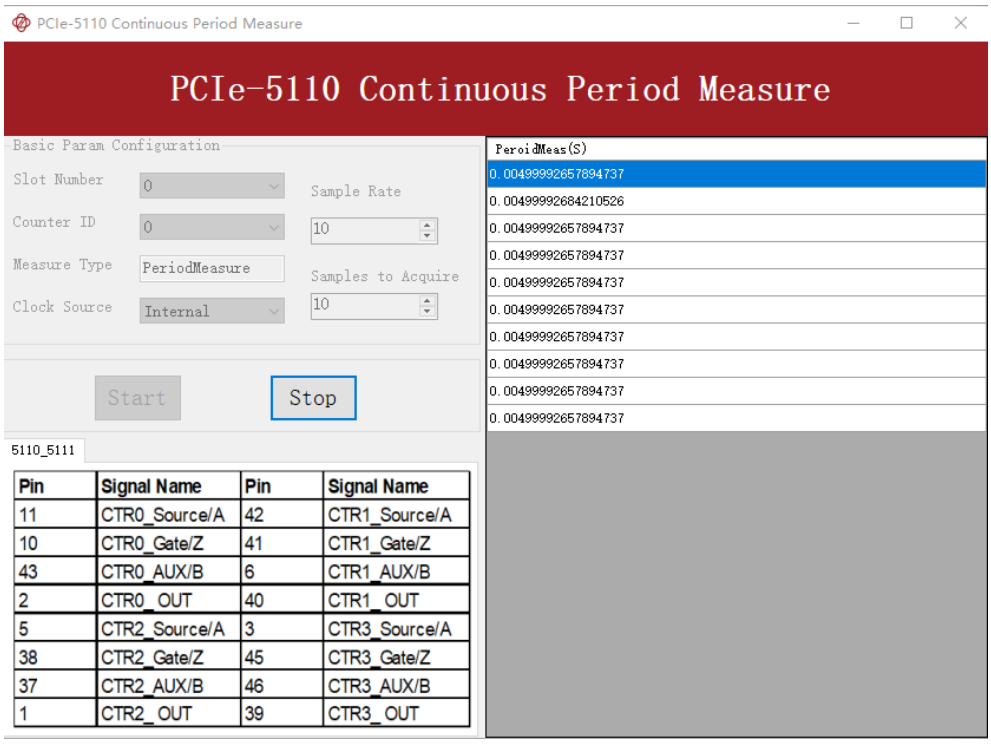


Figure 76 Peroid Measure For Continuous Mode

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

9.8.5. Two-Edge Separation

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

Timing

1) Single Mode

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

The number of rising edges of timebase between previous rising edge of the second signal and current rising edge of the first signal is written to the register on each rising edge of the first signal as shown in Figure 77.

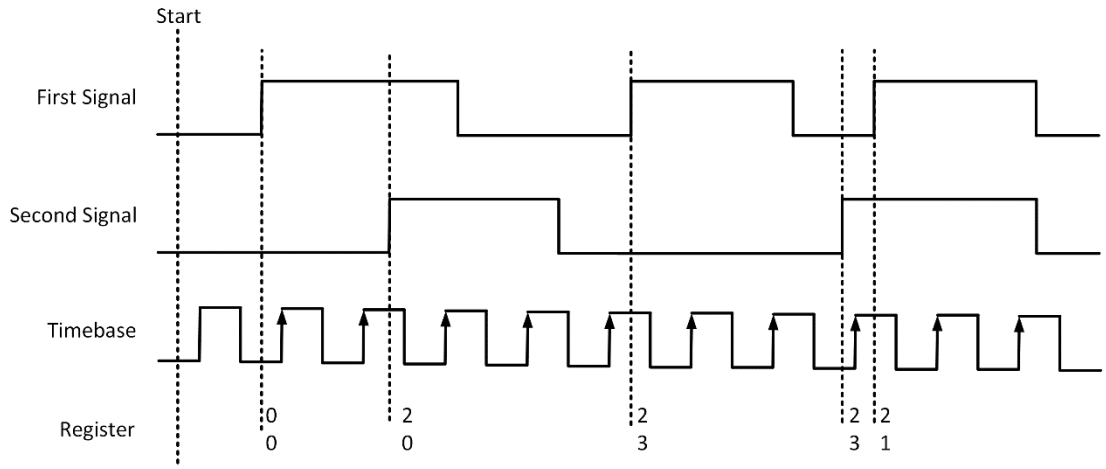


Figure 77 Two-Edge Separation in Single Mode

2) Finite/Continuous Mode with Internal Sample Clock:

The count values of rising edges of timebase between first signal and second signal are stored into buffer on each rising edge of the sample clock, as shown in Figure 78.

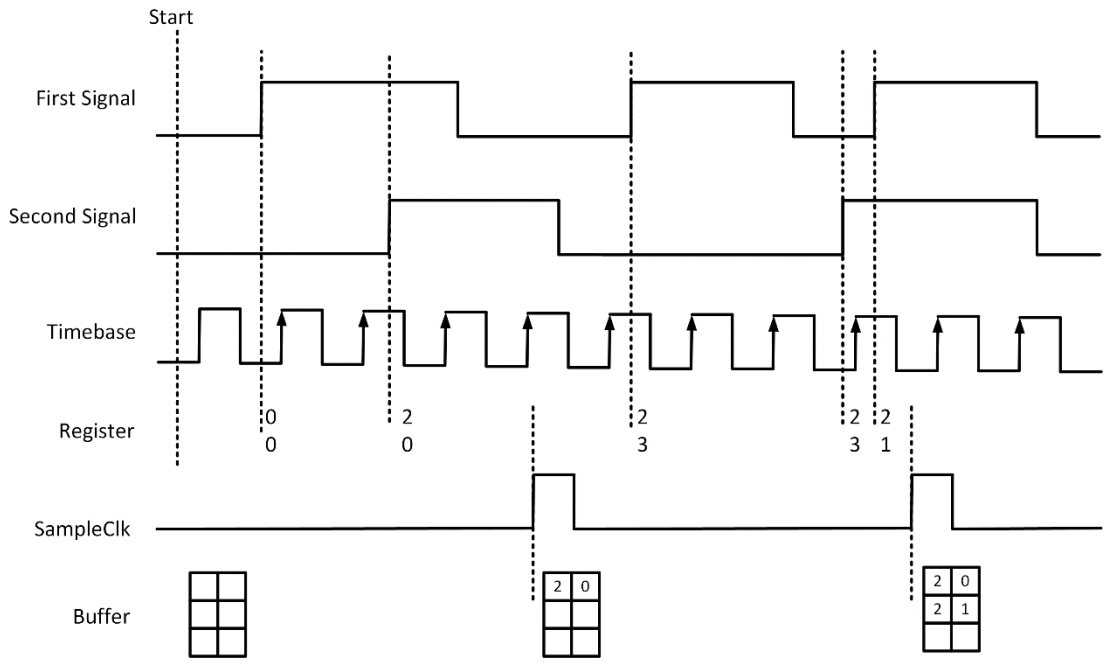


Figure 78 Two-Edge Separation with Internal Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

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

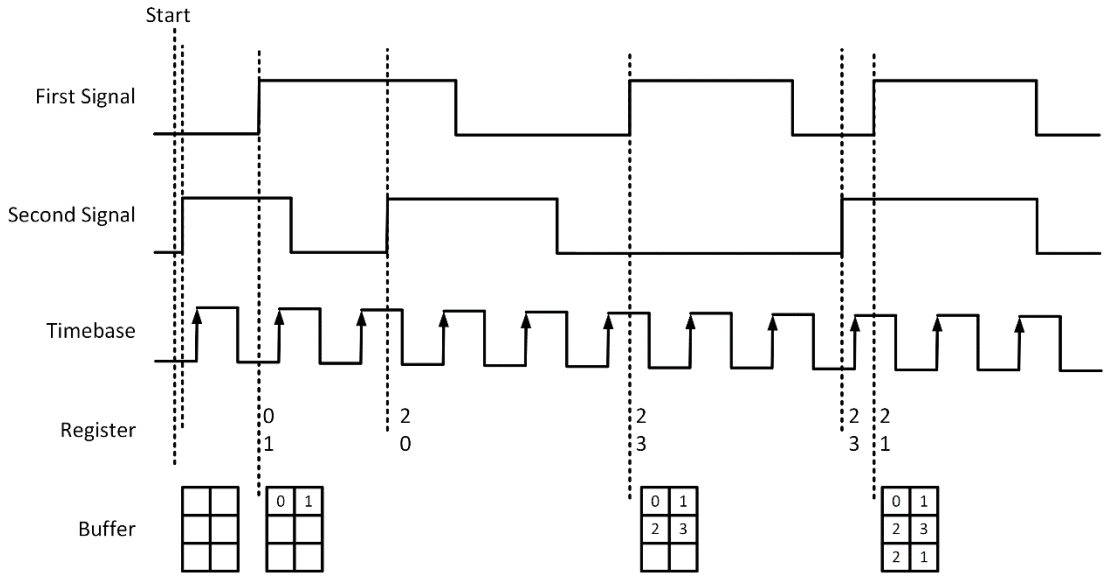


Figure 79 Two-Edge Separation with Implicit Sample Clock

Learn by Examples 9.8.5

- Connect the signal source's two positive terminals to PCIe-5111 first signal input (squarewave, CTR0_Gate/Z, Pin #10) and second signal input (squarewave, CTR0_AUX/B, Pin#43), two negative terminals to the ground (DGND, Pin#44) and (D_GND, Pin#9) as shown in Figure 3 and Figure 4.
- Set a squarewave signal (f=1Hz, Phase=0°) and a squarewave signal (f=1Hz, Phase=135°).

Single Mode

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

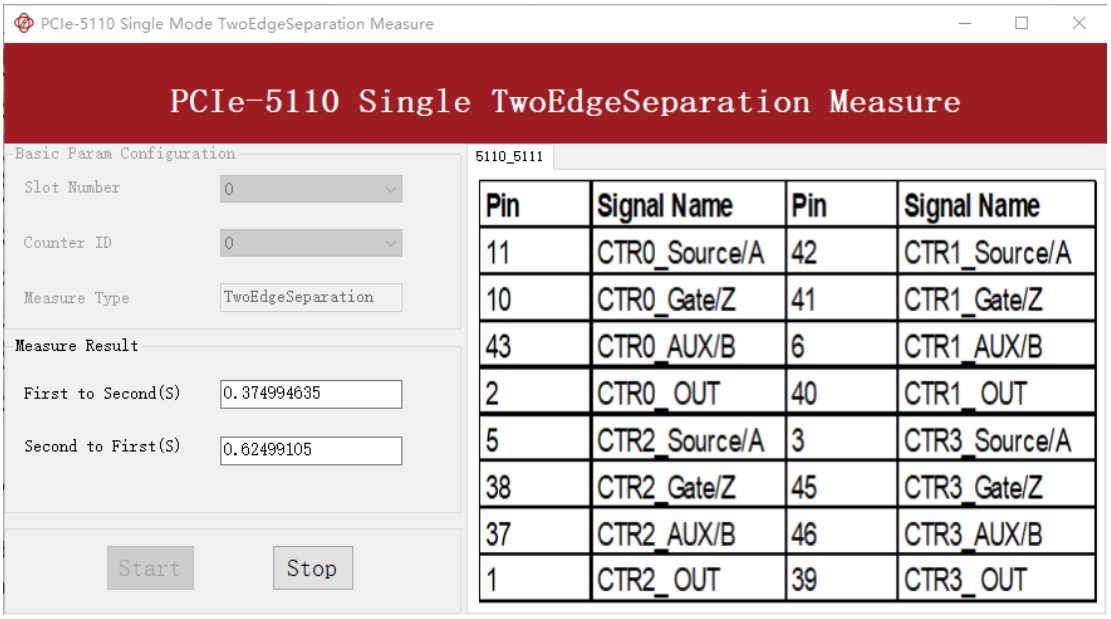


Figure 80 Two-EdgeSeparation Measure For Single Mode

- 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:

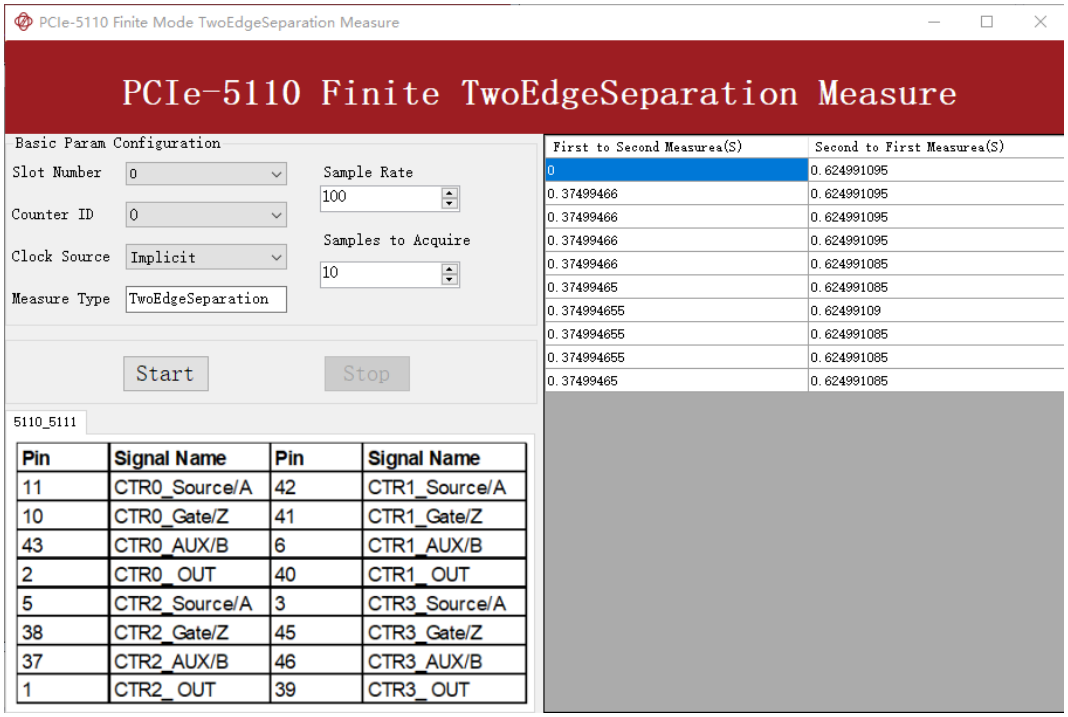


Figure 81 Two-EdgeSeparation Measure For Finite Mode

- 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.

9.8.6. Quadrature Encoder

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

Encoding Type

1) x1 Encoding

When A leads B, the count increase occurs on the rising edge of A; when B leads A, the count decrease occurs on the falling edge of A as shown in Figure 82.

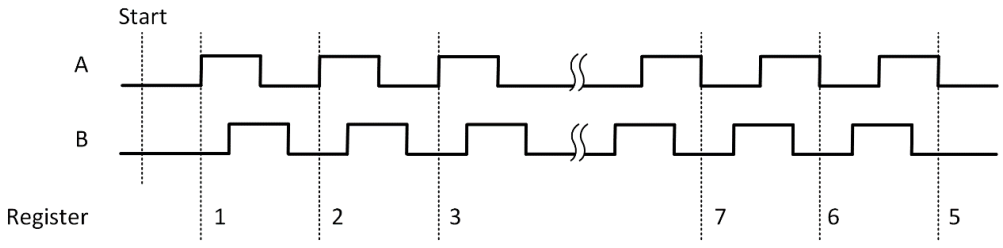


Figure 82 Quadrature Encoder x1 Mode

2) x2 Encoding

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

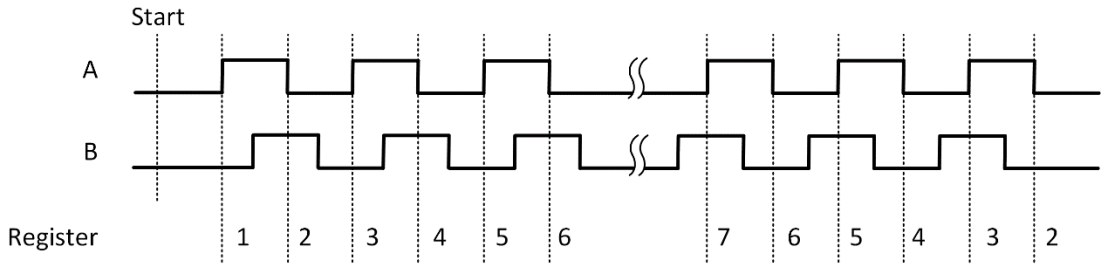


Figure 83 Quadrature Encoder x2 Mode

3) x4 Encoding

When A leads B, the increase of count occurs on the rising and falling edges of A and B. When B leads A, the decrease in count occurs on the rising and falling edges of A and B. As shown in Figure 84.

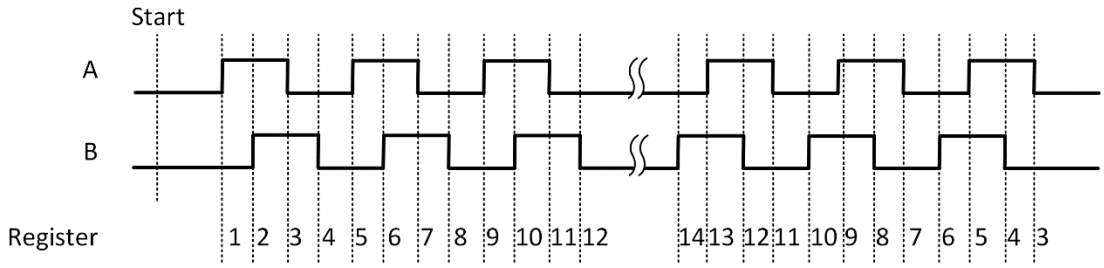


Figure 84 Quadrature Encoder x4 mode

Channel Z Behavior

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

Timing

Take Encoding x1 mode as an example.

1) Single Mode

The count value is written to the register on each rising edge of the signal A, as shown in Figure 56.

To configure the counter to work in this mode, set JY5113CITask. Mode to CIMode.Single.

2) Finite/Continuous Mode with Internal Sample Clock

The count value is stored into the buffer on each rising edge of the sample clock, as shown in Figure 85.

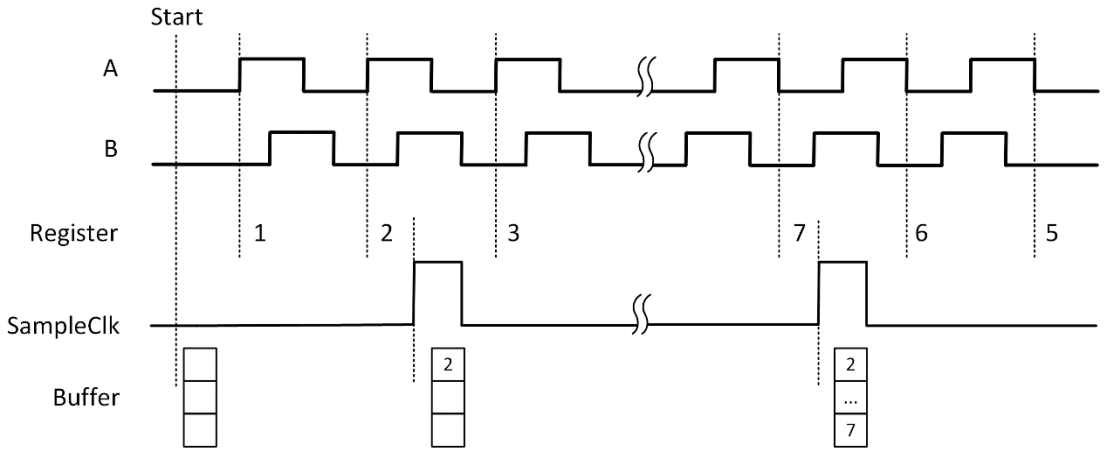


Figure 85 Quadrature Encoder x1 with Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

The count value is stored into the buffer every time it changes, as shown in Figure 86.

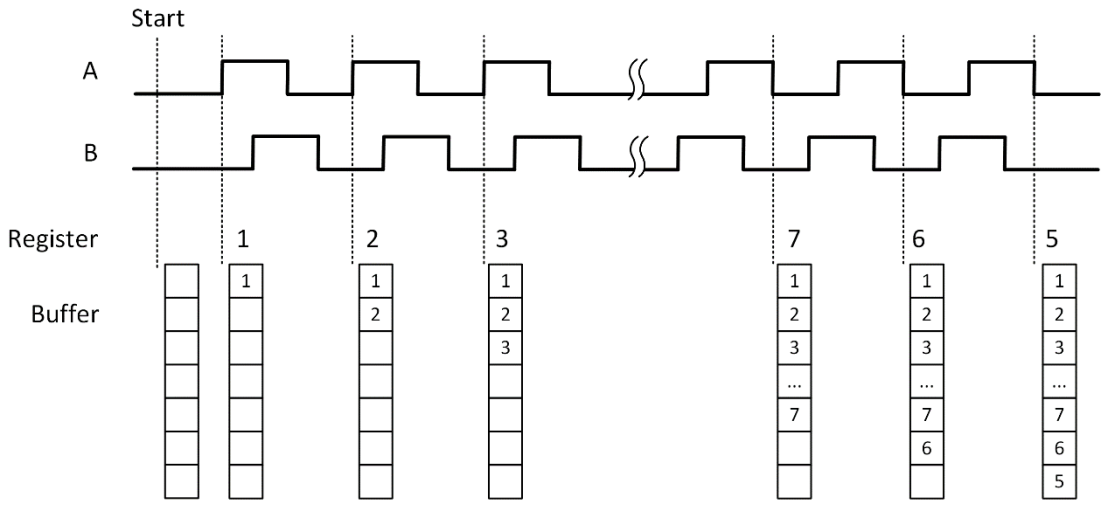


Figure 86 Quadrature Encoder x4 with Implicit Sample Clock

Learn by Examples 9.8.6

- Connect the signal source's two positive terminals to PCIe-5111 first signal input (sinewave, CTR0_Source/A, Pin #11) and second signal input (squarewave , CTR0_AUX/B, Pin#43), two negative terminals to the ground (DGND, Pin#44) and (D_GND, Pin#9) as shown in Figure 3 and Figure 4. (CTR0_Source/A, DGND) consists of the first signal to be measured; (CTR0_AUX/B, D_GND) consists of the second signal to be measured.
- Set a sqaurewave signal (f=10Hz, Phase=90°) and a squarewave signal (f=10Hz, Phase=0°).

Single Mode

- Open **Counter Input--> Winform CI Single QuadEncoder** and click **Start**. The result is shown below by **CounterValue** according to the counting rules explained in 9.8.6:

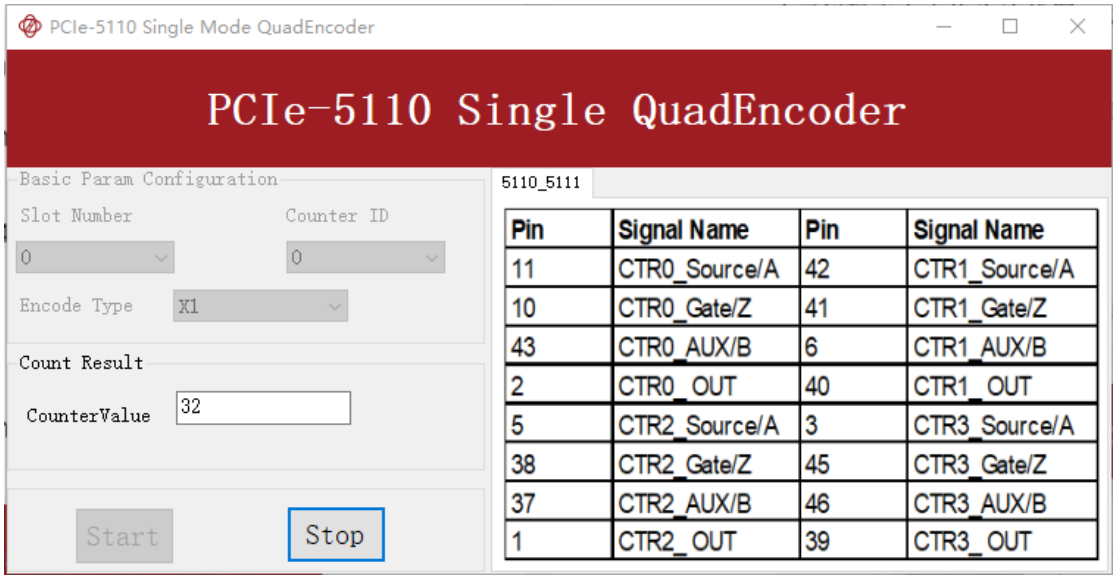
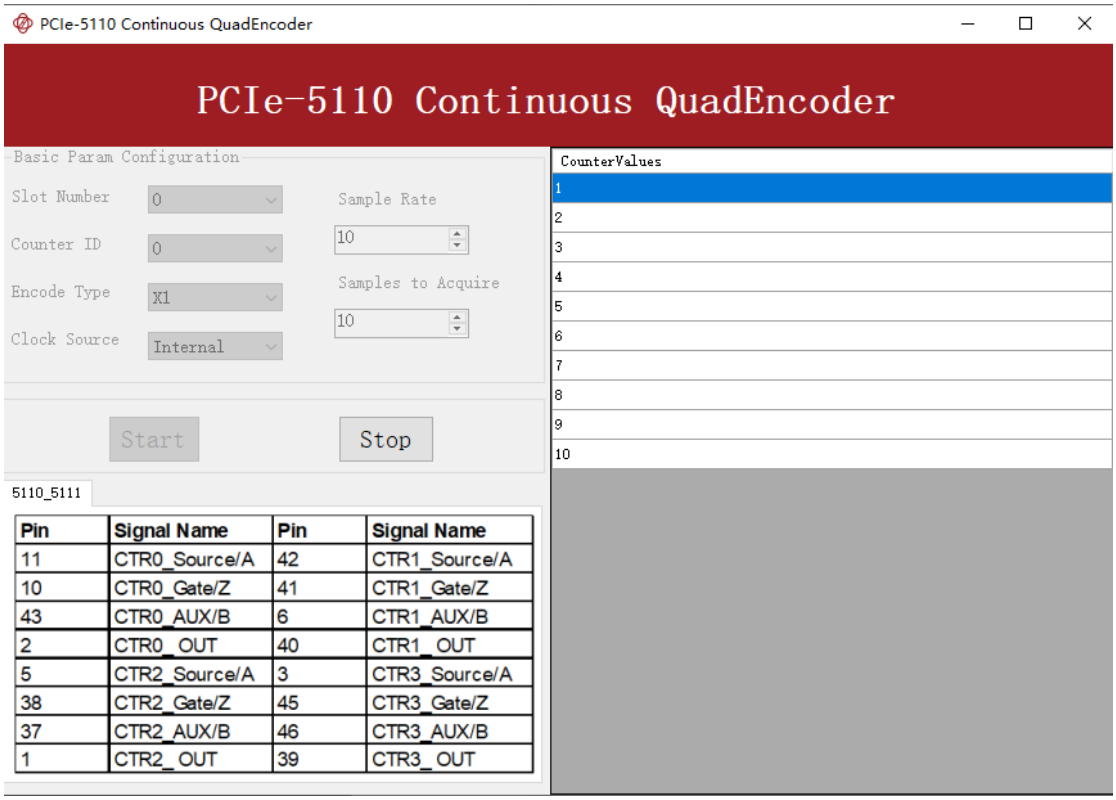


Figure 87 QuadEncoder For Single Mode

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

Continuous Mode

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



Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
2	CTR0_ OUT	40	CTR1_ OUT
5	CTR2_Source/A	3	CTR3_Source/A
38	CTR2_Gate/Z	45	CTR3_Gate/Z
37	CTR2_AUX/B	46	CTR3_AUX/B
1	CTR2_ OUT	39	CTR3_ OUT

Figure 88 QuadEncoder For Continuous Mode

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

9.8.7. Two-Pulse Encoder

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

Timing

1) Single Mode

The count value is written to the register on each rising edge of the signal A, and signal B, as shown in Figure 89.

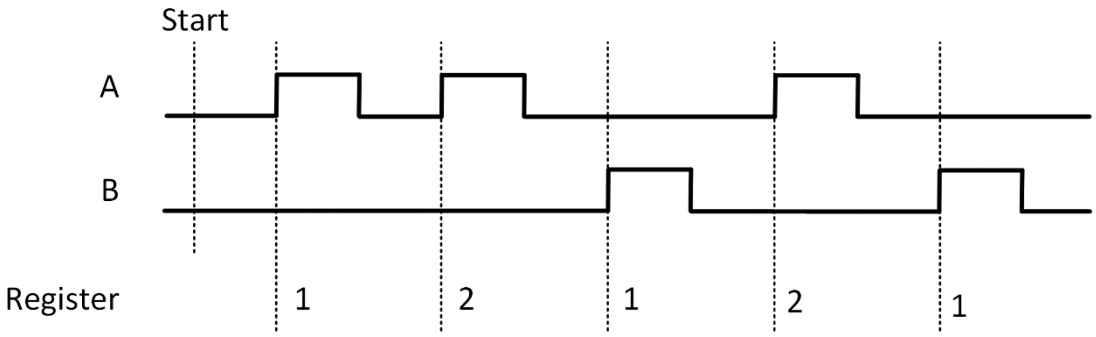


Figure 89 Two-Pulse Encoder in Single Mode

2) Finite/Continuous Mode with Internal Sample Clock

The count value is stored into the buffer on each rising edge of the sample clock, as shown in Figure 90.

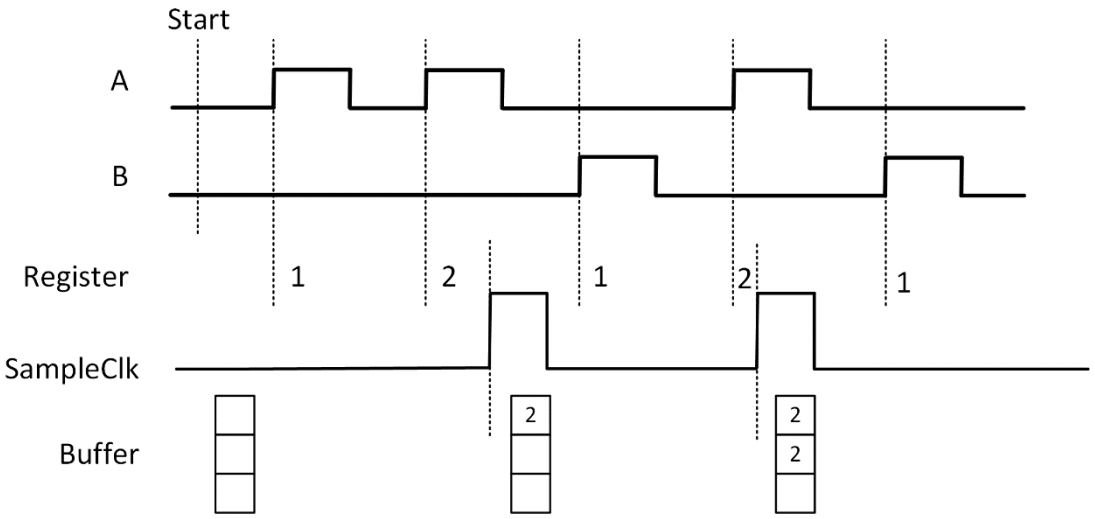


Figure 90 Two-Pulse Encoder with Internal Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

The count value is stored into the buffer every time it changed, as shown in Figure 91.

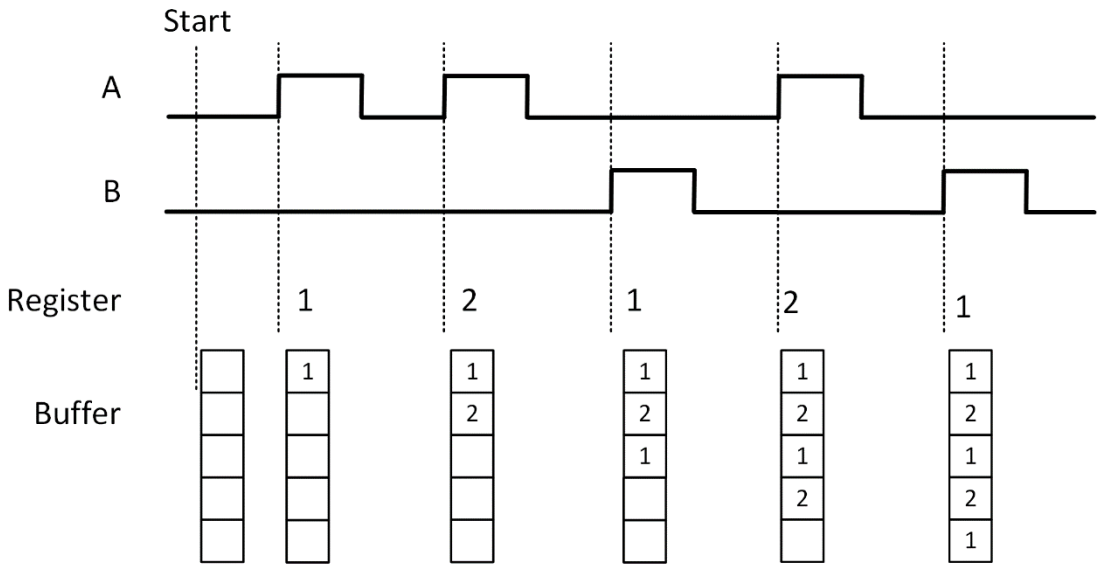


Figure 91 Two-Pulse Encoder with Implicit Sample Clock

Learn by Examples 9.8.7

- Connect the signal source's positive terminal to PCIe-5111 signal input (squarewave, CTR0_Source/A, Pin #11)， negative terminal to the ground (DGND, Pin#44)。
- Connect the PCIe-5111 signal input(CTR0_AUX/B, Pin#43)to ground (DGND, Pin#44).
- Set a sqaurewave signal (f=40Hz)

Single Mode

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

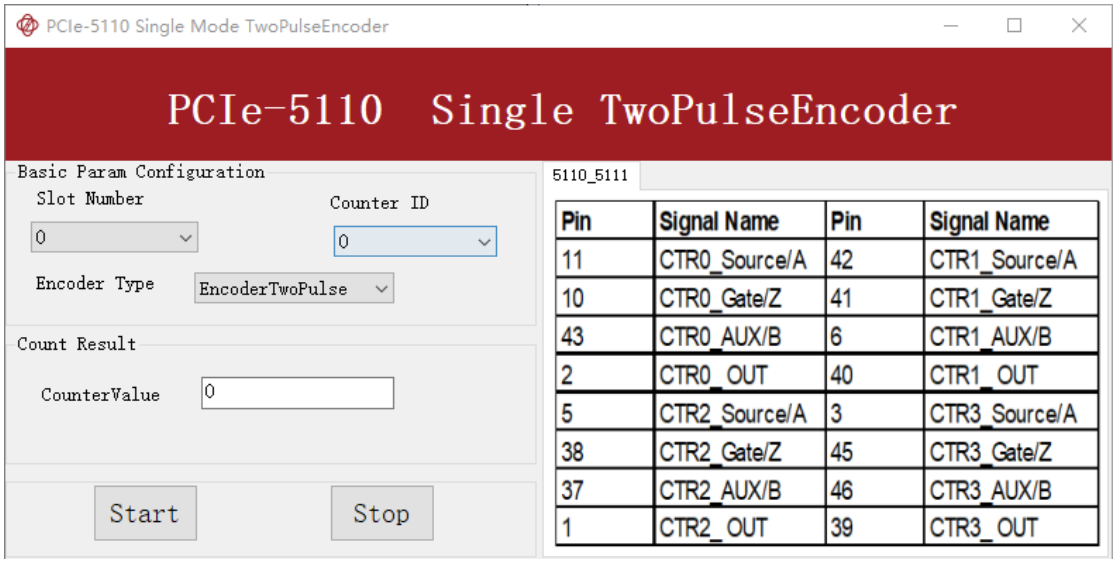


Figure 92 Two-PulseEncoder For Single Mode

➢ 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.

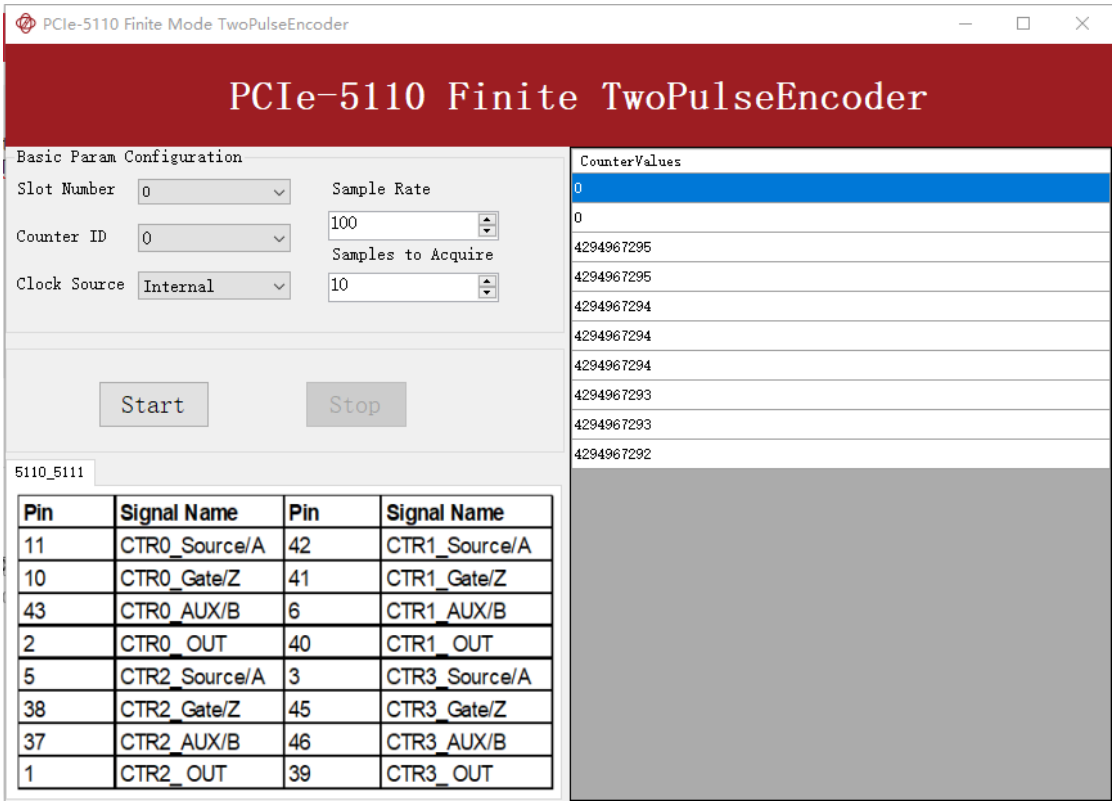
Finite Mode

■ Connect the signal source's positive terminal to PCIe-5111 signal input (squarewave, CTR0_AUX/B, Pin#43)， negative terminal to the ground (DGND, Pin#44)。

■ Connect the PCIe-5111 signal input(CTR0_Source/A, Pin #11)to ground (DGND, Pin#44).

■ Set a sqaurewave signal (f=40Hz).

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



Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
2	CTR0_ OUT	40	CTR1_ OUT
5	CTR2_Source/A	3	CTR3_Source/A
38	CTR2_Gate/Z	45	CTR3_Gate/Z
37	CTR2_AUX/B	46	CTR3_AUX/B
1	CTR2_ OUT	39	CTR3_ OUT

Figure 93 Two-PulseEncoder For Finite Mode

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

■ Click **Start** to start counting. You can see that the **CounterValue** is decreasing**,** 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.

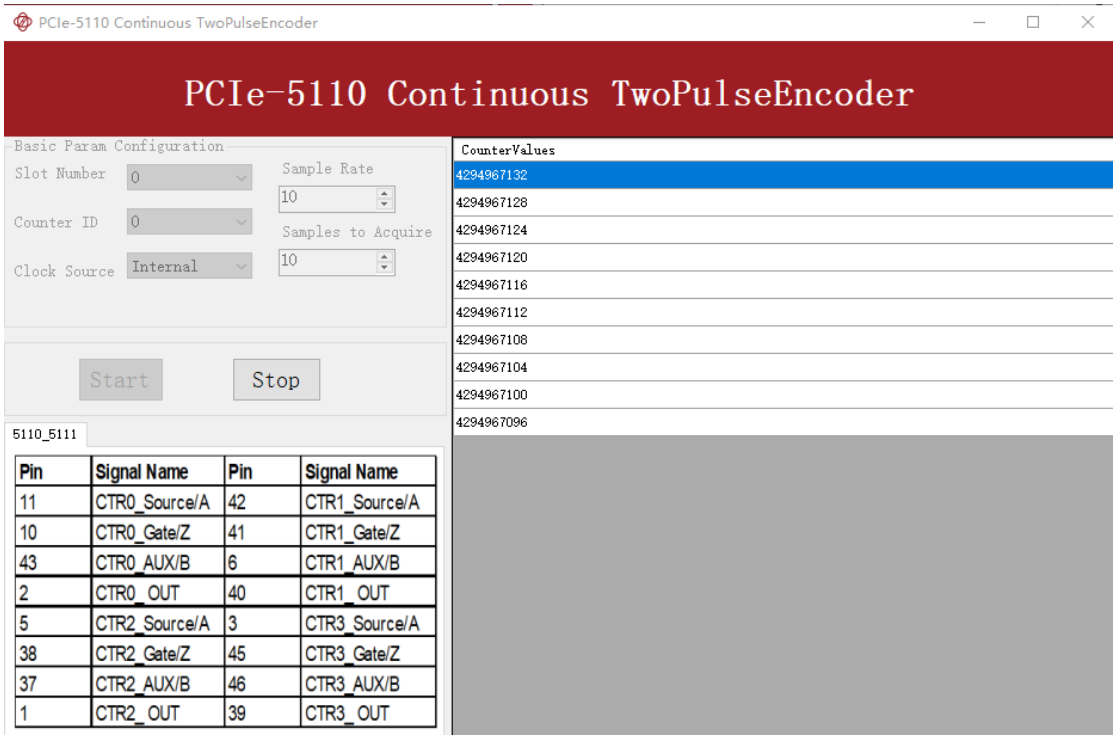


Figure 94 Two-PulseEncoder For Continuous Mode

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

- Click **Start** to start counting. You can see that the **CounterValue** is decreasing**,** which follows the counting rules explained in this chapter.

9.9. Counter Output Operations

9.9.1. Single Pulse Output

The PCIe/PXIe-5113 timer/counter can output a single pulse with a specified pulse width. The timing diagram of the pulse output is shown in Figure 95.

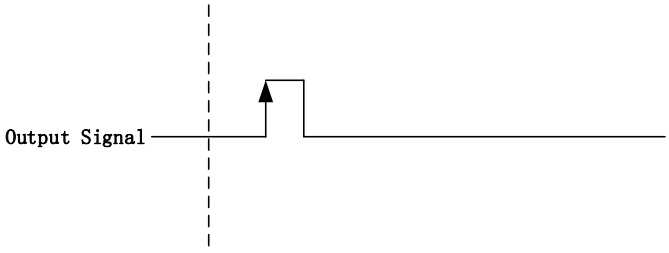


Figure 95 Single Pulse Output

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

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

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

Learn by Example 9.9.1

- To see the signal that PCIe-5111 Counter Output generates, it is recommended to connect PCIe-5111 Counter Output (CTR0_OUT, Pin#2) to PCIe-5111 AI Ch0 input (AI0+, Pin#68). Please note Counter Output and AI Ch0 input share the same ground so only one connection is needed.
- Open **Counter Output-->Winform CO Single** and click **Start** and set the numbers as follow:

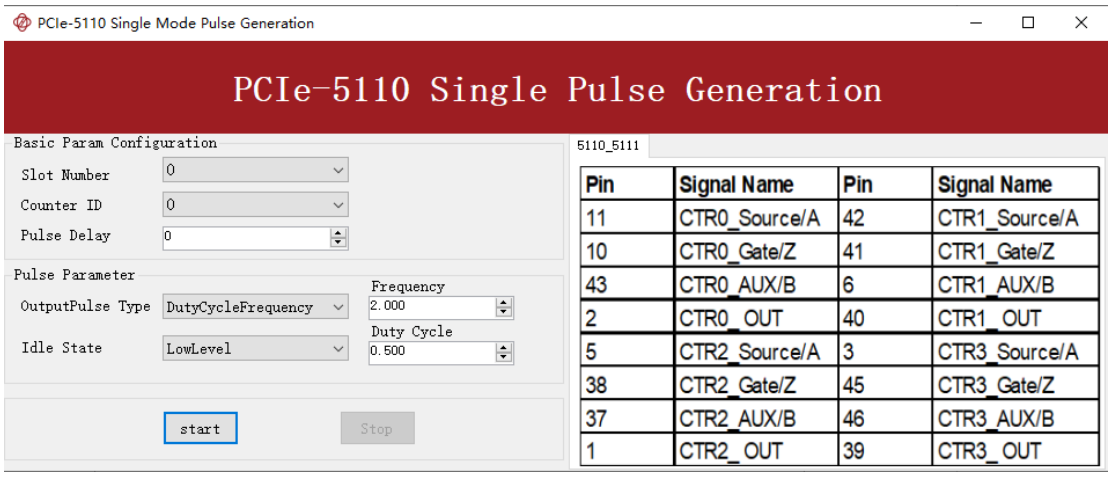


Figure 96 Single Pulse Generation

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

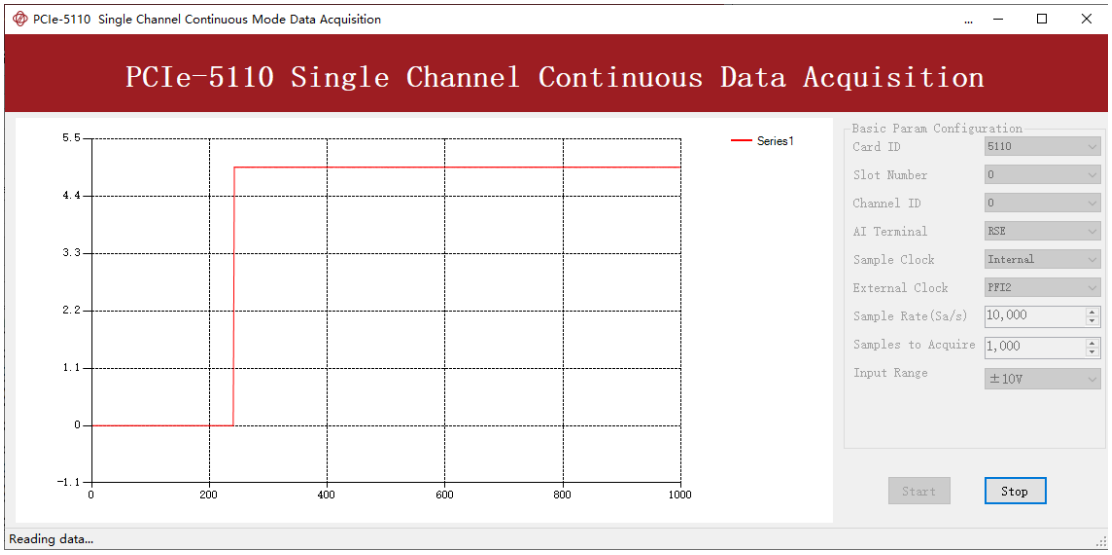


Figure 97 AI Acquisition Single Pulse

9.9.2. Finite Pulse Output

The pulse output timing is as shown in Figure 98.

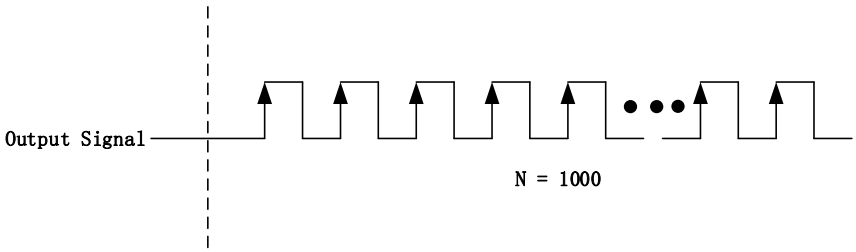


Figure 98 Finite Pulse Output

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

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

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

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

Learn by Example 9.9.2

- To see the signal that PCIe-5111 Counter Output generates, it is recommended to connect PCIe-5111 Counter Output (CTR0_OUT, Pin#2) to PCIe-5111 AI Ch0 input (AI0+, Pin#68). Please note Counter Output and AI Ch0 input share the same ground so only one connection is needed.
- Open **Counter Output-->Winform CO Finite** and click **Start** and set the numbers as follow:

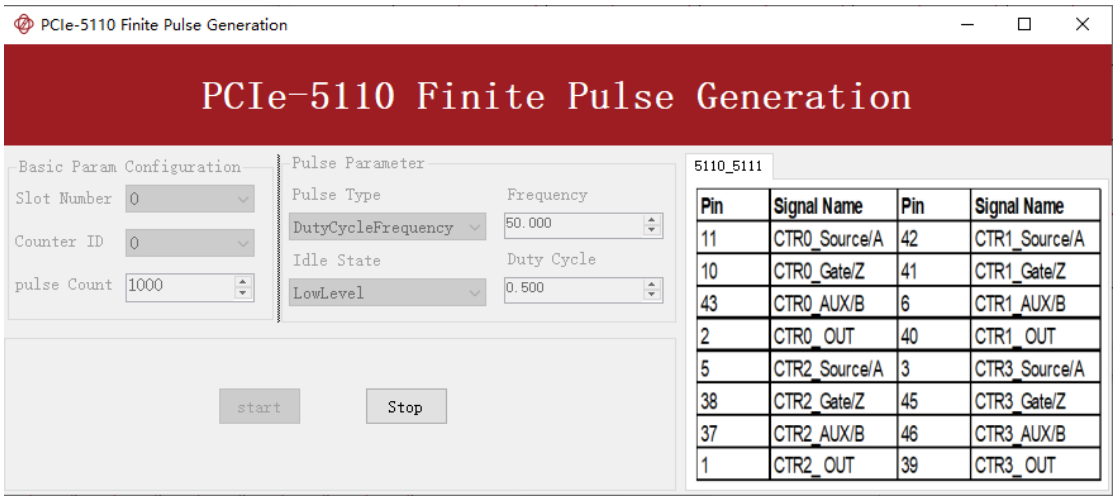


Figure 99 Finite Pulses Generation

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

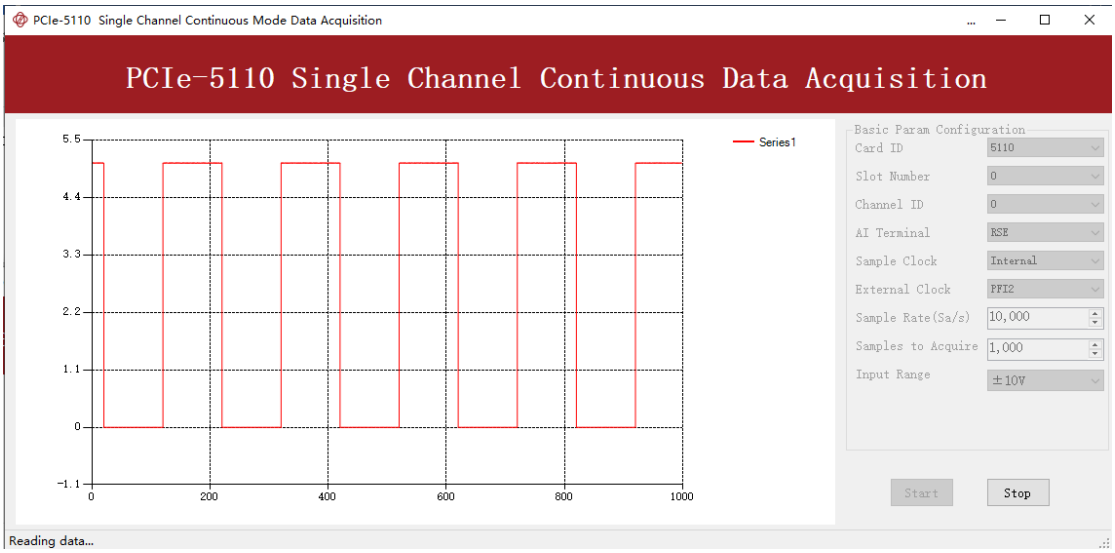


Figure 100 AI Acquisition Finite Pulse

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

9.9.3. Continuous Pulse Output

The pulse output timing is shown in Figure 101 below.

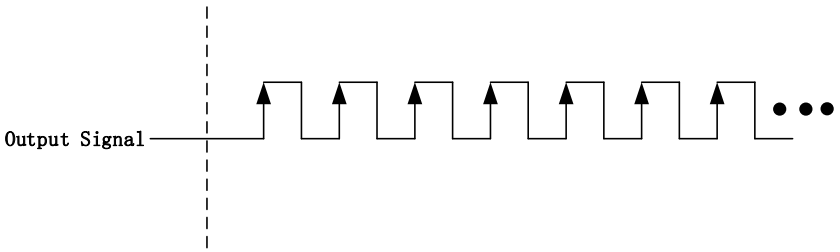


Figure 101 Continuous Pulse Output

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

Learn by Example 9.9.3

- To see the signal that PCIe-5111 Counter Output generates, it is recommended to connect PCIe-5111 Counter Output (CTR0_OUT, Pin#2) to PCIe-5111 AI Ch0 input (AI0+, Pin#68). Please note Counter Output and AI Ch0 input share the same ground so only one connection is needed.

■ Open **Counter Output-->Winform CO Continuous** and click **Start** and set the numbers as follow:

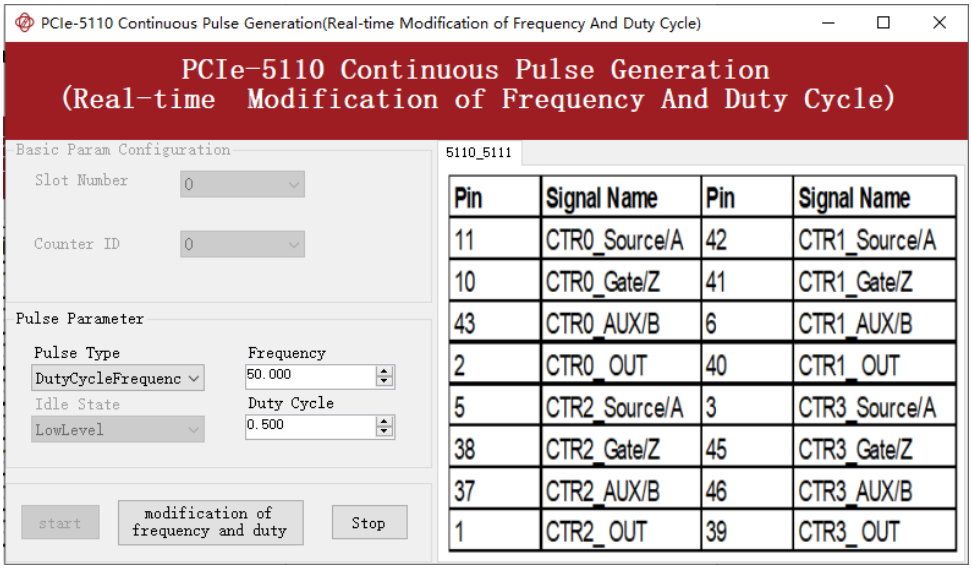


Figure 102 Continuous Pulse Generation

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

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

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

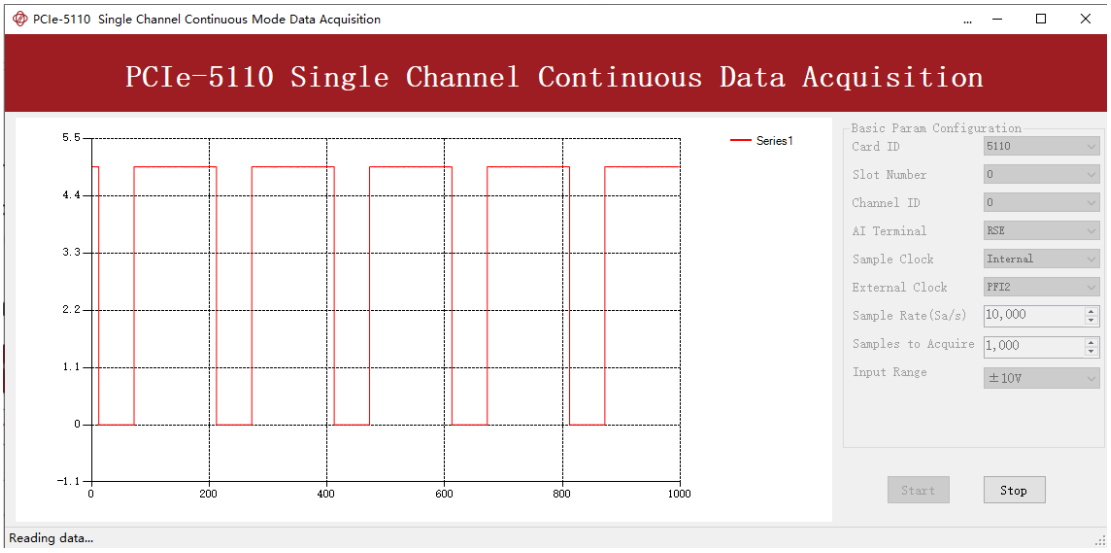


Figure 103 AI Acquisition Continuous Pulse

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

9.10. System Synchronization Interface（SSI）for PCIe Modules

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

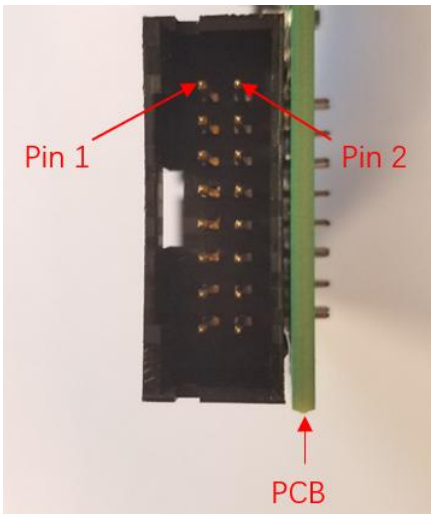


Figure 104 SSI Connector in PCIe-5113

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 15 SSI Connector Pin Assignment for PCIe-5113

9.11. DIP Switch in PCIe/PXIe-5113

PCIe/PXIe-5113 modules have a DIP switch. The card number can be adjusted manually by changing the DIP switch setting, which is used to identify the boards with different slot positions.

10. Calibration

PCIe/PXIe-5113 boards are precalibrated before the shipment. We recommend you recalibrate PCIe/PXIe-5113 boards periodically to ensure the measurement accuracy. A commonly accepted practice is one year. If for any reason, you need to recalibrate your board, please contact JYTEK.

11. Appendix(Measurement Issues)

11.1. Performance and Tests

11.1.1. AI Accuracy

The maximum AI accuracy is limited by three factors: Total Gain Error, Total Offset Error and the Noise Uncertainty as shown in the following table. The PCIe/PXIe-5113 device is first calibrated. The test temperature is 23 ºC±5ºC.

Typical Tested 5110 AI Accuracy							
	Nominal Range						
Total Gain Error (µV)	314.0	153.5	57.2	29.1	16.4	7.1	5.2
Total Offset Error (µV)	154.0	95.0	24.8	23.2	19.1	16.9	15.2
Noise Uncertainty (µV)	2.8	1.6	0.8	0.5	0.4	0.4	0.4
Absolute Accuracy at Full Scale (µV)	470.8	250.1	82.8	52.8	35.9	24.3	20.8

Table 16 Typical AI Accuracy

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

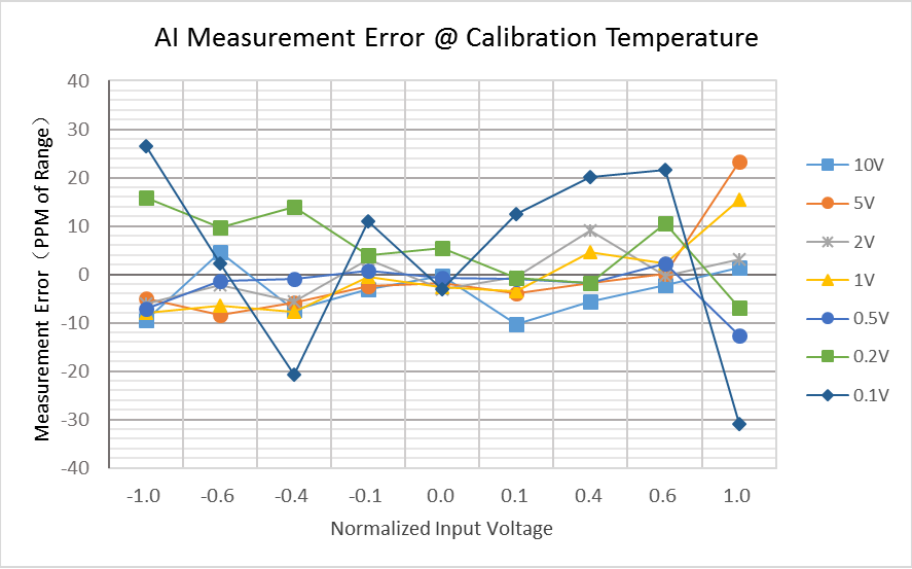


Figure 105 AI Measurement Error

11.1.2. AI Bandwidth

Analog Input Bandwidth	
Nominal Range Full Scale (V)	**-3dB Bandwidth (MHz)**
±10	3.26
±5	2.77
±2	1.39
±1	0.78
±0.5	0.78
±0.2	0.72
±0.1	0.63

Table 17 AI Bandwidth

11.1.3. System Noise

A typical system noise is show below.

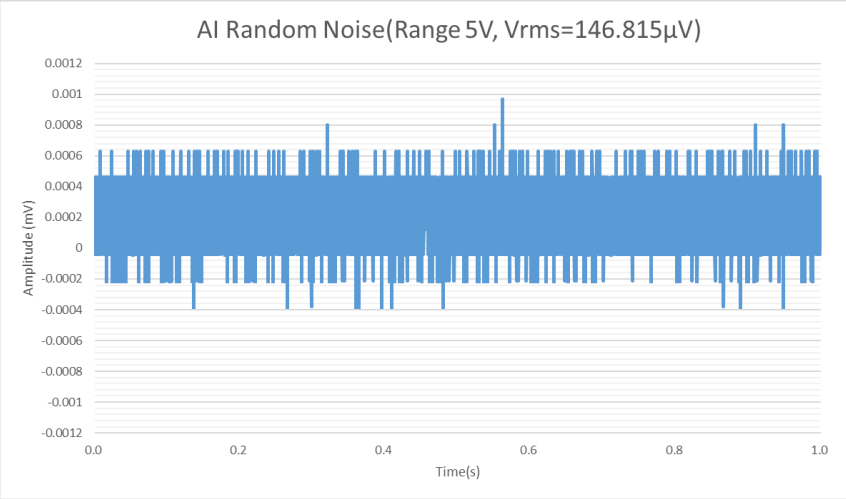


Figure 106 AI Error due to System Random Noise

11.1.4. PCIe/PXIe-5113 CMRR

A typical CMRR performance is shown below.

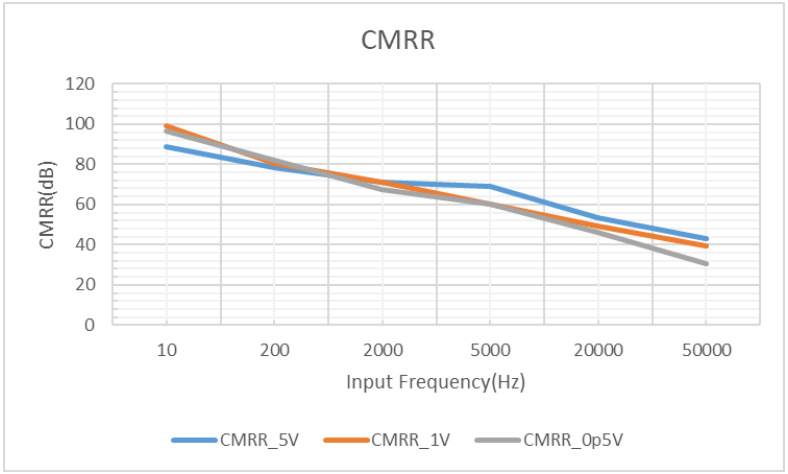


Figure 107 Common Mode Rejection Ratio

11.1.5. AO Accuracy

The maximum AO accuracy is limited by two factors: Total Gain Error, Total Offset Error as shown in the following table. The PCIe/PXIe-5113 device is first calibrated. The test temperature is 2 °C from the calibration temperature.

Typical Tested 5110 AO Accuracy		
Nominal Range Positive Full Scale (V)	10	5
Gain Error	33.7	23.7
Offset Error	77	59.4
Absolute Accuracy at Full Scale (μV)	**1052**	**416**

Table 18 Typical AO Accuracy

11.2. Floating Signals and Ground Referenced Signals

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

11.3. Differential, NRSE, RSE Modes

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

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

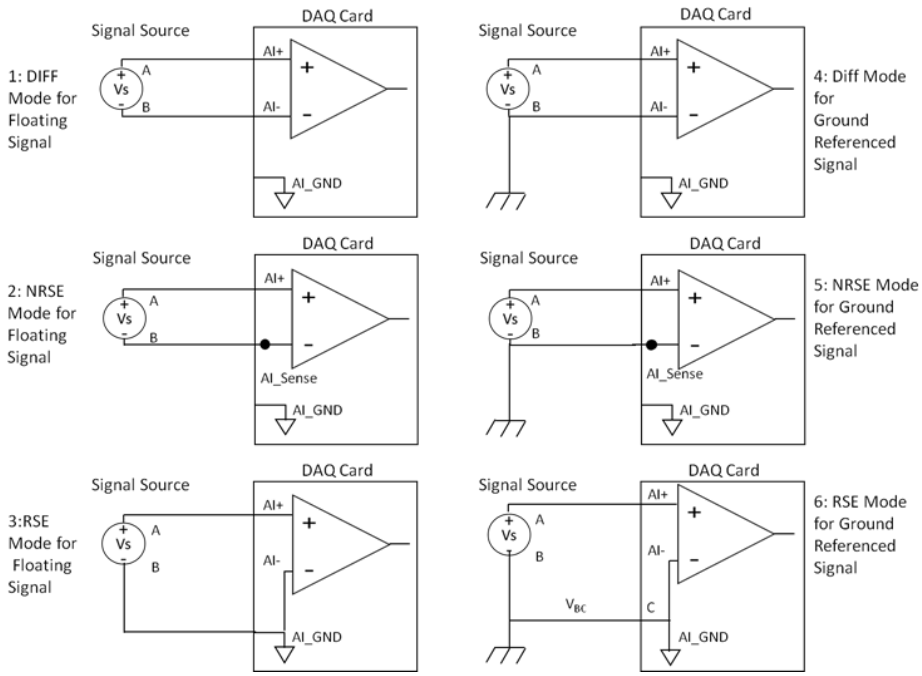


Figure 108 Six Measurement Scenarios

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

11.4. Reducing the Common Mode Voltage Effect

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

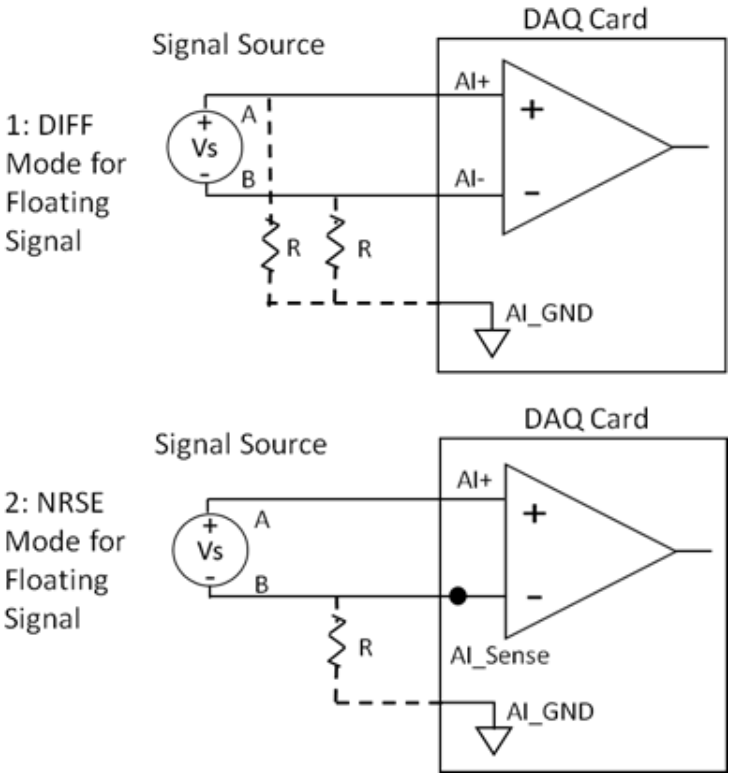


Figure 109 Using Resister to Reduce Common Mode Voltage Effect

12. About JYTEK

12.1. 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.

12.2. 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.

12.3. 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.

13. 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 PCIe/PXIe-5113 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