1. PCIe-5111/5112/5111B/5112B Specifications

Multi-functional Data Acquisition Boards

Overview

JYTEK JY-5110 series includes PCIe-5111/5112/5111B/5112B. PCIe-5111/5111B provide up to 32 channels of analog inputs of up to 500K samples per second, 4 channels of analog outputs, 48 channels of digital IO or 4 32-bit counters/timers ; PCIe-5112/5112B provide up to 16 channels of analog inputs of up to 250K samples per second, 2 channels of analog outputs, 24 channels of digital IO or 2 32-bit counters/timers.

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

1.1. Main Features

- High accuracy: 350 ppm (JY-5111/5111B), 300 ppm (JY-5112/5112B)
- Up to 32 single-ended/16 differential analog input channels(5111/5111B); Up to 16 single-ended/8 differential analog input channels(5112/5112B);
- Sampling rates: 500 kS/s for one channel, 300 kS/s for multi-channel(5111/5111B); 250 kS/s for one channel, 250 kS/s for multi-channel(5112/5112B).
- 16 bits ADC
- Up to 128M samples of analog input FIFO buffer
- JY-5111B/5112B support 60 V Isolation
- 4(5111/5111B)/2(5112/5112B) simultaneous 16-bit analog output channels
- 32M sample FIFO buffer for analog output
- 6 ports digital IO, 8(5111)/4(5112) lines per port
- DIO supports hardware timing up to 10MHz (5111)
- 4(5111)/2(5112) channels 32-bit timer/counter
- DMA for AI, AO and DIO
- Analog/Digital/Software Trigger

1.2. Analog Input

Analog Input	5111	5112	5111B	5112B
Number of channels	32 SE / 16 DIFF	16 SE/ 8 DIFF	32 SE / 16 DIFF	16 SE/ 8 DIFF
ADC resolution (Bits)	16		16	
Single channel maximum sample rate	500 kS/s	250 kS/s	500 kS/s	250 kS/s
Multichannel maximum sample rate with same range(aggregate)	300 kS/s	250 kS/s	300 kS/s	250 kS/s
Clock	100 MHz		100 MHz	
Input range(V)	±10/±5/±2.5/±1.25/±0.625	±10/±5/±2.5/±1.25/±0.625	±10/±5/±2.5/±1.25/±0.625	±10/±5/±2.5/±1.25/±0.625
Input mode	RSE / NRSE / Differential	RSE / NRSE / Differential	RSE / NRSE / Differential	RSE / NRSE / Differential
Input impedance	>1 GΩ||100 pF	>1 GΩ||100 pF	>1 GΩ||100 pF	>1 GΩ||100 pF
Input coupling	DC		DC	
Overvoltage protection	±25 V		±25 V	
CMRR	85 dB		85 dB	
Crosstalk	-80 dB		-80 dB	
DNL	No Missing Code		No Missing Code	
INL	70 ppm of Range Typical		70 ppm of Range Typical	
Input FIFO	64 M Samples	96 K Samples	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	5111	5112	5111B	5112B
Number of channels	4	2	4	2
DAC resolution	16 bits		16 bits	
Maximum update rate(simultaneous)	1 channel 2.86 MS/s 2 channels 2 MS/s 3 channels 1.54 MS/s 4 channels 1.25 MS/s		1 channel 2 MS/s 2 channels 2 MS/s (1 channel per bank)* 3 channels 1 MS/s 4 channels 1 MS/s	
Clock	100 MHz		100 MHz	
Clock accuracy	Jitter <20 ps		Jitter <20 ps	
Output range(V)	±10, ±5	±10	±10	±10
Output mode	RSE		RSE	
Output impedance	2 Ω		2 Ω	
Output coupling	DC		DC	
Output current drive	±10 mA		±10 mA	
Output FIFO	32 M Samples	32 K Samples	32 M Samples	32 M Samples
Trigger type	Digital, Software		Digital, Software	
Trigger mode	StartTrigger		StartTrigger	

* Each bank consists of 2 AO channels. Any channels being used within a single bank will update

Table 2 Analog Output Specifications

1.4. Counter Input/Output

CIO	5111	5112	5111B	5112B
Number of channels	4	2	4	2
Resolution	32 bits			
Input Terminal	Gate(Z) ,Source(A) ,Aux(B) , Digital Trigger External Sample Clock	Gate(Z) ,Source(A) ,Aux(B)	Gate(Z) ,Source(A) ,Aux(B) , Digital Trigger External Sample Clock	
Output Terminal	OUT			
Input Mode	Single,Finite,Continuous	Single	Single,Finite,Continuous	
Input Type	Edge Counting Period Measure Frequency Measure Pulse Measure Two Edge Separation Measure Quadrature Encoder Two Pulse Encoder	Edge Counting Period Measure SemiPeriod Measure Frequency Measure Pulse Measure Two Edge Separation Measure Quadrature Encoder Two Pulse Encoder		
Internal timebase	200 MHz	N/A	100MHz/5MHz/100kHz	
Max SampleClock	10 MHz	N/A	10 MHz	
External timebase	N/A	N/A	0-10M Hz	
Output Mode	Single, Finite and Continuous pulse	Single pulse	Single, Finite and Continuous pulse	
FIFO per channel	4 M Samples	N/A	4 M Samples	

Table 3 Counter Input Operations Specifications

1.5. Digital IO Specifications

DIO	5111	5112	5111B	5112B
Number of channels	Port (0,1,2,3,4,5), 8 Lines per port	Port（0,1,2), 8 Lines per port	48 lines	24 lines
Number of Dynamic DIO channels	Port (0,1,2,3), 8 Lines per port	N/A	32 lines (line0~line24)	24 lines (line0~line23)
Ground reference	D GND		D GND	
Directional control	Independent control of each port		Independent control of each line	
Clock	10 MHz			
DI FIFO	16M Samples	N/A	16M Samples	
DO FIFO	16M Samples	N/A	16M Samples	
Clock	10 MHz	N/A	10 MHz	
Initial state	Input		Input	
Digital Input	Logic Low: V_{IL} Min : 0 / Max : 1.0 V		Logic Low: V_{IL} Min : 0 / Max : 1.0 V	
	Logic High: V_{IH} Min : 2 V / Max : 5.3		Logic High: V_{IH} Min : 2 V / Max : 5.3	
Digital Output	Logic Low : 0~0.55 V, I_{OL}: 0~36 mA		Logic Low : 0~0.55 V, I_{OL}: 0~36 mA	
	Logic High : 3.8 V~5 V, I_{OH}: -36		Logic High : 3.8 V~5 V, I_{OH}: -36 mA~0	
Overvoltage Protection	Continuous 30 mA, -3.9 V~8.9 V Instantaneous 200 mA, ±25 V;		Continuous 40 mA, -0.5 V~6.4 V	
	Duty cycle of instantaneous current pulse does not exceed 15%		Duty cycle of instantaneous current pulse does not exceed 15%	

Table 4 Digital IO Specifications

1.6. PFI Specifications

PFI	5111	5112	5111B	5112B
Number of channels	16			
External digital trigger interface	Trigger voltage : 5V TTL			
	Trigger edge: Rising/Falling			
Initial state	Input			

Table 5 PFI Specifications

1.7. Basic DC AI Accuracy

JY-5111 Basic Accuracy = ±(% Reading+% Range)										
Nominal Range (V)	24 Hour Tcal ±1C°			90 Days Tcal ±5C°			24 Hr Full Scale Accuracy	90 Days Full Scale Accuracy
0.625	0.010	+	0.045	0.024	+	0.120	340 uV	850 uV
1.25	0.004	+	0.024	0.008	+	0.063	330 uV	890 uV
2.5	0.004	+	0.020	0.008	+	0.050	570 uV	1500 uV
5	0.004	+	0.015	0.007	+	0.037	870 uV	2200 uV
10	0.004	+	0.012	0.007	+	0.028	1400 uV	3500 uV
Valid for one channel only. 95% of Confidence Interval Max sampling rates for 5111: 500 kS/s Add 20% to Gain and Offset Errors From 91 Days to 1 Year. Preliminary Testing condition for AO accuracy: within 95% of full scale range.										

JY-5112 Basic Accuracy = ±(% Reading+% Range)								
Nominal Range (V)	24 Hour Tcal ±1C°			90Day Tcal ±5C°			24 Hr Full Scale Accuracy	90 Days Full Scale Accuracy
0.625	0.004	+	0.046	0.010	+	0.120	310 uV	770 uV
1.25	0.003	+	0.032	0.008	+	0.079	430 uV	1100 uV
2.5	0.002	+	0.021	0.004	+	0.053	560 uV	1400 uV
5	0.002	+	0.015	0.005	+	0.037	830 uV	2100 uV
10	0.002	+	0.011	0.003	+	0.027	1200 uV	3000 uV
Valid for one channel only. 95% of Confidence Interval Max sampling rates for 5112: 250 kS/s Add 20% to Gain and Offset Errors From 91 Days to 1 Year. Preliminary Testing condition for AO accuracy: within 95% of full scale range.								

Table 6 Basic Accuracy in DAQ Mode

1.8. AI Bandwidth

Analog Input Bandwidth	
Nominal Range Full Scale (V)	**-3dB Bandwidth (KHz)**
±10	325
±5	390
±2.5	480
±1.25	480
±0.625	480

Table 7 AI Bandwidth Specifications

1.9. Basic AO Accuracy

JY-5111 Basic AO Accuracy = ±(% of Output+% of Range)				
Nominal Range (V)	24 Hour Tcal ±1C°	90 Days Tcal ± 5°	24 Hr Full-Scale Accuracy	90 Days Full-Scale Accuracy
5	0.004 + 0.007	0.009 + 0.017	490 uV	1300 uV
10	0.004 + 0.005	0.009 + 0.013	830 uV	2100 uV
Valid for all update rates. 95% Confidence Interval Testing condition for AO accuracy: within 95% of full scale range. Add accuracy adjustment if temperature is ouside calibration temperature range. Add 20% to Gain and Offset Errors From 91 Days to 1 Year. Preliminary. Maximum update rates(simultaneous) 1 Channel: 2.86 MS/s; 2 Channels: 1 MS/s; 3 Channels 1.54 MS/s; 4 Channels 1.25 MS/s Specs subject to minor changes when more tests become available.				

JY-5112 AO Basic Accuracy = ±(% Reading+% Range)						
Nominal Range (V)	24 Hour Tcal ±1C°		90 Day Tcal ±5C°		24 Hr Full Scale Accuracy	90 Day Full Scale Accuracy
10	0.002 +	0.011	0.004 +	0.028	1300 uV	3100 uV

Table 8 Basic AO Accuracy

1.10. System Noise

	5111	5112	5111B	5112B
Range(V)	SystemNoise(μVrms)	SystemNoise(μVrms)	SystemNoise(μVrms)	SystemNoise(μVrms)
0.625	72		72	
1.25	93		93	
2.5	150		145	
5	230		227	
10	350		335	

Table 9 System Noise for PCIe-5111

1.11. 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

1.12. Front Panel and Pin Definition

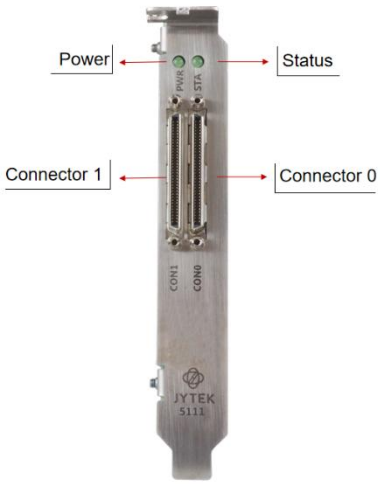


Figure 1 PCIe 5111 Front Panel

5111 Counter Pin Definition on Connector 0 as show in Table 11.

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

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

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

Table 11 5111 Pin Definition

AI_GND	Analog Input Reference Ground
AI<0..31>	Analog Input channel
AI SENSE	Analog Input Signal, Suitable for NRSE mode
AO_GND	Analog Output Reference Ground
AO<0..3>	Analog Output Channel
D_GND	Digital Signal Reference Ground
P<0..3>.<0..7>	Digital I/O Channel
PFI<0..15>	Programmable Function Interface
+5V_OUT	5V power supply
RSVD	Do Not Connect

Table 12 5111 Cable Specification

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

Pin	**Signal Name**	**Pin**	**Signal Name**
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
37	CTR0_ OUT	3	CTR1_ OUT

Table 13 5112 Pin Definition

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

Table 14 5112 Cable Specification

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

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

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

Table 15 5111B Pin Definition

Connector 0			
Pin	Identification	Pin	Identification
1	PFI 14 /DO_ECLK	35	D_GND
2	PFI 12	36	D_GND
3	PFI 9	37	PFI 8
4	D_GND	38	PFI 7
5	PFI 6/ AO_ECLK	39	PFI 15
6	PFI 5	40	PFI 13
7	D_GND	41	PFI 4
8	+5V_OUT	42	PFI 3
9	D_GND	43	PFI 2 /AI_ECLK
10	PFI 1	44	DGND
11	PFI 0	45	PFI 10/DI_ECLK
12	D_GND	46	PFI 11
13	D_GND	47	P0.3
14	+5V_OUT	48	P0.7
15	D_GND	49	P0.2
16	P0.6	50	D_GND
17	P0.1	51	P0.5
18	D_GND	52	P0.0
19	P0.4	53	D_GND
20	APFI_0	54	AO_COM
21	AO 1	55	AO_COM
22	AO 0	56	AI_COM
23	AI 15（AI 7-）	57	AI 7（AI 7+）
24	AI_COM	58	AI 14（AI 6-）
25	AI 6（AI 6+）	59	AI_COM
26	AI 13（AI 5-）	60	AI 5（AI 5+）
27	AI_COM	61	AI 12（AI 4-）
28	AI 4（AI 4+）	62	AI_SENSE 0
29	AI_COM	63	AI 11（AI 3-）
30	AI 3（AI 3+）	64	AI_COM
31	AI10（AI 2-）	65	AI 2（AI 2+）
32	AI_COM	66	AI 9（AI 1-）
33	AI 1（AI 1+）	67	AI_COM
34	AI 8（AI 0-）	68	AI 0（AI 0+）

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

Table 16 5112B Pin Definition

2. Table of Contents

Table of Contents

3. Software

3.1. Supported Operating System

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

3.2. Programming Languages

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

4. Order Information

- PCIe-5111 (PN: JY2953885-01)

  32-ch AI (16-Bit,500 kS/s), 4-ch AO (16-Bit, 2.86 MS/s), 48DIO PCIe Multifunction I/O Card

- PCIe-5112 (PN: JY1922016-01)

  16-ch AI (16-Bit,500 kS/s), 2-ch AO (16-Bit, 2.86 MS/s), 24DIO PCIe Multifunction I/O Card

- PCIe-5111B(PN: JY8379886-01)

  32-ch AI (16-Bit,500 kS/s), 4-ch AO (16-Bit,2 MS/s), 48DIO PCIe Board Isolated Multifunction I/O Module

- PCIe-5112B (PN: JY9786666-01)

  16-ch AI (16-Bit,500 kS/s), 2-ch AO (16-Bit,2 MS/s), 24DIO PCIe Board Isolated Multifunction I/O Module

- Accessories:

  DIN-68 (PN: JY2953885-01)

  ACL-1016868-1 (PN: JY7996916-01) 1 M 68pin VHDC-SCSI twisted pair cable

  ACL-1016868-2 (PN: JY7996916-02) 2 M 68pin VHDC-SCSI twisted pair cable

5. 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.

6. Additional Hardware Information

6.1. 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-5111 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 0.625 V range and 24 Hours column, if your measurement or reading is 0.02V, the accuracy of this measurement is:

$$\pm(0.0013\% * 0.02 + 0.0544\% * 0.625) = \pm 0.00034\ \text{V} = \pm 340\ \mu\text{V}$$

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 0.625 V range and the 24-Hour calibration column is 348 µV.

6.2. Basic AO Accuracy

The AO output accuracy of PCIe-5111 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 5V range and 24 Hours column, if your output is 2V, the accuracy of this measurement is:

$$\pm(0.0014\% * 2 + 0.0043\% * 5) = \pm 0.000243\text{V} = \pm 243\ \mu\text{V}$$

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 285 μV.

7. Additional Software Information

7.1. System Requirements

PCIe-5111 series boards can be used in a Windows or a Linux operating system.

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

Linux Kernel Versions: There are many Linux versions. It is not possible JYTEK can support and test our devices under all different Linux versions. JYTEK 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 17 Supported Linux Versions

7.2. System Software

When using the PCIe-5111 series 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-5111 be with .NET Framework 4.0 with Microsoft Visual Studio 2015. JYTEK relies on Microsoft to maintain the compatibility for the newer versions.

7.3. 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.

7.4. PCIe-5111 Series Hardware Driver

After installing the required application development environment as described above, you need to install the PCIe-5111 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-5111 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.

7.5. Install the SeeSharpTools from JYTEK

To efficiently and effectively use PCIe-5111 boards, you need to install a set of free C# utilities, SeeSharpTools from JYTEK. The SeeSharpTools offers rich user interface functions you will find convenient in developing your applications. They are also

needed to run the examples come with PCIe-5111 hardware. Please register and download the latest SeeSharpTools from our website, www.jytek.com.

7.6. 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.

8. Operating JY5111

This chapter provides the operation guides for PCIe-5111, 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-5111 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.

8.1. 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-5111 products.

If you are already familiar with Microsoft Visual Studio C#, the quickest way to use PCIe-5111 boards is to go through our extensive examples. We provide source code of our examples. In many cases, you can modify the source code and start to write your applications.

We also provide **Learn by Example** in the following sections. These examples will help you navigate and learn how to use this PCIe-5111.

8.2. Data Acquisition Methods

PCIe-5100 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-5111 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-5111 provides 4 acquisition modes, **Continuous**, **Finite**, **Single Point**, **Record,** which will be described in details in Section 8.2.1-8.2.4.

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 Section8.2.1; in the finite acquisition mode, it is the total number of samples to capture, please see Section8.2.2.

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.

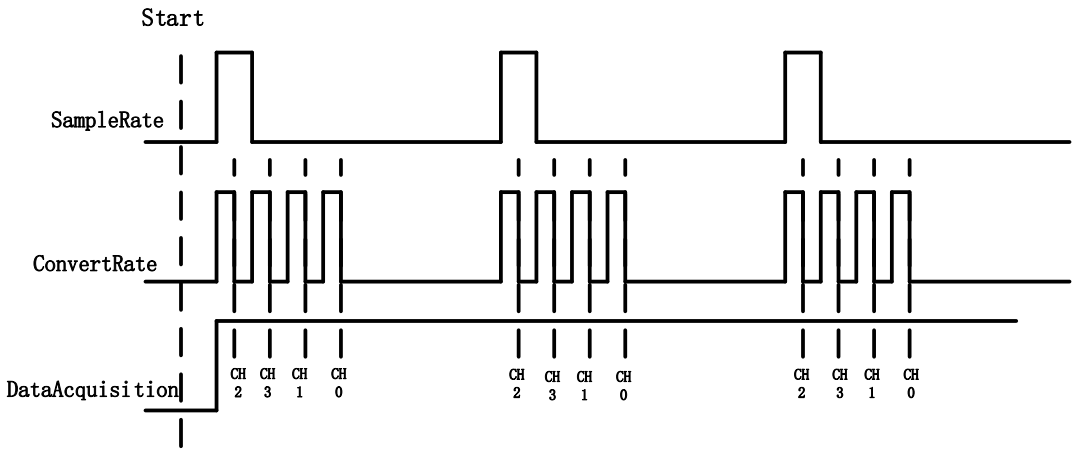


Figure 2 Sample Rate and Internal AD Conversion

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 8.2

- 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.

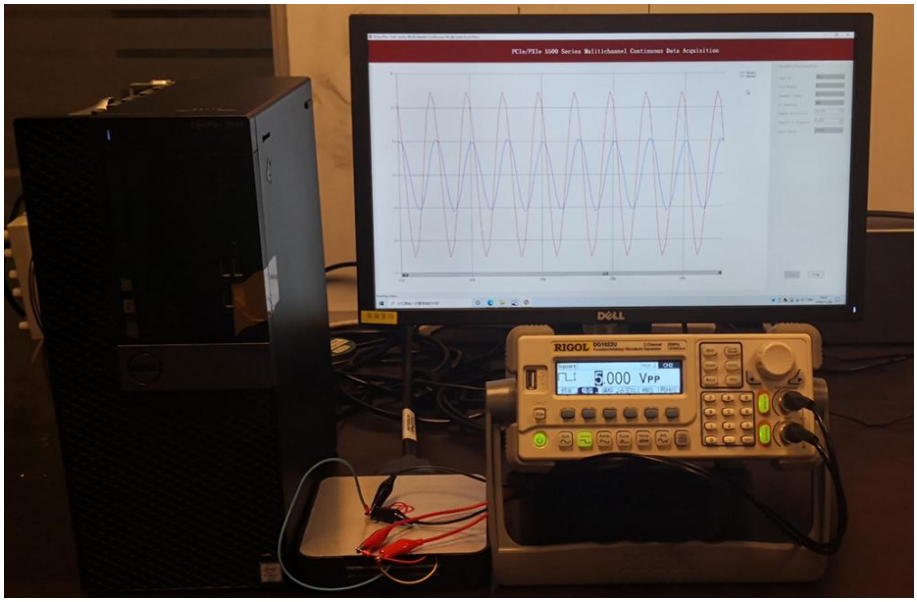


Figure 3 PCIe-5111 experiment

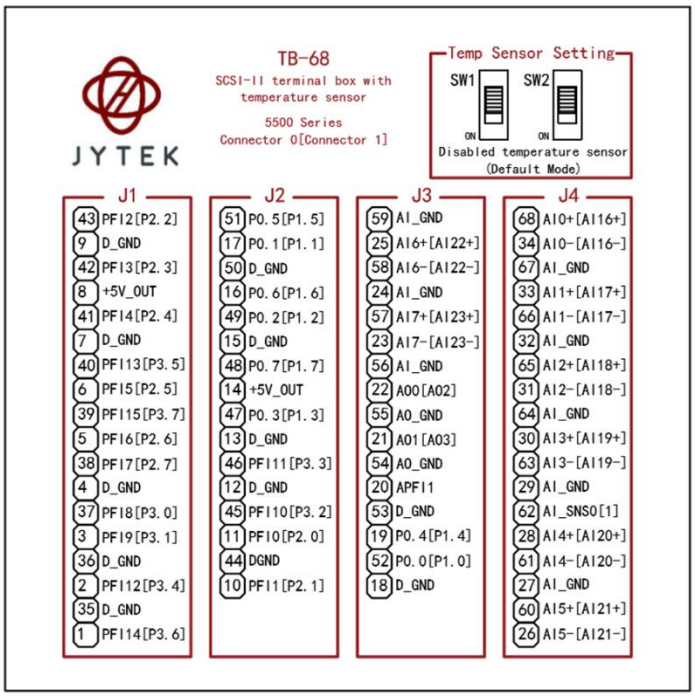


Figure 4 TB-68 Terminal Block

- 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.

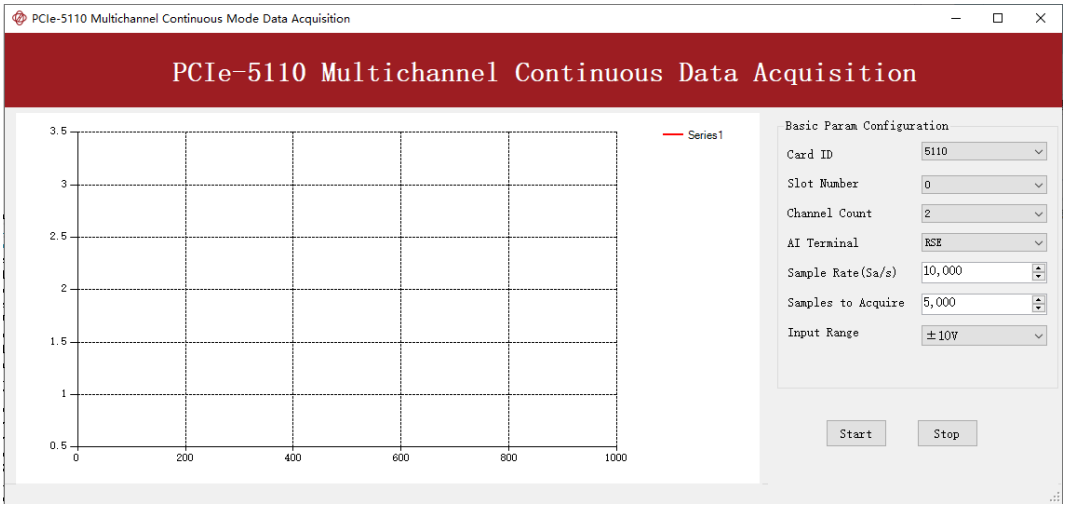


Figure 5 Continuous MultiChannel Paraments

➢ 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.

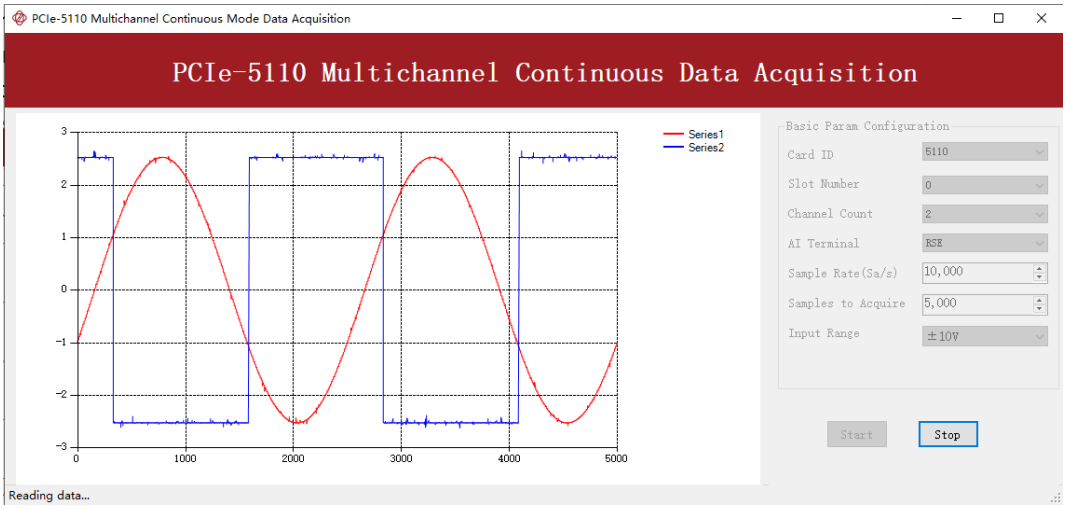


Figure 6 MultiChannel Continuous Acquisition

8.2.1. Continuous Acquisition

An AI acquisition task will acquire the data continuously until the task is stopped. The PCIe-5111 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.

8.2.2. 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.

8.2.3. 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.

8.2.4. 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.

8.3. Analog Input Terminal Type

The PCIe-5111 provide 3 analog input terminal types:

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

The DIFF connection is recommended for ground-referenced signal sources and it is usually better in rejecting the common-mode noise. However, to acquire one input signal, two AI channels are required to form the differential pair. The RSE and NRSE are recommended when the input signal sources are floating signals. In RSE and NRSE modes, these floating signal sources all share the same ground reference (AI_GND). Because of it, the RSE and NRSE modes can acquire twice as many channels than the DIFF mode. Appendix has more details on these 3 modes.

8.3.1. 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.

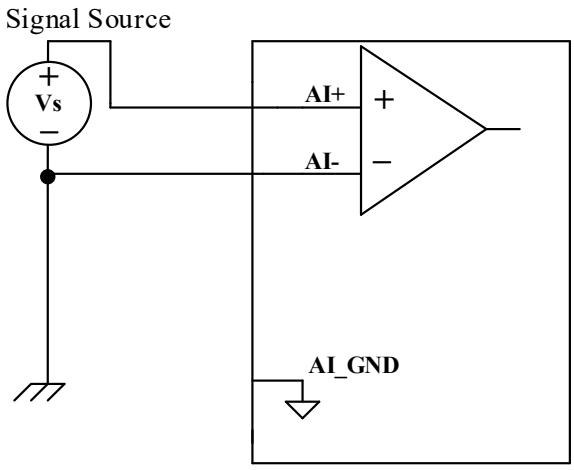


Figure 7 Differential Mode for Grounding Signals

Learn by Example 8.3.1

- 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**.

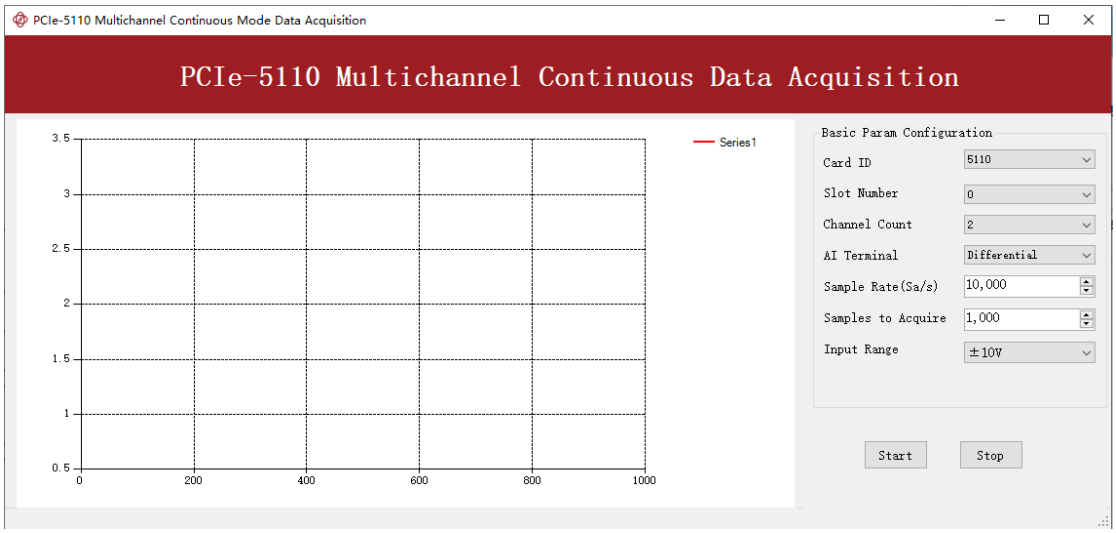


Figure 8 Choose Differential in AI Terminal

8.3.2. RSE Mode

In the RSE mode, all input signals' negative sides are connected to the AI ground of Instrumentation Amplifier, as shown in Figure 9. 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.

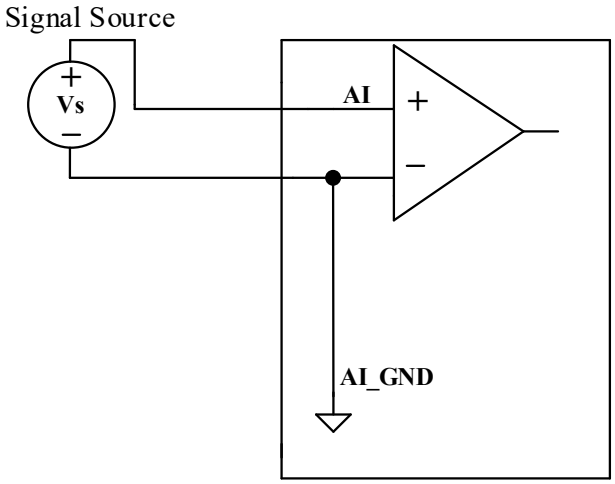


Figure 9 RSE Mode for Floating Signals

8.3.3. NRSE Mode

The NRSE mode is recommended for the measurement of ground-referenced signals, as shown in Figure 10. NRSE is also called the pseudo differential mode, because it looks very similar to a DIFF connection. In this mode, the PCIe-5111 device offers a special reference point, AI SENSE. Instead of connecting two grounds directly, signal's ground and PXI device's ground, the input signals' ground is connected to AI SENSE to avoid the ground loop bias. The PCIe-5111 is also designed to better reject the common mode noise than the RSE mode. Therefore, the NRSE model still offers twice many channels as the DIFF mode. Please see Appendix for more explanations.

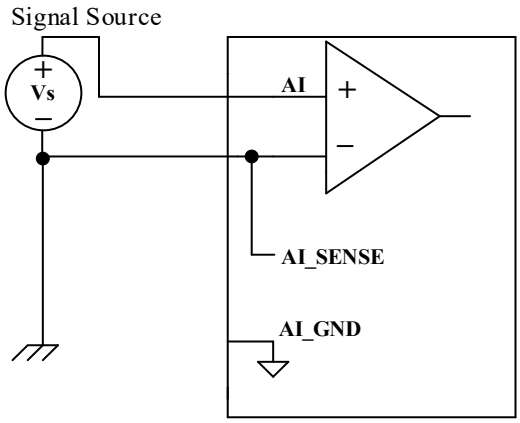


Figure 10 NRSE Mode for Grounding Signals

Learn by Example 8.3.3

- Open the program **Analog Input-->Winform AI Continuous MultiChannel**.
- This Example needs two TB-68 terminal blocks, Connector0 and Connector1 and two cables, which are connected to PCIe-5111. 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_SENSE 0 (Pin#62) of the first TB-68 and AI_SENSE 1 (Pin#62) of the second TB-68 as shown in Figure 3 and Figure 4. (AI0+, AI_SENSE 0) and (AI1+, AI_SENSE 1) consist of two channels of NRSE inputs.

- Choose the NRSE in **AI Terminal**
- Set other numbers as shown and click **start**.

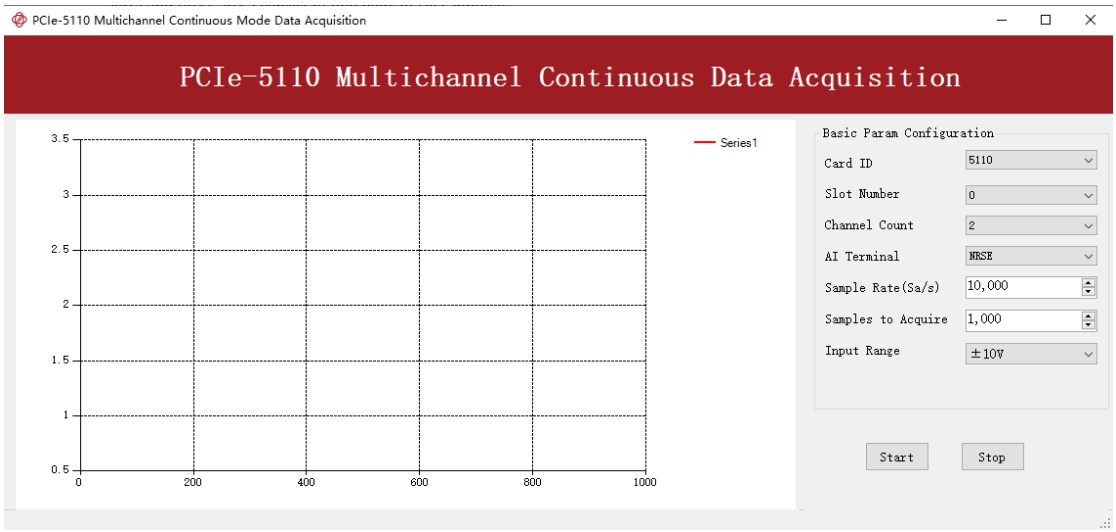


Figure 11 Choose NRSE In AI Terminal

8.4. 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.

8.4.1. 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 8.4.1

- Use the same program and connection as in **Learn by Example8.2.**

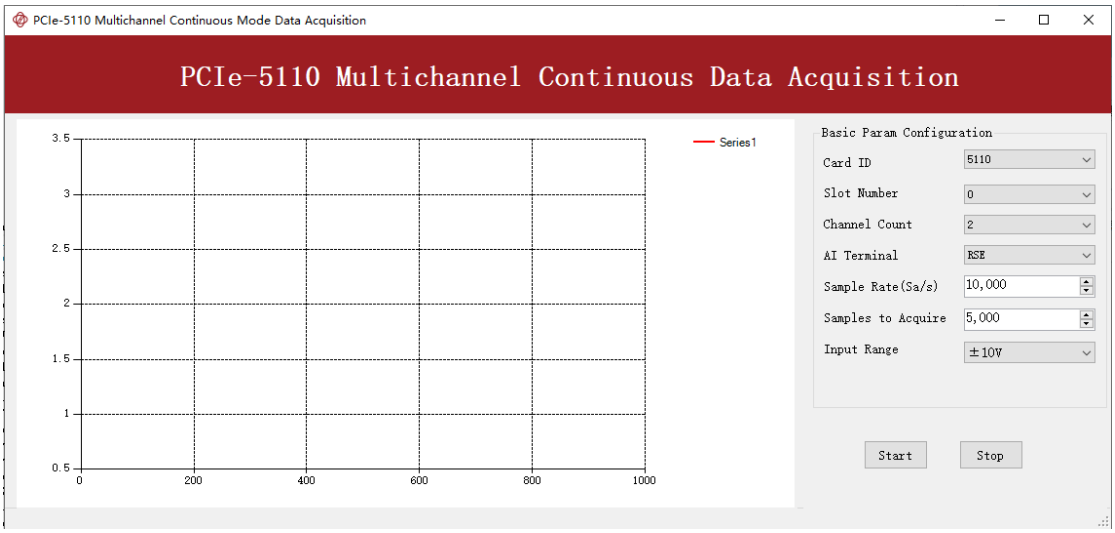


Figure 12 Immediate trigger Paraments

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

8.4.2. 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 8.4.2

- 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.

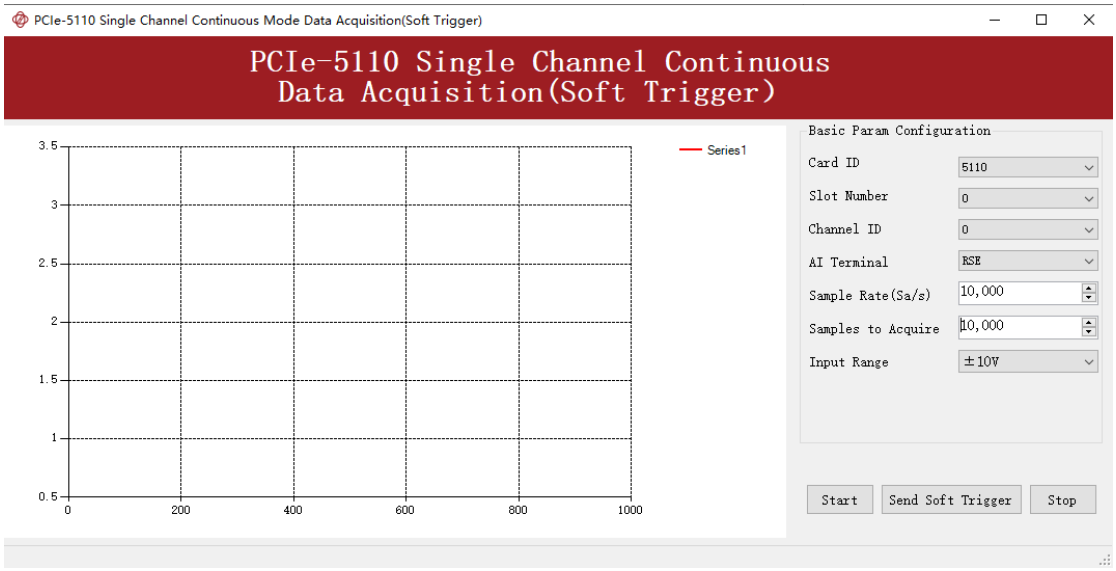


Figure 13 Software trigger Paraments

➢ 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:

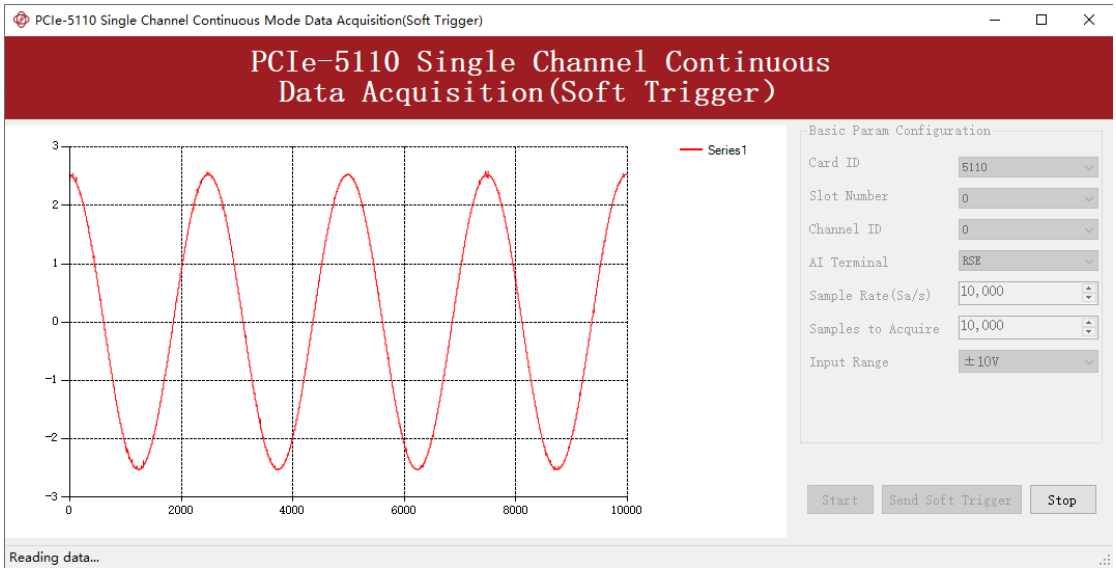


Figure 14 Software trigger Acquisition

8.4.3. External Analog Trigger

You can assign one of measurement channels as the analog trigger source. PCIe-5111 provides three analog trigger modes:

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

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

Edge comparator

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

Rising Slope Trigger: The Edge comparator output is high when the signal goes above the threshold; the output is low when the signal goes below the threshold as shown in Figure 15.

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

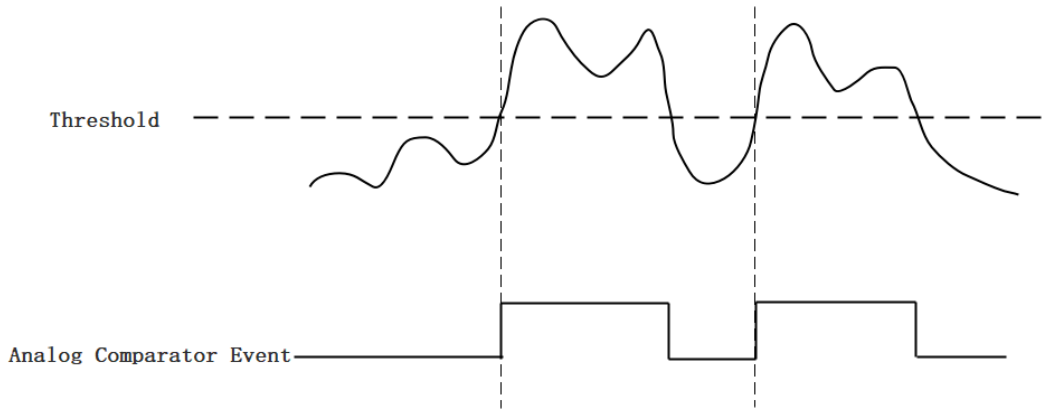


Figure 15 Rising Slope Trigger

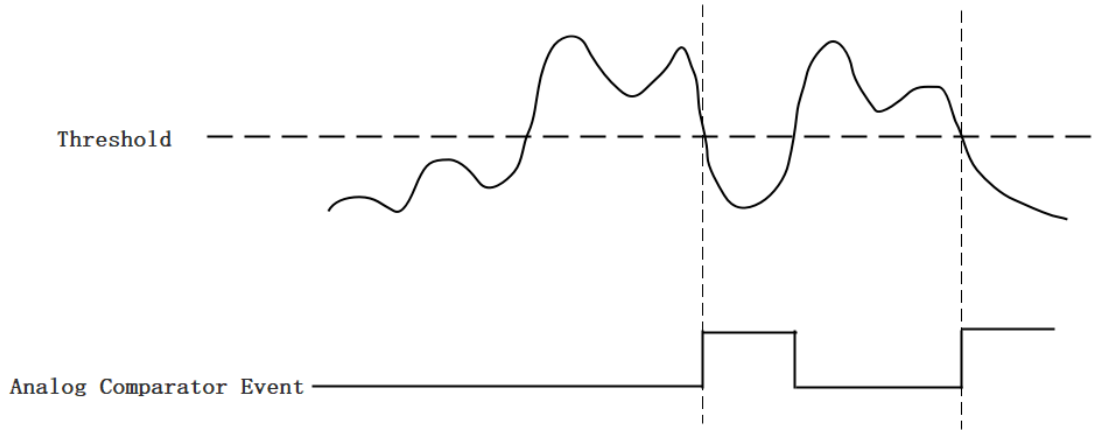


Figure 16 Falling Slope Trigger

Hysteresis Comparator

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

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

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

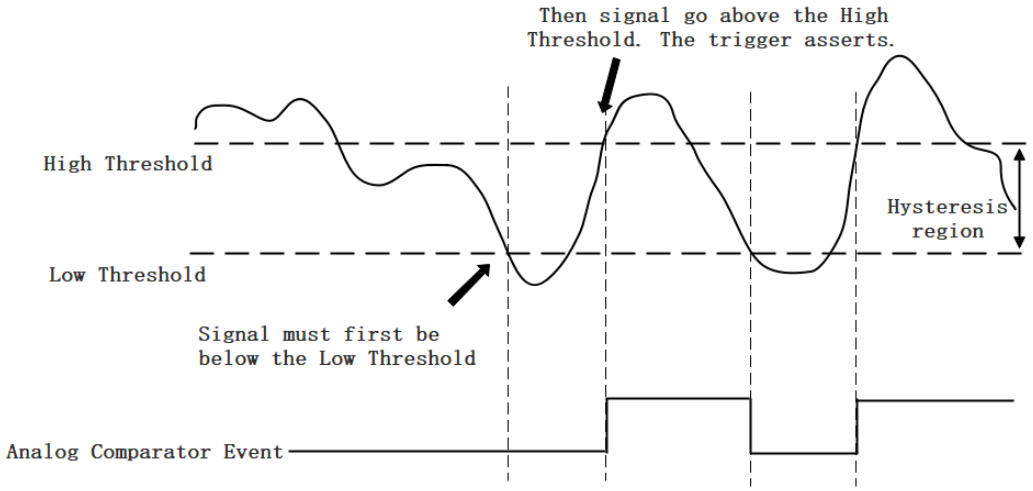


Figure 17 Hysteresis with Rising Slope Trigger

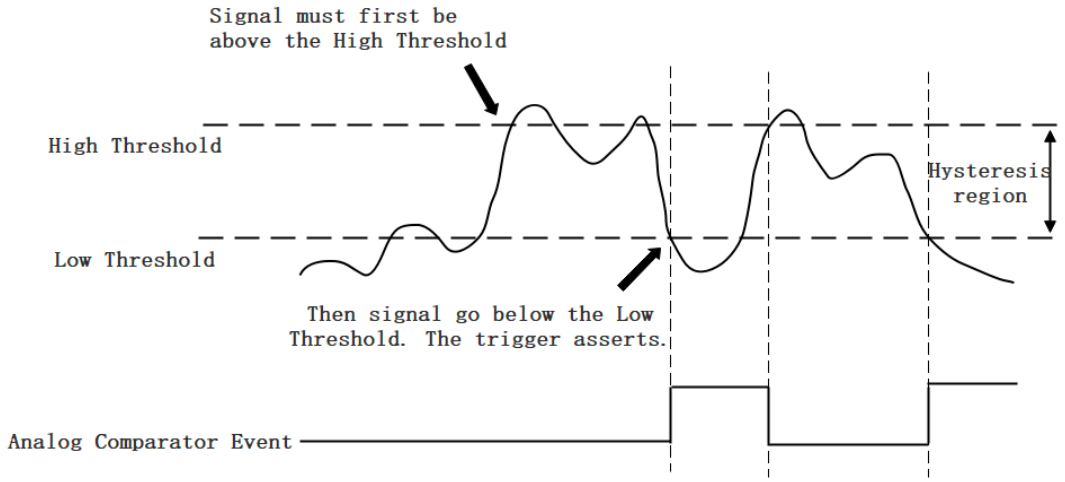


Figure 18 Hysteresis with Falling Slope Trigger

Window comparator

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

Entering Window Trigger: The window comparator output is high when the signal enters the window defined by the *Low Threshold* and *High Threshold*. The output will change to low when the signal leaves the window as shown in Figure 19.

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

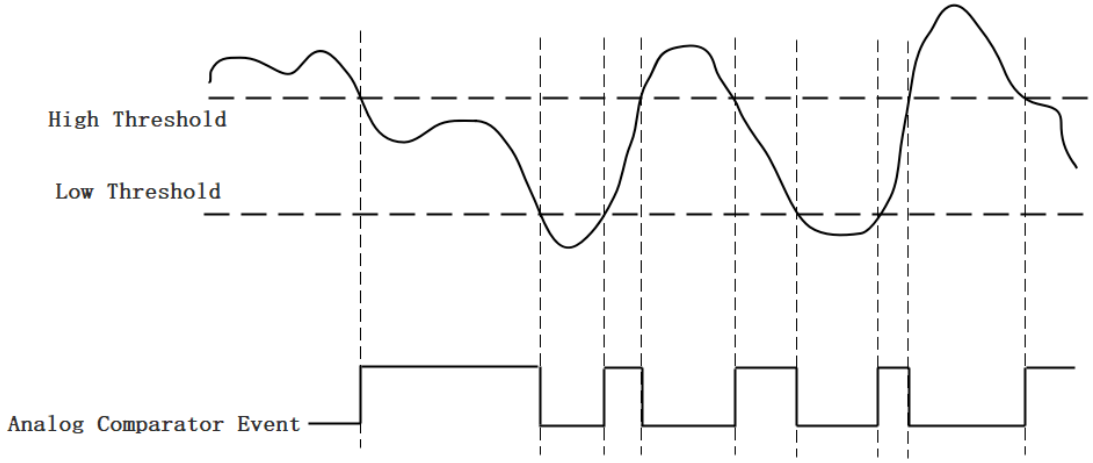


Figure 19 Entering Window Trigger

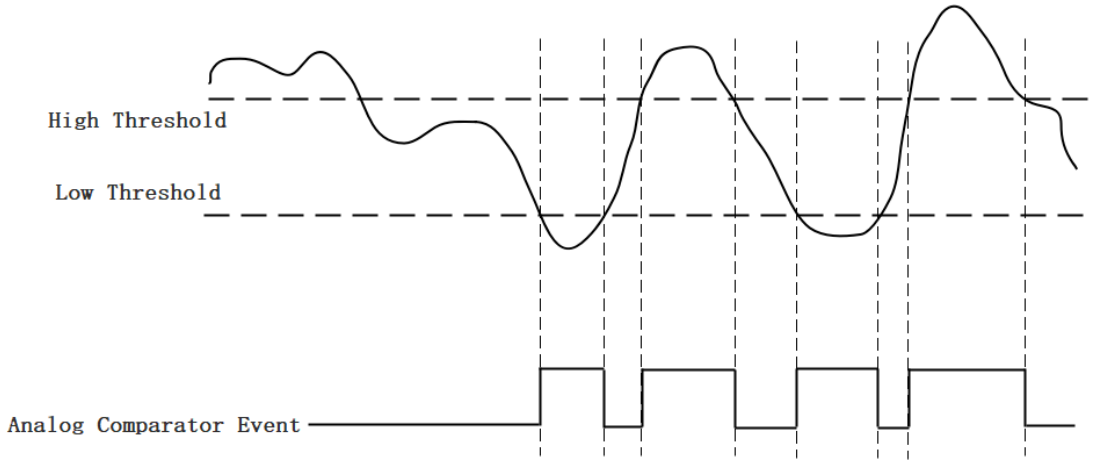


Figure 20 Leaving Window Trigger

Learn by Example 8.4.3

- 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.

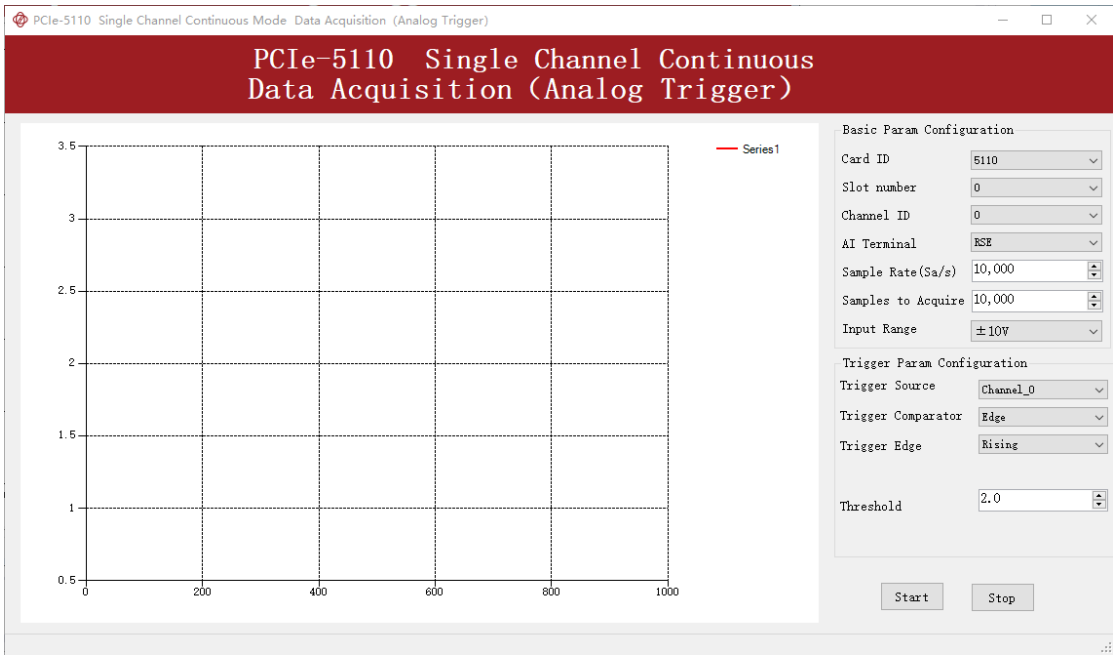


Figure 21 Analog Trigger Paraments

- 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:

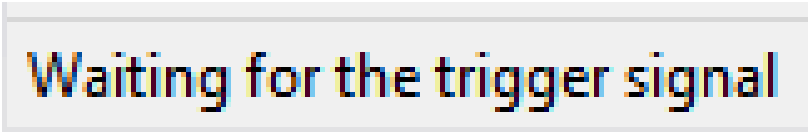


Figure 22 Waiting For Trigger

➢ 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 **8.4.3**.

■ The result is shown below:

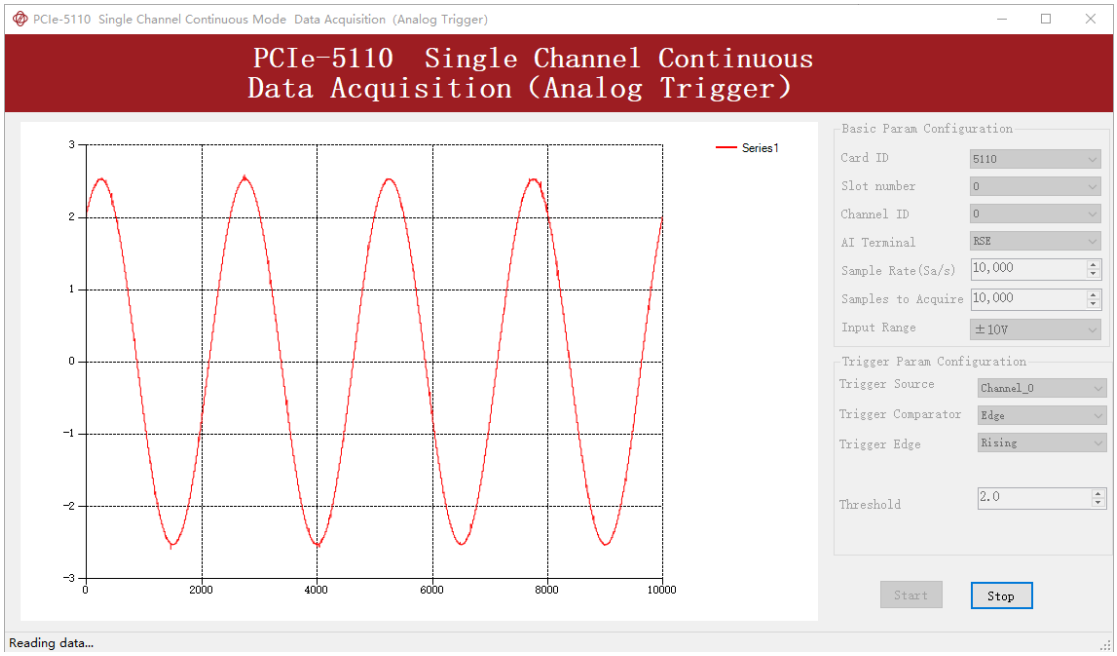


Figure 23 Analog Trigger Acquisition

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

8.4.4. External Digital Trigger

PCIe-5100 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 24

Figure 24 External Digital Trigger

Learn by Example 8.4.4

- 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.

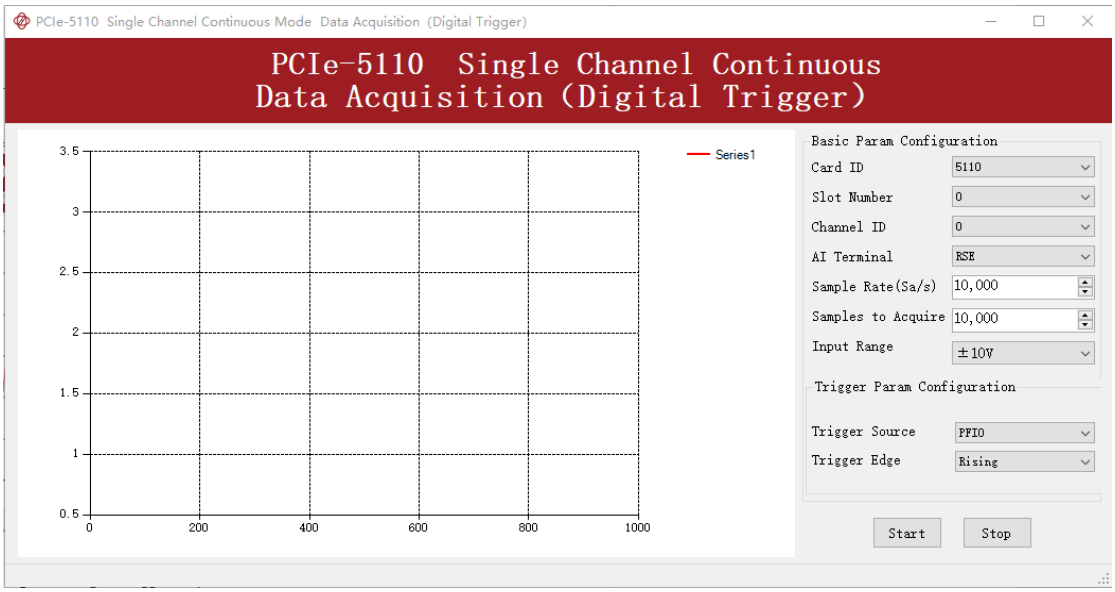


Figure 25 Digital Trigger Paraments

- **Trigger Source** must match the pin on 5110.

- There are two **Trigger Edge**: **Rising** and **Falling**.

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

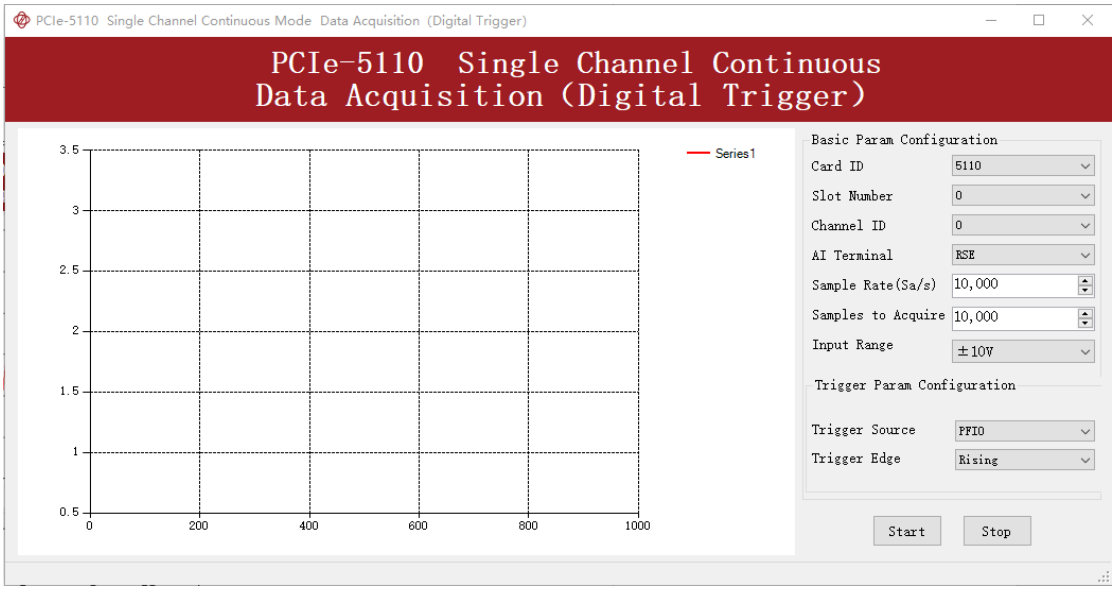


Figure 26 Digital Trigger Acquisition

- 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.

8.5. Trigger Mode

The PCIe-5111's analog inputs support several trigger modes: start trigger, reference trigger, and re-trigger.

8.5.1. 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 27.

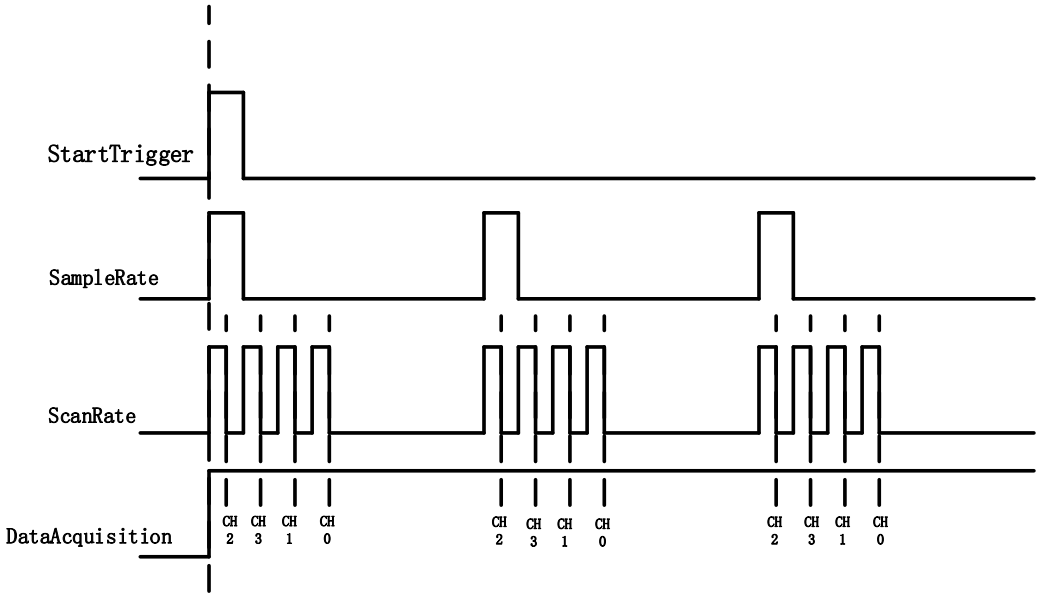


Figure 27 Start Trigger

8.5.2. 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 28.

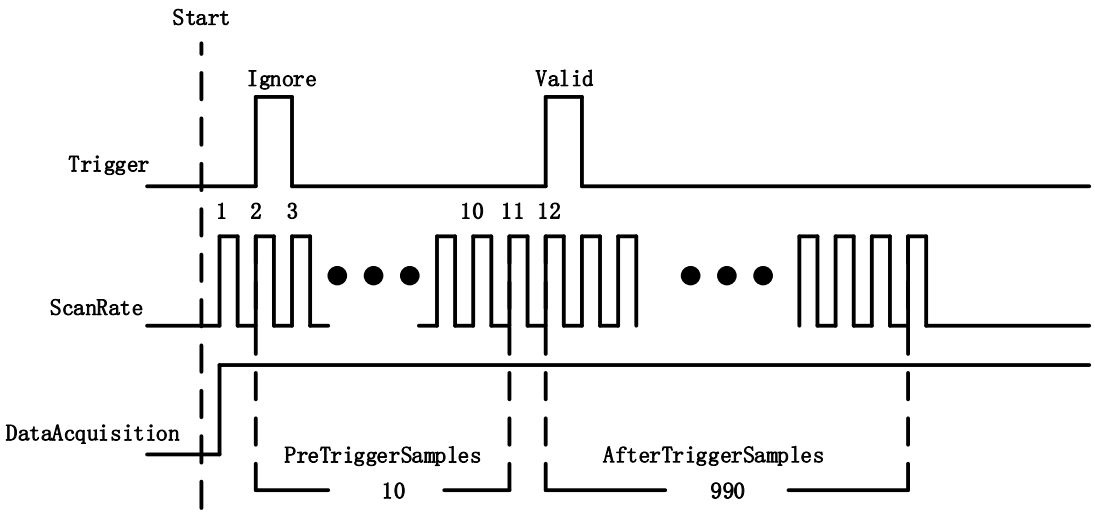


Figure 28 Reference Trigger

8.5.3. ReTrigger

PCIe-5111 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 29.

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

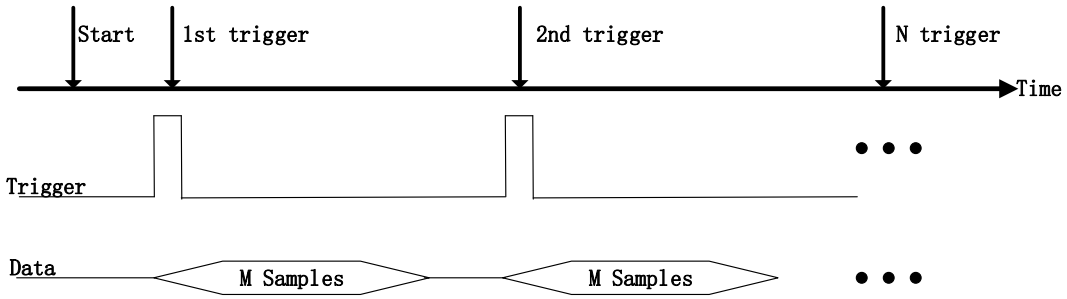


Figure 29 ReTrigger

Learn by Example 8.5

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.

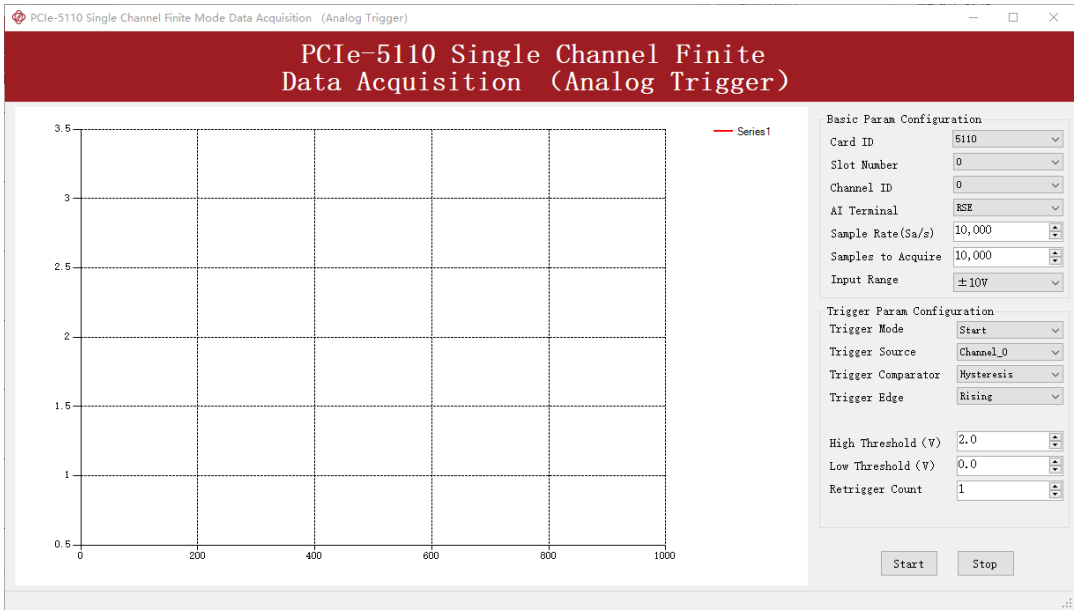


Figure 30 Retrigger Paraments

- You can use three different kinds of triggers in this program as mentioned in **8.5**. *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:

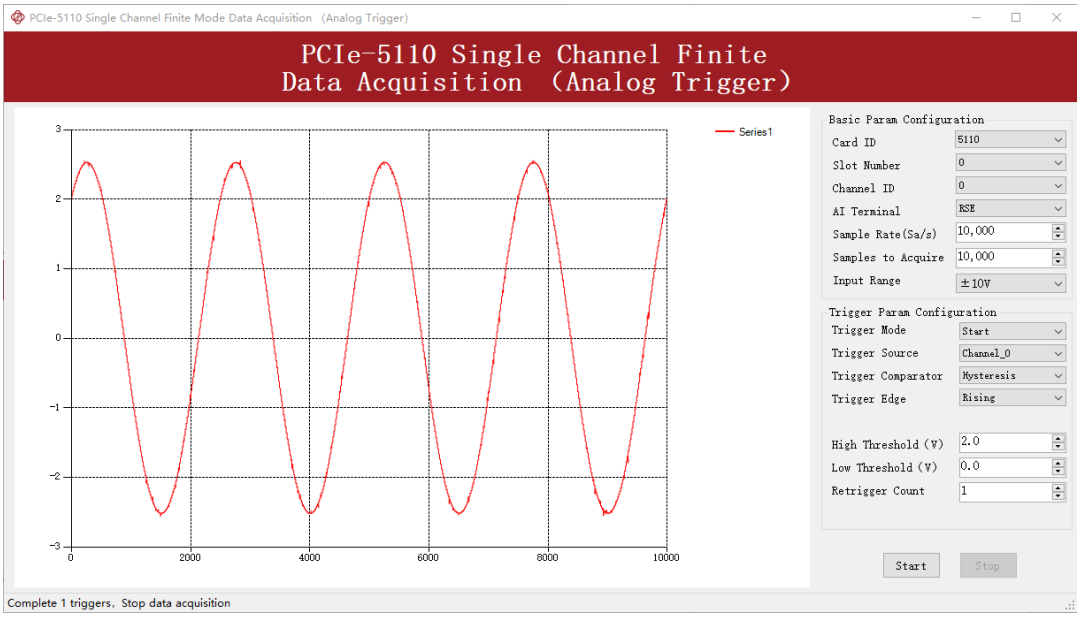


Figure 31 Retrigger In Start Trigger Mode

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

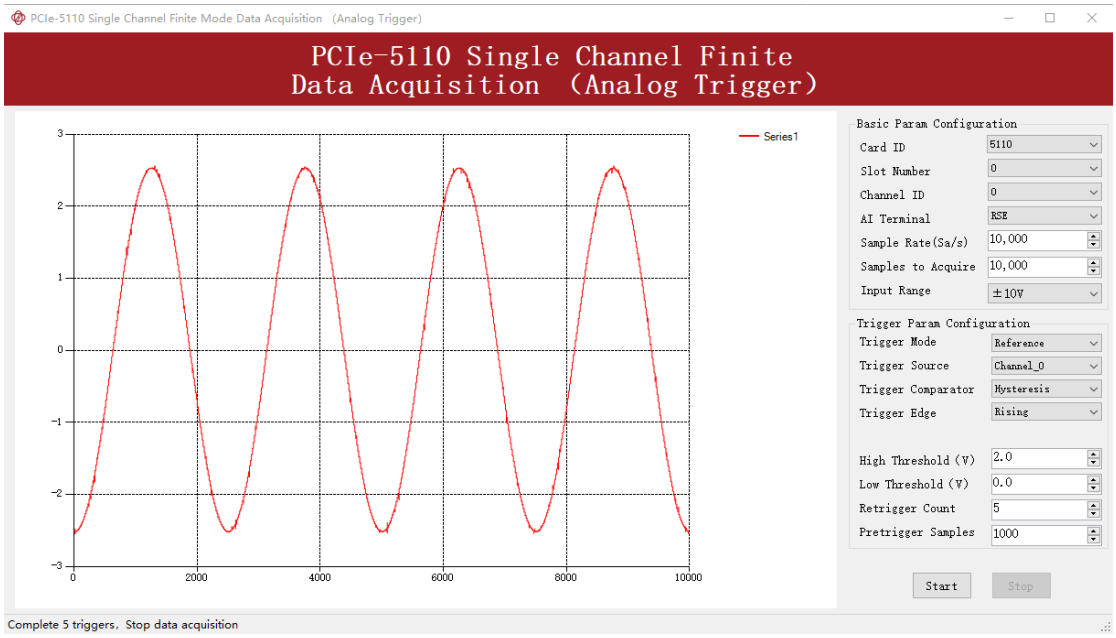


Figure 32 Retrigger In Reference Trigger Mode

- 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".

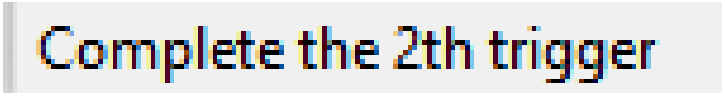


Figure 33 Complete Retrigger Count

  - It shows the acquisition process through every trigger signal.

8.6. AO Operations

The PCIe-5111 AO provides 16-bit simultaneous outputs. The analog output has three modes of operation: Finite, ContinuousWrapping, and ContinuousNoWrapping.

8.6.1. 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 8.6.1

- 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.

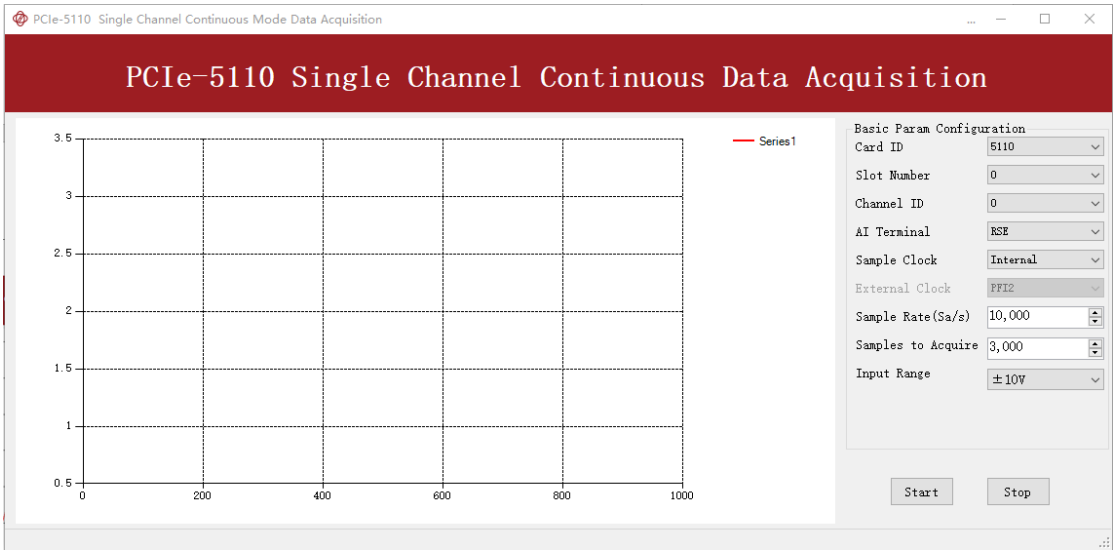


Figure 34 AI Continuous Paraments

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

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

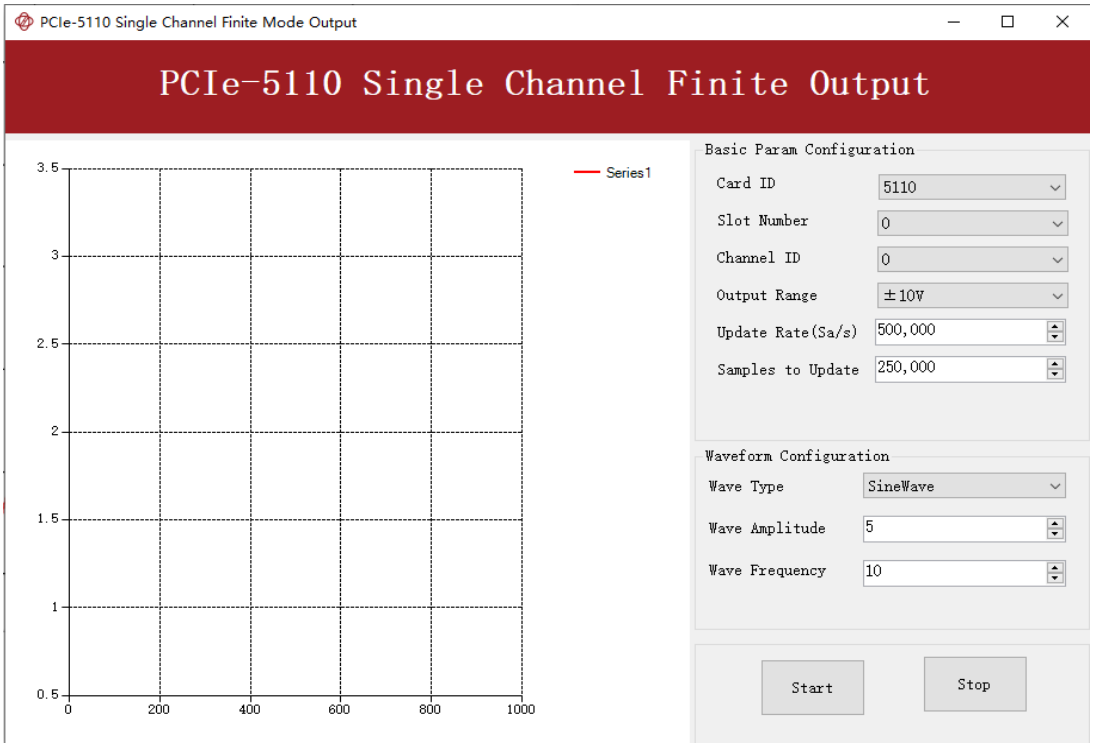


Figure 35 AO Finite Output Paraments

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

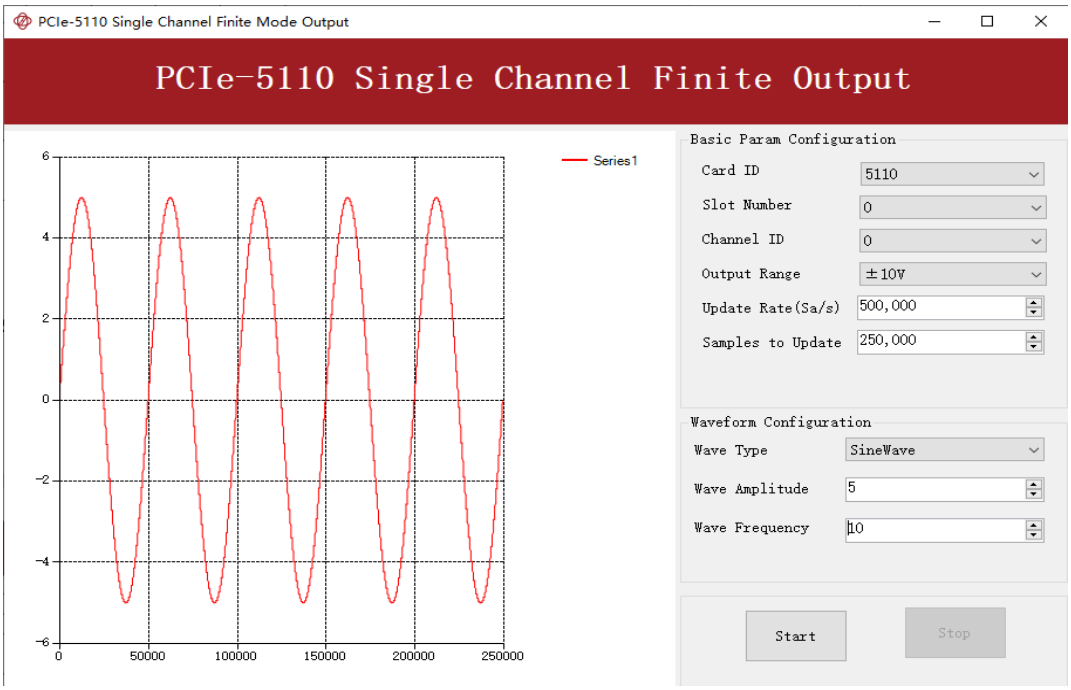


Figure 36 AO Finite Signal

■ And the received signal is shown below.

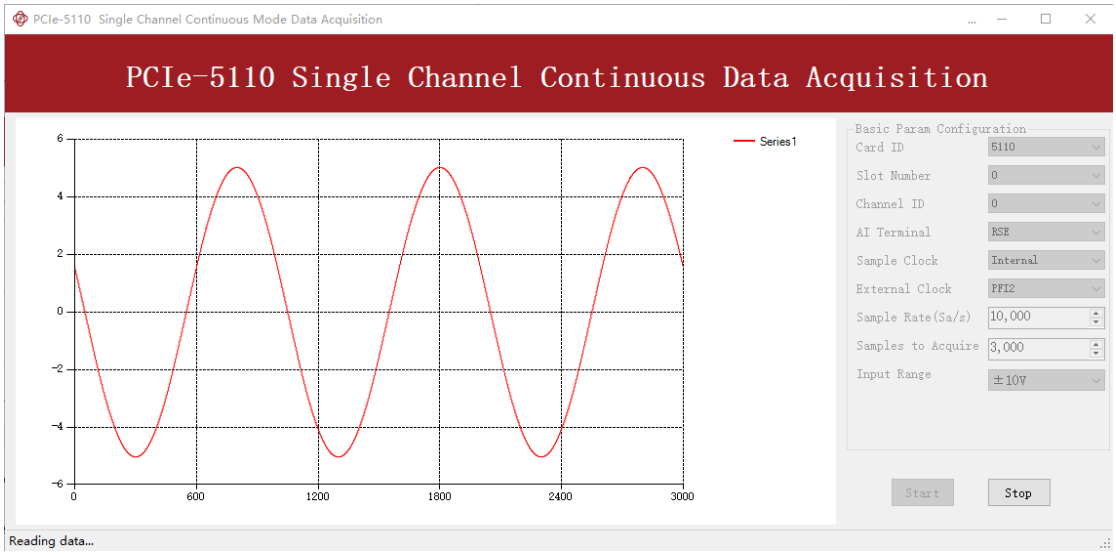


Figure 37 AI Acquisition Signal

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

8.6.2. 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 8.6.2

- 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.

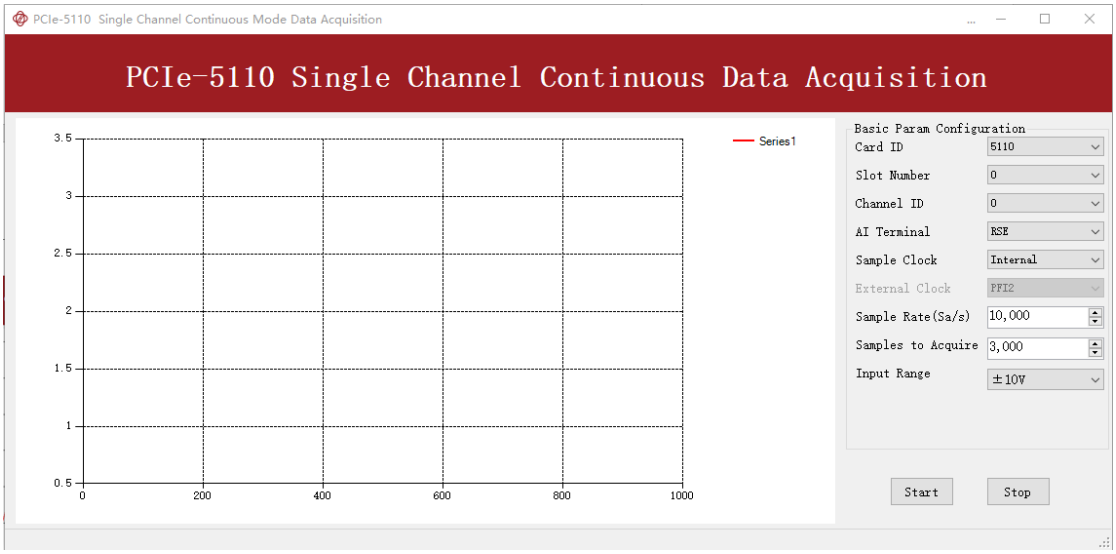


Figure 38 AI Continuous Paraments

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

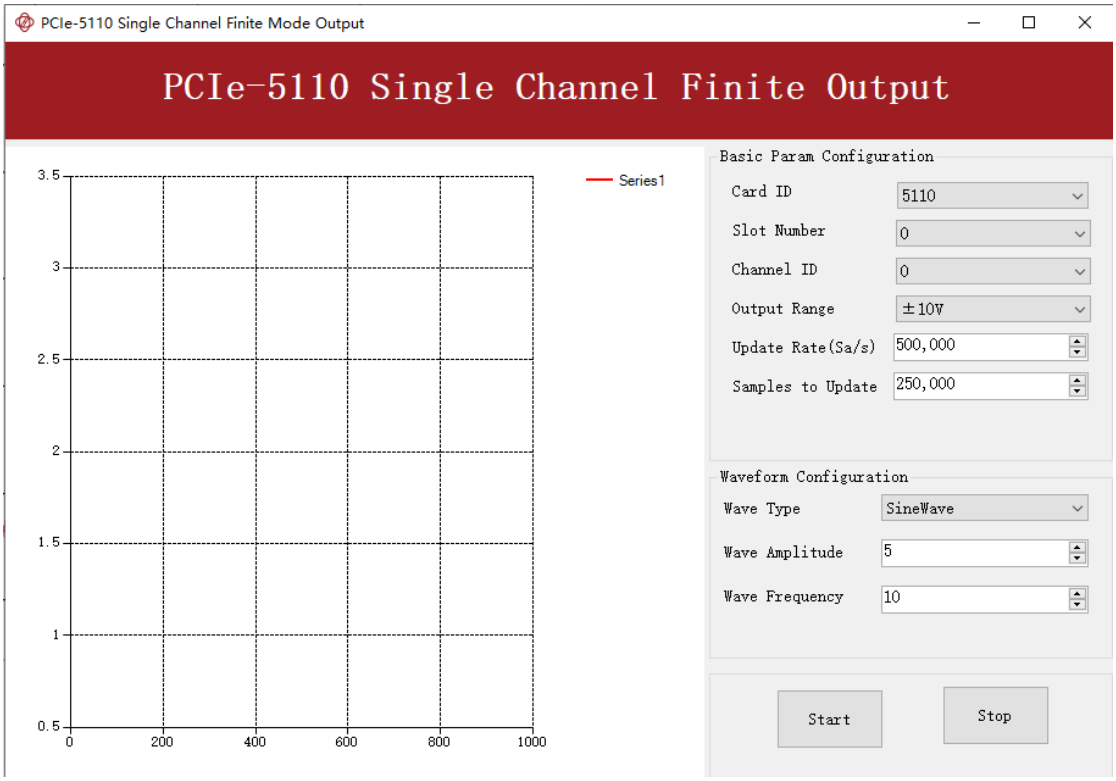


Figure 39 AO ContinuousNoWrapping Output Paraments

- 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.

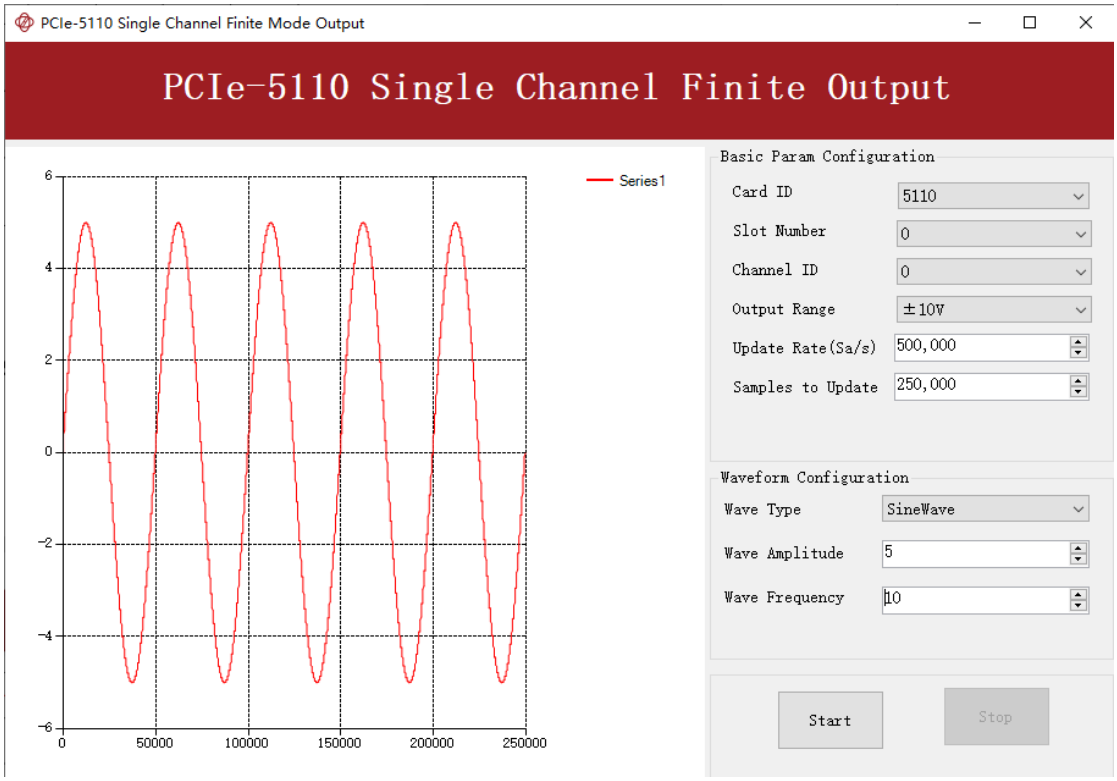


Figure 40 AO ContinuousNoWrapping Signal

- And the received signal is shown below.

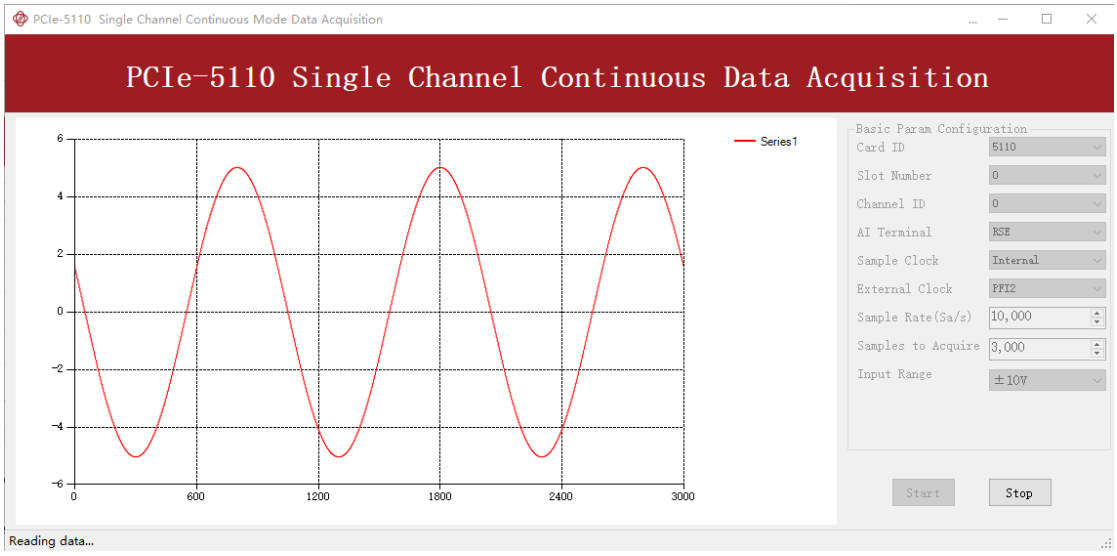


Figure 41 AI Acquisition AO Sin Signal

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

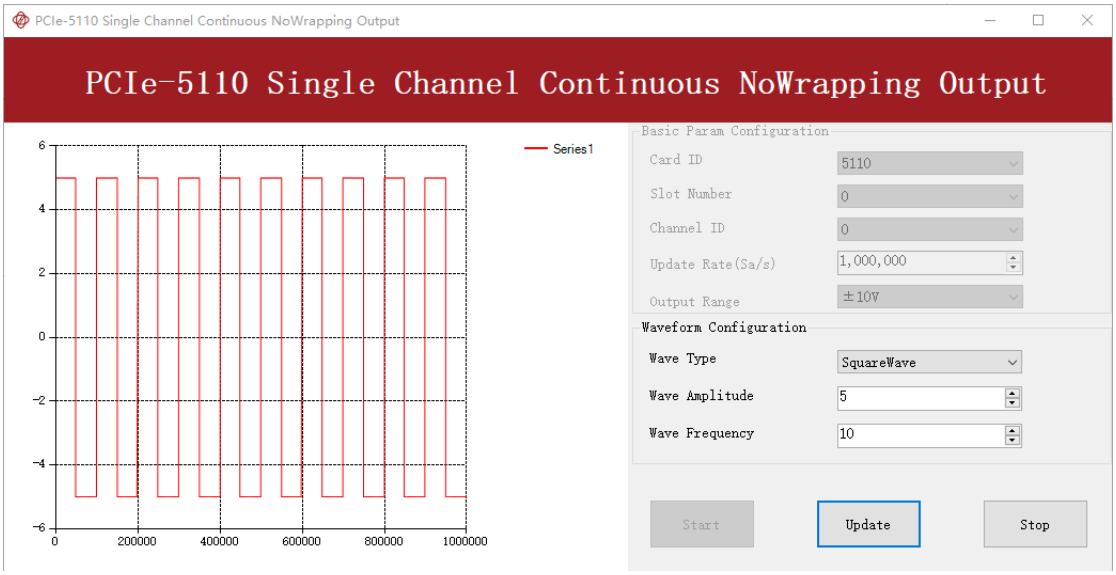


Figure 42 Update AO Square Signal

- And the received signal is shown below.

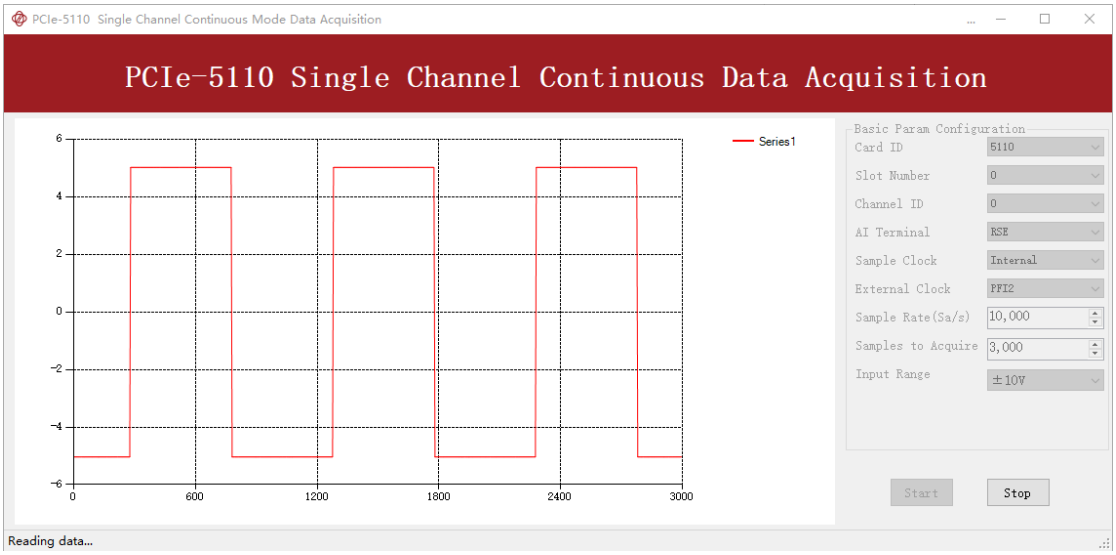


Figure 43 AI Acquisition AO Square Signal

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

8.6.3. 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 8.6.3

- 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.

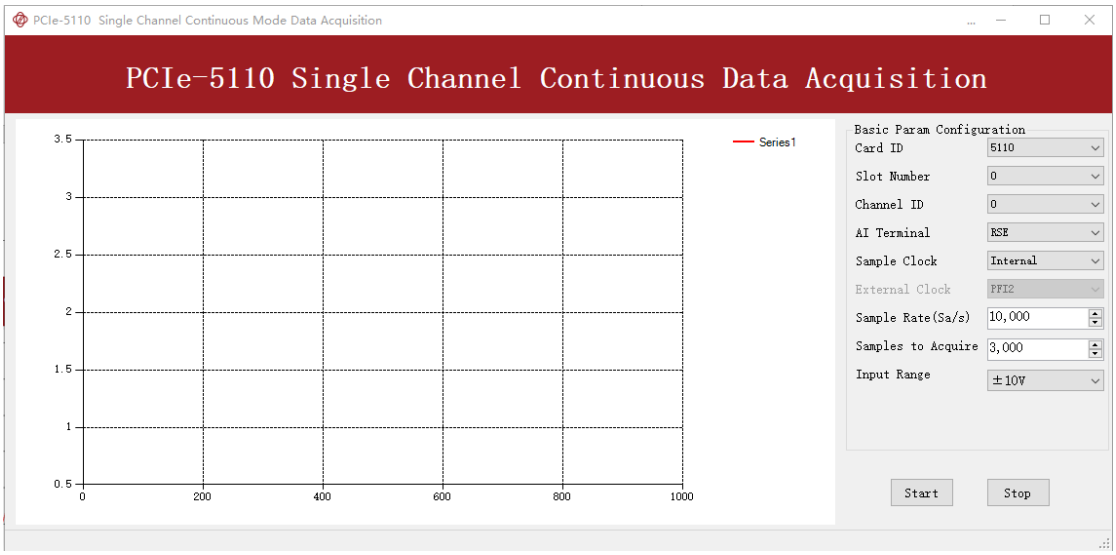


Figure 44 AI Continuous Paraments

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

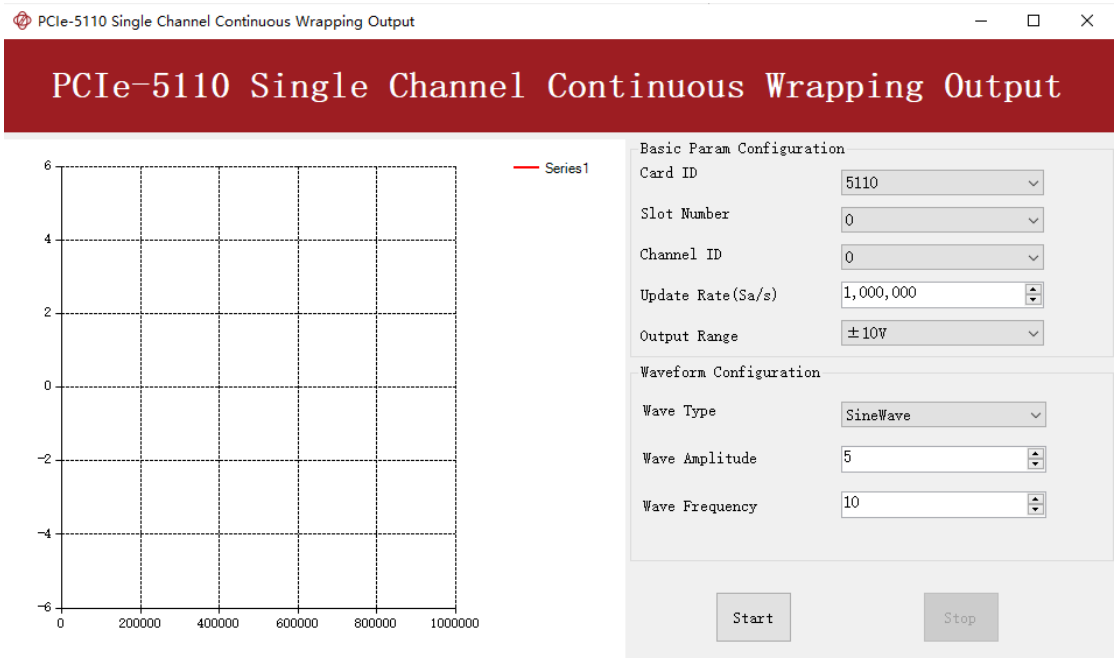


Figure 45 AO Continuous Wrapping Paraments

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

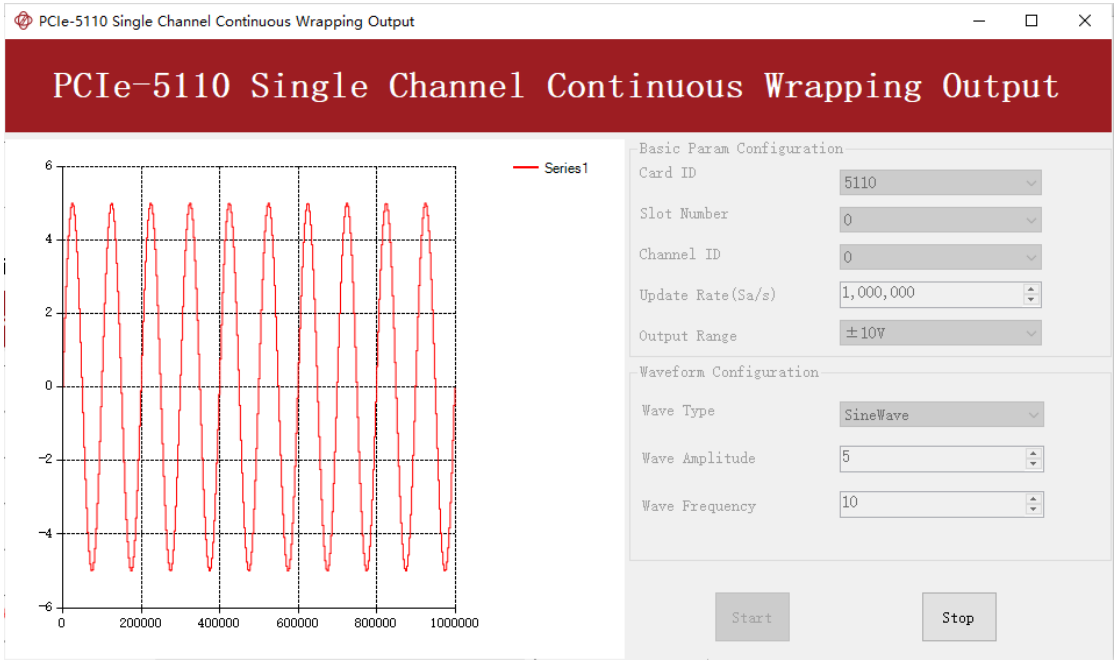


Figure 46 AO Continuous Wrapping Signal

- And the received signal is shown below.

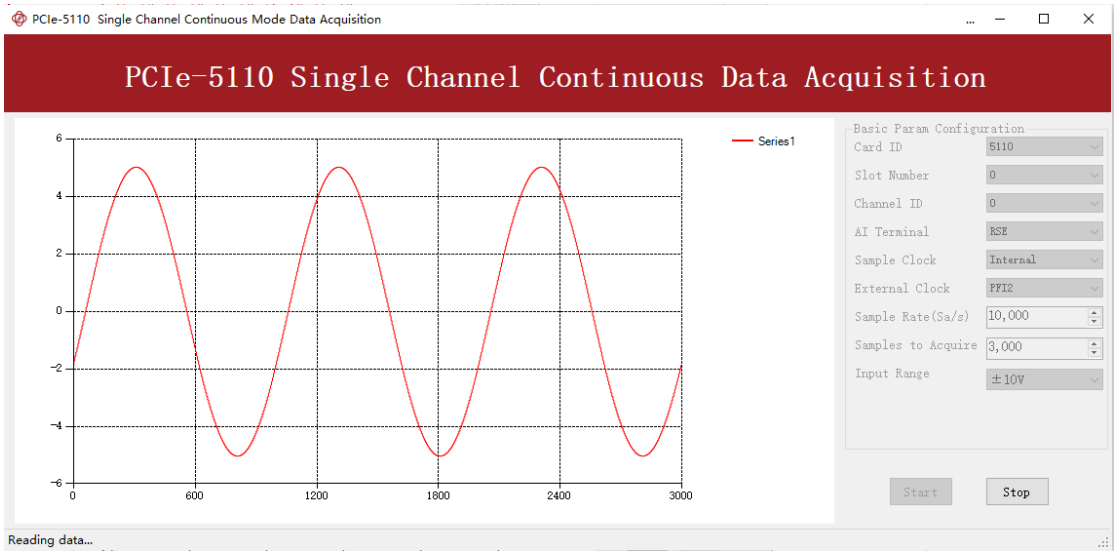


Figure 47 AI Acquisition AO Signal

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

8.7. Digital I/O Operations

The PCIe-5111 provides powerful programmable digital I/O functions.

8.7.1. Static DI/DO

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

Learn by Example 8.7.1

- 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.

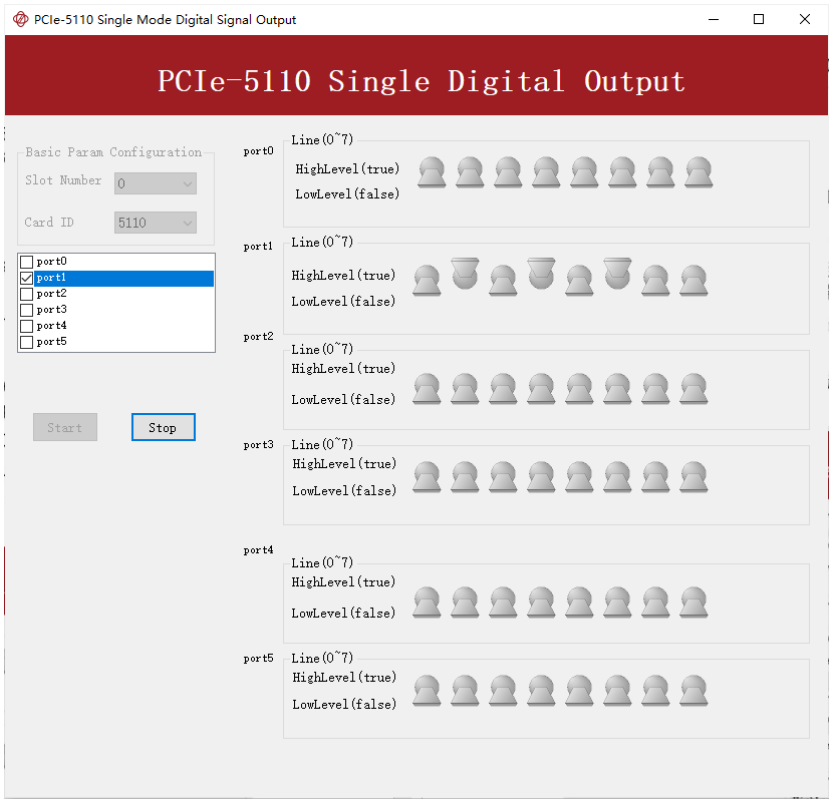


Figure 48 Single Digital Output

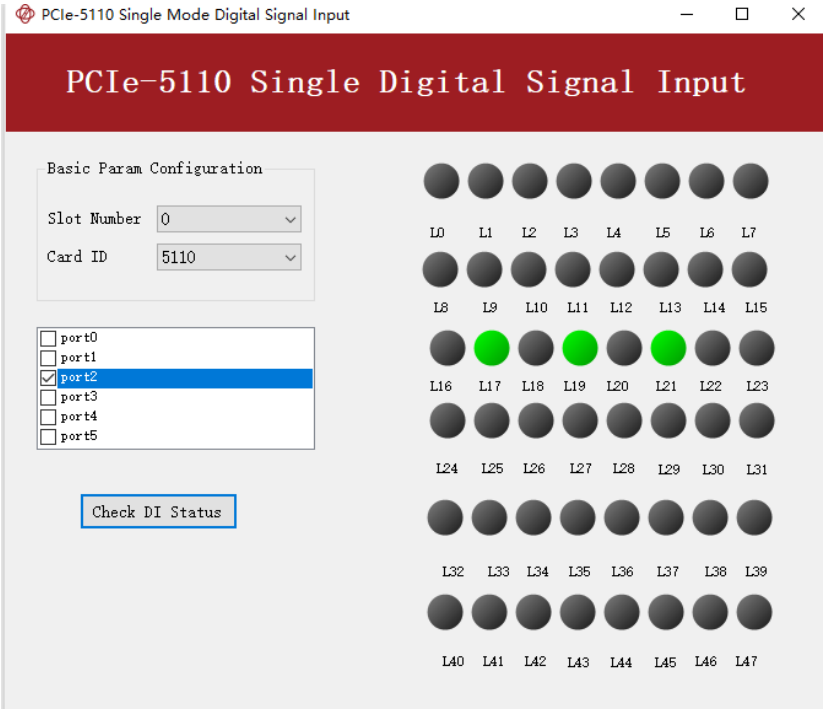


Figure 49 Single Digital Input

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

8.7.2. Dynamic DI/DO

The PCIe-5111 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 8.7.2

- 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-5111/5111 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**(PCIe-5110/5111) or **port 1**(PCIe-5110/5111).

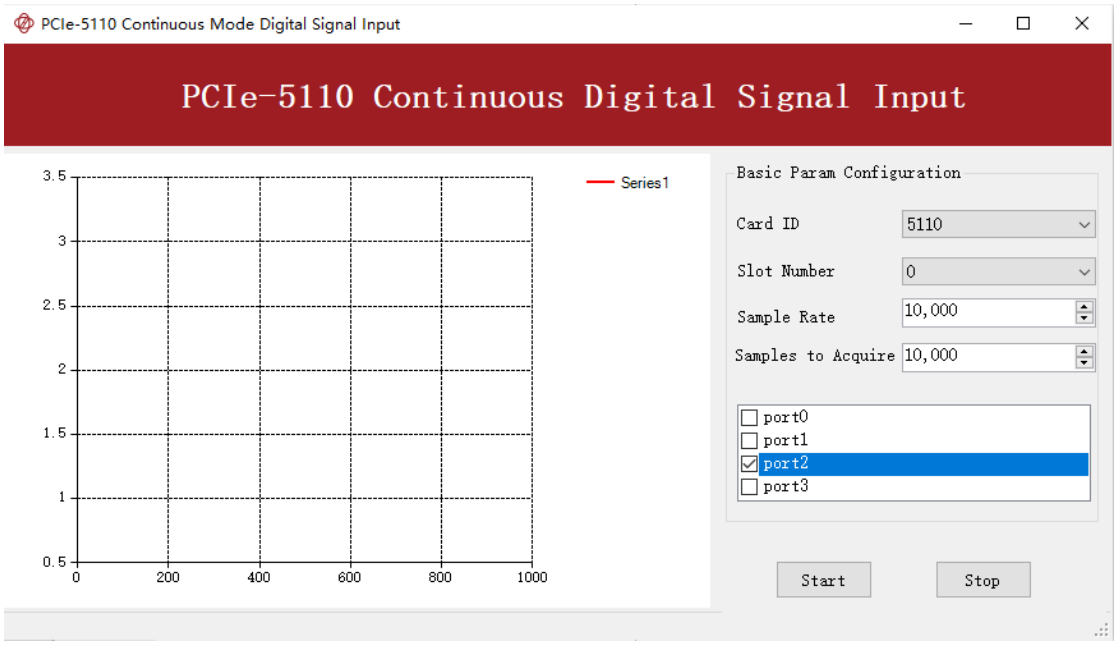


Figure 50 DI Continuous Paraments

- 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.

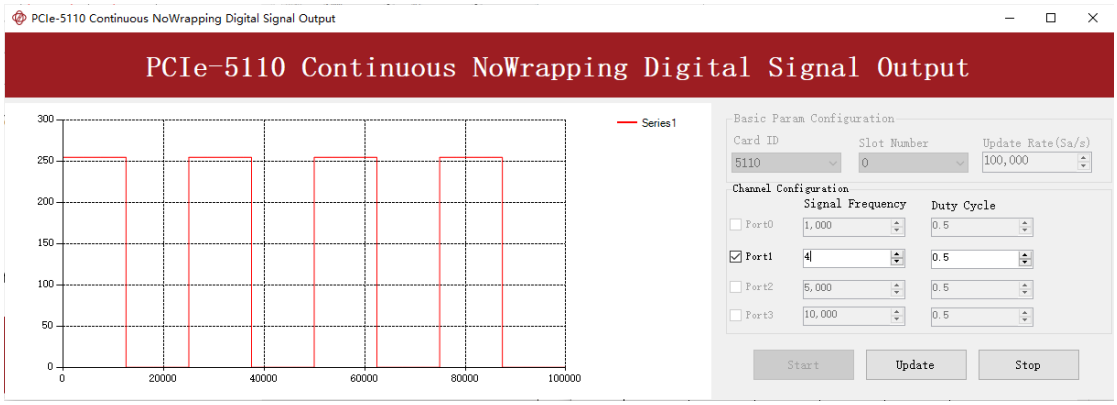


Figure 51 DO ContinuousNoWrapping Output

- In program **Winform DI Continuous**, you can see the acquired signal. Select **port 1**(PCIe-5110/5111) or **port 0**(PCIe-5110/5111).

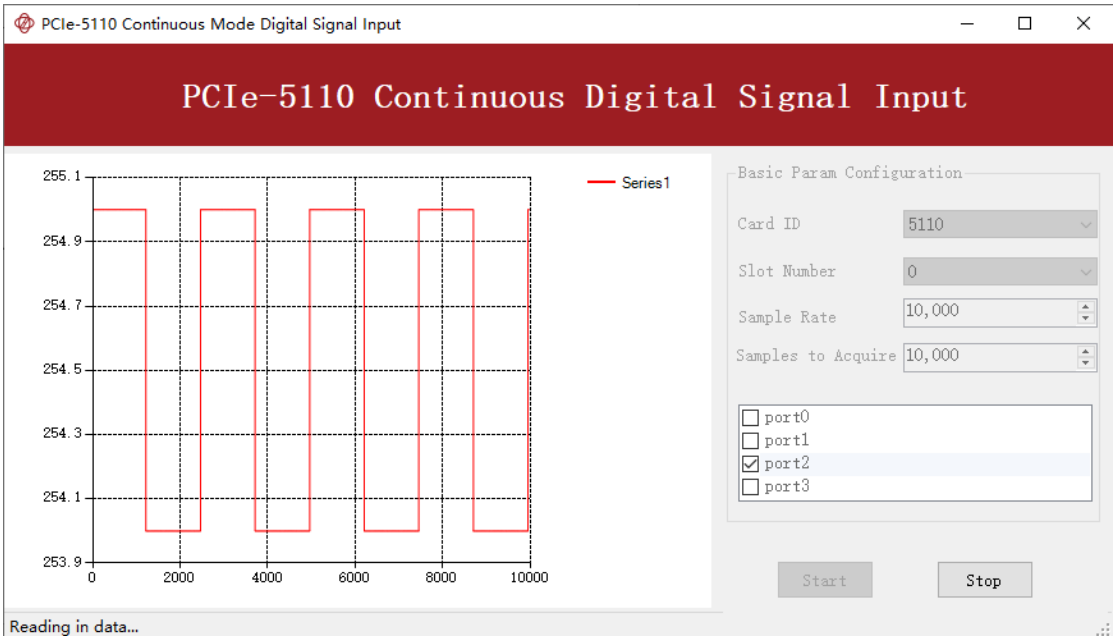


Figure 52 DI Continuous Acquisition

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

8.8. Counter Input Operations

The PCIe-5111 has four or two identical 32 bits timers/counters.

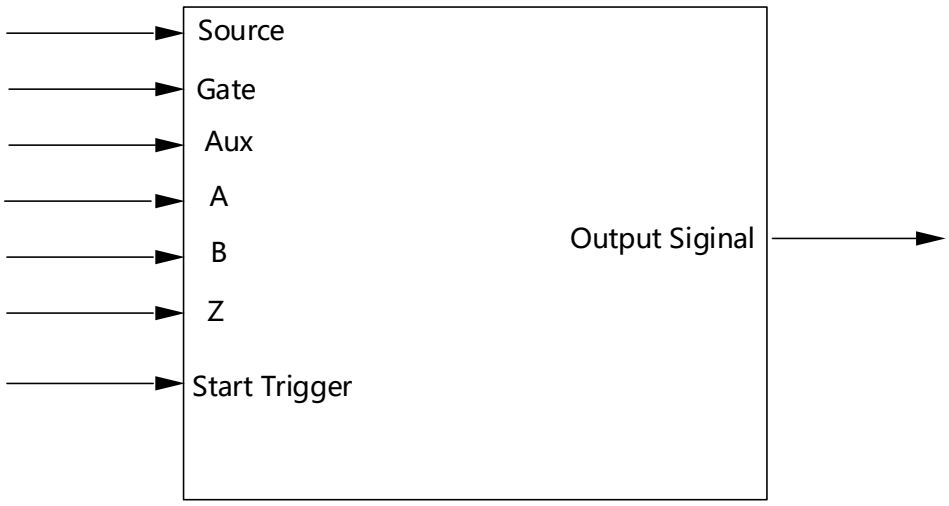


Figure 53 Counter Terminal

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 54 Counter Signal Wiring Instruction

8.8.1. 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 55.

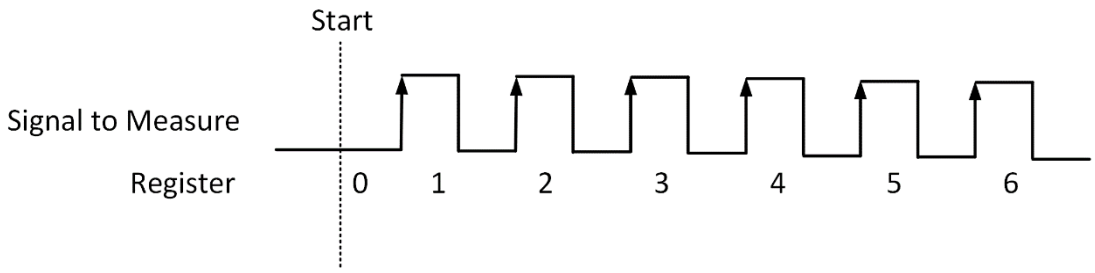


Figure 55 Simple Edge Counting in Single Mode

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 56.

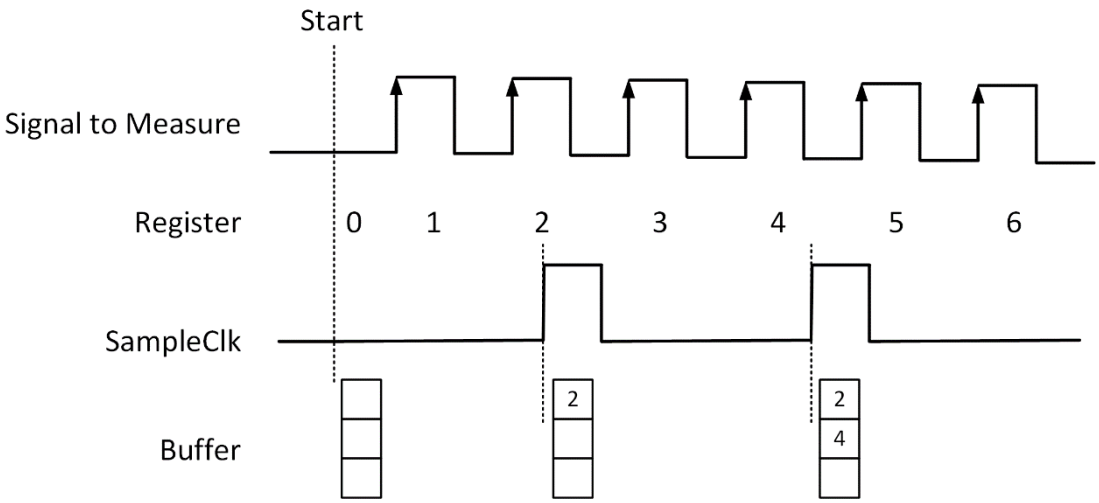


Figure 56 Buffered Edge Counting with Internal Sample Clock

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 57.

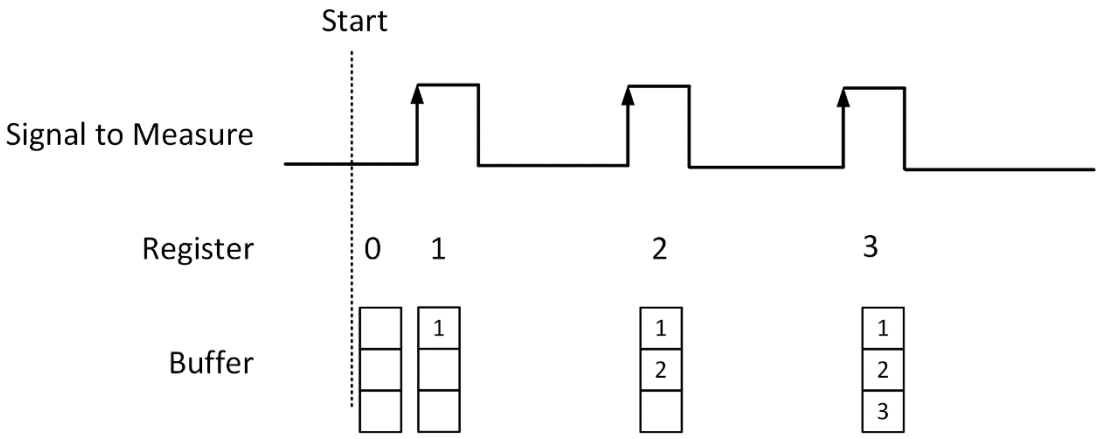


Figure 57 Simple Edge Counting with Implicit SampleClk

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 58.

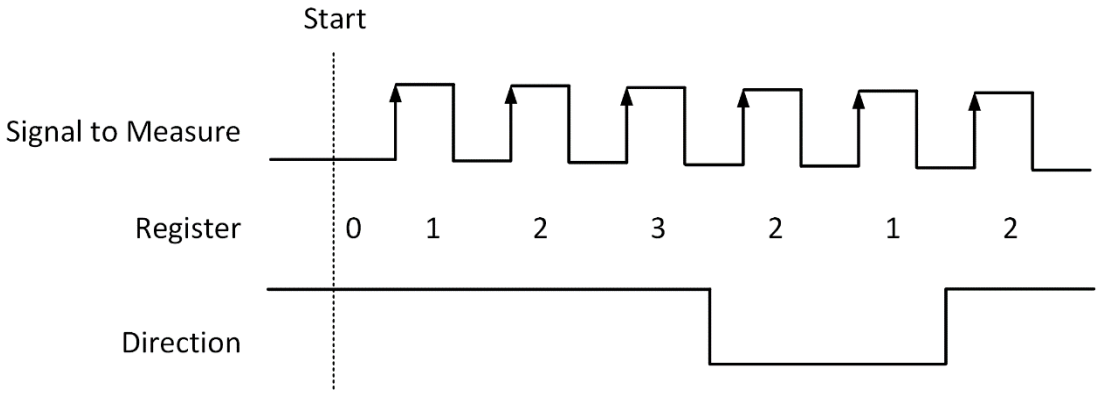


Figure 58 Count Direction

Learn by Examples8.8.1

- 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:

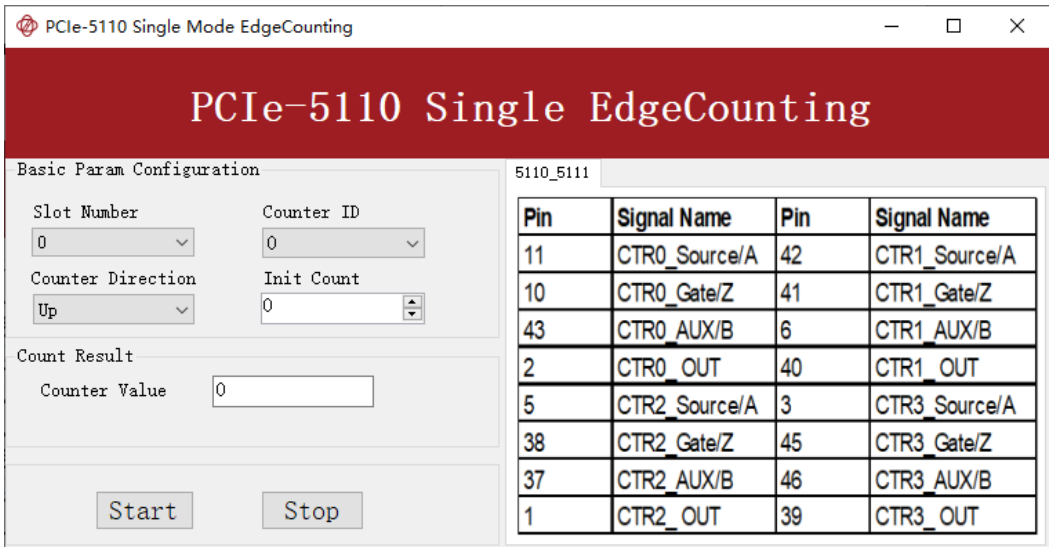


Figure 59 EdgeCounting For Single Mode

- 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:

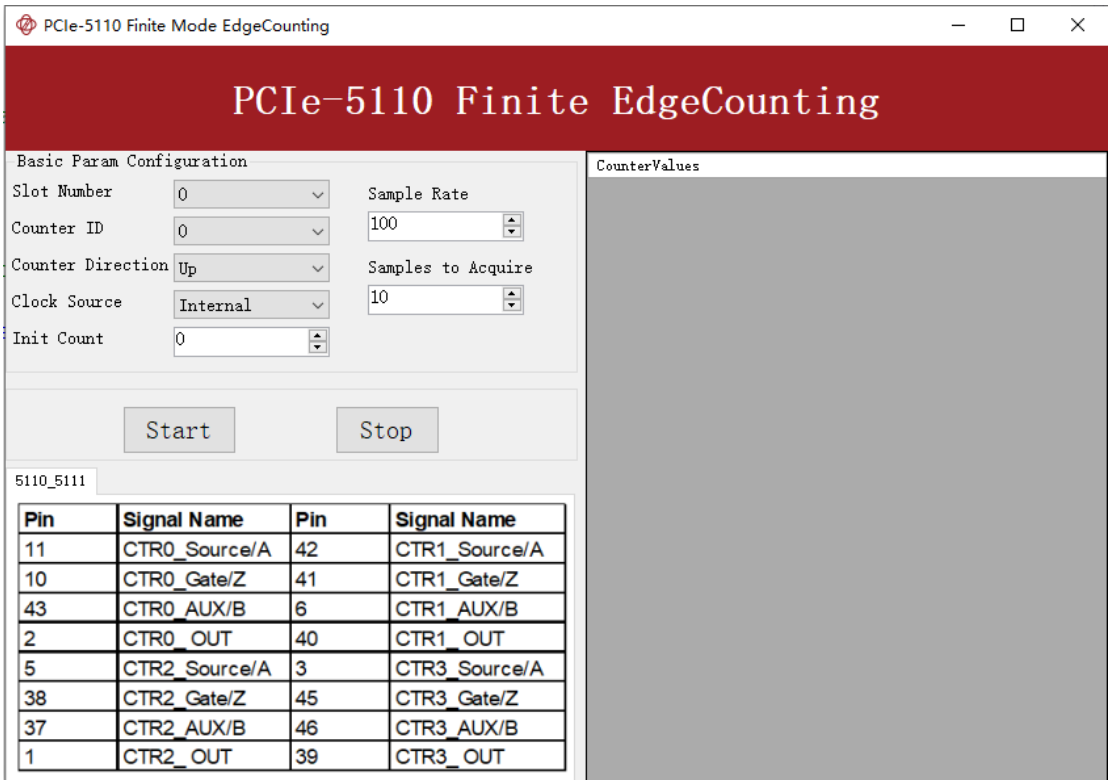


Figure 60 EdgeCounting For Finite Mode

- 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:

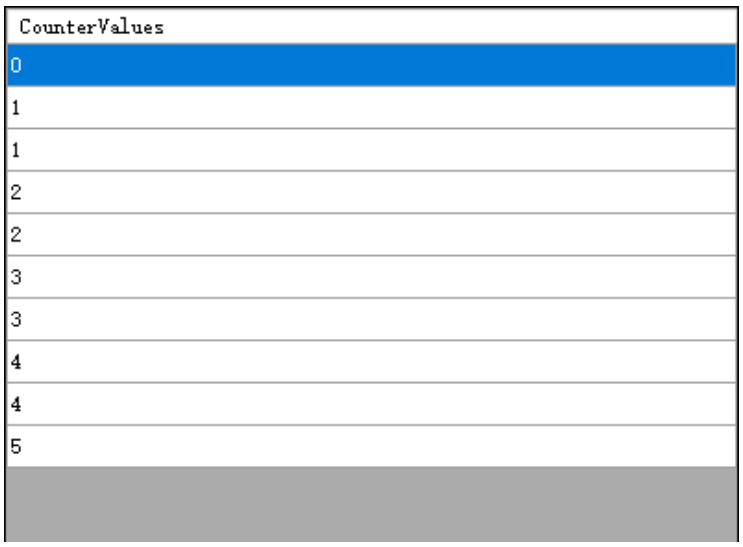


Figure 61 Counter Values For Internal Clock

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

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

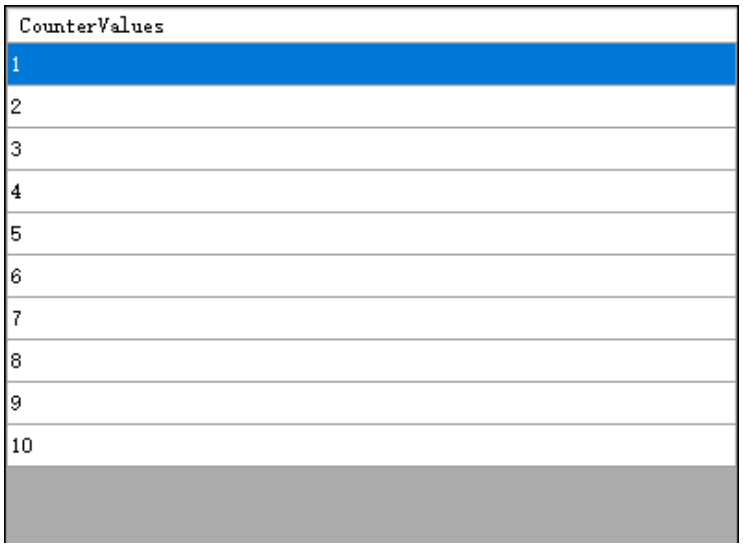


Figure 62 Counter Values For Implicit Clock

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

8.8.2. 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 63.

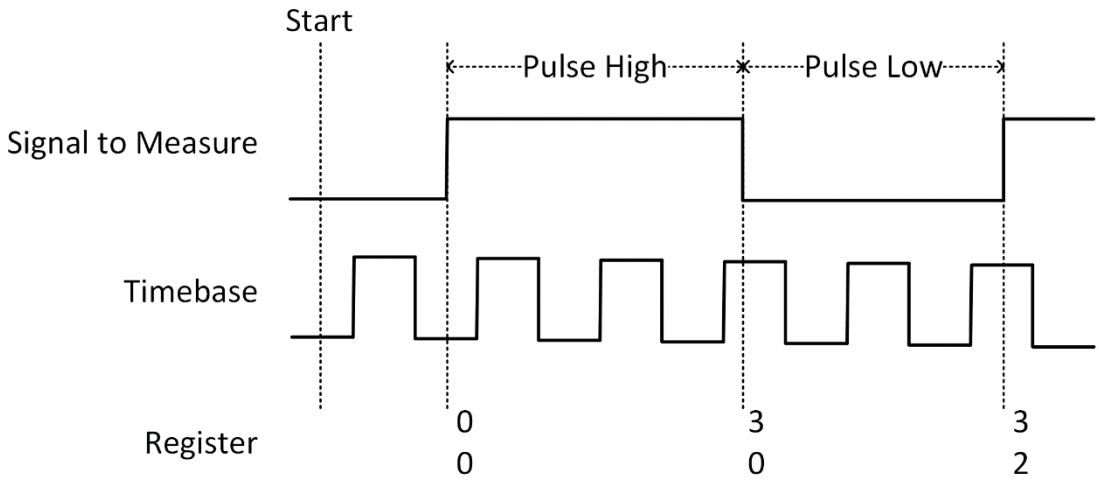


Figure 63 Pulse Measurement in Single Mode

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 64.

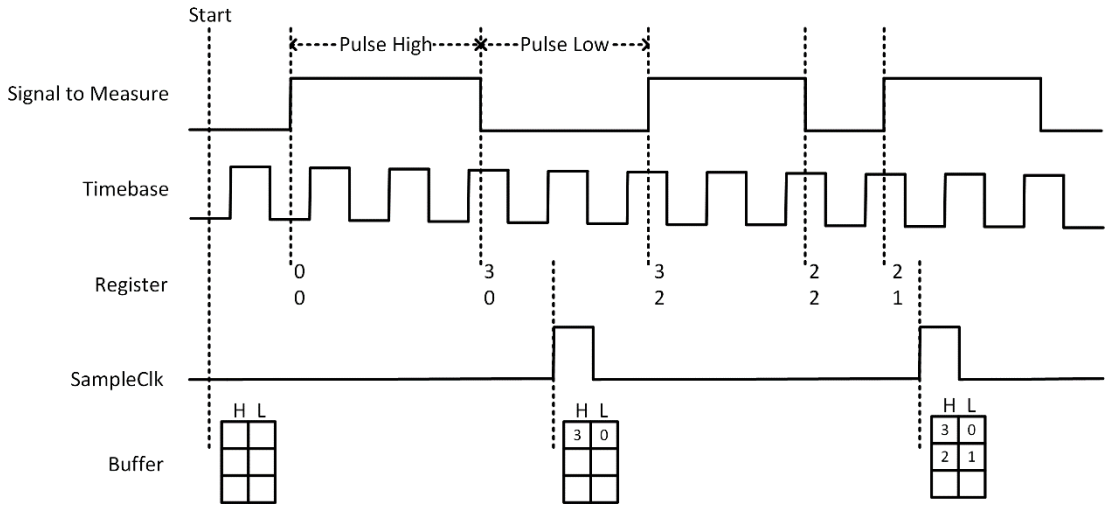


Figure 64 Pulse Measurement with Internal SampleClk

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 65.

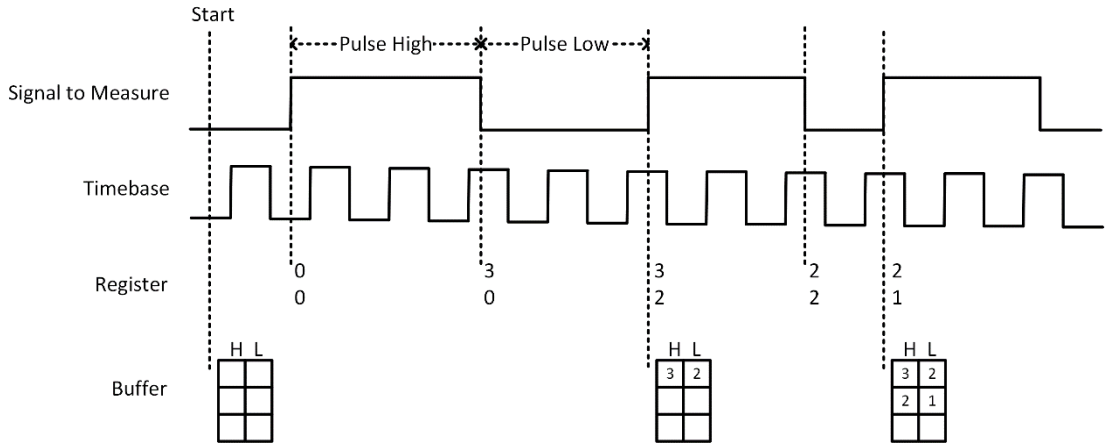


Figure 65 Pulse Measurement with Implicit SampleClk

Learn by Examples 8.8.2

- 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:

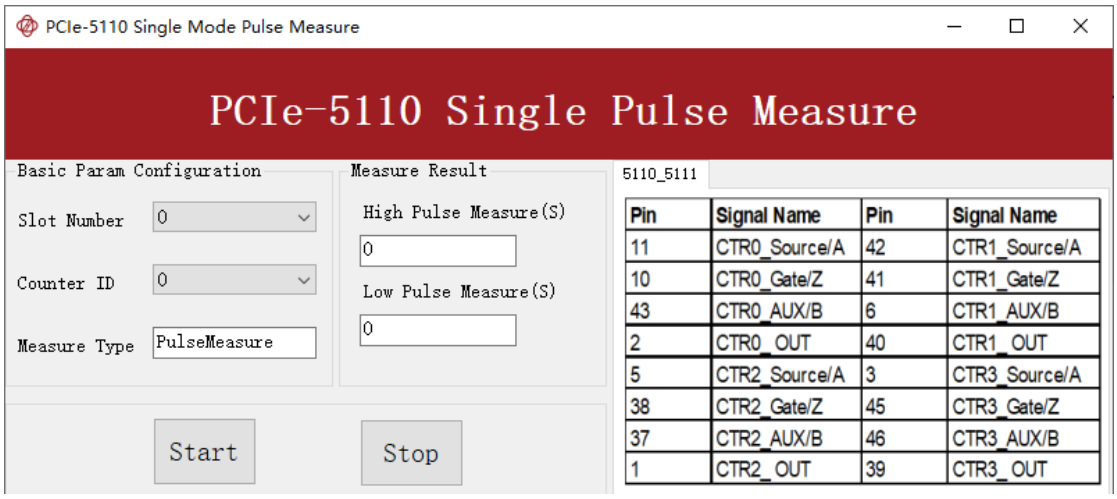


Figure 66 Pulse Measure For Single Mode

➢ 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)**:

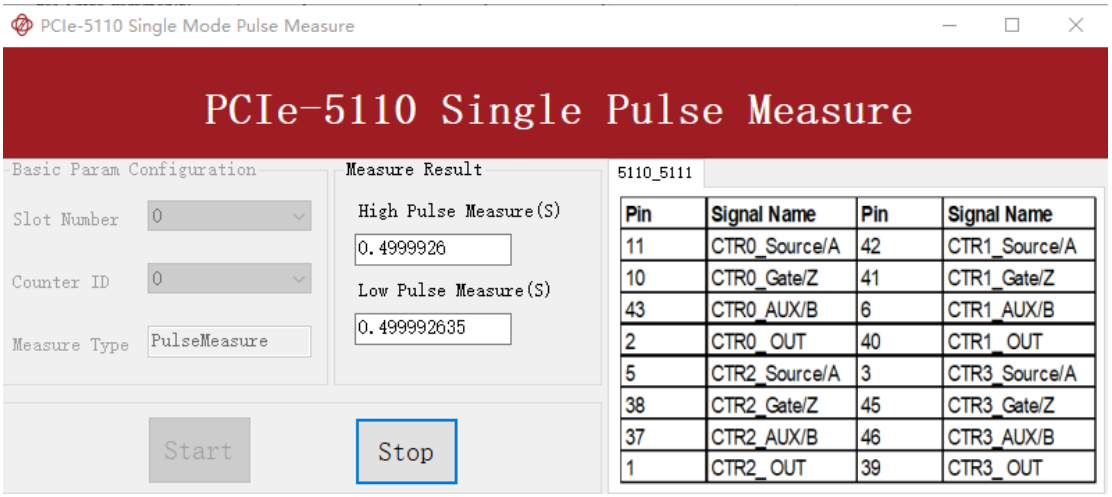


Figure 67 Pulse Measure Value For Single Mode

➢ 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:

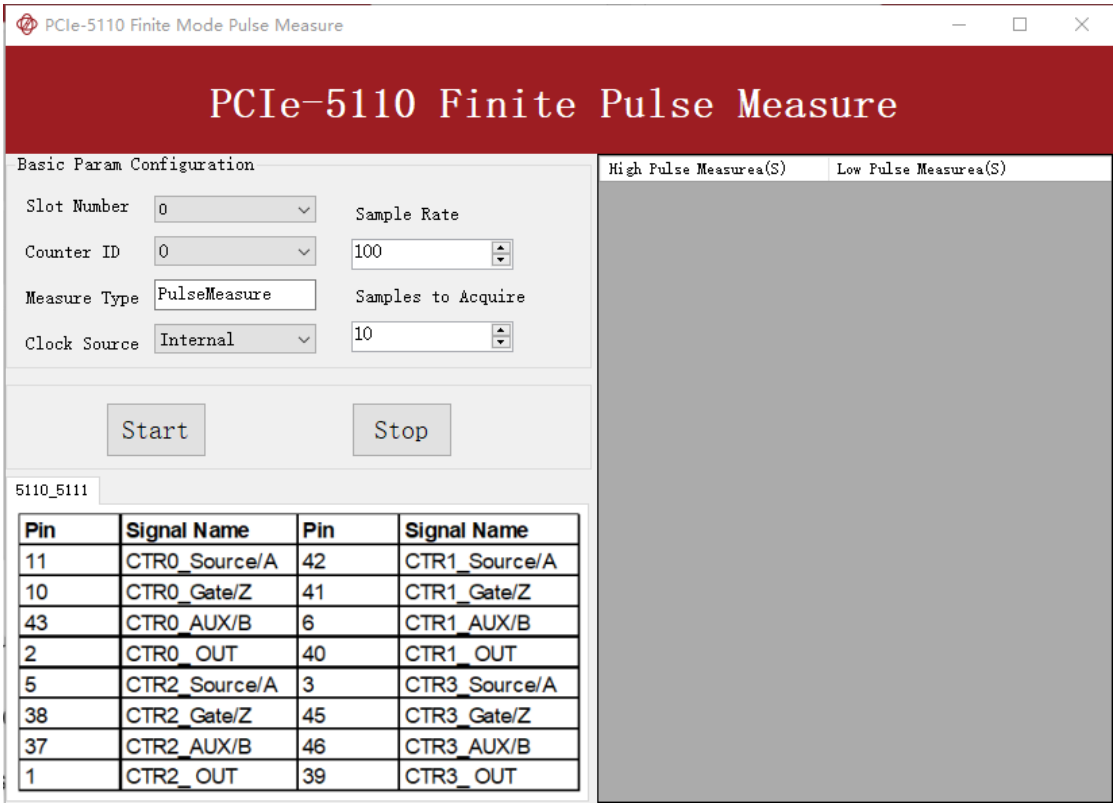


Figure 68 Pulse Measure For Finite Mode

- 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 69 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 Examples8.8.1 Finite/Continuous** Mode about the difference between Internal and Implicit.

8.8.3. 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 8.8.2 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 JY5111CITask.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 70.

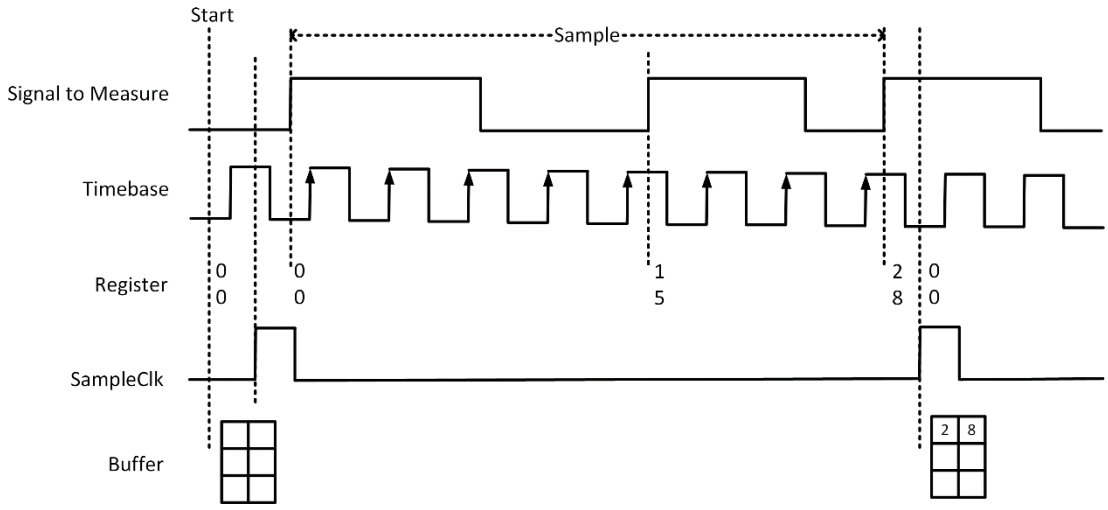


Figure 70 Frequency Measurement with Internal Sample Clock

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 8.8.2 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 8.8.3

- 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)**:

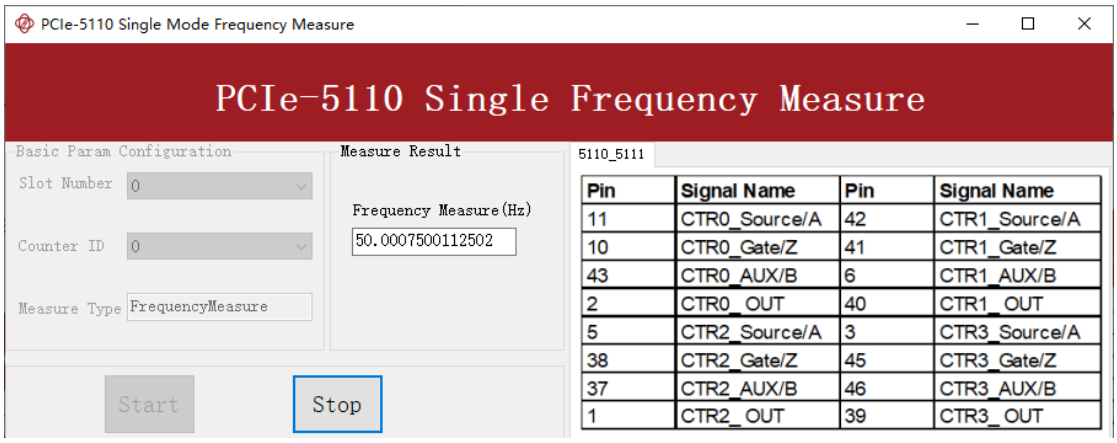


Figure 71 Frequency Measure For Single Mode

- 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.**

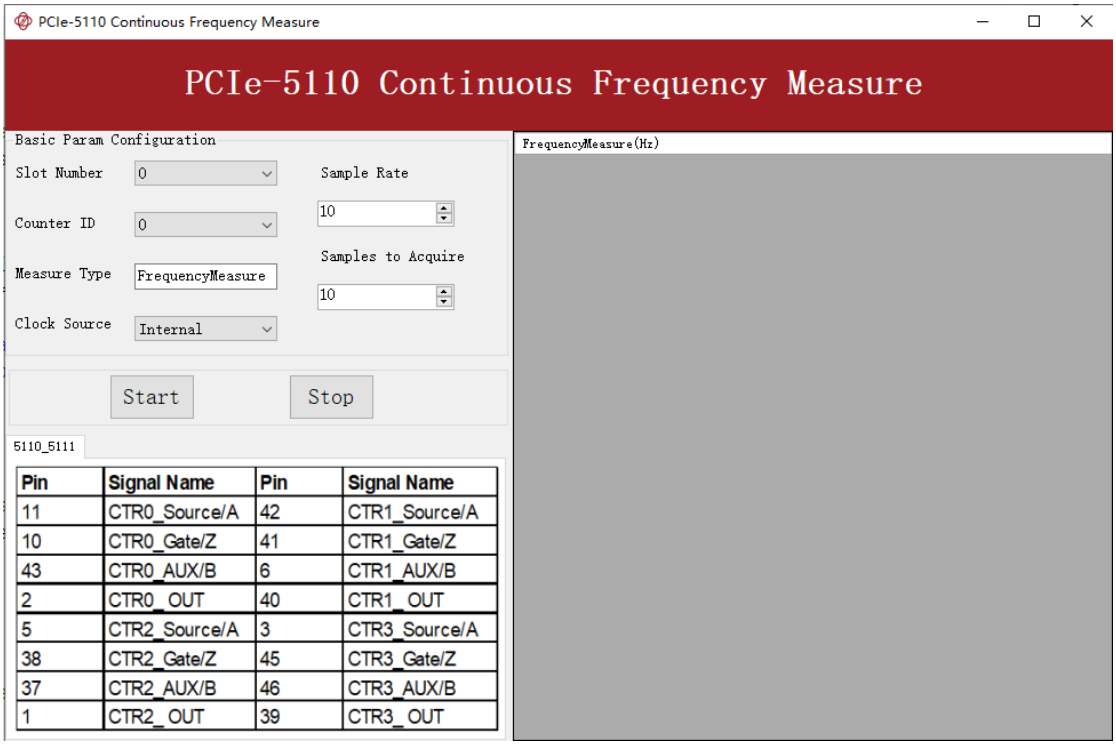


Figure 72 Frequency Measure For Continuous Mode

- 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.

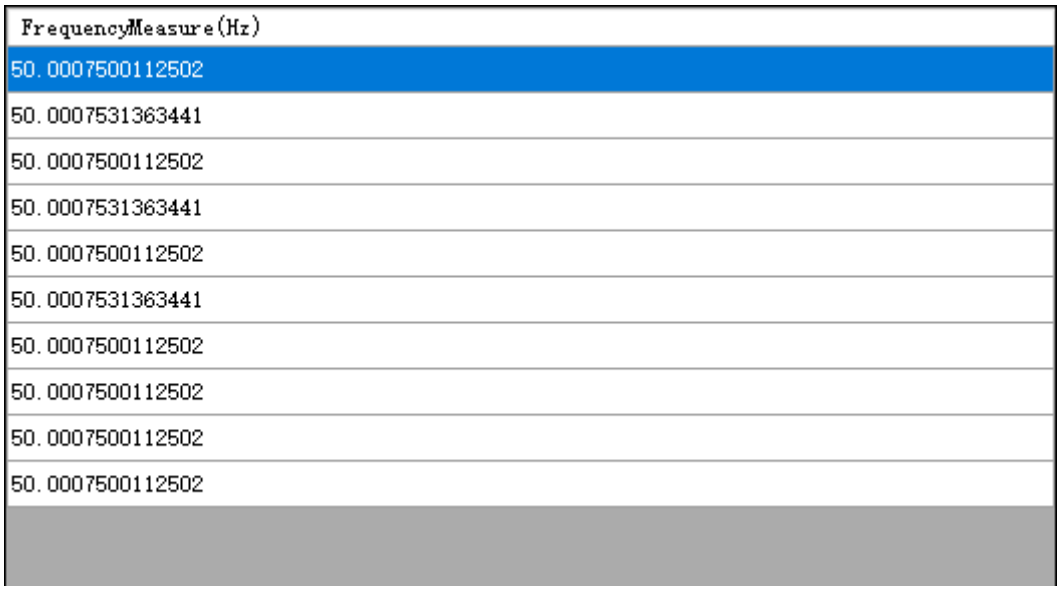


Figure 73 Frequency Measure Values

8.8.4. 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 8.8.3 for more information.

Learn by Examples 8.8.4

- 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)**:

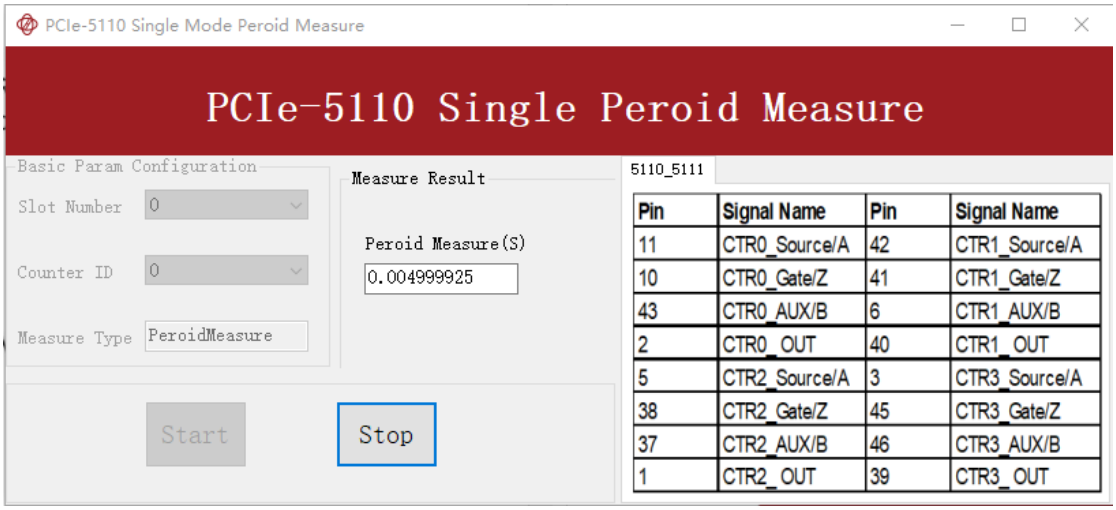
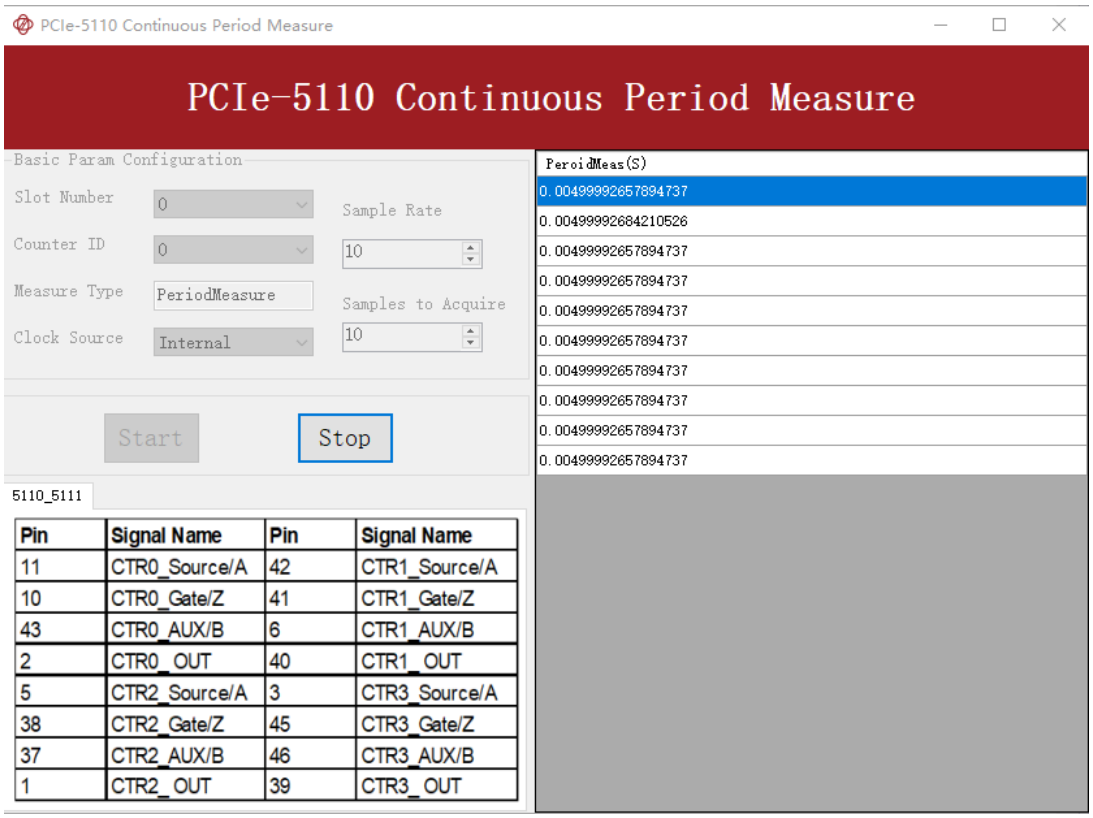


Figure 74 Peroid Measure For Single Mode

- 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).**



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

Figure 75 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.

8.8.5. 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 76.

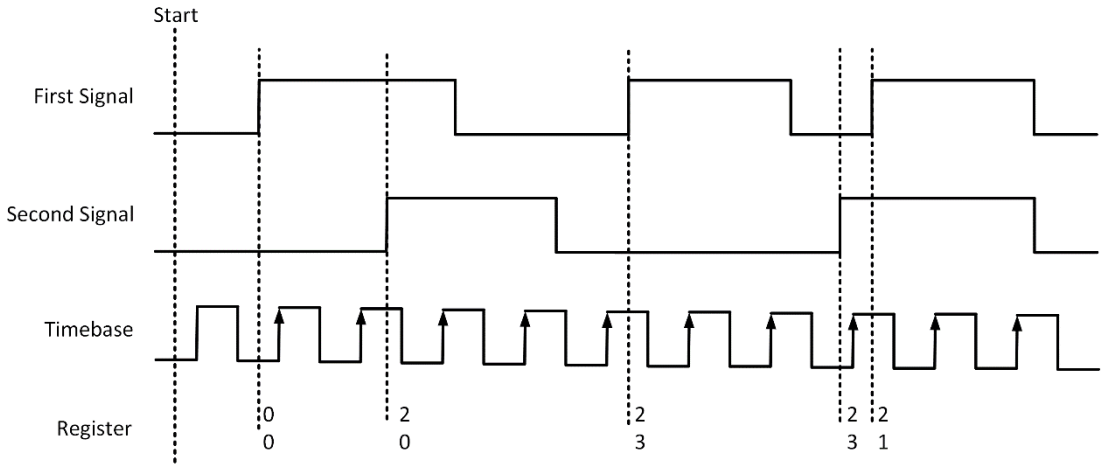


Figure 76 Two-Edge Separation in Single Mode

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 77.

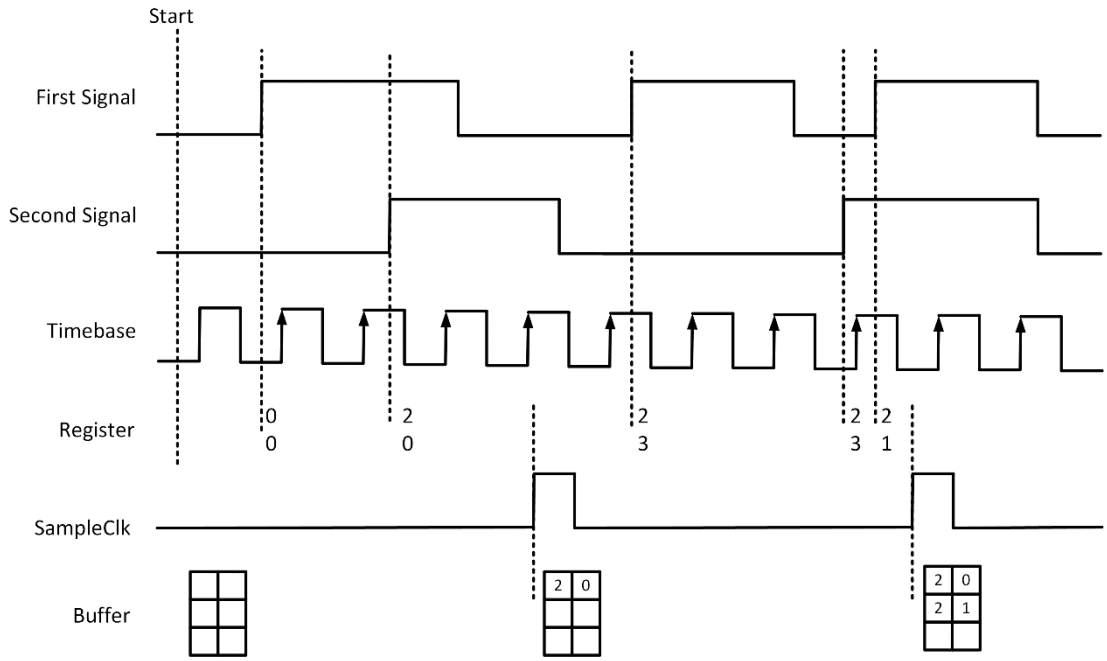


Figure 77 Two-Edge Separation with Internal Sample Clock

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 78.

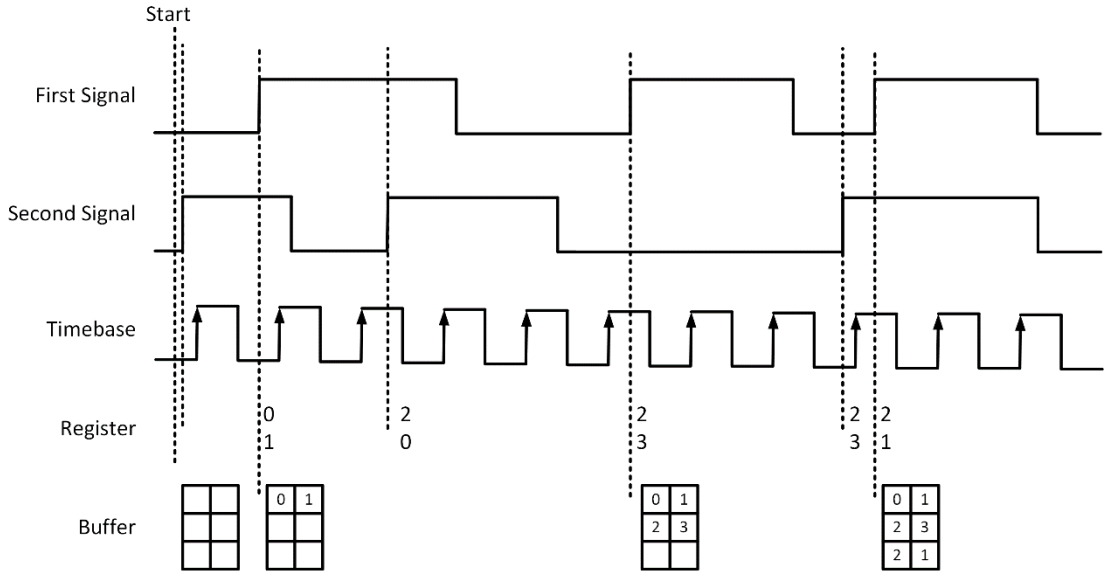


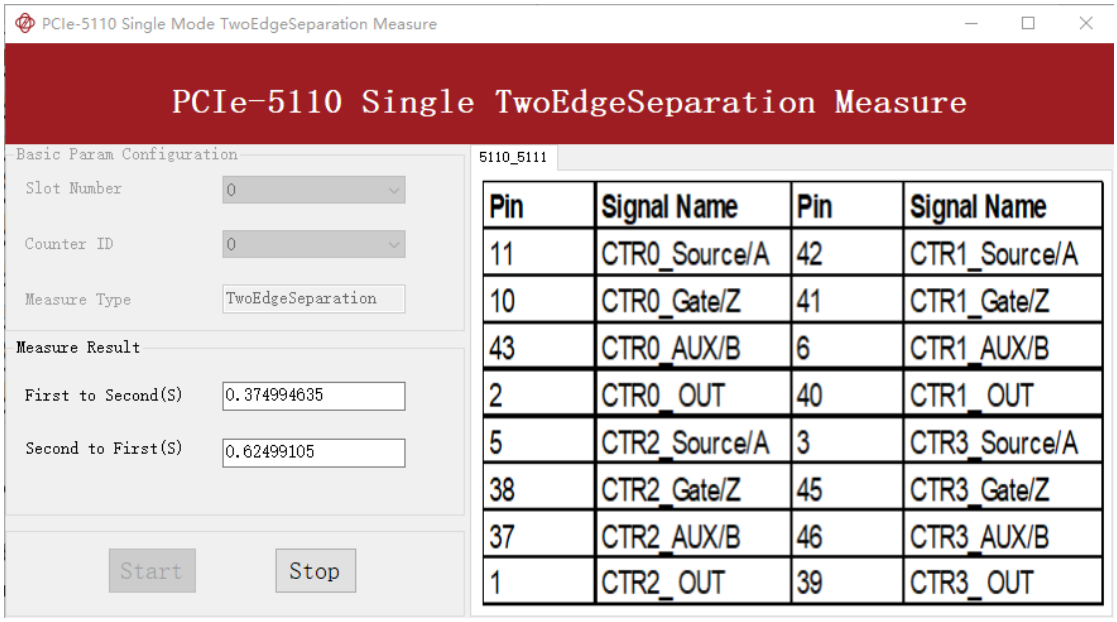
Figure 78 Two-Edge Separation with Implicit Sample Clock

Learn by Examples 8.8.5

- 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:



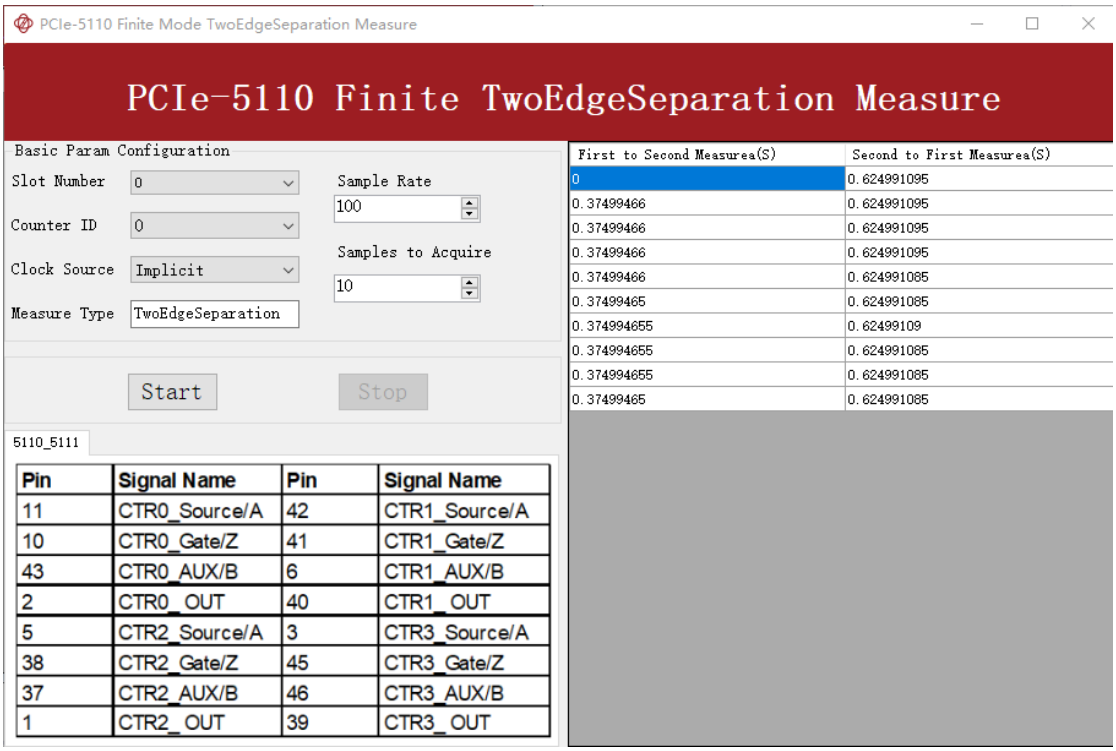
Pin	Signal Name	Pin	Signal Name
11	CTR0_Source/A	42	CTR1_Source/A
10	CTR0_Gate/Z	41	CTR1_Gate/Z
43	CTR0_AUX/B	6	CTR1_AUX/B
2	CTR0_OUT	40	CTR1_OUT
5	CTR2_Source/A	3	CTR3_Source/A
38	CTR2_Gate/Z	45	CTR3_Gate/Z
37	CTR2_AUX/B	46	CTR3_AUX/B
1	CTR2_OUT	39	CTR3_OUT

Figure 79 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:



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 80 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.

8.8.6. 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 81.

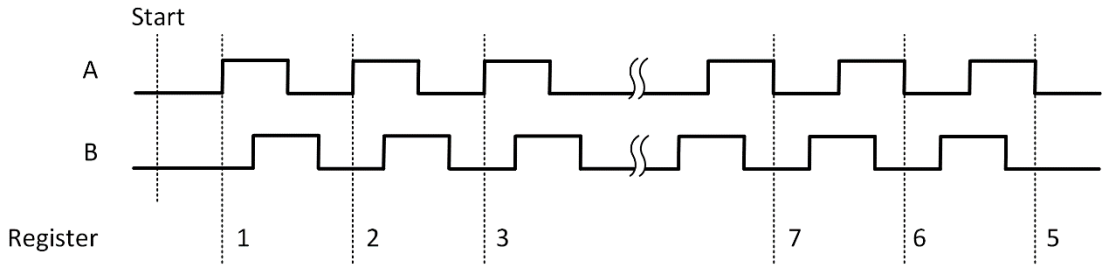


Figure 81 Quadrature Encoder x1 Mode

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 82.

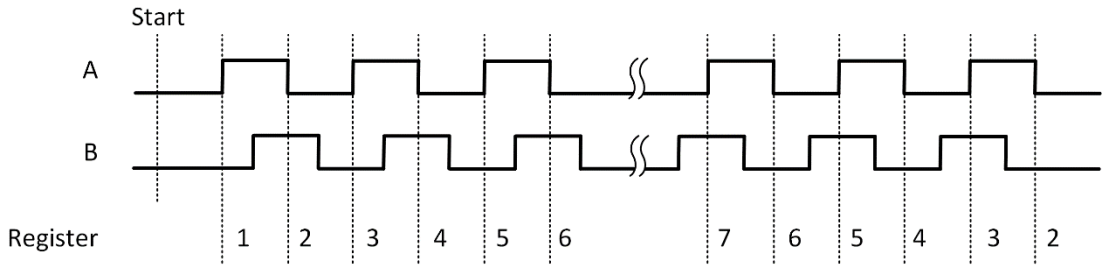


Figure 82 Quadrature Encoder x2 Mode

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 83.

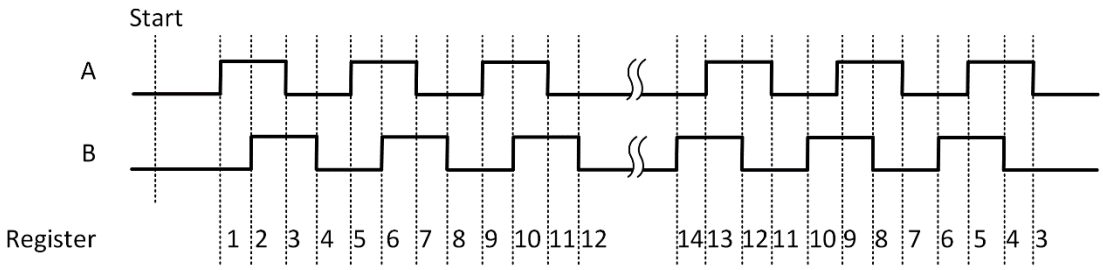


Figure 83 Quadrature Encoder x4 mode

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 55.

To configure the counter to work in this mode, set JY5111CITask. 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 84.

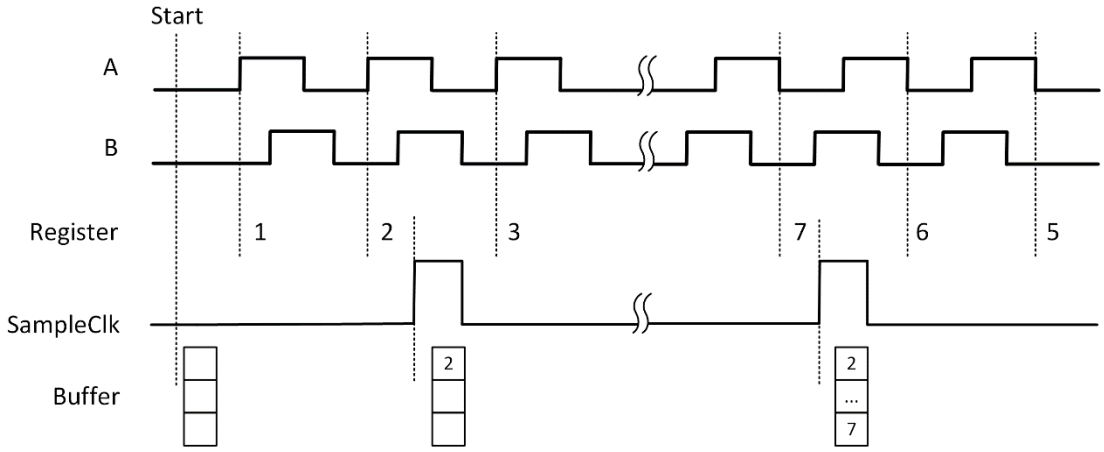


Figure 84 Quadrature Encoder x1 with Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

The count value is stored into the buffer every time it changes, as shown in Figure 85.

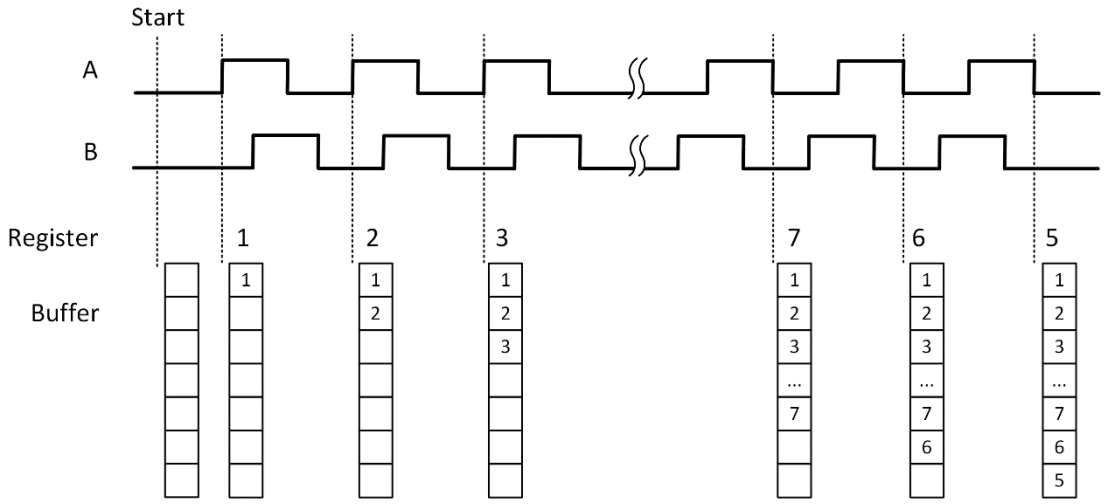


Figure 85 Quadrature Encoder x1 with Implicit Sample Clock

Learn by Examples 8.8.6

- Connect the signal source's two positive terminals to PCIe-5111 first signal input (sinewave, CTR0_Source/A, Pin #11) and second signal input (square wave , 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 8.8.6:

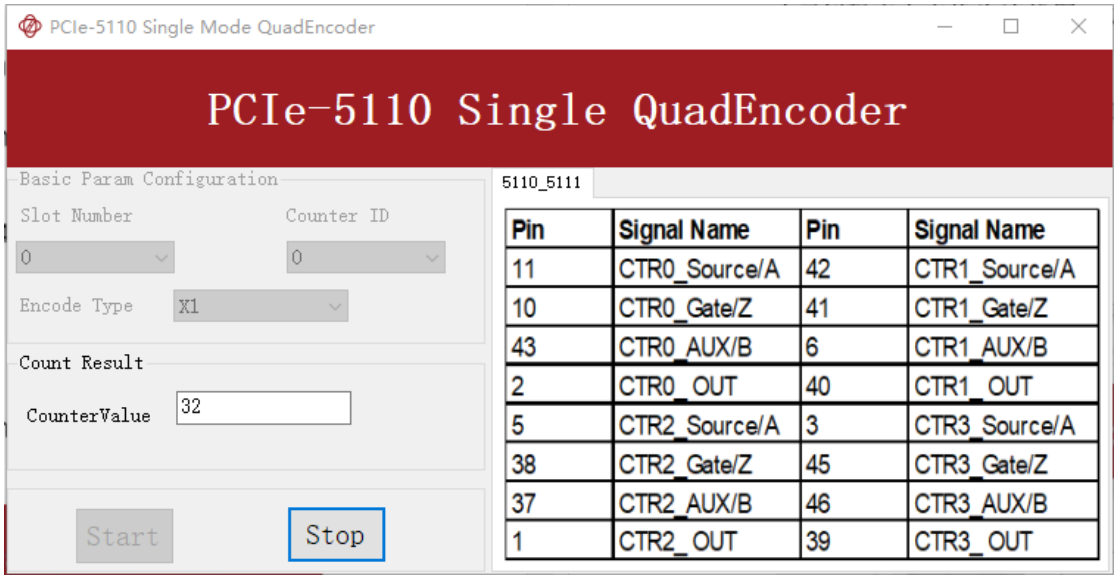


Figure 86 QuadEncoder For Single Mode

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

Continuous Mode

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

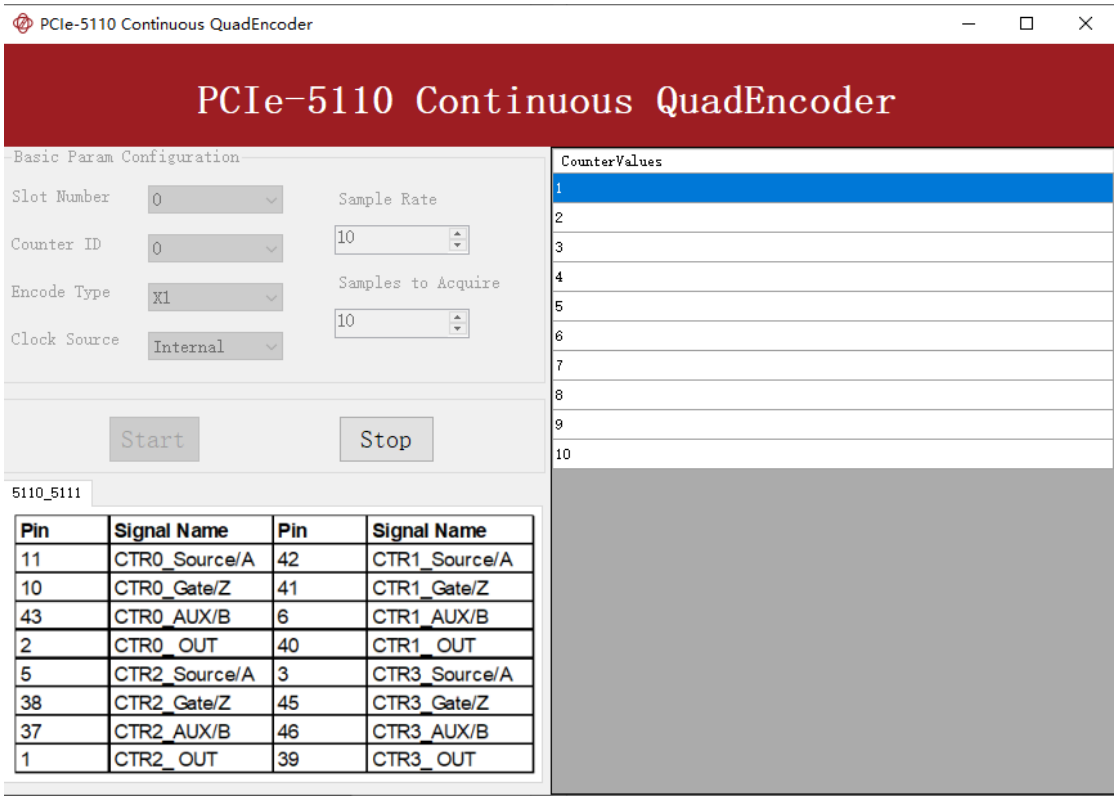


Figure 87 QuadEncoder For Continuous Mode

- 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.

8.8.7. 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 88.

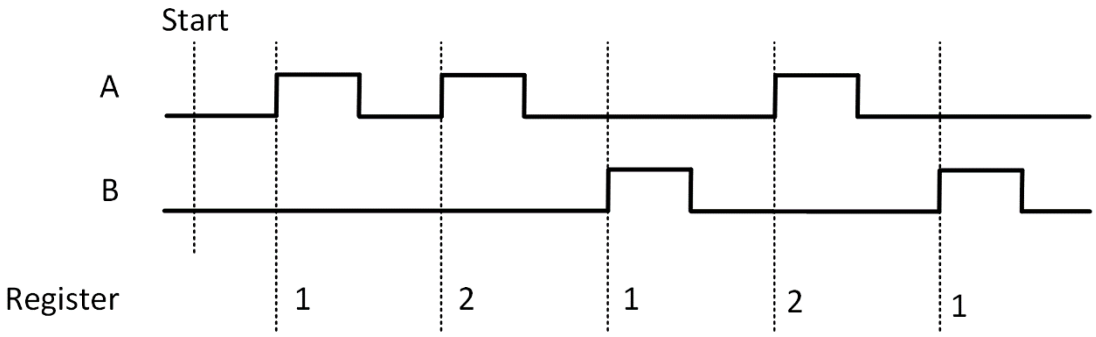


Figure 88 Two-Pulse Encoder in Single Mode

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 89.

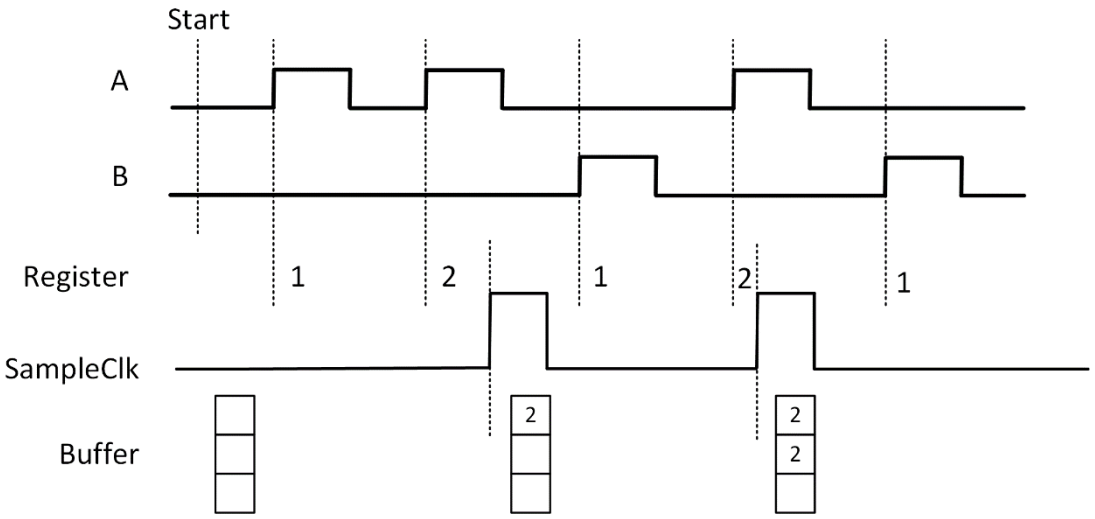


Figure 89 Two-Pulse Encoder with Internal Sample Clock

3) Finite/Continuous Mode with Implicit Sample Clock

The count value is stored into the buffer every time it changed, as shown in Figure 90.

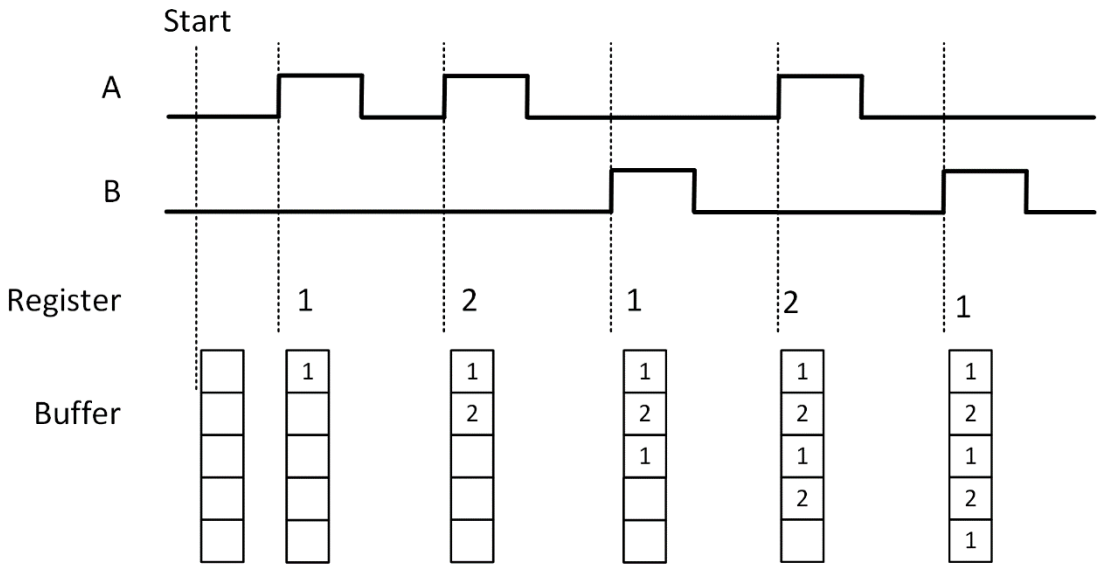


Figure 90 Two-Pulse Encoder with Implicit Sample Clock

Learn by Examples 8.8.7

- 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.

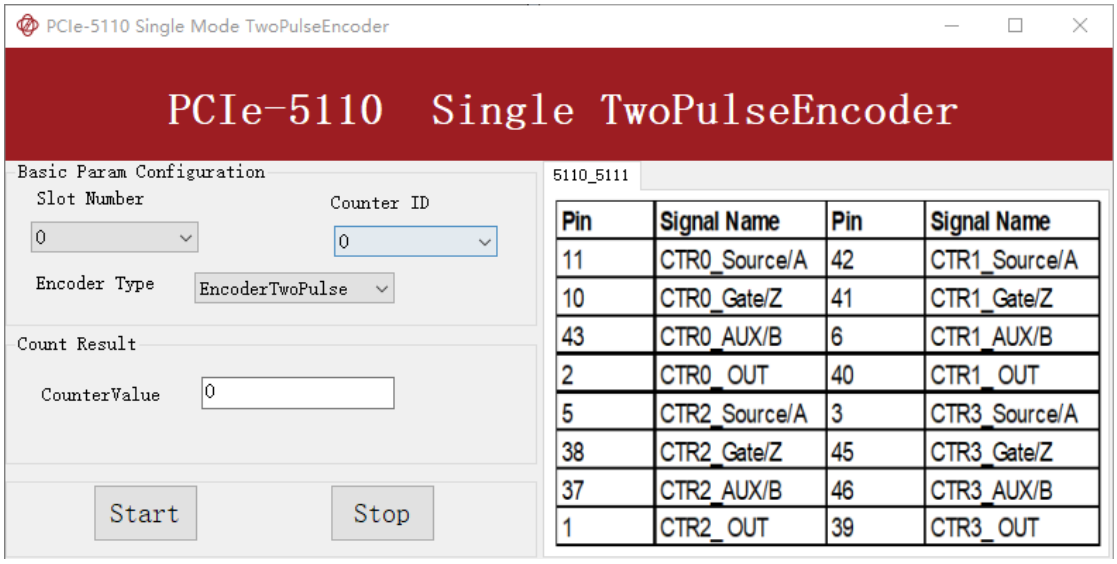


Figure 91 Two-PulseEncoder For Single Mode

➢ 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.

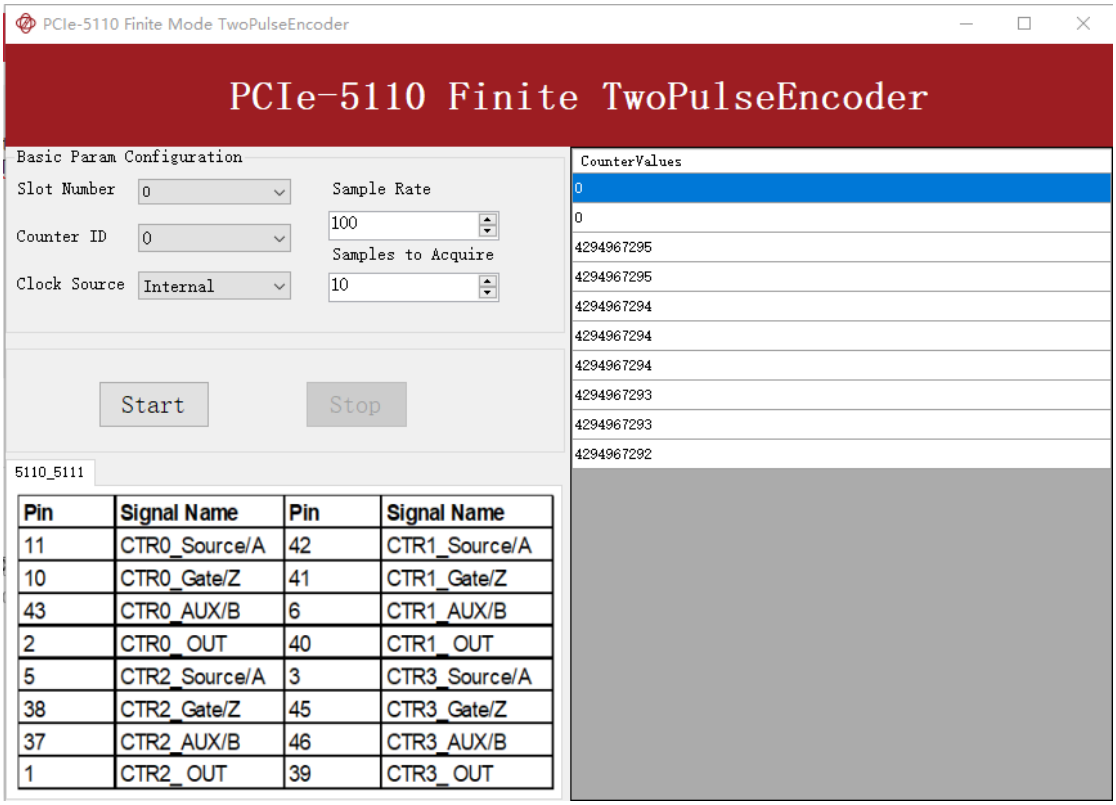
Finite Mode

■ 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 92 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.

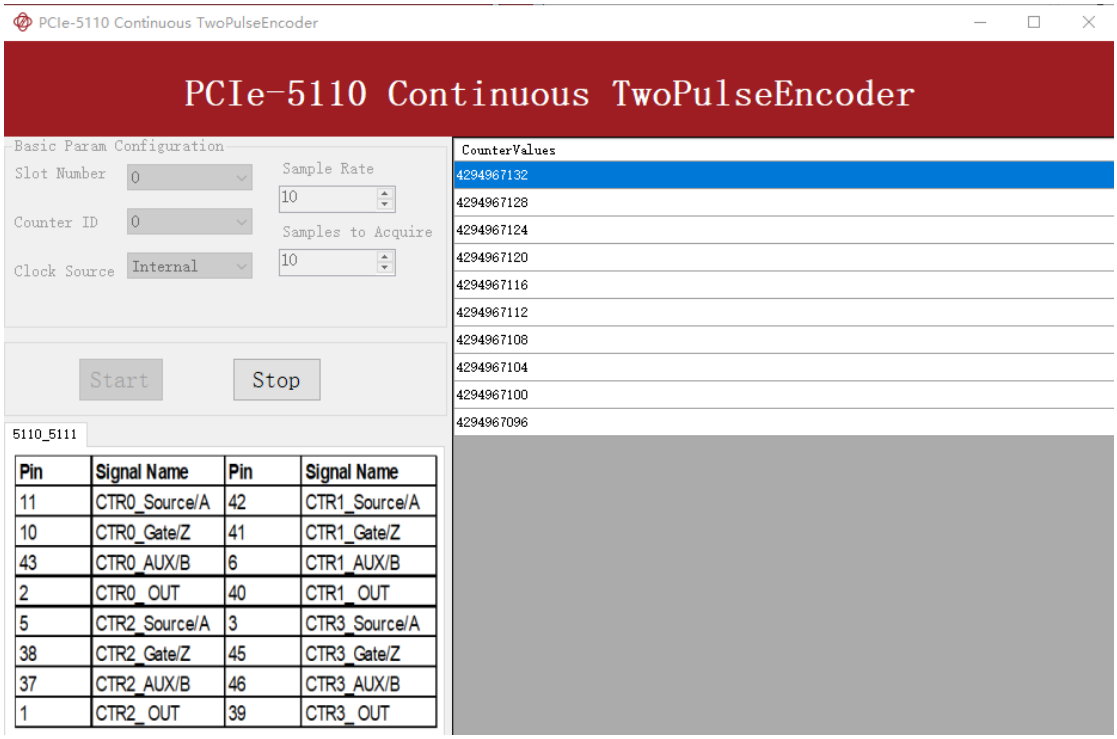


Figure 93 Two-PulseEncoder For Continuous Mode

- 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.

8.9. Counter Output Operations

8.9.1. Single Pulse Output

The PCIe-5111 timer/counter can output a single pulse with a specified pulse width. The timing diagram of the pulse output is shown in Figure 94.

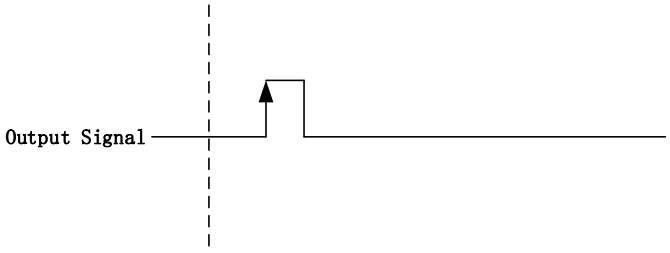


Figure 94 Single Pulse Output

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 8.9.1

- To see the signal thatPCIe-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:

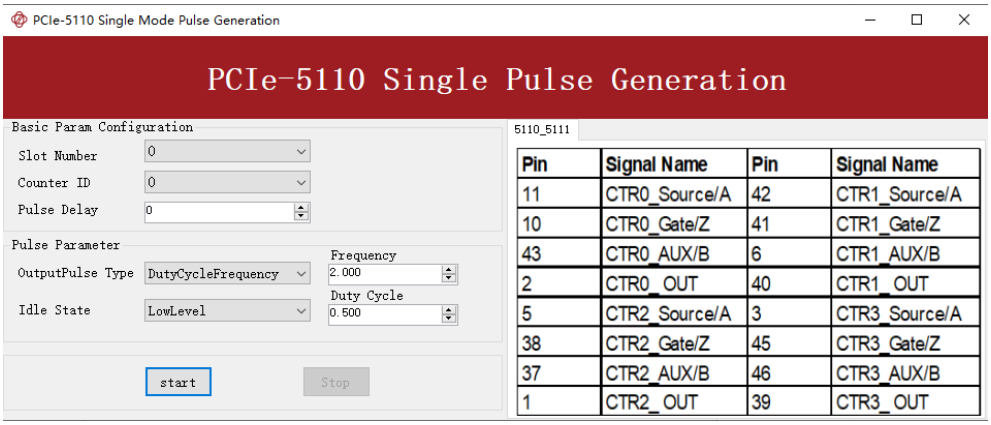


Figure 95 Single Pulse Generation

- 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.

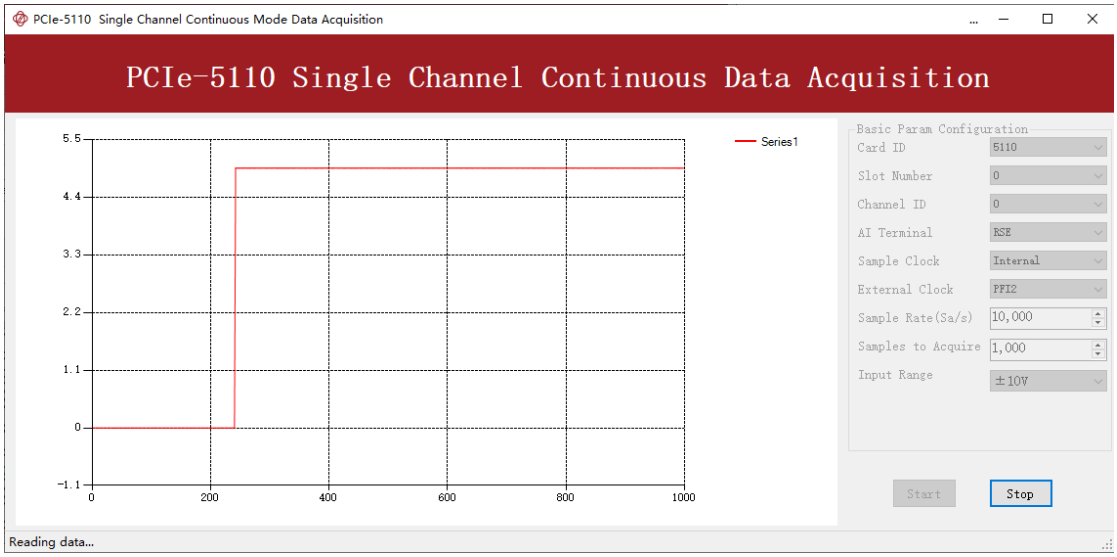


Figure 96 AI Acquisition Single Pulse

8.9.2. Finite Pulse Output

The pulse output timing is as shown in Figure 97.

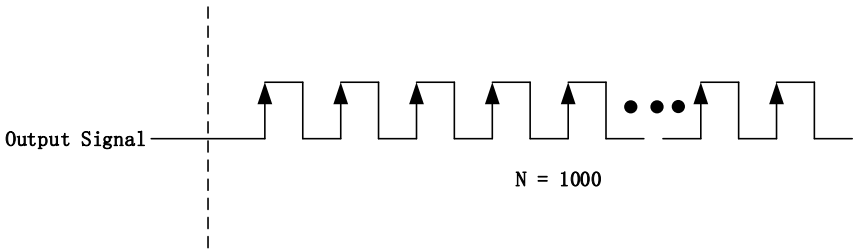


Figure 97 Finite Pulse Output

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 8.9.2

- 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:

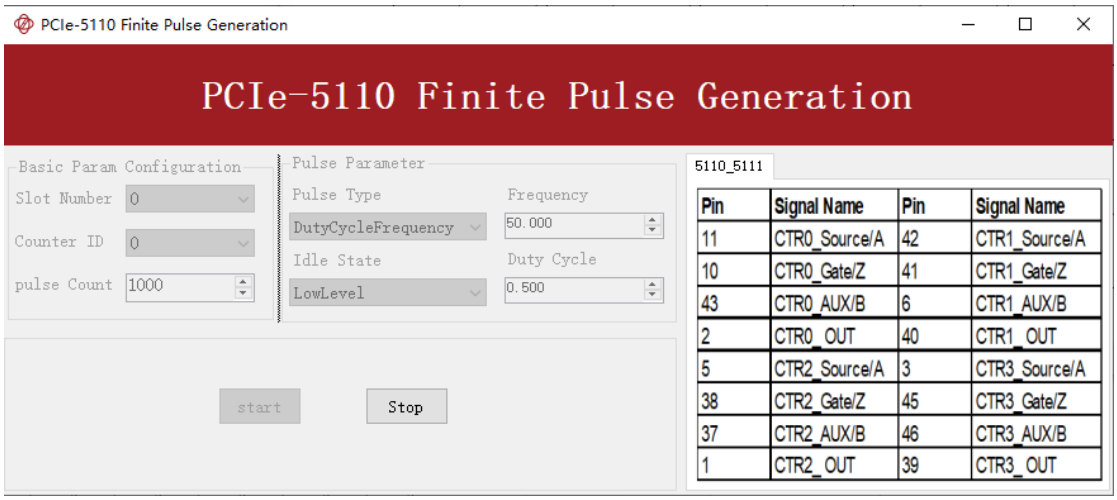


Figure 98 Finite Pulses Generation

- 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.

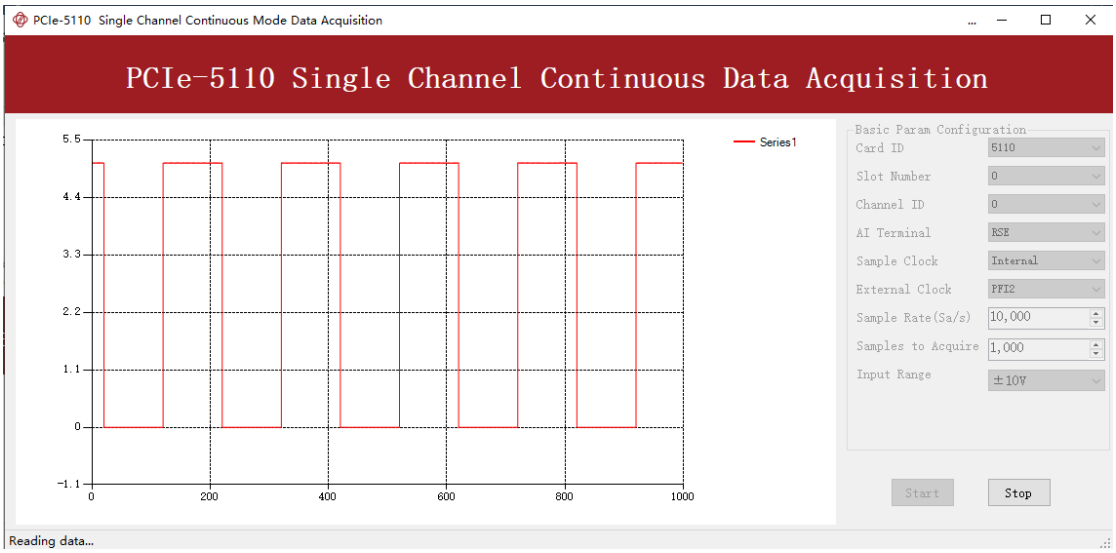


Figure 99 AI Acquisition Finite Pulse

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

8.9.3. Continuous Pulse Output

The pulse output timing is shown in Figure 100 below.

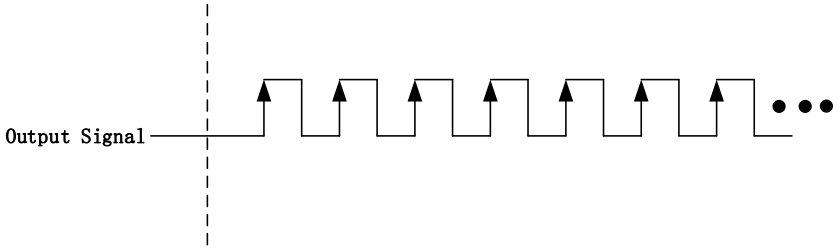


Figure 100 Continuous Pulse Output

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 8.9.3

- 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:

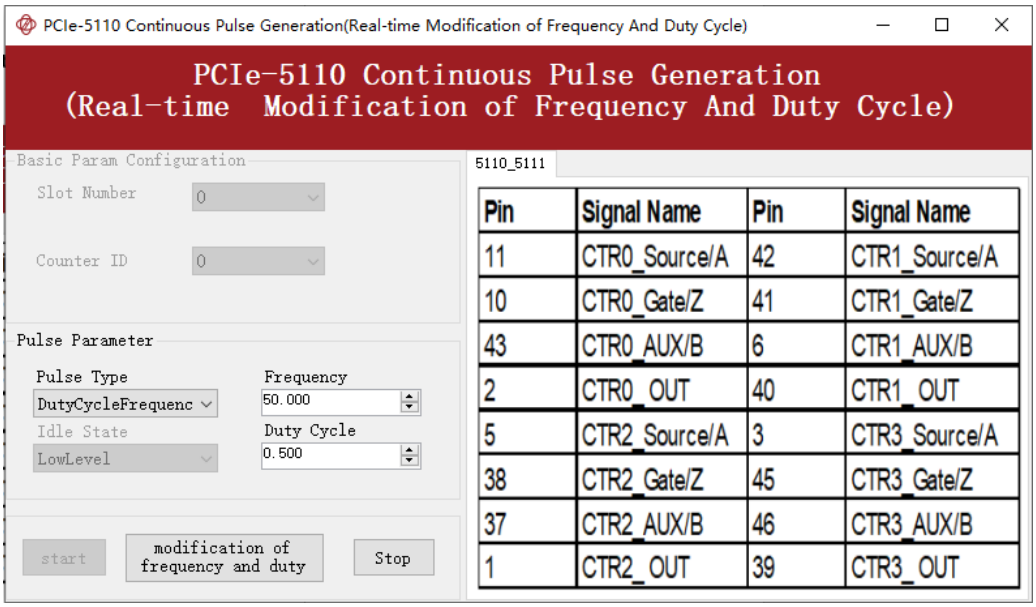


Figure 101 Continuous Pulse Generation

➢ 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.

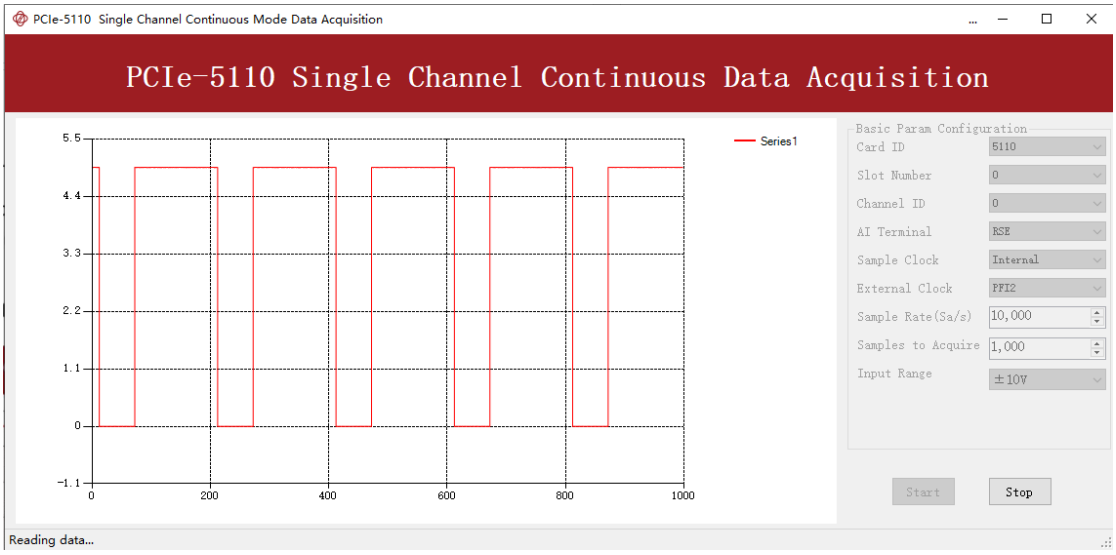


Figure 102 AI Acquisition Continuous Pulse

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

8.10. 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.

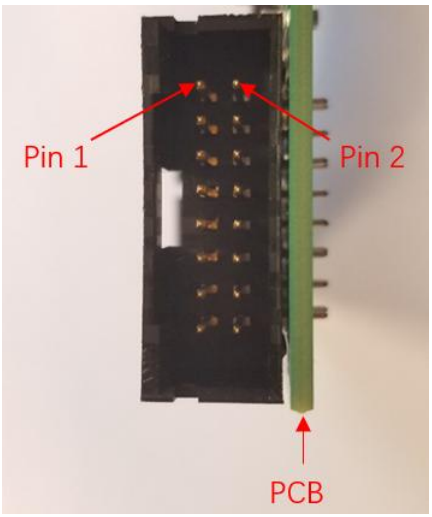


Figure 103 SSI Connector in PCIe-5111

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 18 SSI Connector Pin Assignment for PCIe-5111

8.11. DIP Switch in PCIe-5111

PCIe-5111 module have a DIP switch. The card number can be adjusted manually by changing the DIP switch setting, which is used to identify the boards with different slot positions. For example, if you want to set the card number to 3, you could turn the position 2 and 1 of the DIP switch to the ON position and the orthers to OFF. See below for details.

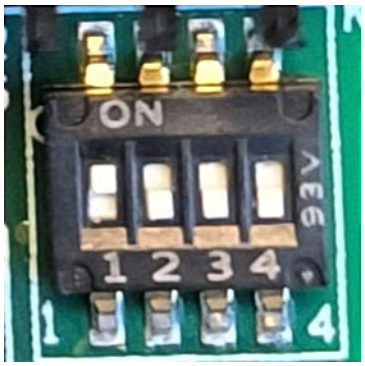


Figure 104 DIP swich in PCIe-5111

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

Table 19 Relationship between switch position and slot number

9. Calibration

PCIe-5111 Series boards are precalibrated before the shipment. We recommend you recalibrate PCIe-5111 board periodically to ensure the measurement accuracy. A commonly accepted practice is one year. If for any reason, you need to recalibrate your board, please contact JYTEK.

10. Appendix(Measurement Issues)

10.1. Floating Signals and Ground Referenced Signals

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

10.2. Differential, NRSE, RSE Modes

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

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

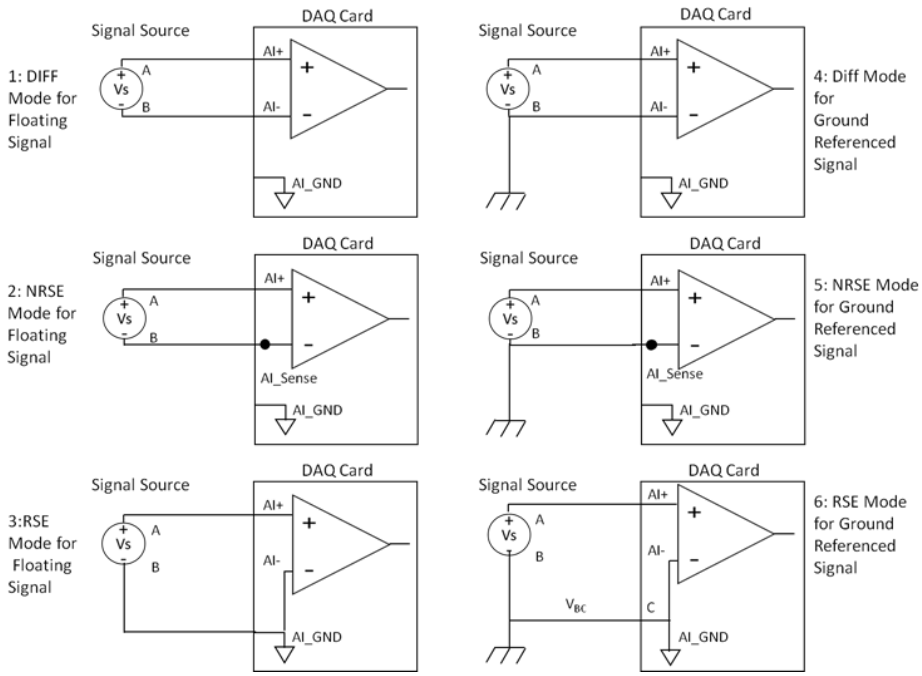


Figure 105 Six Measurement Scenarios

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.

10.3. Reducing the Common Mode Voltage Effect

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

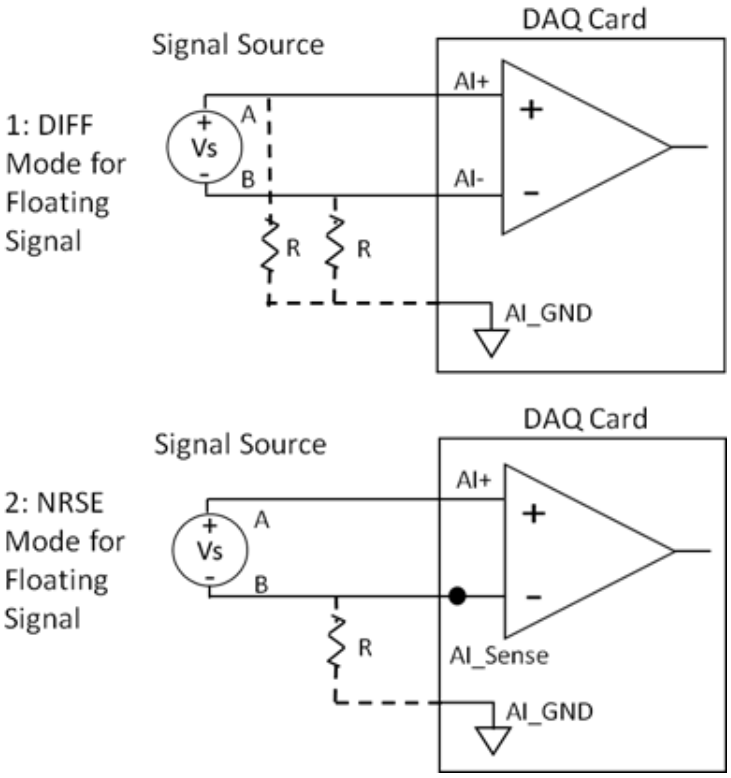


Figure 106 Using Resister to Reduce Common Mode Voltage Effect

11. About JYTEK

11.1. 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.

11.2. 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.

11.3. 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.

12. 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-5111 family of temperature sensor data acquisition cards. The manual is copyrighted by JYTEK.

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

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

Shanghai Jianyi Technology Co., Ltd.

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

Post Code: 201203

Tel: 021-5047 5899

Website: www.jytek.com