PCIe/PXIe-5211

Counter/Timer Module

User Manual

User Manual Version: V1.6.6

Revision Date: April 2, 2025

Table of Contents

1. Overview

This chapter presents the information how to use this manual and quick start if you are already familiar with Microsoft Visual Studio and C# programming language.

1.1 Introduction

PCIe/PXIe-5211 series modules are multi-purpose counter data acquisition modules, which can provide 8 channels of counter, supporting edge counting, frequency measurement, period measurement, pulse measurement, two-edge separation measurement, encoder and pulse generation. The device utilizes a high-throughput PCI Express bus and multi-core optimized drivers and application software to provide high-performance capabilities.

Please check with JYTEK website for the latest 5211 series offering.

1.2 Main Features

- 8 channels of counter
- 40 channels multi-purpose PFI
- Up to 100 MHz internal clock rate
- 1.8 V / 2.5 V / 3.3 V / 5 V level
- Edge Counting / Frequency Measurement / Pulse measurement / Period Measurement / Two-Edge seperation
- Quadrature (x1/x2/x4) encoder
- Two-Pulse encoder
- Dynamic reconfigurable counter output function
- Support multi-card synchronization
- On-board high-performance TCXO clock
- On-board clock generator for sampling

1.3 Abbreviations

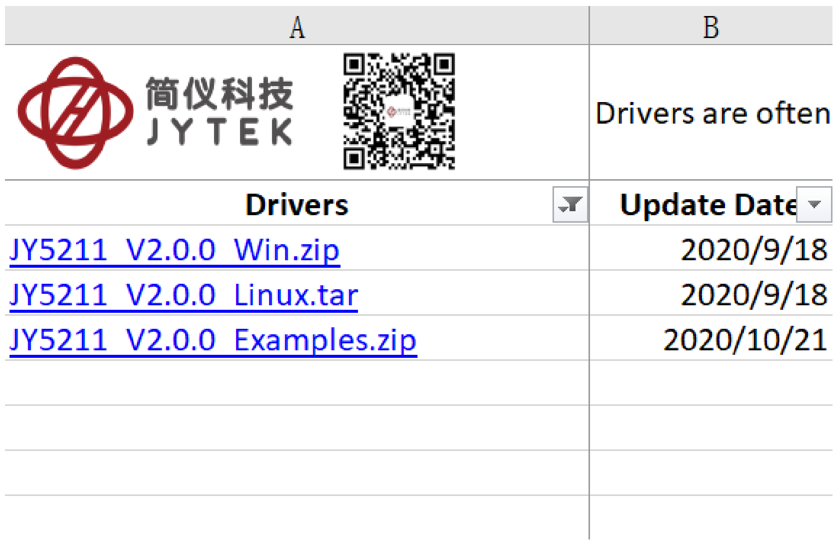
DI:	Digital Input
DO:	Digital Output
CI:	Counter Input
CO:	Counter Output
DAQ:	Data AcQuisition

PFI: Programmable Function Interface

TXCO: Temperature Compensate X'tal (crystal) Oscillator

1.4 Learn by Example

JYTEK has added **Learn by Example** in this manual. We provide many sample programs for this device. Please download and install the sample programs for this device. You can download a JYPEDIA excel file from our web www.jytel,com. Open JYPEDIA and search for JY5211 in the driver sheet, select **JY5211.Examples.zip**. This will lead you to download the sample program for this device. In addition to the download information, JYPEDIA also has a lot of other valuable information, JYTEK highly recommend you use this file to obtain information from JYTEK.



A	B
简仪科技 JYTEK	Drivers are often
Drivers	**Update Date**
JY5211_V2.0.0_Win.zip	2020/9/18
JY5211_V2.0.0_Linux.tar	2020/9/18
JY5211_V2.0.0_Examples.zip	2020/10/21

Figure 1 JYPEDIA Information

In a **Learn by Example** section, the sample program is in bold style such as **Counter Input-->Winform CI Single Edge Counting**; the property name in the sample program is also in bold style such as **SamplesToAcquire**; the technical names used in the manual is in italic style such as *SampleRate*. You can easily relate the property names in the example program with the manual documentation.

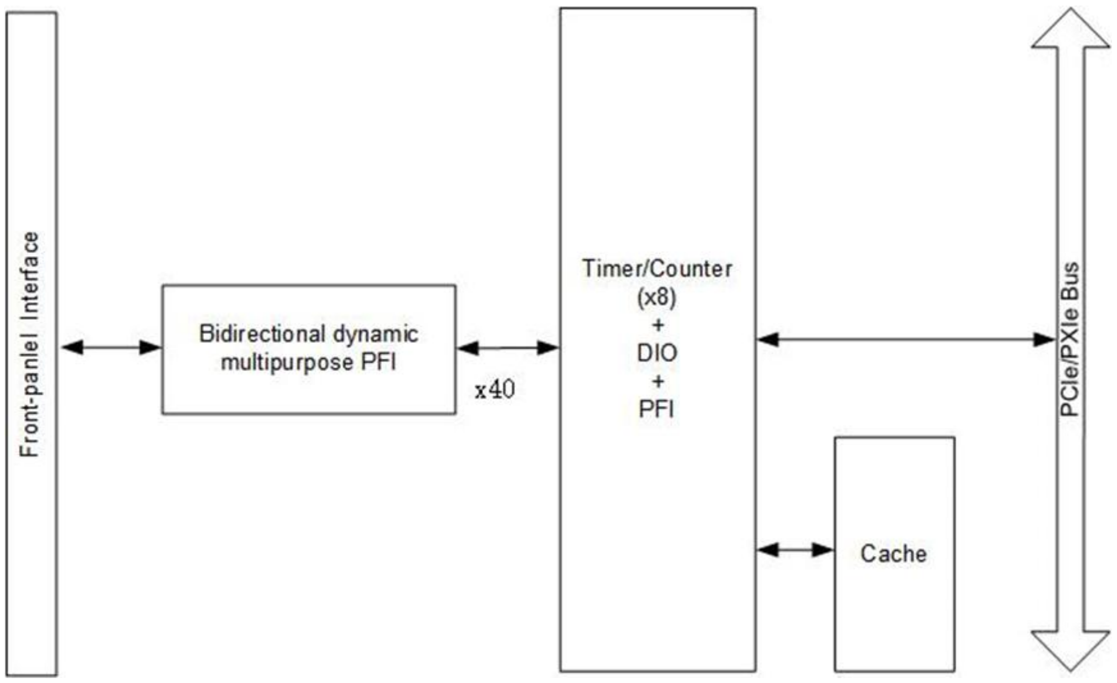
In an **Learn by Example** section, the experiment is set up as follow. A PCIe/PXIe-5211 card is plugged in a desktop computer. The PCIe/PXIe-5211 is connected to a TB-68/DIN-68S-01 terminal block. A signal source is also connected to the same terminal block.

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

Tip: PCIe/PXIe-5211 also has the counter output capability. If you do not have a signal source, you can use the outputs of PCIe/PXIe-5211 as the signal source. In this case you need first run example program ***Counter Output--> Winform CO Continuous Wrapping*** *to generate the output.*

2. Hardware Specifications

2.1 System Diagram



Front-panlel Interface
Bidirectional dynamic multipurpose PFI
x40
Timer/Counter (x8) + DIO + PFI
Cache
PCIe/PXIe Bus

Figure 2 PCIe/PXIe-5211 Series System Block Diagram

The system block diagram of PCIe/PXIe-5211 series is shown in Figure 2.

It is mainly composed of one DIO module and eight Counter/Timer modules, providing digital input, digital output, counter input, counter output functions. JYTEK's FPGA-based driver FirmDrive provides a stable and efficient PCIe / PXIe / USB interface.

2.2 Digital IO Specifications

Basic

Number of channels	40
Ground reference	GND
Direction control	Independently Controlled
Pull-down resister	NONE
Logic signal levels	1.8 V 2.5 V 3.3 V 5 V
Input voltage protection	-0.5-(UserVcc+0.5 V)
Output impedance	50 Ω
UserVcc ： user select the logic signal level	

DIO

Terminals used	DIO <0..39>
Port width	32 bits (Maximum)

PFI

Available PFI	PFI<0..39>
Supported functions	Static input Static output

Recommended Operating Conditions

Input high voltage (V_{IH})	1.2 V (UserVcc=1.8 V) 1.7 V (UserVcc=2.5 V) 2 V (UserVcc=3.3 V) 3.5 V (UserVcc=5 V)
Input low voltage (V_{IL})	0.65 V (UserVcc=1.8 V) 0.7 V (UserVcc=2.5 V) 0.8 V (UserVcc=3.3 V) 1.5 V (UserVcc=5 V)
Maximum high-level input voltage	UserVcc+0.5 V
Minimum low-level input level	-0.5 V
Output High current (I_{OH})	-4 mA(UserVcc=1.8 V) -8 mA(UserVcc=2. 5V) -24 mA(UserVcc=3.3 V) -32 mA(UserVcc=5 V)
Output Low current (I_{OL})	4 mA(UserVcc=1.8 V) 8 mA(UserVcc=2.5 V) 24 mA(UserVcc=3.3 V) 32 mA(UserVcc=5 V)

Electrical

Maximum input current of low-level voltage	-2 μA
Maximum input current of high-level voltage	2 μA

Table 1 Digital IO Specifications

2.3 Counter/Timer Specifications

Basic

Number of counters/timers	8
Input Mode	Single,Finite,Continuous
Input Type	Edge Counting Frequency Period SemiPeriod Pulse Two Edge Separation Quadrature Encoder Two Pulse Encoder
Output Mode	Single,Finite and Continuous pulse

Maximum Frequency of the Source

	PFI
Frequency Measurement	50 MHz
Edge Counting	50 MHz

Minimum Pulse Measurement

	PFI
Period Measurement	15 ns
Pulse Measurement	20 ns
Two-Edge Separation	20 ns

Other Functions

Internal timebase	100MHz/5MHz/100kHz
MaxSampleClock	10 MHz @ Internal timebase 100MHz
Internal timebase accuracy	2 ppm
External timebase	0-50 MHz
MaxSampleClock	10 MHz @External timebase 50MHz

Input Terminal	Gate(Z)
	Source(A)
	Aux(B)
	Digital Trigger
	External Sample Clock

Output Terminal	OUT
FIFO per channel	15M Samples

Table 2 Counter/Timer Specifications

2.1 Front Panel and Pin Definition

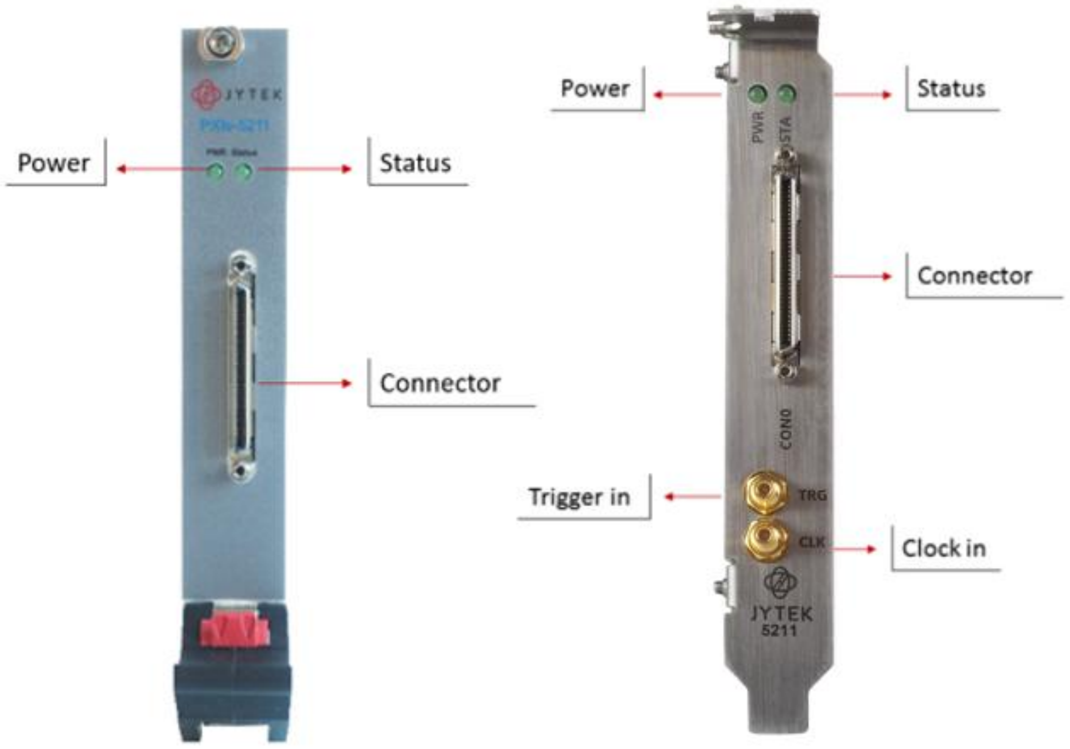


Figure 3 Front Pannel

SCSI-VHDCI 68pin			
Signal	Pin	Pin	Signal
PFI30 / DIO 30 / CTR 7 Aux	35	1	PFI31 / DIO 31 / CTR 7 Out
GND	36	2	GND
PFI28 / DIO 28 / CTR 7 Source	37	3	PFI29 / DIO 29 / CTR 7 Gate
GND	38	4	GND
PFI26 / DIO 26 / CTR 6 Aux	39	5	PFI27 / DIO 27 / CTR 6 Out
GND	40	6	GND
PFI24 / DIO 24 / CTR 6 Source	41	7	PFI25 / DIO 25 / CTR 6 Gate
GND	42	8	PFI32 / DIO 32 / Start Trigger
PFI22 / DIO 22 / CTR 5 Aux	43	9	PFI23 / DIO 23 / CTR 5 Out
GND	44	10	GND
PFI20 / DIO 20 / CTR 5 Source	45	11	PFI21 / DIO 21 / CTR 5 Gate
GND	46	12	GND
PFI18 / DIO 18 / CTR 4 Aux	47	13	PFI19 / DIO 19 / CTR 4 Out
GND	48	14	GND
PFI16 / DIO 16 / CTR 4 Source	49	15	PFI17 / DIO 17 / CTR 4 Gate
GND	50	16	GND
PFI14 / DIO 14 / CTR 3 Aux	51	17	PFI15 / DIO 15 / CTR 3 Out
PFI33 / DIO 33 / ECLK	52	18	GND
PFI12 / DIO 12 / CTR 3 Source	53	19	PFI13 / DIO 13 / CTR 3 Gate
GND	54	20	GND
PFI10 / DIO 10 / CTR 2 Aux	55	21	PFI11 / DIO 11 / CTR 2 Out
GND	56	22	GND
PFI8 / DIO 8 / CTR 2 Source	57	23	PFI9 / DIO 9 / CTR 2 Gate
GND	58	24	GND
PFI6 / DIO 6 / CTR 1 Aux	59	25	PFI7 / DIO 7 / CTR 1 Out
PFI35 / DIO 35	60	26	PFI34 / DIO 34
PFI4 / DIO 4 / CTR 1 Source	61	27	PFI5 / DIO 5 / CTR 1 Gate
GND	62	28	GND
PFI2 / DIO 2 / CTR 0 Aux	63	29	PFI3 / DIO 3 / CTR 0 Out
PFI37 / DIO 37	64	30	PFI36 / DIO 36
PFI0 / DIO 0 / CTR 0 Source	65	31	PFI1 / DIO 1 / CTR 0 Gate
GND	66	32	GND
PFI39 / DIO 39	67	33	PFI38 / DIO 38
GND	68	34	GND

Table 3 Pin Defination

2.2 Other Specifications

PLL(Phase lock loop)

Number of PLL	1
Reference clock source	PXIe_DSTAR A：　100 MHz (Max)
	PXIe_CLK100：　100 MHz
	Onboard TCXO：　10 MHz
Output	200 MHz base clock
	Counter internal sample clock
	100 KHz timebase
	5 MHz timebase
	200 MHz timebase

TCXO

Basic Property	Nominal frequency	10 MHz
	Warm-up time	15 minutes
	Temperature drift	±20 ppb
	Temperature drift and 1 year drift	±0.5 ppm

External Digital Trigger

Trigger functions	Trigger source	PXI_TRIG <0..7> PXI_STAR PFI<0..39>
	Polarity	Rising Edge
	Counter/Timer functions	Start trigger
Device to device trigger bus	Input source	PXI_TRIG <0..7> PXI_STAR
	Output destination	PXI_TRIG <0..7>
	Output options	Sync Trigger Routing
	Debounce filter settings	Not Support

Bus and Power

Bus interface	PXIe standard	x4 PXI Express peripheral module Specification V1.0 compliant
	Slot supported	x1 and x4 PXI Express or PXI Express hybrid slots
Calibration	Recommended warm-up time	>=15 min
	Recommended calibration interval	One year

Physical Size and Environment

Size	External physical size	3U PXIe
	Weight	190 g
Operating Environment	Indoor only	Yes
	Ambient temperature range	0 °C to 50 °C
	Relative humidity range	20% to 80%, noncondensing
Storage Environment	Ambient temperature range	-20 ˚C to 80 ˚C
	Relative humidity range	10% to 90%, noncondensing

Table 4 Other Specifications

2.3 Default Routing for Counter Input/Output Signals

All counter input and output terminals are routed to a certain PFI by default as shown in Table 5.

Application	Signal Type	Ctr0(Name(Pin#))	Ctr1(Name(Pin#))	Ctr2(Name(Pin#))	Ctr3(Name(Pin#))	Ctr4(Name(Pin#))	Ctr5(Name(Pin#))	Ctr6(Name(Pin#))	Ctr7(Name(Pin#))
Edge Counting	Signal To Measure(Source)	PFI_In0(65)	PFI_In4(61)	PFI_In8(57)	PFI_In12(53)	PFI_In16(49)	PFI_In20(45)	PFI_In24(41)	PFI_In28(37)
	Pause Trigger(Gate)	PFI_In1(31)	PFI_In5(27)	PFI_In9(23)	PFI_In13(19)	PFI_In17(15)	PFI_In21(11)	PFI_In25(7)	PFI_In29(3)
	Count Direction(Aux)	PFI_In2(63)	PFI_In6(59)	PFI_In10(55)	PFI_In14(51)	PFI_In18(47)	PFI_In22(43)	PFI_In26(39)	PFI_In30(35)
	Output(Out)	PFI_In3(29)	PFI_In7(25)	PFI_In11(21)	PFI_In15(17)	PFI_In19(13)	PFI_In23(9)	PFI_In27(5)	PFI_In31(1)
Frequency Measurement	Signal To Measure(Gate)	PFI_In1(31)	PFI_In5(27)	PFI_In9(23)	PFI_In13(19)	PFI_In17(15)	PFI_In21(11)	PFI_In25(7)	PFI_In29(3)
	External Timebase(Source)	PFI_In0(65)	PFI_In4(61)	PFI_In8(57)	PFI_In12(53)	PFI_In16(49)	PFI_In20(45)	PFI_In24(41)	PFI_In28(37)
Period Measurement	Signal To Measure(Gate)	PFI_In1(31)	PFI_In5(27)	PFI_In9(23)	PFI_In13(19)	PFI_In17(15)	PFI_In21(11)	PFI_In25(7)	PFI_In29(3)
	External Timebase(Source)	PFI_In0(65)	PFI_In4(61)	PFI_In8(57)	PFI_In12(53)	PFI_In16(49)	PFI_In20(45)	PFI_In24(41)	PFI_In28(37)
Pulse Measurement	Signal To Measure(Gate)	PFI_In1(31)	PFI_In5(27)	PFI_In9(23)	PFI_In13(19)	PFI_In17(15)	PFI_In21(11)	PFI_In25(7)	PFI_In29(3)
	External Timebase(Source)	PFI_In0(65)	PFI_In4(61)	PFI_In8(57)	PFI_In12(53)	PFI_In16(49)	PFI_In20(45)	PFI_In24(41)	PFI_In28(37)
Two-Edge Separation Measuremer	First Signal(Gate)	PFI_In1(31)	PFI_In5(27)	PFI_In9(23)	PFI_In13(19)	PFI_In17(15)	PFI_In21(11)	PFI_In25(7)	PFI_In29(3)
	Second Signal(Aux)	PFI_In2(63)	PFI_In6(59)	PFI_In10(55)	PFI_In14(51)	PFI_In18(47)	PFI_In22(43)	PFI_In26(39)	PFI_In30(35)
	External Timebase(Source)	PFI_In0(65)	PFI_In4(61)	PFI_In8(57)	PFI_In12(53)	PFI_In16(49)	PFI_In20(45)	PFI_In24(41)	PFI_In28(37)
Quad Encoder	A Signal(Source)	PFI_In0(65)	PFI_In4(61)	PFI_In8(57)	PFI_In12(53)	PFI_In16(49)	PFI_In20(45)	PFI_In24(41)	PFI_In28(37)
	B Signal(Aux)	PFI_In2(63)	PFI_In6(59)	PFI_In10(55)	PFI_In14(51)	PFI_In18(47)	PFI_In22(43)	PFI_In26(39)	PFI_In30(35)
	Z Signal(Gate)	PFI_In1(31)	PFI_In5(27)	PFI_In9(23)	PFI_In13(19)	PFI_In17(15)	PFI_In21(11)	PFI_In25(7)	PFI_In29(3)
Two-Pulse Encoder	A Signal(Source)	PFI_In0(65)	PFI_In4(61)	PFI_In8(57)	PFI_In12(53)	PFI_In16(49)	PFI_In20(45)	PFI_In24(41)	PFI_In28(37)
	B Signal(Aux)	PFI_In2(63)	PFI_In6(59)	PFI_In10(55)	PFI_In14(51)	PFI_In18(47)	PFI_In22(43)	PFI_In26(39)	PFI_In30(35)
Pulse Generation	Output(Out)	PFI_In3(29)	PFI_In7(25)	PFI_In11(21)	PFI_In15(17)	PFI_In19(13)	PFI_In23(9)	PFI_In27(5)	PFI_In31(1)
	External Timebase(Source)	PFI_In0(65)	PFI_In4(61)	PFI_In8(57)	PFI_In12(53)	PFI_In16(49)	PFI_In20(45)	PFI_In24(41)	PFI_In28(37)
Note: Start Trigger: PFI 32 (For all Counter), External Sample Clock: PFI 33 (For all Counter)									

Table 5 Counter Input/Output Default Routing

3. Software

3.1 System Requirements

PCIe/PXIe-5211 modules can be used in a Windows or a Linux operating system.

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

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

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

Table 6 Supported Linux Versions

3.2 System Software

When using the PCIe/PXIe-5211 in the Window environment, you need to install the following software from Microsoft website:

Microsoft Visual Studio Version 2015 or above,

.NET Framework version is 4.0 or above.

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

Given the resources limitation, JYTEK only tested PCIe/PXIe-5211 with .NET Framework 4.0 with Microsoft Visual Studio 2015. JYTEK relies on Microsoft to maintain the compatibility for the newer versions.

3.3 C# Programming Language

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

3.4 PCIe/PXIe-5211 Series Hardware Driver

After installing the required application development environment as described above, you need to install the PCIe/PXIe-5211 hardware driver.

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

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

Specific Hardware Driver: Each JYTEK hardware has a C# specific hardware driver. This driver provides rich and easy-to-use C# interfaces for users to operate various PCIe/PXIe-5211 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 module works, you should be able to know how to use all other DAQ hardware by using the same methods.

3.5 Install the SeeSharpTools from JYTEK

To efficiently and effectively use PCIe/PXIe-5211 boards, you need to install a set of free C# utilities, SeeSharpTools from JYTEK. The SeeSharpTools offers rich user interface functions you will find convenient in developing your applications. They are also needed to run the examples come with PCIe/PXIe-5211 hardware. Please register and down load the latest SeeSharpTools from our website, www.jytek.com.

3.6 Running C# Programs in Linux

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

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

4. Operating PCIe/PXIe-5211

This chapter provides the operation guides for PCIe/PXIe-5211, including Timer and programmable I/O interface, etc.

JYTEK provides extensive examples, on-line help and documentation to assist you to use the PCIe/PXIe-5211 module. JYTEK strongly recommends you go through these examples before writing your own application. In many cases, an example can also be a good starting point for a user application.

4.1 Quick Start

After you have installed the driver software and the SeeSharpTools, you are ready to use Microsoft Visual Studio C# to operate the PCIe/PXIe-5211 products.

If you are already familiar with Microsoft Visual Studio C#, the quickest way to use PCIe/PXIe-5211 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.

4.1 Digital I/O Operations

The PCIe/PXIe-5211 provides programable I/O function, and contains 40 channels of Programmable Function Interface (PFI).

All 40 PFIs can be used as static digital inputs or outputs. The direction and output state of each PFI can be controlled independently. Users can configure these PFIs by the driver.

In addition, the PCIe/PXIe-5211 also provides an Debounce Filter for each PFI, effectively eliminate the false pulse signal introduced by jitter.

Set JY5211CITask.Device.PFI.Filter.Enable to use this function.

This function enables the filter on Specified Group of Terminals and set the minPulseWidth (ns) of a specified length. Jitter smaller than this length will be ignored. The minPulseWidth is 8 ns in the range of 8 ns to 2040 ns, 128 ns in the range of 2040 ns to 32640 ns, and 2048 ns in the range of 32640 ns to 522240 ns

4.2 Counter Measurement Operations

The PCIe/PXIe-5211 has eight identical 32-bit channels of timer/counter as shown in Figure 4.

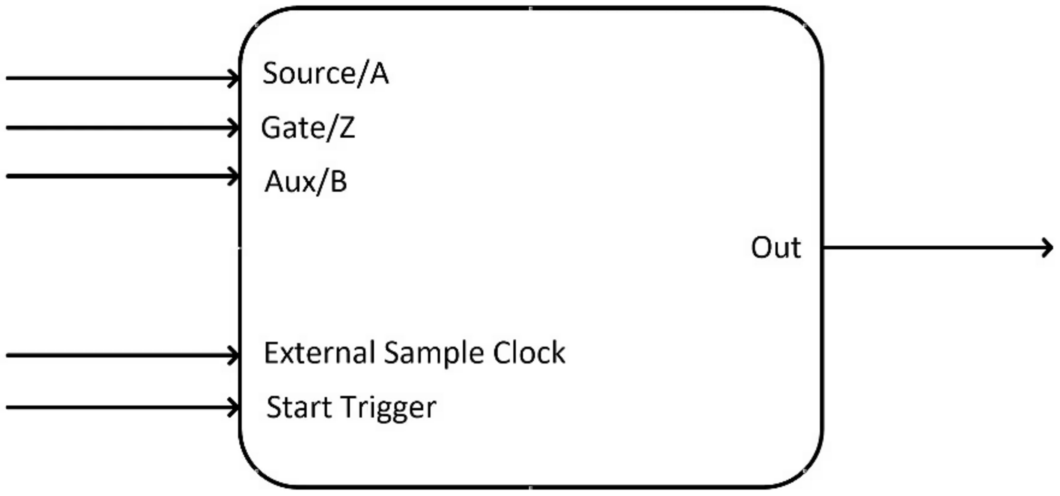


Figure 4 Counter Input Diagram

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

- Edge Counting
- Pulse Measurement
- Frequency Measurement
- Period Measurement
- Two-Edge Separation
- Quadrature Encoder (x1, x2, x4)
- Two-Pulse Encoder

For buffered acquisition, each counter has a separate DDR storage space and requires a sample clock. For more information about sample clock, please refer to chapter 4.4.2.

4.2.1 Edge Counting

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

Set JY5211CITask.Type to CIType.EdgeCounting to use this function.

Timing

1) Single Mode

The counting value is written to the register on each rising edge or falling edge of the measured signal as shown in Figure 5.

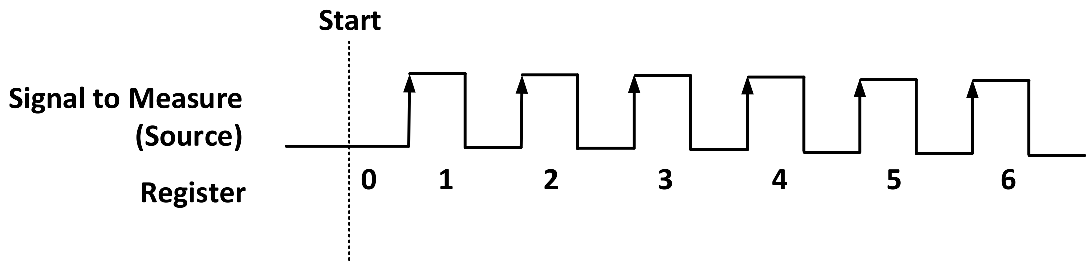


Figure 5 Simple Edge Counting in Single Mode

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

2) Finite/Continuous Mode with Explicit Sample Clock

The counting value is stored into the buffer on each rising edge of the sample clock as shown in Figure 6.

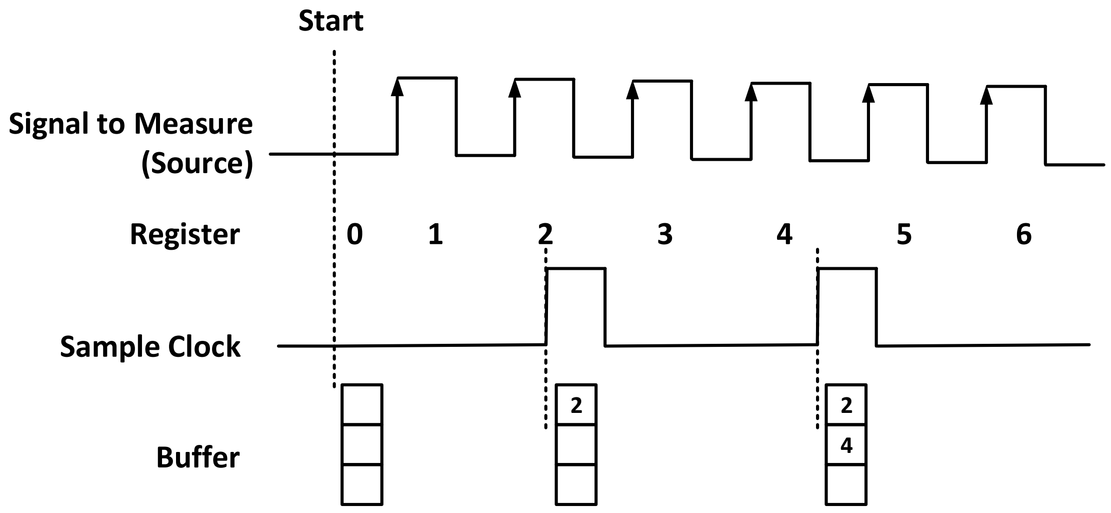


Figure 6 Buffered Edge Counting with Explicit Sample Clock

To configure the counter to work in this mode, set JY5211CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5211CITask.SampleClock.Source to CISampleClockSource.Internal or CISampleClockSource.External.

3) Finite/Continuous Mode with Implicit Sample Clock

In implicit mode, the signal active edge as the implicit sample clock edge. The counting value is stored into the buffer on each rising edge of the measured signal as shown in Figure 7.

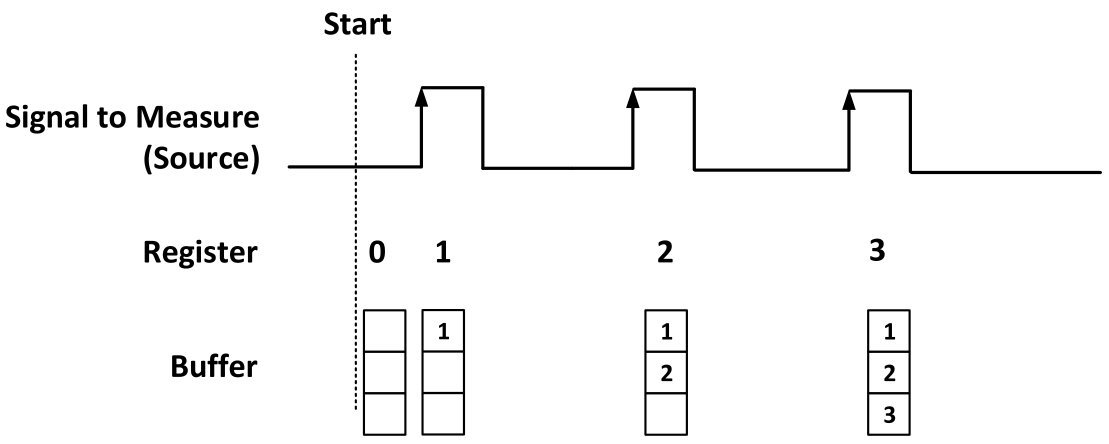


Figure 7 Simple Edge Counting with Implicit Sample Clock

To configure the counter to work in this mode, set JY5211CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5211CITask.SampleClock.Source to CISampleClockSource.Implicit.

Pause Trigger

Pause trigger is used to pause counting when the input signal is active depending on active polarity configuration as shown in Figure 8. Default, the Pause Trigger signal must be connected to Counter Gate terminal.

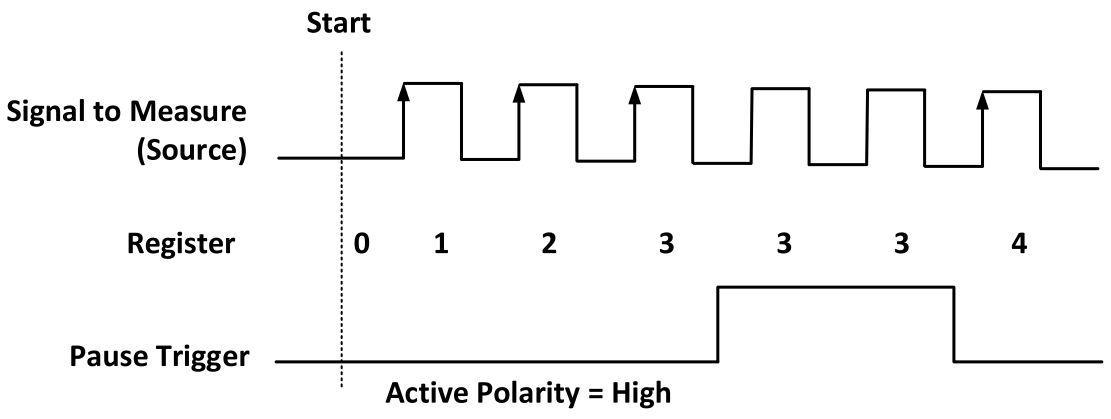


Figure 8 Pause Trigger

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

- JY5211CITask.EdgeCounting.Pause.ActivePolarity – To set active level (high or low) to pause counting.

Count Direction

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

When using an input signal to control the counting direction, the counter counts up when the signal is high and counts down when the signal is low as shown in Figure 9.

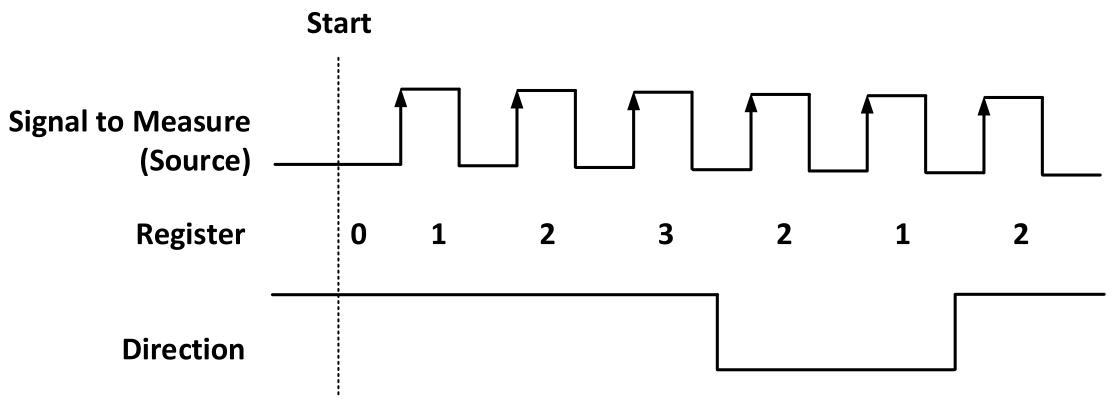


Figure 9 Count Direction

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

- JY5211CITask.EdgeCounting.Direction – To specify count up, count down, or controled by an external signal.

Exporting Count Event

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

- JY5211CITask.EdgeCounting.OutEvent.Threshold

Terminals

To change the terminal of signals instead of using its default value as shown in chapter 2.3, using following properties:

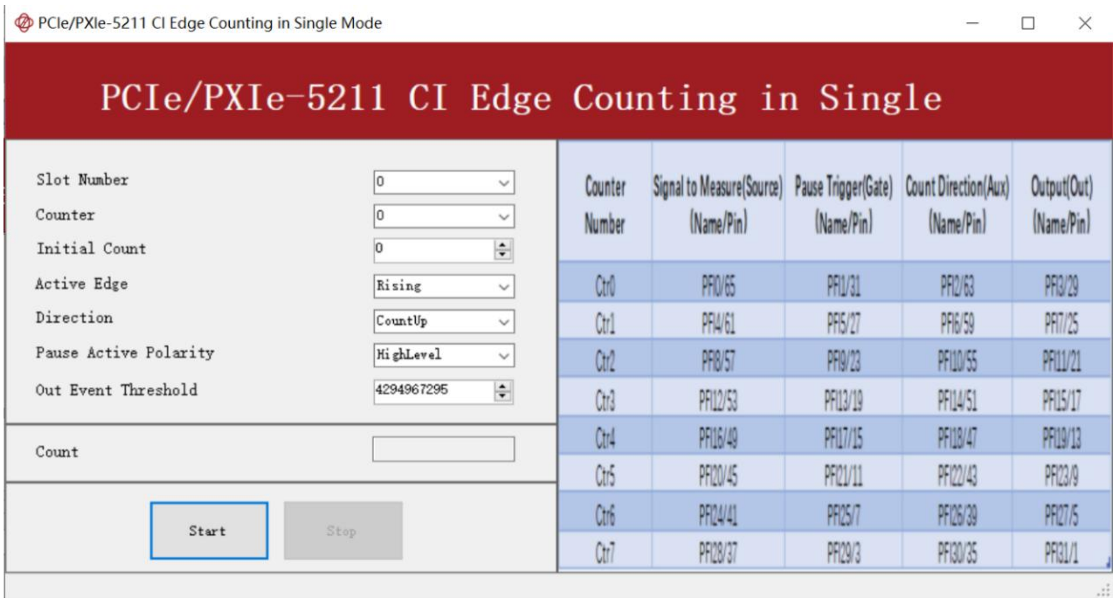
- JY5211CITask.EdgeCounting.InputTerminal – Signal-to-measure input terminal.
- JY5211CITask.EdgeCounting.Pause.Terminal – Pause signal input terminal.
- JY5211CITask.EdgeCounting.DirTerminal – External direction control signal input terminal.
- JY5211CITask.EdgeCounting.OutEvent.Terminal – Count event output terminal.

Learn by Examples 4.2.1

- Connect the signal source's positive terminal Ch1 of a signal source to PCIe/PXIe-5211 counter0's edge counting source (CTR0_Source, Pin#65), negative terminal to the ground (GND, Pin#66) as shown in Table 2-15 Pin Defination. (CTR0_Source, GND) consists of an edge counting counter input.
- Set the signal source Ch1's output to squarewave signal (f=1Hz, V_H=3.3v, V_L=0v).

Single Mode

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


PCIe/PXIe-5211 CI Edge Counting in Single Mode

PCIe/PXIe-5211 CI Edge Counting in Single

Slot Number	0
Counter	0
Initial Count	0
Active Edge	Rising
Direction	CountUp
Pause Active Polarity	HighLevel
Out Event Threshold	4294967295
Count	

Counter Number	Signal to Measure(Source) (Name/Pin)	Pause Trigger(Gate) (Name/Pin)	Count Direction(Aux) (Name/Pin)	Output(Out) (Name/Pin)
Ctr0	PFI0/65	PFI1/31	PFI2/63	PFI3/29
Ctr1	PFI4/61	PFI5/27	PFI6/59	PFI7/25
Ctr2	PFI8/57	PFI9/23	PFI10/55	PFI11/21
Ctr3	PFI12/53	PFI13/19	PFI14/51	PFI15/17
Ctr4	PFI16/49	PFI17/15	PFI18/47	PFI19/13
Ctr5	PFI20/45	PFI21/11	PFI22/43	PFI23/9
Ctr6	PFI24/41	PFI25/7	PFI26/39	PFI27/5
Ctr7	PFI28/37	PFI29/3	PFI30/35	PFI31/1

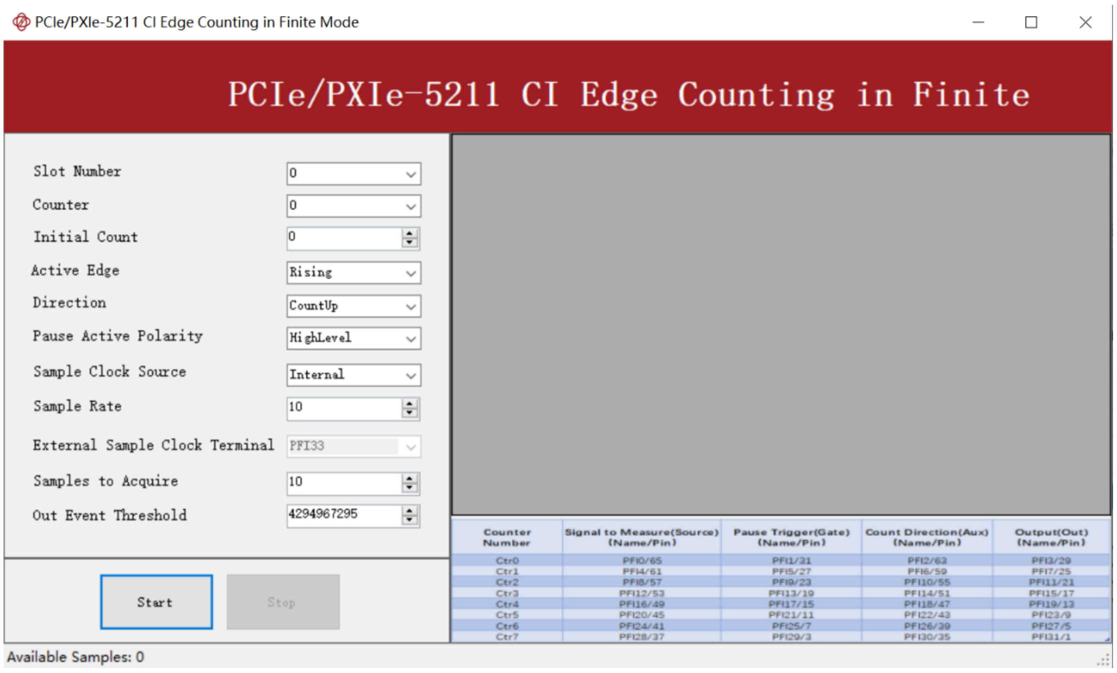
Figure 10 Edge Counting In Single Mode

- *Count direction* is set by **Direction.**
- The table in the sample program is a connection diagram for your convenience.
- Click **Start**, and the result is shown by **Count**. In this example **Count** increases by 1 every second for a 1Hz squarewave.

Finite/Continuous Mode

- Change the squarewave frequency to 50 Hz.

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



PCIe/PXIe-5211 CI Edge Counting in Finite Mode

PCIe/PXIe-5211 CI Edge Counting in Finite

Slot Number	0
Counter	0
Initial Count	0
Active Edge	Rising
Direction	CountUp
Pause Active Polarity	HighLevel
Sample Clock Source	Internal
Sample Rate	10
External Sample Clock Terminal	PFI33
Samples to Acquire	10
Out Event Threshold	4294967295

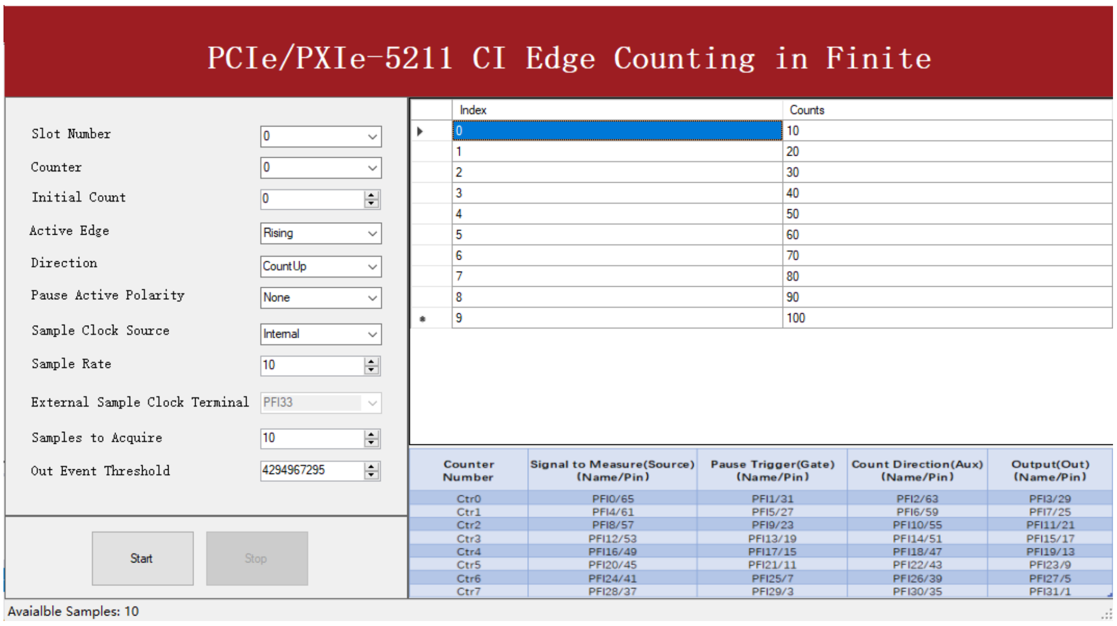
Counter Number	Signal to Measure(Source) (Name/Pin)	Pause Trigger(Gate) (Name/Pin)	Count Direction(Aux) (Name/Pin)	Output(Out) (Name/Pin)
Ctr0	PFI0/65	PFI1/31	PFI2/63	PFI3/29
Ctr1	PFI4/61	PFI5/27	PFI6/59	PFI7/25
Ctr2	PFI8/57	PFI9/23	PFI10/55	PFI11/21
Ctr3	PFI12/53	PFI13/19	PFI14/51	PFI15/17
Ctr4	PFI16/49	PFI17/15	PFI18/47	PFI19/13
Ctr5	PFI20/45	PFI21/11	PFI22/43	PFI23/9
Ctr6	PFI24/41	PFI25/7	PFI26/39	PFI27/5
Ctr7	PFI28/37	PFI29/3	PFI30/35	PFI31/1

Available Samples: 0

Figure 11 Edge Counting In Finite Mode

- The table in the sample program is a connection diagram for your convenience.
- *Direction* is set by **Direction.**
- There are three sample clock sources in PCIe/PXIe-5211 set by **Sample Clock Source**: **Internal**, **Implicit** and **External**.

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



PCIe/PXIe-5211 CI Edge Counting in Finite

Slot Number	0
Counter	0
Initial Count	0
Active Edge	Rising
Direction	CountUp
Pause Active Polarity	None
Sample Clock Source	Internal
Sample Rate	10
External Sample Clock Terminal	PFI33
Samples to Acquire	10
Out Event Threshold	4294967295

Index	Counts
0	10
1	20
2	30
3	40
4	50
5	60
6	70
7	80
8	90
9	100

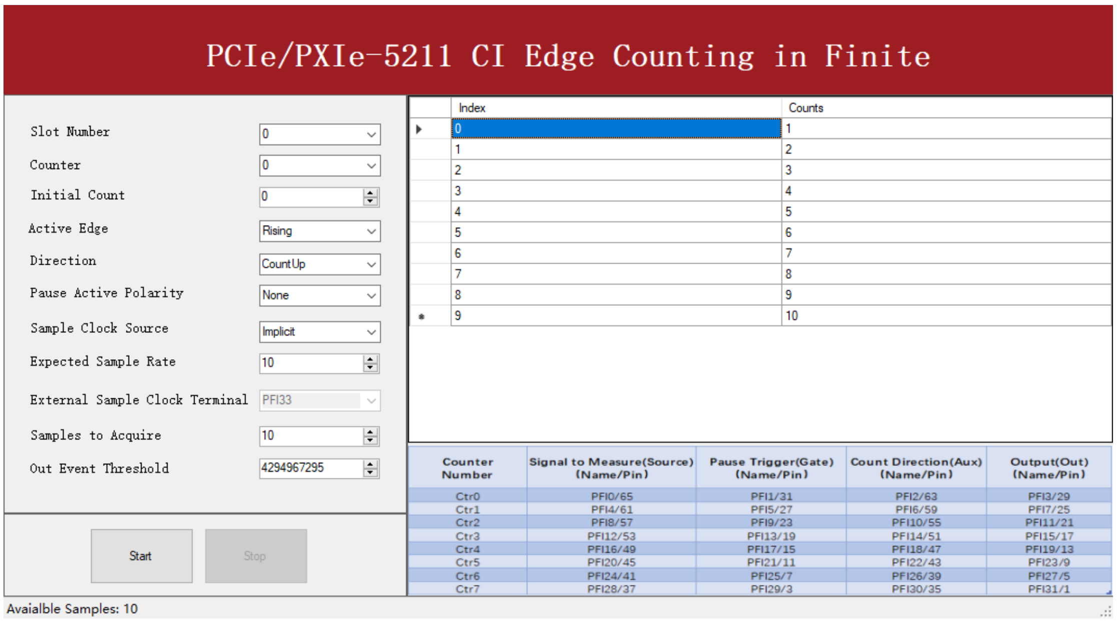
Counter Number	Signal to Measure(Source) (Name/Pin)	Pause Trigger(Gate) (Name/Pin)	Count Direction(Aux) (Name/Pin)	Output(Out) (Name/Pin)
Ctr0	PFI0/65	PFI1/31	PFI2/63	PFI3/29
Ctr1	PFI4/61	PFI5/27	PFI6/59	PFI7/25
Ctr2	PFI8/57	PFI9/23	PFI10/55	PFI11/21
Ctr3	PFI12/53	PFI13/19	PFI14/51	PFI15/17
Ctr4	PFI16/49	PFI17/15	PFI18/47	PFI19/13
Ctr5	PFI20/45	PFI21/11	PFI22/43	PFI23/9
Ctr6	PFI24/41	PFI25/7	PFI26/39	PFI27/5
Ctr7	PFI28/37	PFI29/3	PFI30/35	PFI31/1

Avaialble Samples: 10

Figure 12 Edge Counting In Continuous Mode

- ➢ The numbers are stored in a buffer **Counts**.

- ■ Change the Sample Clock Source to Implicit:



PCIe/PXIe-5211 CI Edge Counting in Finite

Slot Number	0
Counter	0
Initial Count	0
Active Edge	Rising
Direction	CountUp
Pause Active Polarity	None
Sample Clock Source	Implicit
Expected Sample Rate	10
External Sample Clock Terminal	PFI33
Samples to Acquire	10
Out Event Threshold	4294967295

Index	Counts
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10

Counter Number	Signal to Measure(Source) (Name/Pin)	Pause Trigger(Gate) (Name/Pin)	Count Direction(Aux) (Name/Pin)	Output(Out) (Name/Pin)
Ctr0	PFI0/65	PFI1/31	PFI2/63	PFI3/29
Ctr1	PFI4/61	PFI5/27	PFI6/59	PFI7/25
Ctr2	PFI8/57	PFI9/23	PFI10/55	PFI11/21
Ctr3	PFI12/53	PFI13/19	PFI14/51	PFI15/17
Ctr4	PFI16/49	PFI17/15	PFI18/47	PFI19/13
Ctr5	PFI20/45	PFI21/11	PFI22/43	PFI23/9
Ctr6	PFI24/41	PFI25/7	PFI26/39	PFI27/5
Ctr7	PFI28/37	PFI29/3	PFI30/35	PFI31/1

Avaialble Samples: 10

Figure 13 Edge Counting With Implicit Clock

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

4.2.2 Pulse Measurement

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

Set JY5211CITask.Type to CIType.Pulse to use this function.

Timing

1) Single Mode

The counting value of the duration of the high-level or low-level is written to the register on each rising or falling edge of the pulse to measure, as shown in Figure 14.

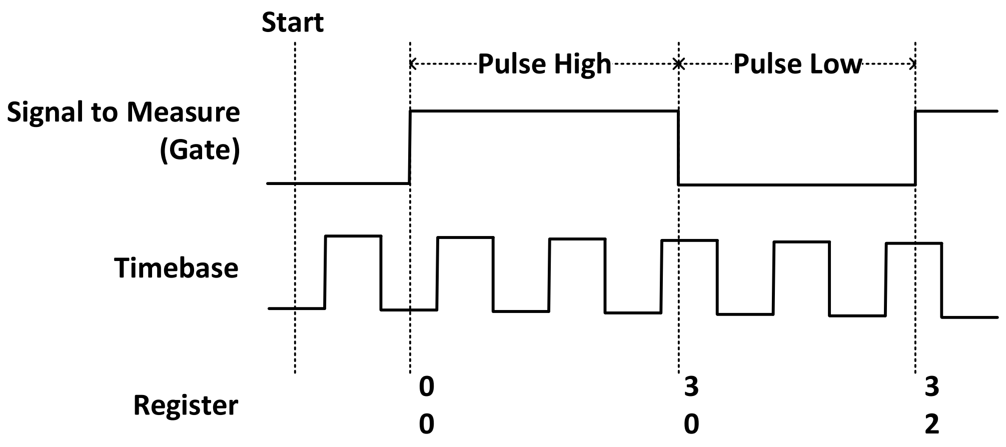


Figure 14 Pulse Measurement in Single Mode

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

2) Finite/Continuous Mode with Explicit Sample Clock

The counting value of the duration of the high-level or low-level is stored into the buffer on each rising edge of the sample clock, as shown in Figure 15.

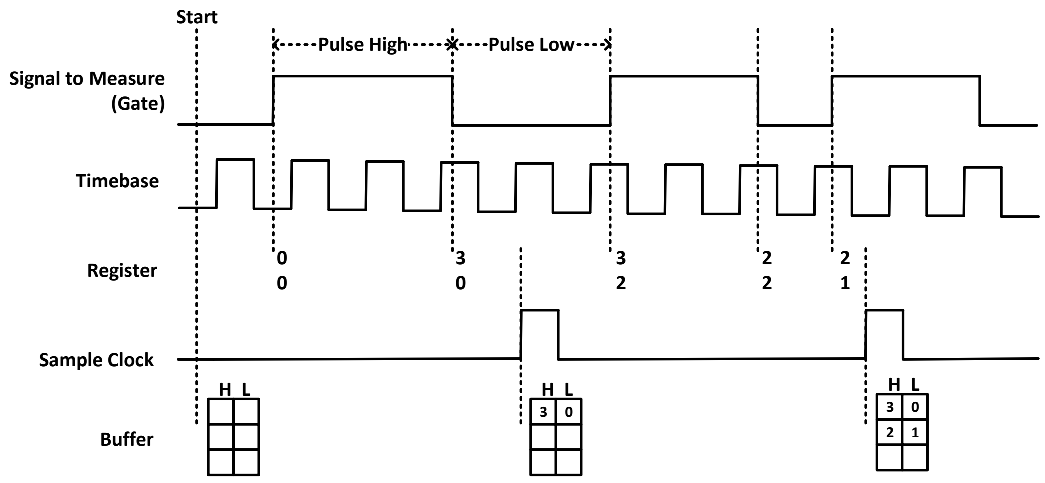


Figure 15 Pulse Measurement with Explicit Sample Clock

To configure the counter to work in this mode, set JY5211CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5211CITask.SampleClock.Source to CISampleClockSource.Internal or CISampleClockSource.External.

3) Finite/Continuous Mode with Implicit Sample Clock

In implicit mode, the signal active edge as the implicit sample clock edge. The counting value of the duration of the high-level or low-level is stored into the buffer on each rising edge of the measured pulse, as shown in Figure 16.

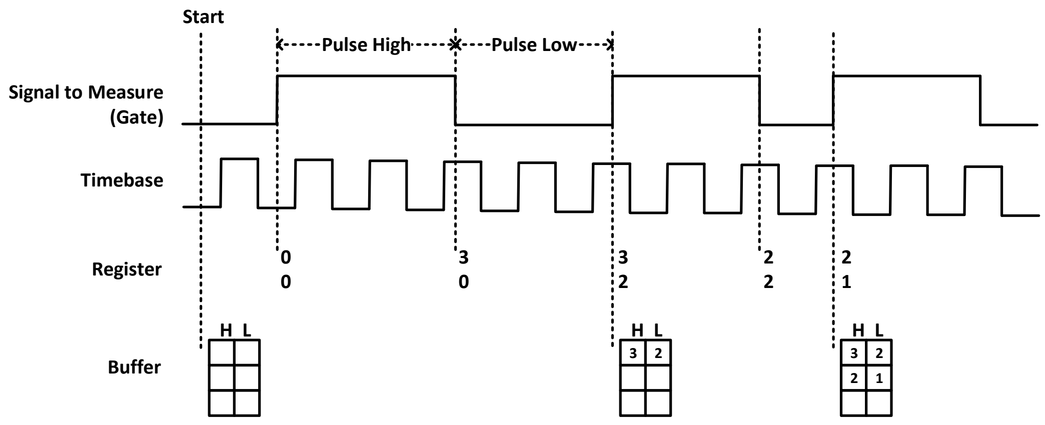


Figure 16 Pulse Measurement with Implicit Sample Clock

To configure the counter to work in this mode, set JY5211CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5211CITask.SampleClock.Source to CISampleClockSource.Implicit.

Timebase

By default, the counter uses the onboard 200MHz timebase to measure pulses. Use the property JY5211CITask.PulseMeas.Timebase to configure the timebase.

Please refer to chapter 4.4.3 for more information about timebase.

Terminals

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

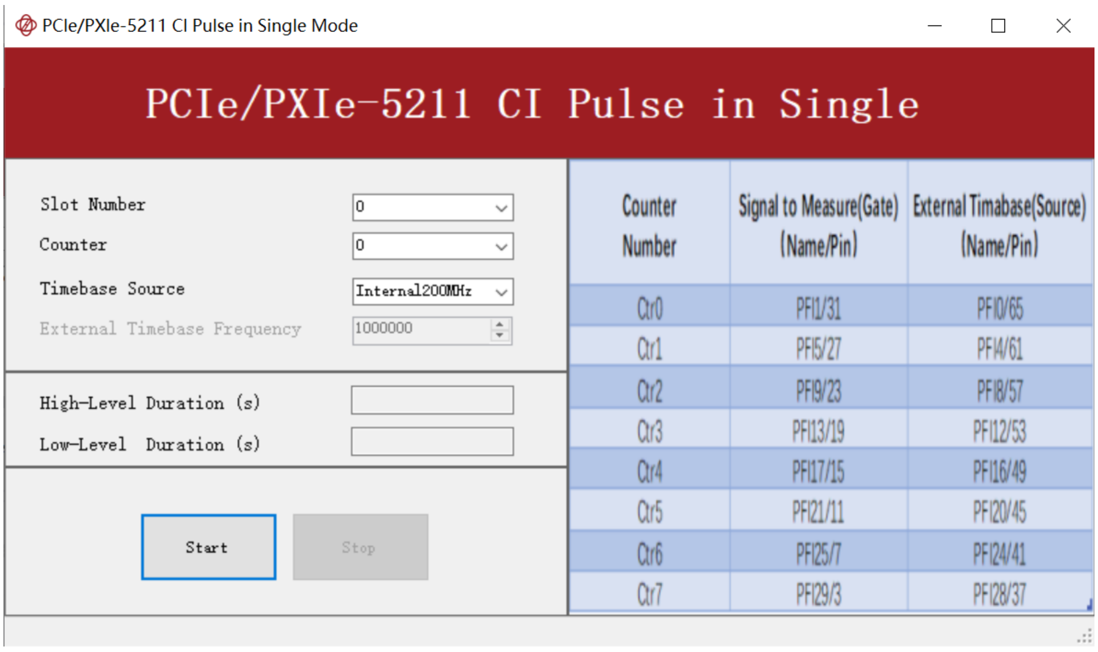
- JY5211CITask.PulseMeas.InputTerminal – Signal-to-measure input terminal.
- JY5211CITask.PulseMeas.Timebase.External.Terminal – External timebase input terminal.

Learn by Examples 4.2.2

- Connect the signal source's positive terminal Ch1 to PCIe/PXIe-5211 counter0's pulse measure source (CTR0_Gate, Pin#31), negative terminal to the ground (GND, Pin#32) as shown in Table 2-15 Pin Defination. (CTR0_Gate, GND) consists of a pulse measure counter input.
- Set the signal source Ch1's output to squarewave signal (f=1Hz, Duty Cycle=50%, V_H=3.3v, V_L=0v).

Single Mode

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



PCIe/PXIe-5211 CI Pulse in Single Mode

PCIe/PXIe-5211 CI Pulse in Single

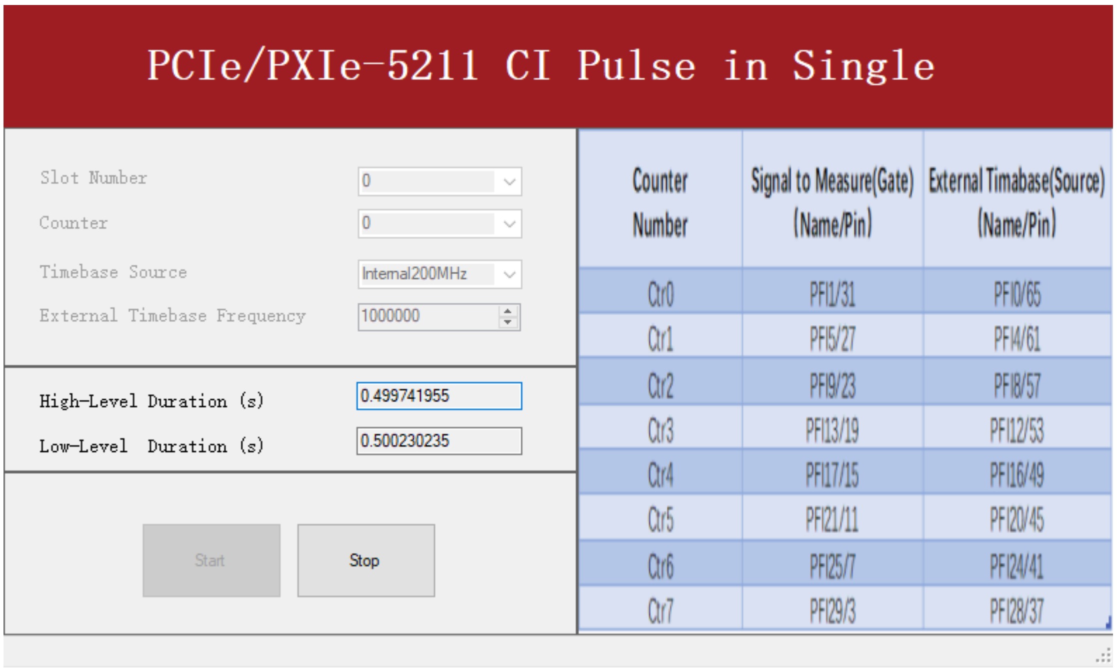
Slot Number	0
Counter	0
Timebase Source	Internal200MHz
External Timebase Frequency	1000000
High-Level Duration (s)	
Low-Level Duration (s)	

Counter Number	Signal to Measure(Gate) (Name/Pin)	External Timabase(Source) (Name/Pin)
Ctr0	PFI1/31	PFI0/65
Ctr1	PFI5/27	PFI4/61
Ctr2	PFI9/23	PFI8/57
Ctr3	PFI13/19	PFI12/53
Ctr4	PFI17/15	PFI16/49
Ctr5	PFI21/11	PFI20/45
Ctr6	PFI25/7	PFI24/41
Ctr7	PFI29/3	PFI28/37

Figure 17 Pulse Measure In 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-Level Duration (s)** and **Low-Level Duration (s)**:


PCIe/PXIe-5211 CI Pulse in Single

Slot Number	0
Counter	0
Timebase Source	Internal200MHz
External Timebase Frequency	1000000
High-Level Duration (s)	0.499741955
Low-Level Duration (s)	0.500230235

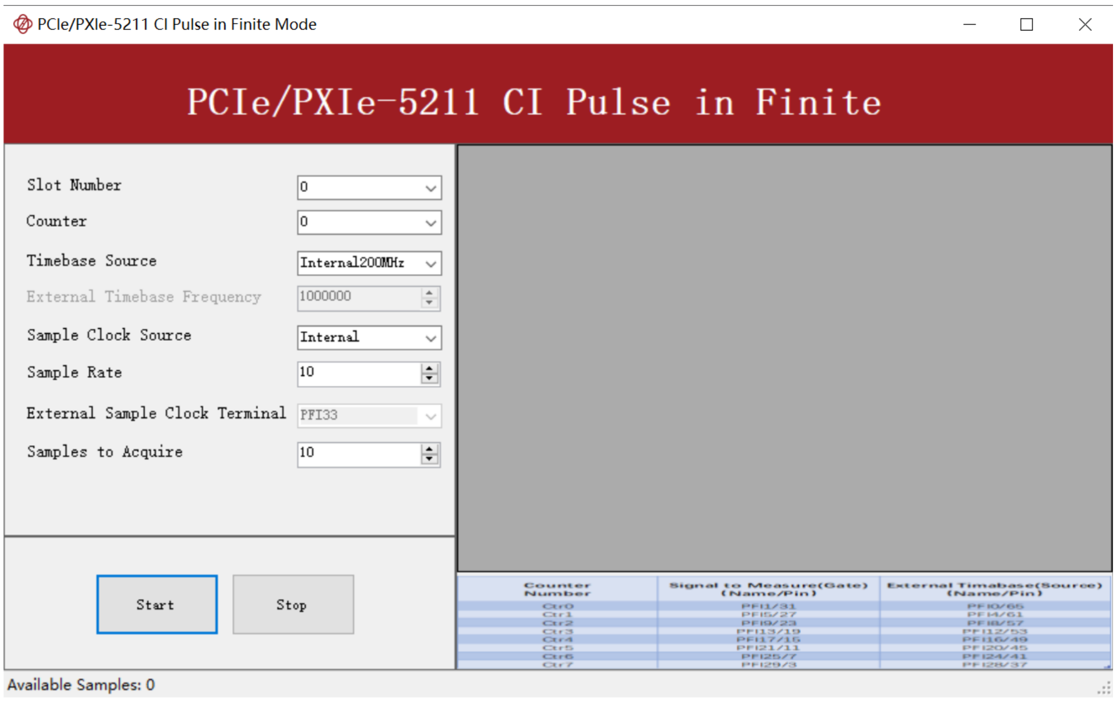
Counter Number	Signal to Measure(Gate) (Name/Pin)	External Timabase(Source) (Name/Pin)
Ctr0	PFI1/31	PFI0/65
Ctr1	PFI5/27	PFI4/61
Ctr2	PFI9/23	PFI8/57
Ctr3	PFI13/19	PFI12/53
Ctr4	PFI17/15	PFI16/49
Ctr5	PFI21/11	PFI20/45
Ctr6	PFI25/7	PFI24/41
Ctr7	PFI29/3	PFI28/37

Figure 18 Pulse Measure Value In Single Mode

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

Finite/Continuous Mode

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


PCIe/PXIe-5211 CI Pulse in Finite Mode

PCIe/PXIe-5211 CI Pulse in Finite

Slot Number	0
Counter	0
Timebase Source	Internal200MHz
External Timebase Frequency	1000000
Sample Clock Source	Internal
Sample Rate	10
External Sample Clock Terminal	PFI33
Samples to Acquire	10

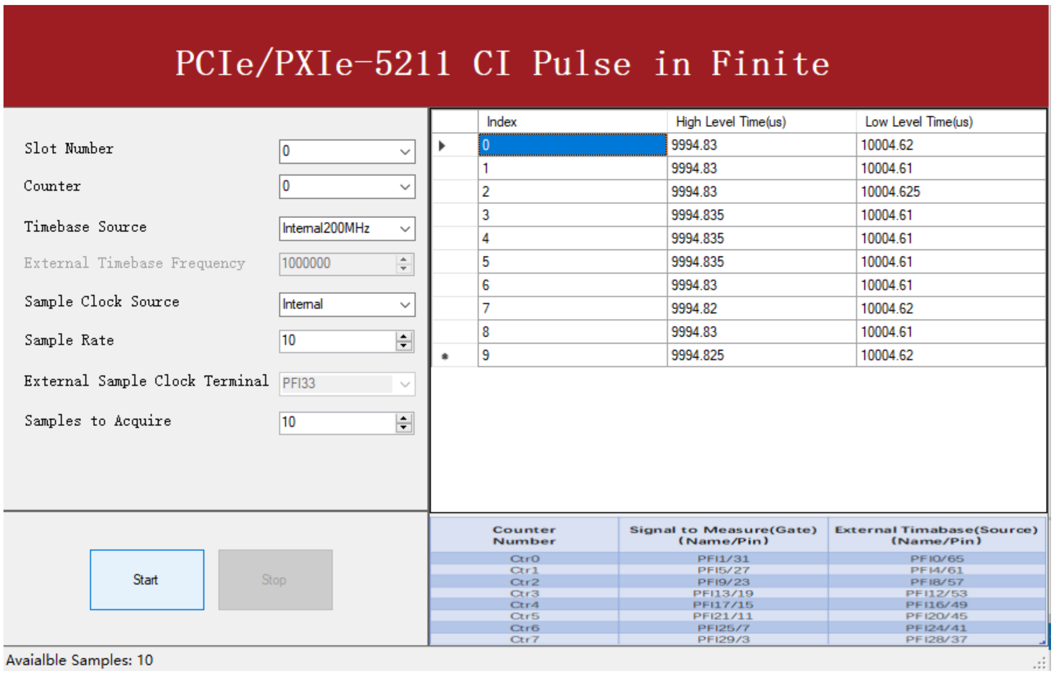
Counter Number	Signal to Measure(Gate) (Name/Pin)	External Timabase(Source) (Name/Pin)
Ctr0	PFI1/31	PFI0/65
Ctr1	PFI5/27	PFI4/61
Ctr2	PFI9/23	PFI8/57
Ctr3	PFI13/19	PFI12/53
Ctr4	PFI17/15	PFI16/49
Ctr5	PFI21/11	PFI20/45
Ctr6	PFI25/7	PFI24/41
Ctr7	PFI29/3	PFI28/37

Available Samples: 0

Figure 19 Pulse Measure In 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:


PCIe/PXIe-5211 CI Pulse in Finite

Slot Number	0
Counter	0
Timebase Source	Internal200MHz
External Timebase Frequency	1000000
Sample Clock Source	Internal
Sample Rate	10
External Sample Clock Terminal	PFI33
Samples to Acquire	10

Index	High Level Time(us)	Low Level Time(us)
0	9994.83	10004.62
1	9994.83	10004.61
2	9994.83	10004.625
3	9994.835	10004.61
4	9994.835	10004.61
5	9994.835	10004.61
6	9994.83	10004.61
7	9994.82	10004.62
8	9994.83	10004.61
9	9994.825	10004.62

Counter Number	Signal to Measure(Gate) (Name/Pin)	External Timabase(Source) (Name/Pin)
Ctr0	PFI1/31	PFI0/65
Ctr1	PFI5/27	PFI4/61
Ctr2	PFI9/23	PFI8/57
Ctr3	PFI13/19	PFI12/53
Ctr4	PFI17/15	PFI16/49
Ctr5	PFI21/11	PFI20/45
Ctr6	PFI25/7	PFI24/41
Ctr7	PFI29/3	PFI28/37

Avaialble Samples: 10

Figure 20 Pulse Measure Values In Finite Mode

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

Please refer to **Learn by Examples 4.3.1 Finite/Continuous Mode** about the difference between *Explicit* and *Implicit.*

4.2.3 Frequency Measurement

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

Set JY5211CITask.Type to CIType.Frequency to use this function.

Timing

1) Single Mode

Frequency Measurement without sample clock is actually using Pulse Width Measuement internally, refer to chapter 4.2.2 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 fomular and return the signal frequency to the user.

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

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

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

Between every two rising edges of the sample clock, the counter counts the number of full periods ($T1$) of the signal, and the number of rising edges of timebase ($T2$) during those full periods. These two values are stored into the buffer on each rising edge of the sample clock, as shown in Figure 21.

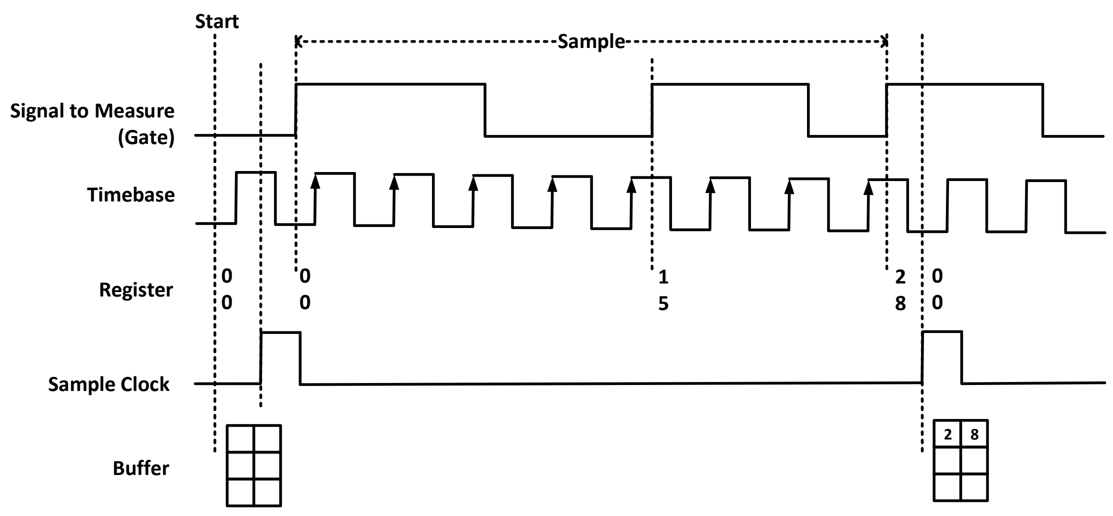


Figure 21 Frequency Measurement with Explicit Sample Clock

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

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

To configure the counter to work in this mode, set JY5211CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5211CITask.SampleClock.Source to CISampleClockSource.Internal or CISampleClockSource.External.

3) Finite/Continuous Mode with Implicit Sample Clock

Frequency Measurement with implicit sample clock is actually using Pulse Measuement internally. Refer to chapter 4.2.2 for more information.

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

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

To configure the counter to work in this mode, set JY5211CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5211CITask.SampleClock.Source to CISampleClockSource.Implicit.

Timebase

By default, the counter uses the onboard 200MHz timebase to measure pulses. Use the property JY5211CITask.FrequencyMeas.Timebase to configure the timebase.

Please refer to chapter 4.4.3 for more information about timebase.

Terminals

To change the terminal of signals instead of using its default value shown in chapter 2.3, using following properties:

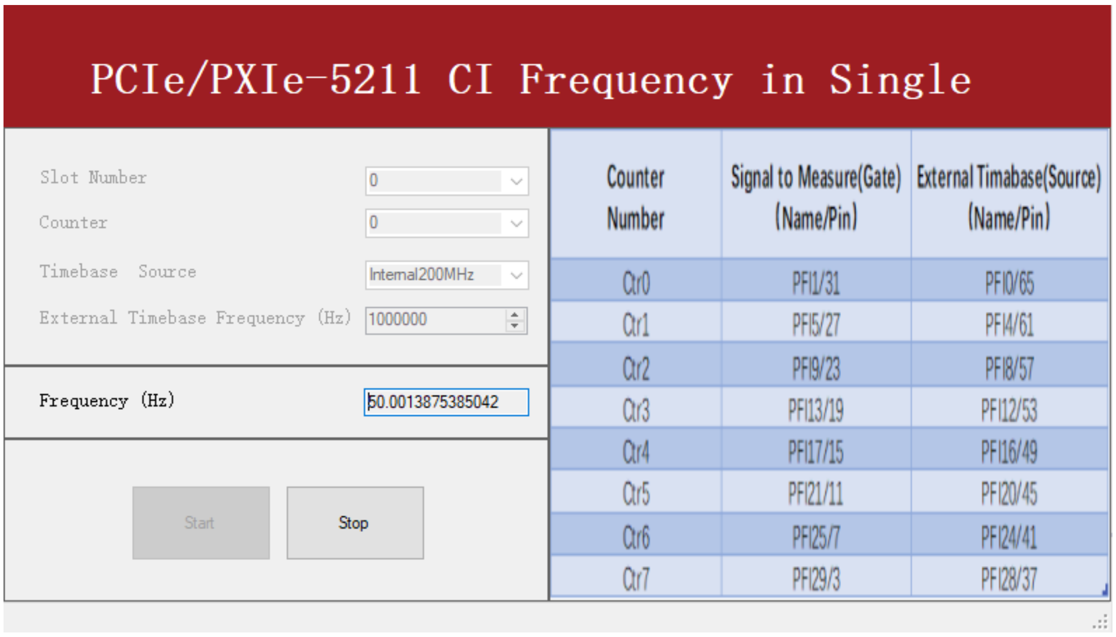
- JY5211CITask.FrequencyMeas.InputTerminal – Signal-to-measure input terminal.
- JY5211CITask. FrequencyMeas.Timebase.External.Terminal – External timebase input terminal.

Learn by Examples 4.2.3

- Connect the signal source's positive terminal Ch1 to PCIe/PXIe-5211 counter0's frequency measure source (CTR0_Gate, Pin#31), negative terminal to the ground (GND, Pin#32) as shown in Table 2-15 Pin Defination. (CTR0_Gate, GND) consists of a frequency measure counter input.
- Set the signal source Ch1's output to squarewave signal (f=50Hz, Duty Cycle=50%, V_H=3.3v, V_L=0v).

Single Mode

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



PCIe/PXIe-5211 CI Frequency in Single

Slot Number: 0
Counter: 0
Timebase Source: Internal200MHz
External Timebase Frequency (Hz): 1000000

Frequency (Hz): 50.0013875385042

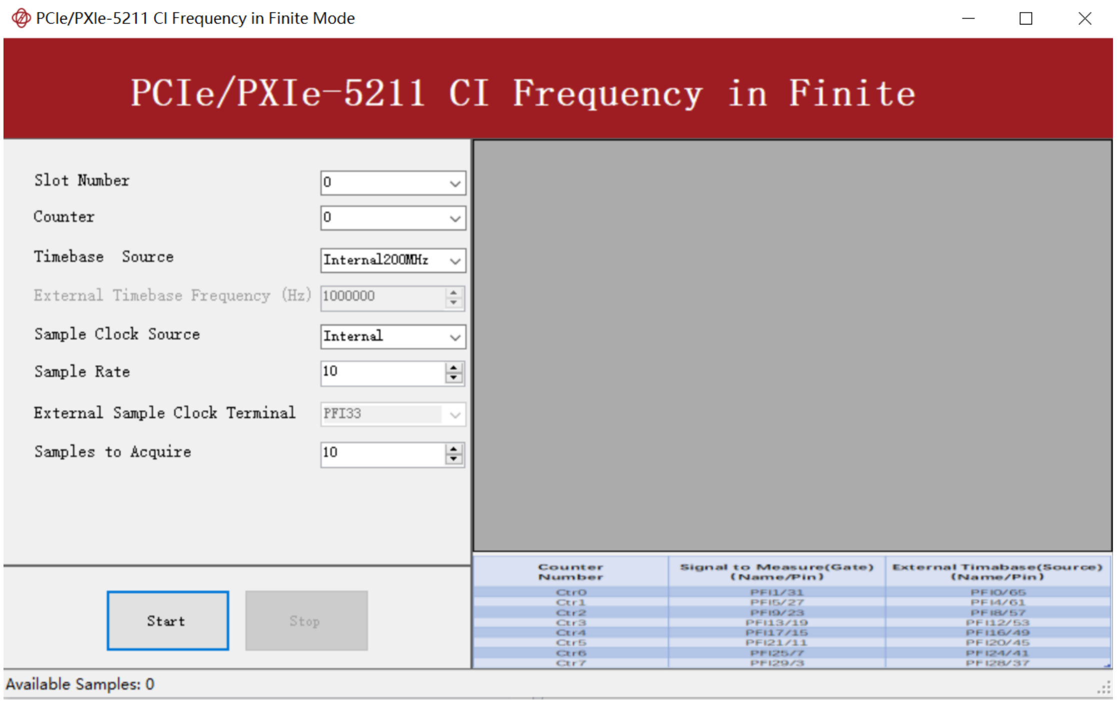
Counter Number	Signal to Measure(Gate) (Name/Pin)	External Timabase(Source) (Name/Pin)
Ctr0	PFI1/31	PFI0/65
Ctr1	PFI5/27	PFI4/61
Ctr2	PFI9/23	PFI8/57
Ctr3	PFI13/19	PFI12/53
Ctr4	PFI17/15	PFI16/49
Ctr5	PFI21/11	PFI20/45
Ctr6	PFI25/7	PFI24/41
Ctr7	PFI29/3	PFI28/37

Figure 22 Frequency Measure In 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.**


PCIe/PXIe-5211 CI Frequency in Finite Mode

PCIe/PXIe-5211 CI Frequency in Finite

Slot Number	0
Counter	0
Timebase Source	Internal200MHz
External Timebase Frequency (Hz)	1000000
Sample Clock Source	Internal
Sample Rate	10
External Sample Clock Terminal	PFI33
Samples to Acquire	10

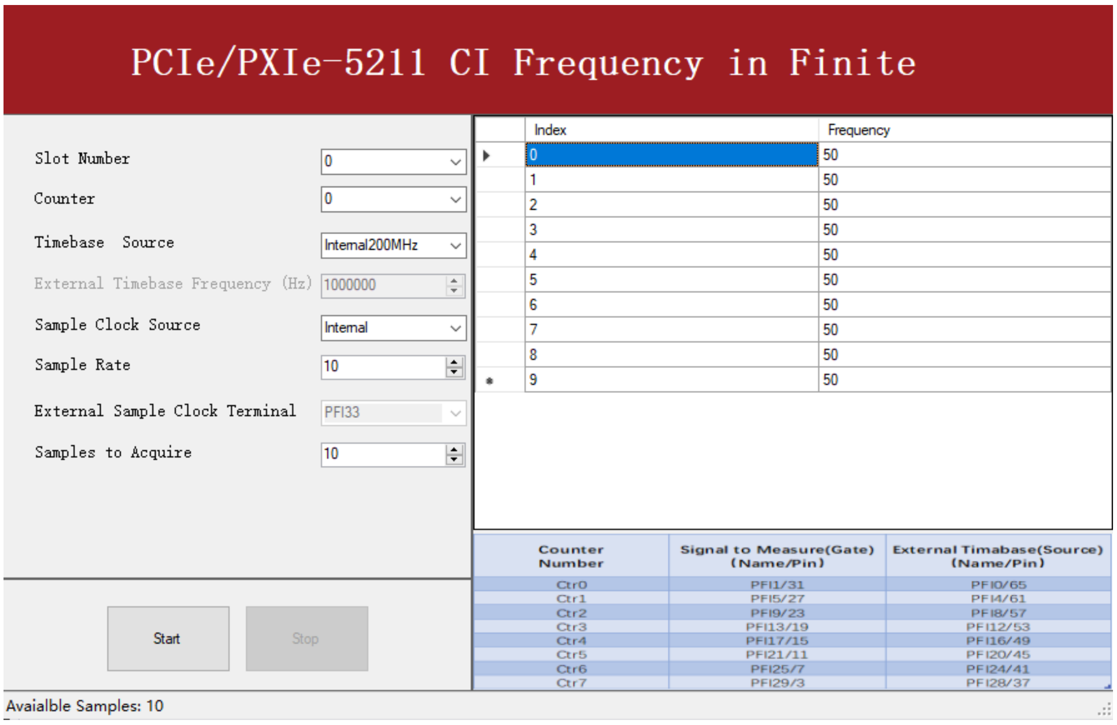
Counter Number	Signal to Measure(Gate) (Name/Pin)	External Timabase(Source) (Name/Pin)
Ctr0	PFI1/31	PFI0/65
Ctr1	PFI5/27	PFI4/61
Ctr2	PFI9/23	PFI8/57
Ctr3	PFI13/19	PFI12/53
Ctr4	PFI17/15	PFI16/49
Ctr5	PFI21/11	PFI20/45
Ctr6	PFI25/7	PFI24/41
Ctr7	PFI29/3	PFI28/37

Available Samples: 0

Figure 23 Frequency Measure In Finite Mode

- ➢ The table in the sample program is a connection diagram for your convenience.
- ➢ Please refer to **Learn by Examples 4.3.1 Finite/Continuous Mode** about the difference between *Explicit* and *Implicit*.

- ■ Click **Start** and it will show the frequency 50 as set in the signal source.



PCIe/PXIe-5211 CI Frequency in Finite

Slot Number	0
Counter	0
Timebase Source	Internal200MHz
External Timebase Frequency (Hz)	1000000
Sample Clock Source	Internal
Sample Rate	10
External Sample Clock Terminal	PFI33
Samples to Acquire	10

Index	Frequency
0	50
1	50
2	50
3	50
4	50
5	50
6	50
7	50
8	50
9	50

Counter Number	Signal to Measure(Gate) (Name/Pin)	External Timabase(Source) (Name/Pin)
Ctr0	PFI1/31	PFI0/65
Ctr1	PFI5/27	PFI4/61
Ctr2	PFI9/23	PFI8/57
Ctr3	PFI13/19	PFI12/53
Ctr4	PFI17/15	PFI16/49
Ctr5	PFI21/11	PFI20/45
Ctr6	PFI25/7	PFI24/41
Ctr7	PFI29/3	PFI28/37

Avaialble Samples: 10

Figure 24 Frequency Measure Values In Single Mode

4.2.4 Period Measurement

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

Set JY5211CITask.Type to CIType.Period to use this function.

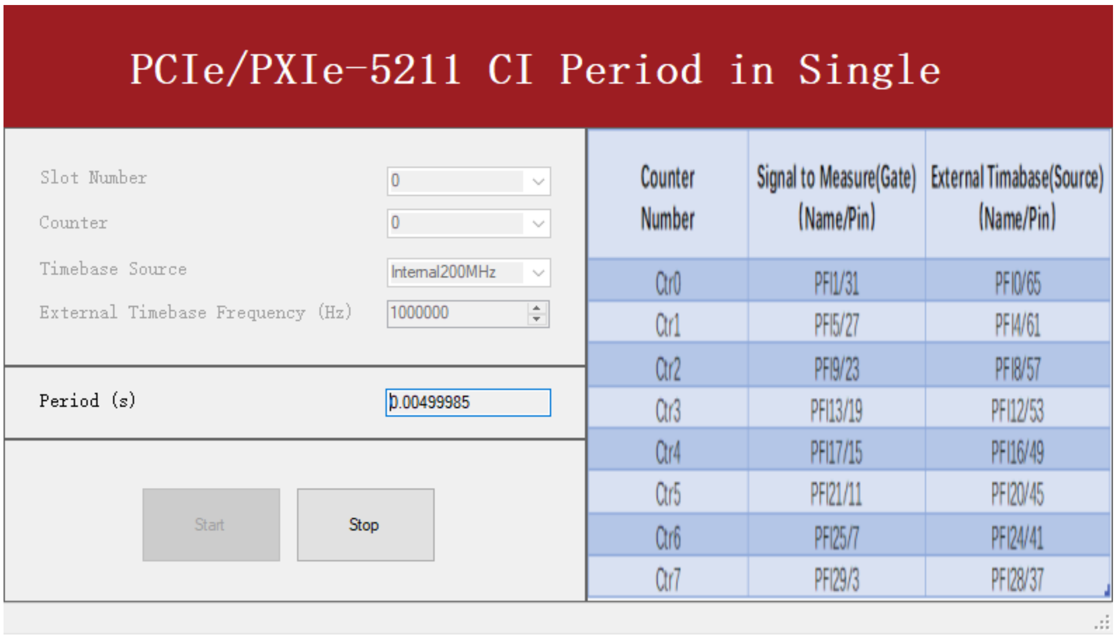
Period Measurements is using Frequency Measurement internally and returns the reciprocal of Frequency Measuremnt. Refer to chapter for more information.

Learn by Examples 4.2.4

- Connect the signal source's positive terminal Ch1 to PCIe/PXIe-5211 counter0's period measure source (CTR0_Gate, Pin#31), negative terminal to the ground (GND, Pin#32) as shown in Table 2-15 Pin Defination. (CTR0_Gate, GND) consists of a period measure counter input.
- Set the signal source Ch1's output to squarewave signal (f=200Hz, Duty Cycle=50%, V_H=3.3v, V_L=0v).

Single Mode

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


PCIe/PXIe-5211 CI Period in Single

Slot Number: 0
Counter: 0
Timebase Source: Internal200MHz
External Timebase Frequency (Hz): 1000000

Period (s): 0.00499985

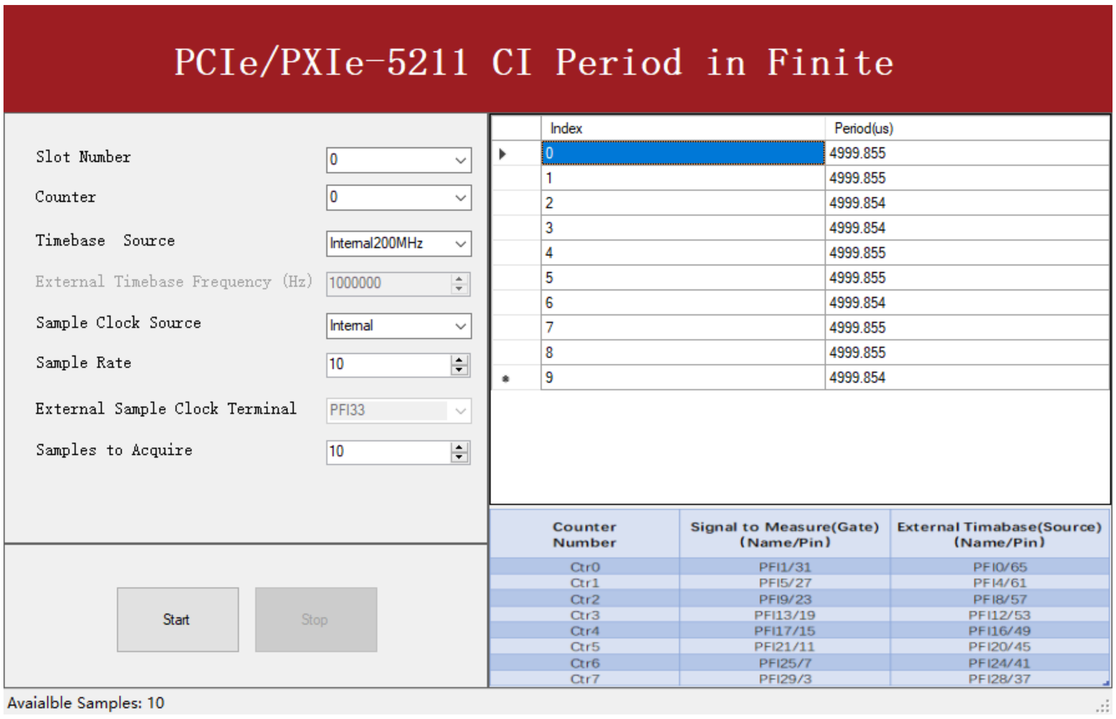
Counter Number	Signal to Measure(Gate) (Name/Pin)	External Timabase(Source) (Name/Pin)
Ctr0	PFI1/31	PFI0/65
Ctr1	PFI5/27	PFI4/61
Ctr2	PFI9/23	PFI8/57
Ctr3	PFI13/19	PFI12/53
Ctr4	PFI17/15	PFI16/49
Ctr5	PFI21/11	PFI20/45
Ctr6	PFI25/7	PFI24/41
Ctr7	PFI29/3	PFI28/37

Figure 25 Period Measure In Single Mode

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

Finite/Continuous Mode

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



PCIe/PXIe-5211 CI Period in Finite

Slot Number	0
Counter	0
Timebase Source	Internal200MHz
External Timebase Frequency (Hz)	1000000
Sample Clock Source	Internal
Sample Rate	10
External Sample Clock Terminal	PFI33
Samples to Acquire	10

Index	Period(us)
0	4999.855
1	4999.855
2	4999.854
3	4999.854
4	4999.855
5	4999.855
6	4999.854
7	4999.855
8	4999.855
9	4999.854

Counter Number	Signal to Measure(Gate) (Name/Pin)	External Timabase(Source) (Name/Pin)
Ctr0	PFI1/31	PFI0/65
Ctr1	PFI5/27	PFI4/61
Ctr2	PFI9/23	PFI8/57
Ctr3	PFI13/19	PFI12/53
Ctr4	PFI17/15	PFI16/49
Ctr5	PFI21/11	PFI20/45
Ctr6	PFI25/7	PFI24/41
Ctr7	PFI29/3	PFI28/37

Avaialble Samples: 10

Figure 26 Period Measure In Finite Mode

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

4.2.5 Two-Edge Separation

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

Set JY5211CITask.Type to CIType.TwoEdgeSeparation to use this function.

Timing

1) Single Mode

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

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

As shown in Figure 27.

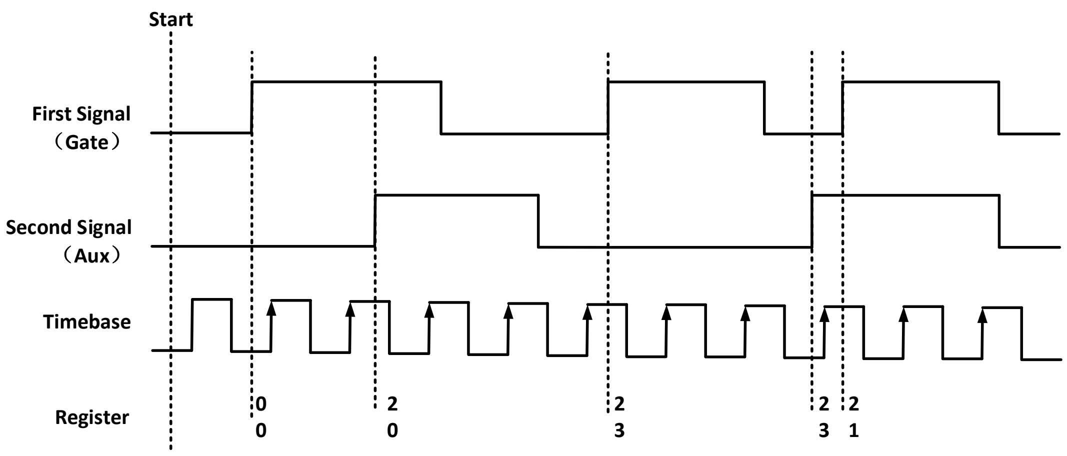


Figure 27 Two-Edge Separation in Single Mode

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

2) Finite/Continuous Mode with Explicit Sample Clock:

The counting 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 28.

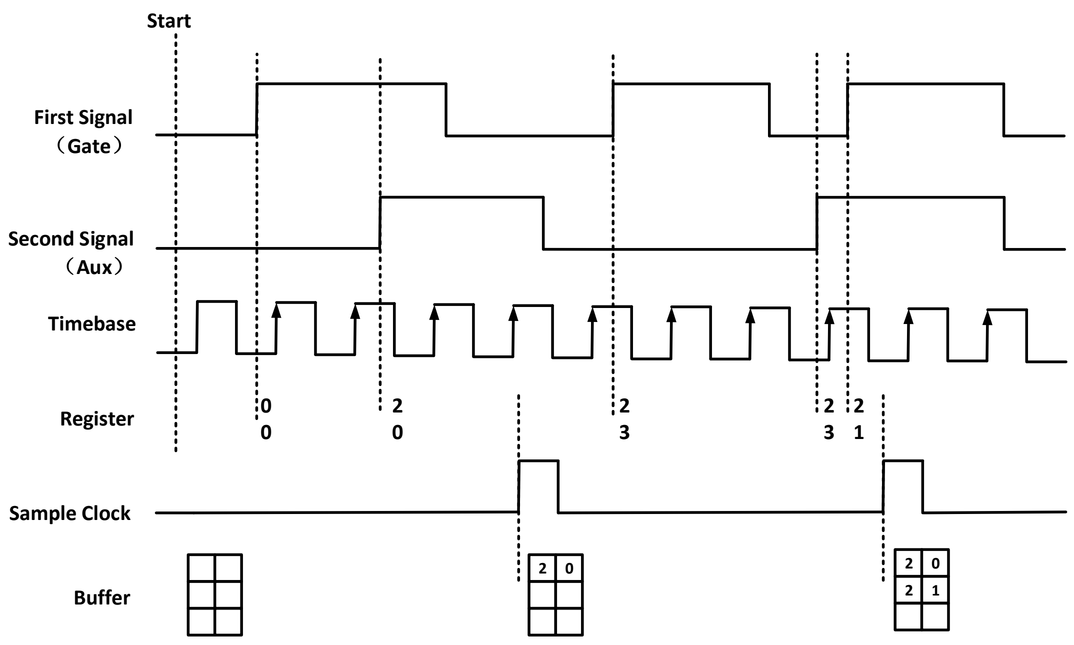


Figure 28 Two-Edge Seperation with Explicit Sample Clock

To configure the counter to work in this mode, set JY5211CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5211CITask.SampleClock.Source to CISampleClockSource.Internal or CISampleClockSource.External.

3) Finite/Continuous Mode with Implicit Sample Clock

In implicit mode, the signal active edge as the implicit sample clock edge. The counting values of rising edges of timebase between first signal and second signal are stored into buffer on each rising edge of the first signal, as shown in Figure 29.

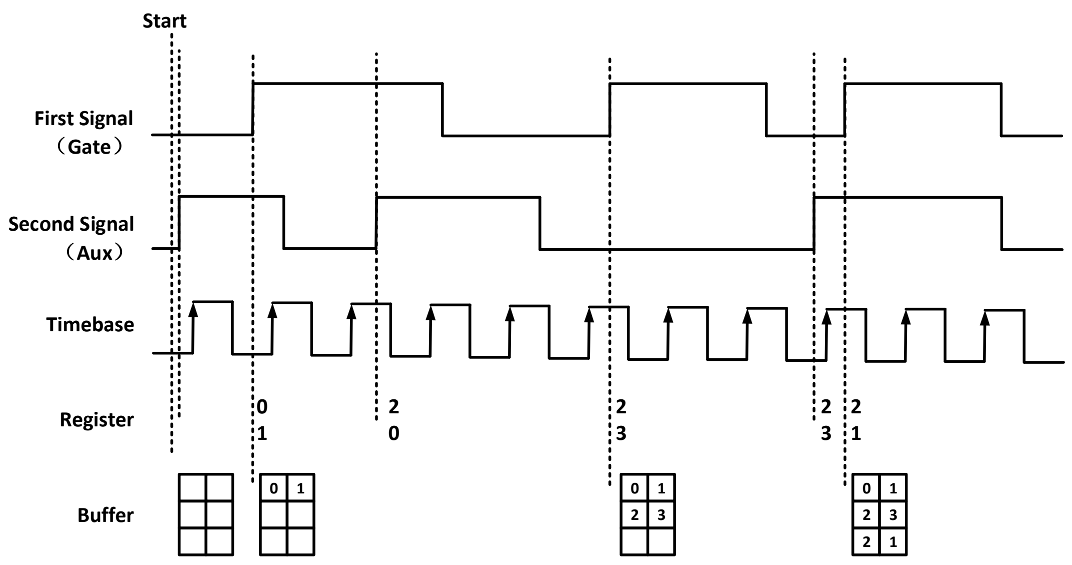


Figure 29 Two-Edge Seperation with Implicit Sample Clock

To configure the counter to work in this mode, set JY5211CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5211CITask.SampleClock.Source to CISampleClockSource.Implicit.

Timebase

By default, the counter uses the onboard 200MHz timebase to measure pulses. Use the property JY5211CITask.TwoEdgeSeparation.Timebase to configure the timebase.

Please refer to chapter 4.4.3 for more information about timebase.

Terminals

To change the terminal of signals instead of using its default value shown in chapter 2.3, using following properties:

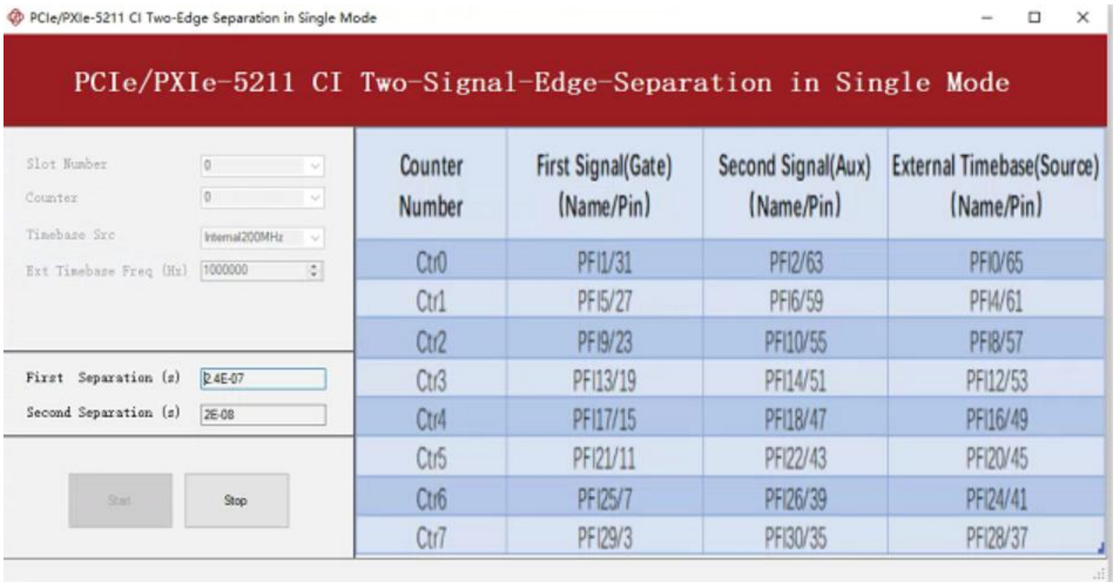
- JY5211CITask.TwoEdgeSeparation.FirstInputTerminal – First signal-to-measure input terminal.
- JY5211CITask.TwoEdgeSeparation.SecondInputTerminal – First signal-to-measure input terminal.
- JY5211CITask.TwoEdgeSeparation.Timebase.External.Terminal – External timebase input terminal.

Learn by Examples 4.2.5

- Connect the signal source's two positive terminals Ch1, Ch2 to PCIe/PXle-5211 first signal input (Gate, Pin#31) and second signal input (AUX, Pin#63), two negative terminals to the ground (GND, Pin#64) and (GND, Pin#28).
- Set the signal source Ch1's to squarewave signal (f=20Hz, Phase=0°), Ch2's output to squarewave signal (f=20Hz, Phase=90°).

Single Mode

- Open **Counter Input-->Winform CI Single Two-Edge Separation** and click **Start**. The result is shown below by **First Separation** and **Second Separation**, which represent the number of rising edges on the timebase difference between the rising edges of the two signals:


PCIe/PXIe-5211 CI Two-Edge Separation in Single Mode

PCIe/PXIe-5211 CI Two-Signal-Edge-Separation in Single Mode

Slot Number	0
Counter	0
Timebase Src	Internal200MHz
Ext Timebase Freq (Hz)	1000000
First Separation (s)	2.4E-07
Second Separation (s)	2E-08

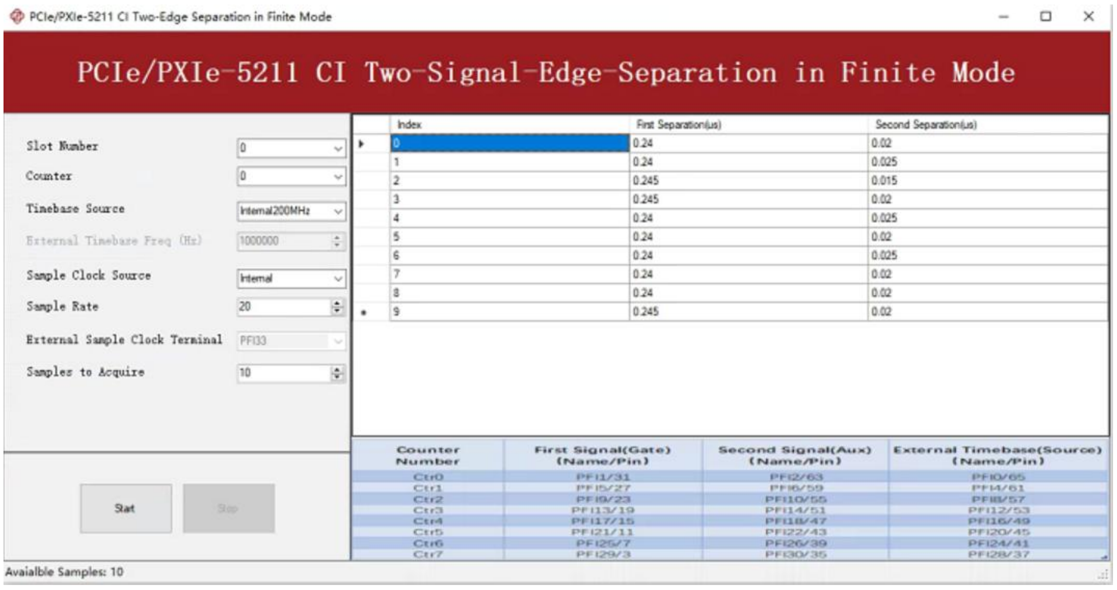
Counter Number	First Signal(Gate) (Name/Pin)	Second Signal(Aux) (Name/Pin)	External Timebase(Source) (Name/Pin)
Ctr0	PFI1/31	PFI2/63	PFI0/65
Ctr1	PFI5/27	PFI6/59	PFI4/61
Ctr2	PFI9/23	PFI10/55	PFI8/57
Ctr3	PFI13/19	PFI14/51	PFI12/53
Ctr4	PFI17/15	PFI18/47	PFI16/49
Ctr5	PFI21/11	PFI22/43	PFI20/45
Ctr6	PFI25/7	PFI26/39	PFI24/41
Ctr7	PFI29/3	PFI30/35	PFI28/37

Figure 30 Two-Edge Separation Measure In 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 Separation(s)** and **Second Separation(s)** are different.

Finite/Continuous Mode

- ■ Open **Counter Input-->Winform CI Finite/Continuous Two-Edge Separation** and click **Start.** The result is shown below by **First Separation(s)** and **Second Separation(s)**, which represent the number of rising edges on the timebase difference between the rising edges of the two signals:


PCIe/PXIe-5211 CI Two-Edge Separation in Finite Mode

PCIe/PXIe-5211 CI Two-Signal-Edge-Separation in Finite Mode

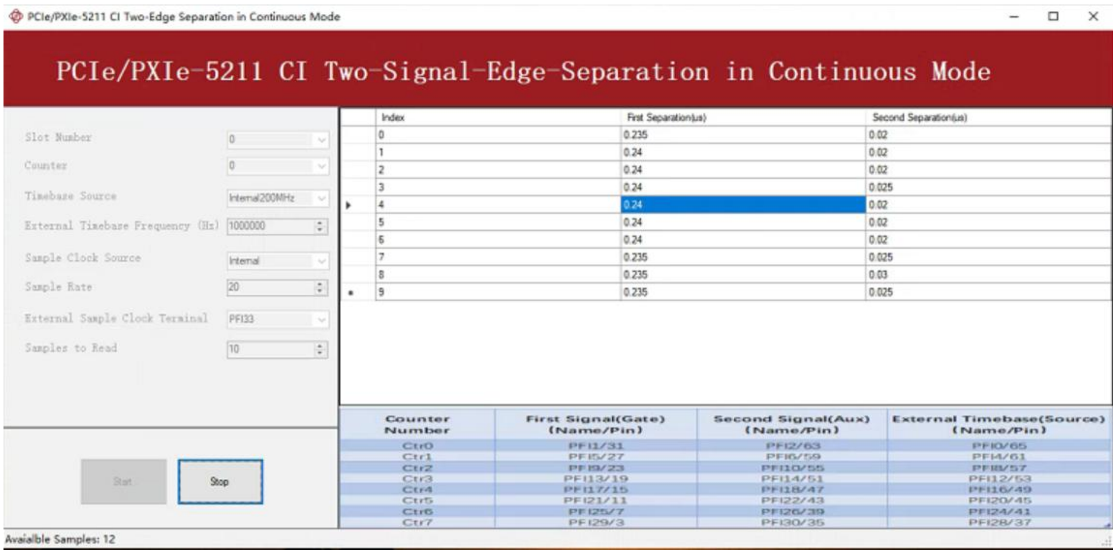
Slot Number	0
Counter	0
Timebase Source	Internal200MHz
External Timebase Freq (Hz)	1000000
Sample Clock Source	Internal
Sample Rate	20
External Sample Clock Terminal	PFI33
Samples to Acquire	10

Index	First Separation(us)	Second Separation(us)
0	0.24	0.02
1	0.24	0.025
2	0.245	0.015
3	0.245	0.02
4	0.24	0.025
5	0.24	0.02
6	0.24	0.025
7	0.24	0.02
8	0.24	0.02
9	0.245	0.02

Counter Number	First Signal(Gate) (Name/Pin)	Second Signal(Aux) (Name/Pin)	External Timebase(Source) (Name/Pin)
Ctr0	PFI1/31	PFI2/63	PFI0/65
Ctr1	PFI5/27	PFI6/59	PFI4/61
Ctr2	PFI9/23	PFI10/55	PFI8/57
Ctr3	PFI13/19	PFI14/51	PFI12/53
Ctr4	PFI17/15	PFI18/47	PFI16/49
Ctr5	PFI21/11	PFI22/43	PFI20/45
Ctr6	PFI25/7	PFI26/39	PFI24/41
Ctr7	PFI29/3	PFI30/35	PFI28/37

Avaialble Samples: 10

Figure 31 Two-Edge Separation Measure In Finite Mode


PCIe/PXIe-5211 CI Two-Edge Separation in Continuous Mode

PCIe/PXIe-5211 CI Two-Signal-Edge-Separation in Continuous Mode

Slot Number	0
Counter	0
Timebase Source	Internal200MHz
External Timebase Frequency (Hz)	1000000
Sample Clock Source	Internal
Sample Rate	20
External Sample Clock Terminal	PFI33
Samples to Read	10

Index	First Separation(us)	Second Separation(us)
0	0.235	0.02
1	0.24	0.02
2	0.24	0.02
3	0.24	0.025
4	0.24	0.02
5	0.24	0.02
6	0.24	0.02
7	0.235	0.025
8	0.235	0.03
9	0.235	0.025

Counter Number	First Signal(Gate) (Name/Pin)	Second Signal(Aux) (Name/Pin)	External Timebase(Source) (Name/Pin)
Ctr0	PFI1/31	PFI2/63	PFI0/65
Ctr1	PFI5/27	PFI6/59	PFI4/61
Ctr2	PFI9/23	PFI10/55	PFI8/57
Ctr3	PFI13/19	PFI14/51	PFI12/53
Ctr4	PFI17/15	PFI18/47	PFI16/49
Ctr5	PFI21/11	PFI22/43	PFI20/45
Ctr6	PFI25/7	PFI26/39	PFI24/41
Ctr7	PFI29/3	PFI30/35	PFI28/37

Avaialble Samples: 12

Figure 32 Two-Edge Separation Measure In Continuous Mode

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

4.2.6 Quadrature Encoder

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

Set JY5211CITask.Type to CIType. QuadEncoder to use this function, and use property JY5211CITask.QuadEncoder.EncodingType to change the type of encoding. Default, the A signal must be connected to Counter A terminal, the B signal mest be connected to Counter B terminal and the Z signal mest be connected to Counter Z terminal. For terminal Z, you can connect Z signal to it, or you can disable Z with property "ZReloadEnabled".

Encoding Type

1) x1 Encoding

When A signal leads B signal, the counter increases the count value on the rising edge of A signal; when B signal leads A signal, the counter decreases the count value on the falling edge of A signal as shown in Figure 33.

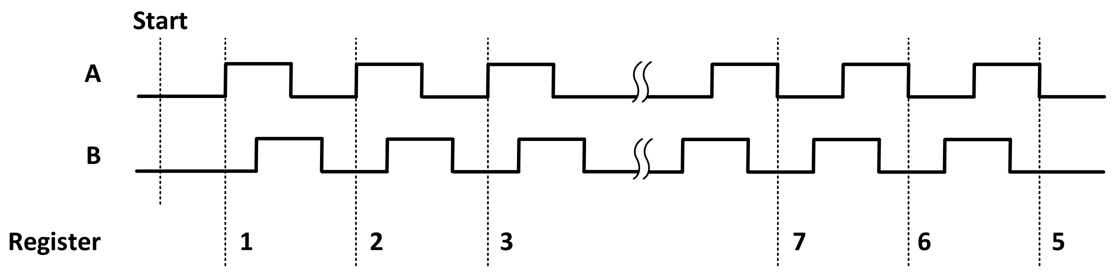


Figure 33 Quadrature Endcoder x1 Mode

2) x2 Encoding

When A signal leads B signal, the counter increases the count on the rising edge and the falling edge of A signal; when B signal leads A signal, the counter decreases the count value the rising edge and falling edge of A signal as shown in Figure 34.

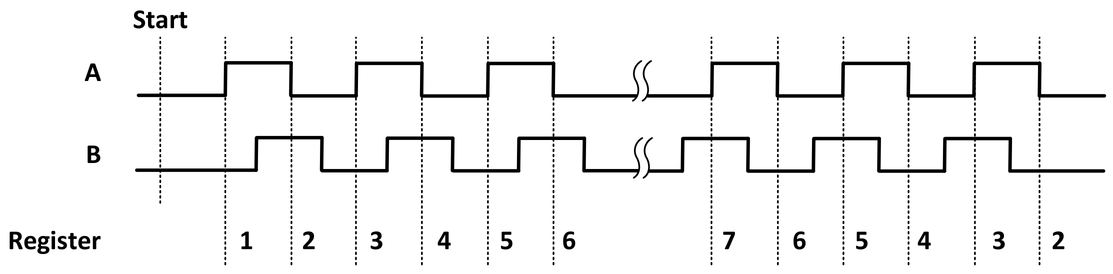


Figure 34 Quadrature Encoder x2 Mode

3) x4 Encoding

When A signal leads B signal, the counter increases the count value on the rising and falling edges of A signal and B signal. When B signal leads A signal, the counter decreases the count value on the rising and falling edges of A signal and B signal. As shown in Figure 35.

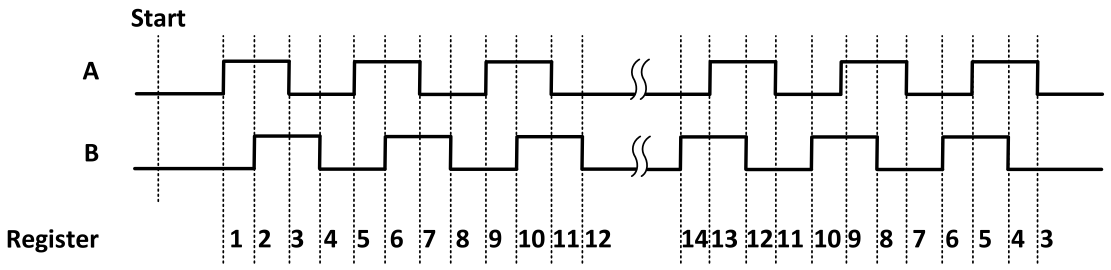


Figure 35 Quadrature Encoder x4 mode

Channel Z Behavior

The reload phase is when Z signal is high and A signal and B signal are low.

Timing

Take Encoding x1 mode as an example.

1) Single Mode

The count value is written to the register on each rising edge of the A signal, as shown in Figure 33.

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

2) Finite/Continuous Mode with Explicit Sample Clock

The count value is stored into the buffer on each rising edge of the sample clock, as shown in Figure 36.

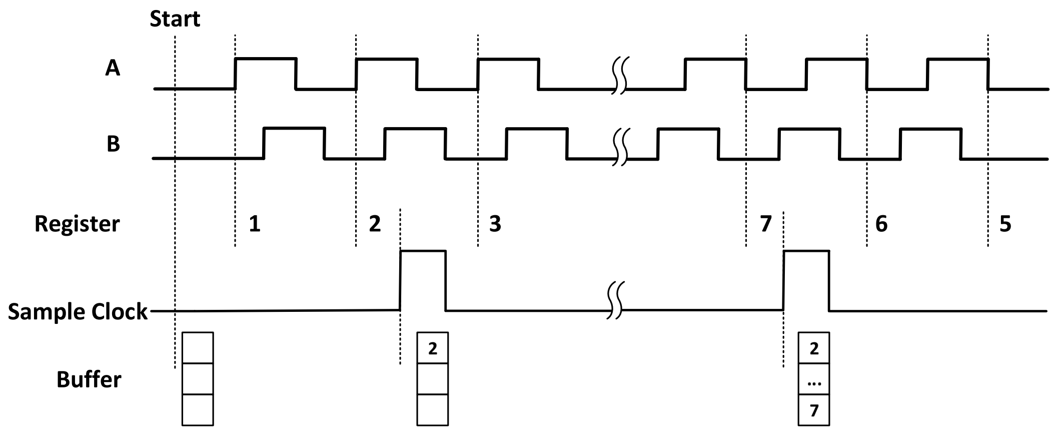


Figure 36 Quadrature Encoder x4 with Explicit Sample Clock

To configure the counter to work in this mode, set JY5211CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5211CITask.SampleClock.Source to CISampleClockSource.Internal or CISampleClockSource.External.

3) Finite/Continuous Mode with Implicit Sample Clock

In implicit mode, the signal active edge as the implicit sample clock edge. The count value is stored into the buffer on the transition of signal as shown in Figure 37.

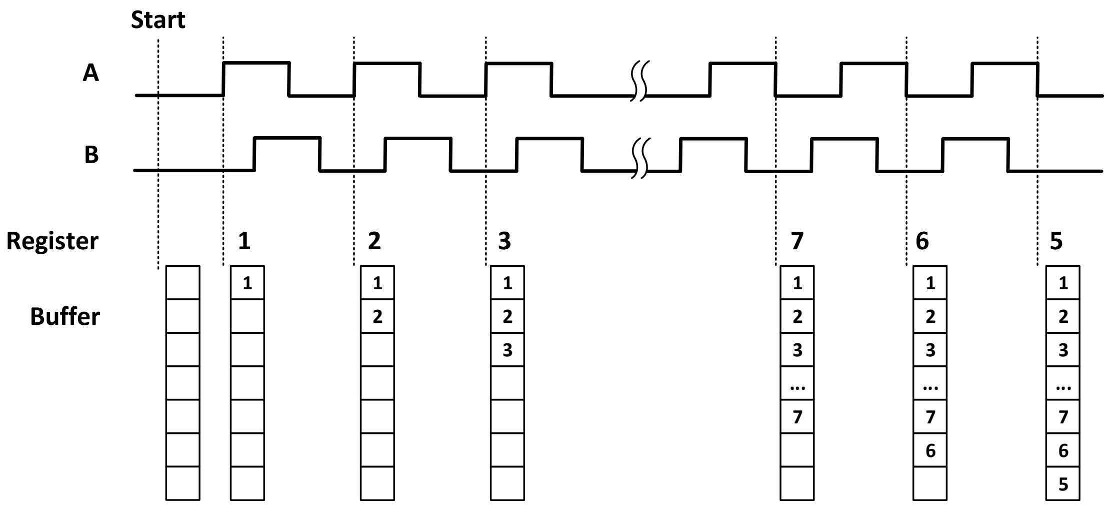


Figure 37 Quadrature Encoder x4 with Implicit Sample Clock

To configure the counter to work in this mode, set JY5211CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5211CITask.SampleClock.Source to CISampleClockSource.Implicit.

Terminals

To change the terminal of signals instead of using its default value as shown in chapter 2.3, use following properties:

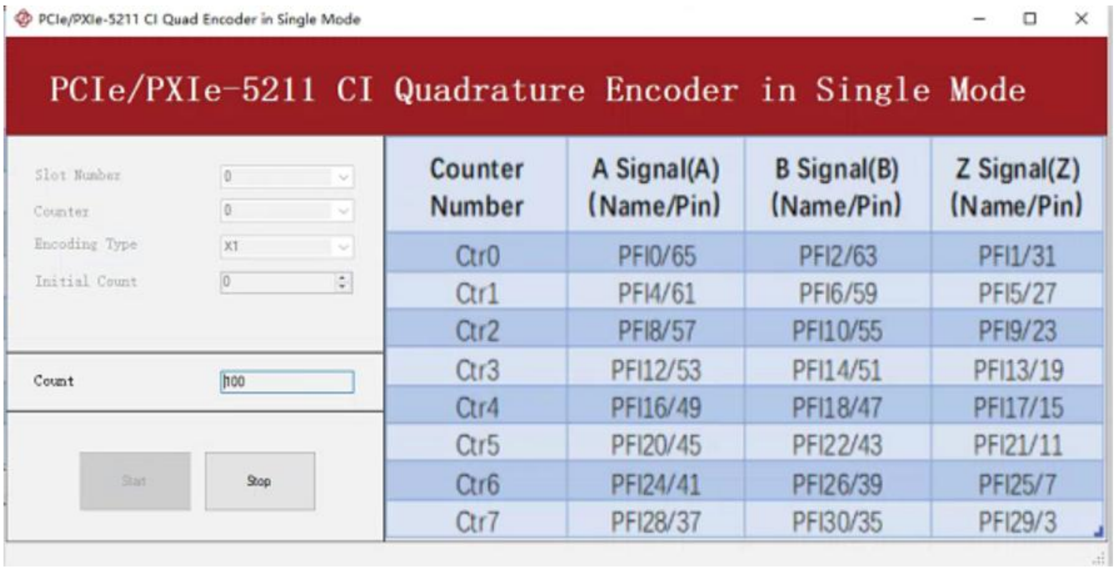
- JY5211CITask.QuadEncoder.AInputTerminal – Signal A input terminal.
- JY5211CITask.QuadEncoder.ZInputTerminal – Signal Z input terminal.
- JY5211CITask.QuadEncoder.BInputTerminal – Signal B input terminal.

Learn by Examples 4.2.6

- Connect the signal source's two positive terminals Ch1, Ch2 to PCIe/PXle-5211 first signal input (A, Pin#65) and second signal input (B, Pin#63), two negative terminals to the ground (GND, Pin#32) and (GND, Pin#28).
- Set the signal source Ch1's to squarewave signal (f=5Hz, Phase=0˚) and Ch2's output to squarewave signal (f=5Hz, Phase=90˚).

Single Mode

- Open **Counter Input--> Winform CI Single Quadrature Encoder** and click **Start.** The result is shown below by **Count.**


PCIe/PXIe-5211 CI Quad Encoder in Single Mode
PCIe/PXIe-5211 CI Quadrature Encoder in Single Mode
Start
Stop

Slot Number	0
Counter	0
Encoding Type	X1
Initial Count	0
Count	100

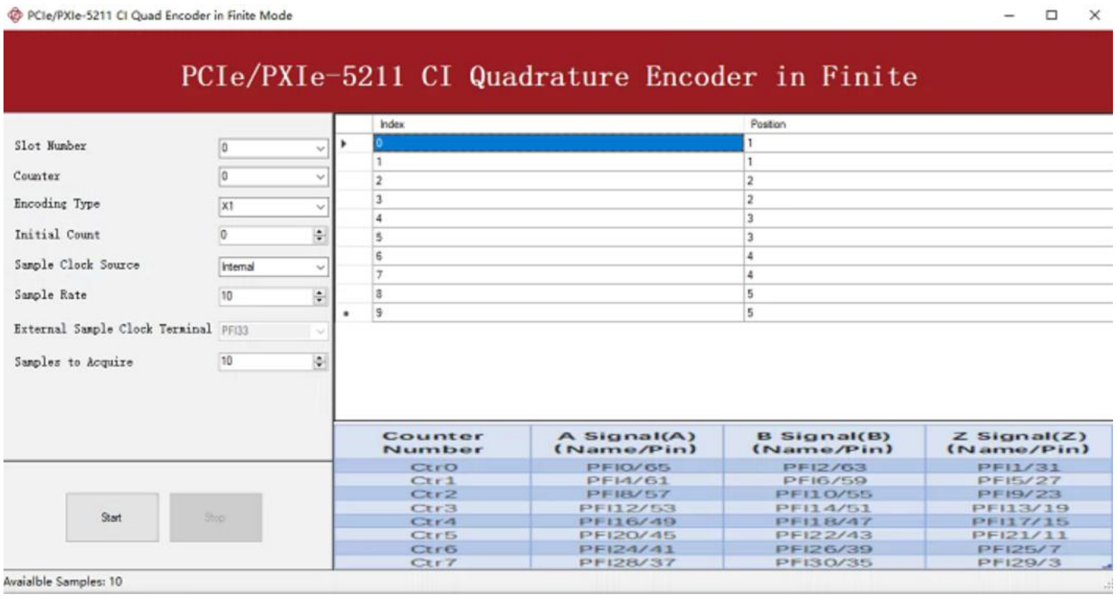
Counter Number	A Signal(A) (Name/Pin)	B Signal(B) (Name/Pin)	Z Signal(Z) (Name/Pin)
Ctr0	PFI0/65	PFI2/63	PFI1/31
Ctr1	PFI4/61	PFI6/59	PFI5/27
Ctr2	PFI8/57	PFI10/55	PFI9/23
Ctr3	PFI12/53	PFI14/51	PFI13/19
Ctr4	PFI16/49	PFI18/47	PFI17/15
Ctr5	PFI20/45	PFI22/43	PFI21/11
Ctr6	PFI24/41	PFI26/39	PFI25/7
Ctr7	PFI28/37	PFI30/35	PFI29/3

Figure 38 Quadrature Encoder In 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 *Encoding type* is changed from x1 to x2 and x4, you can see the rising speed of **Count** is twice and four times than x1Mode.

Finite/Continuous Mode

- Open **Counter Input--> Winform CI Single Quadrature Encoder** and click **Start**. The result is shown below by **Count**.



PCIe/PXIe-5211 CI Quad Encoder in Finite Mode

PCIe/PXIe-5211 CI Quadrature Encoder in Finite

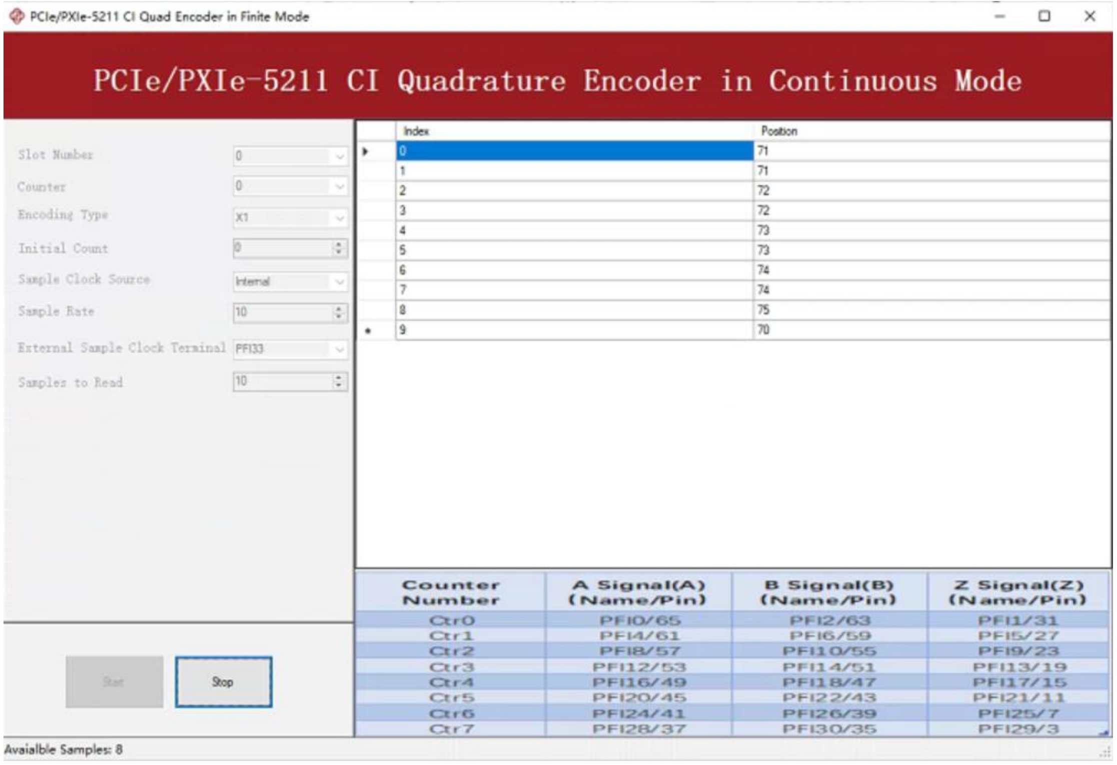
Slot Number	0
Counter	0
Encoding Type	X1
Initial Count	0
Sample Clock Source	Internal
Sample Rate	10
External Sample Clock Terminal	PFI33
Samples to Acquire	10

Index	Position
0	1
1	1
2	2
3	2
4	3
5	3
6	4
7	4
8	5
9	5

Counter Number	A Signal(A) (Name/Pin)	B Signal(B) (Name/Pin)	Z Signal(Z) (Name/Pin)
Ctr0	PFI0/65	PFI2/63	PFI1/31
Ctr1	PFI4/61	PFI6/59	PFI5/27
Ctr2	PFI8/57	PFI10/55	PFI9/23
Ctr3	PFI12/53	PFI14/51	PFI13/19
Ctr4	PFI16/49	PFI18/47	PFI17/15
Ctr5	PFI20/45	PFI22/43	PFI21/11
Ctr6	PFI24/41	PFI26/39	PFI25/7
Ctr7	PFI28/37	PFI30/35	PFI29/3

Avaialble Samples: 10

Figure 39 Quadrature Encoder In Finite Mode



PCIe/PXIe-5211 CI Quad Encoder in Finite Mode

PCIe/PXIe-5211 CI Quadrature Encoder in Continuous Mode

Slot Number	0
Counter	0
Encoding Type	X1
Initial Count	0
Sample Clock Source	Internal
Sample Rate	10
External Sample Clock Terminal	PFI33
Samples to Read	10

Index	Position
0	71
1	71
2	72
3	72
4	73
5	73
6	74
7	74
8	75
9	70

Counter Number	A Signal(A) (Name/Pin)	B Signal(B) (Name/Pin)	Z Signal(Z) (Name/Pin)
Ctr0	PFI0/65	PFI2/63	PFI1/31
Ctr1	PFI4/61	PFI6/59	PFI5/27
Ctr2	PFI8/57	PFI10/55	PFI9/23
Ctr3	PFI12/53	PFI14/51	PFI13/19
Ctr4	PFI16/49	PFI18/47	PFI17/15
Ctr5	PFI20/45	PFI22/43	PFI21/11
Ctr6	PFI24/41	PFI26/39	PFI25/7
Ctr7	PFI28/37	PFI30/35	PFI29/3

Avaialble Samples: 8

Figure 40 Quadrature Encoder In 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 **Count** is twice and four times than x1Mode.

4.2.7 Two-Pulse Encoder

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

Set JY5211CITask.Type to CIType.TwoPulseEncoder to use this function

Timing

1) Single Mode

The count value is written to the register on each rising edge of the A signal, and B signel, as shown in Figure 41.

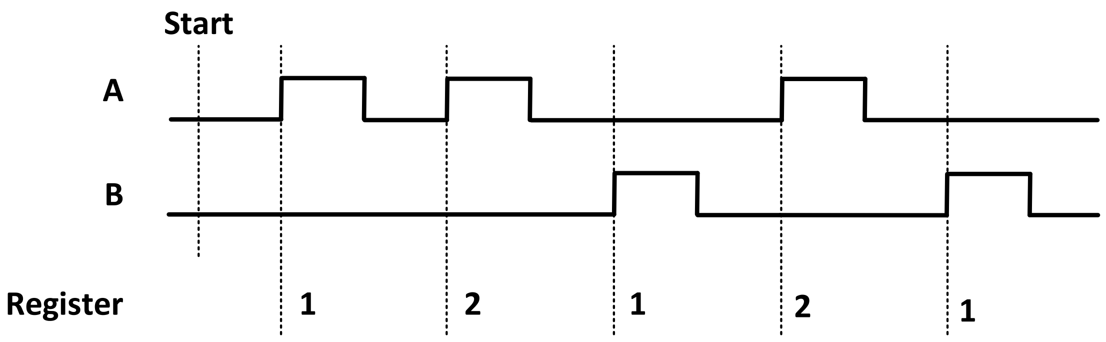


Figure 41 Two-Pulse Encoder in Single Mode

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

2) Finite/Continuous Mode with Explicit Sample Clock

The count value is stored into the buffer on each rising edge of the sample clock, as shown in Figure 42.

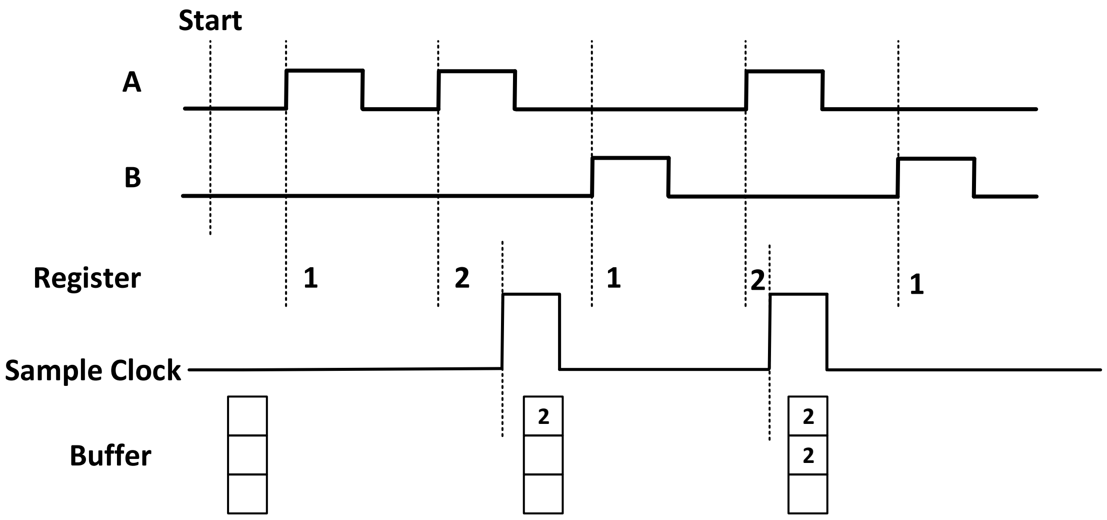


Figure 42 Two-Pulse Encoder with Explicit Sample Clock

To configure the counter to work in this mode, set JY5211CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5211CITask.SampleClock.Source to CISampleClockSource.Internal or CISampleClockSource.External.

3) Finite/Continuous Mode with Implicit Sample Clock

In implicit mode, the signal active edge as the implicit sample clock edge. The count value is stored into the buffer on the rising edge of A signal and B signal, as shown in Figure 43.

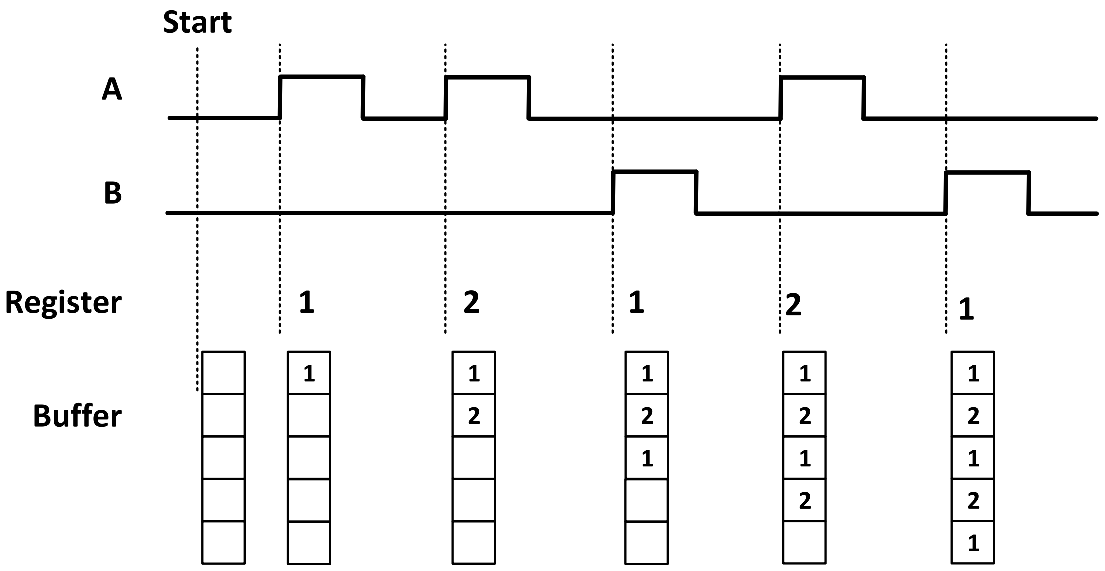


Figure 43 Two-Pulse Encoder with Implicit Sample Clock

To configure the counter to work in this mode, set JY5211CITask.Mode to CIMode.Finite or CIMode.Continuous, and set JY5211CITask.SampleClock.Source to CISampleClockSource.Implicit.

Terminals

To change the terminal of signals instead of using its default value as shown in chapter 2.3, use following properties:

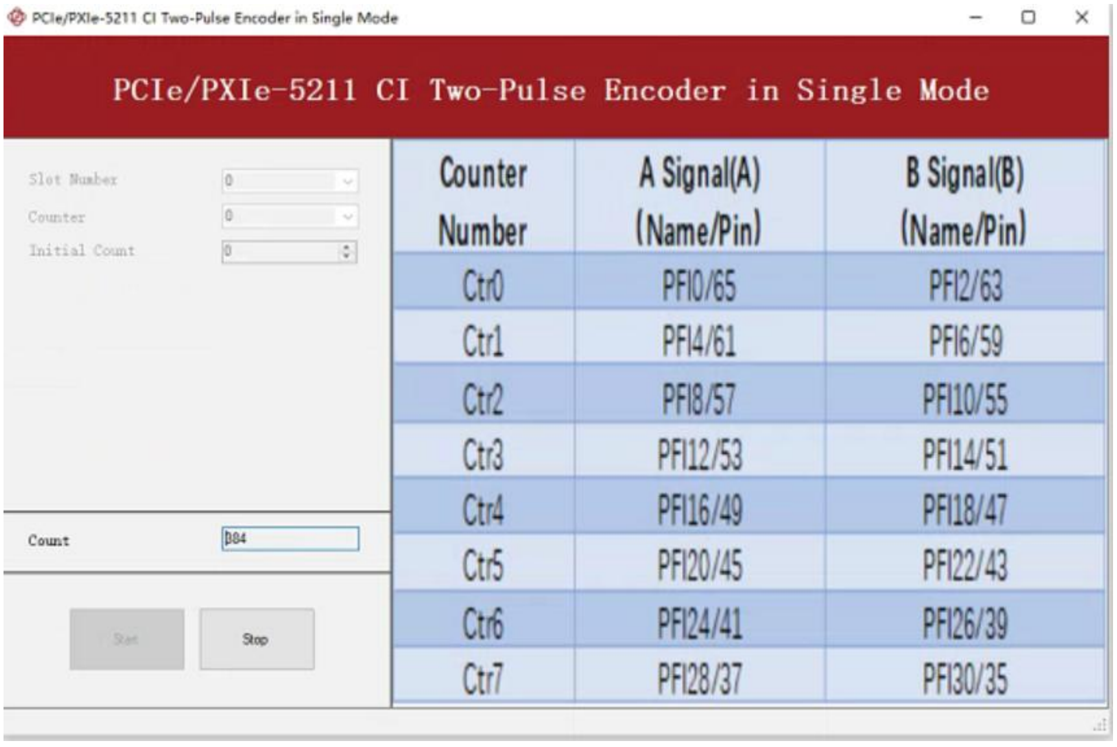
- JY5211CITask.TwoPulseEncoder.AInputTerminal – Signal A input terminal.
- JY5211CITask. TwoPulseEncoder.BInputTerminal – Signal B input terminal.

Learn by Examples 4.2.7

- Connect the signal source's two positive terminals Ch1, Ch2 to PCIe/PXIe-5211 first signal input (A, Pin#65) and second signal input (B, Pin#63), two negative terminals to the ground (GND, Pin#32) and (GND, Pin#28).
- Set the signal source Ch1's to squarewave signal (f=10Hz, Phase=0˚) and Ch2's output to squarewave signal (f=5Hz, Phase=90˚).

Single Mode

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


PCIe/PXIe-5211 CI Two-Pulse Encoder in Single Mode
PCIe/PXIe-5211 CI Two-Pulse Encoder in Single Mode
Start
Stop

Slot Number	0
Counter	0
Initial Count	0
Count	384

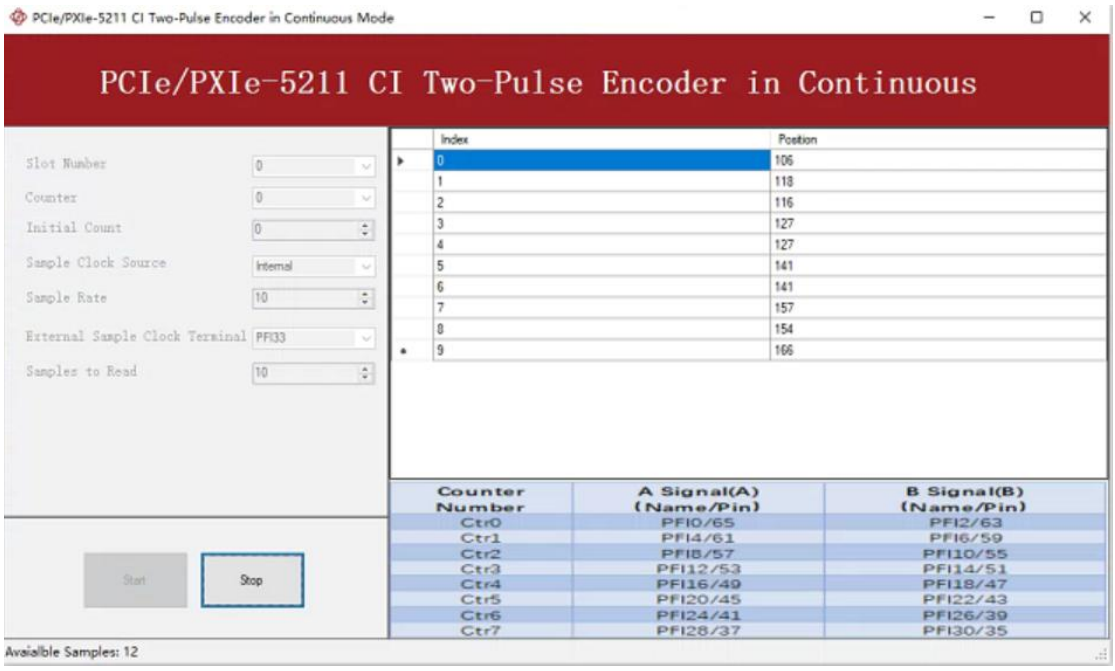
Counter Number	A Signal(A) (Name/Pin)	B Signal(B) (Name/Pin)
Ctr0	PFI0/65	PFI2/63
Ctr1	PFI4/61	PFI6/59
Ctr2	PFI8/57	PFI10/55
Ctr3	PFI12/53	PFI14/51
Ctr4	PFI16/49	PFI18/47
Ctr5	PFI20/45	PFI22/43
Ctr6	PFI24/41	PFI26/39
Ctr7	PFI28/37	PFI30/35

Figure 44 Two-Pulse Encoder In 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 **Count**.

Finite/Continuous Mode

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


PCIe/PXIe-5211 CI Two-Pulse Encoder in Continuous Mode
PCIe/PXIe-5211 CI Two-Pulse Encoder in Continuous
Start
Stop

Slot Number	0
Counter	0
Initial Count	0
Sample Clock Source	Internal
Sample Rate	10
External Sample Clock Terminal	PFI33
Samples to Read	10

Index	Position
0	106
1	118
2	116
3	127
4	127
5	141
6	141
7	157
8	154
9	166

Counter Number	A Signal(A) (Name/Pin)	B Signal(B) (Name/Pin)
Ctr0	PFI0/65	PFI2/63
Ctr1	PFI4/61	PFI6/59
Ctr2	PFI8/57	PFI10/55
Ctr3	PFI12/53	PFI14/51
Ctr4	PFI16/49	PFI18/47
Ctr5	PFI20/45	PFI22/43
Ctr6	PFI24/41	PFI26/39
Ctr7	PFI28/37	PFI30/35

Avaialble Samples: 12

Figure 45 Two-Pulse Encoder In Continuous Mode

- The table in the sample program is a connection diagram for your convenience.
- Click **Start** and you can see a group of rising numbers in position**,** which follows the counting rules explained in4.2.7.

4.3 Counter Generation Operations

The PCIe/PXIe-5211 can generate multiple forms of output signals according to different timing modes, including:

- Pulse generation with dynamic update
- Buffered pulse sequence generation

Timing

1) Single Mode

5211 can output a single pulse with a specified pulse configuration. The timing diagram of the pulse output is shown in Figure 46.

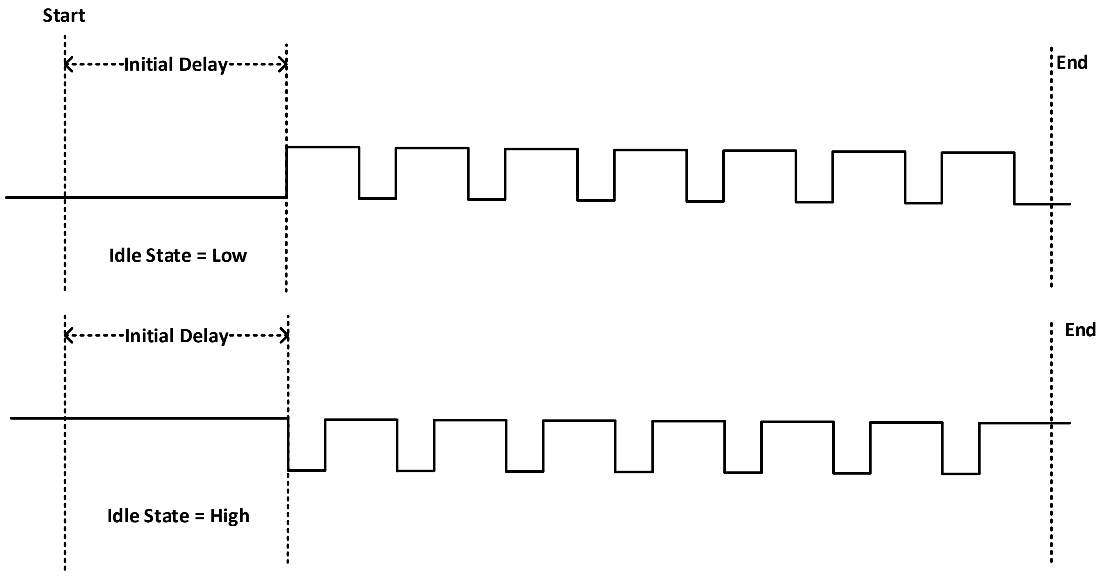


Figure 46 Pulse Output in Single Mode

Dynamic Update

If the number of pulses is set to -1, the pulses will be output continuously until requesting to stop. In this case, it is allowed to change the frequency and duty cycle on the fly. The timing diagram is shown as Figure 47.

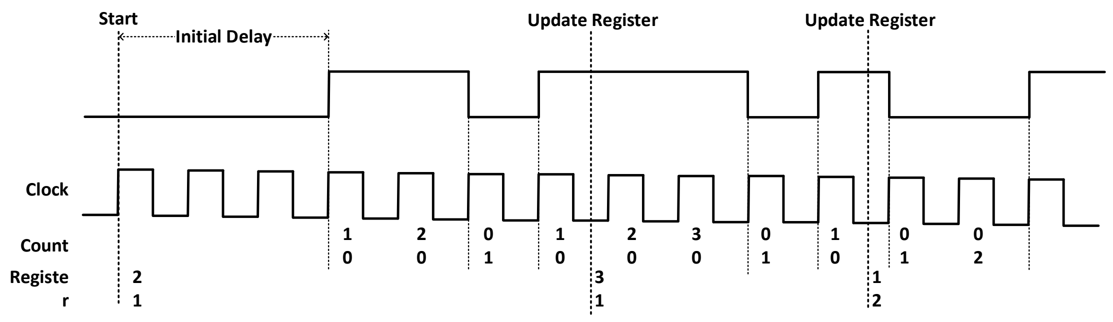


Figure 47 Pulse Output in Single Mode with Dynamic Update

2) Finite/Continuous Mode

This mode allows the user to write all configurations of pulses to be output to the buffer in advance. After send the currently configured pulses, the counter will automatically read the configuration of the next set of pulses to be sent from the buffer and start output. The timing diagram is shown in Figure 48.

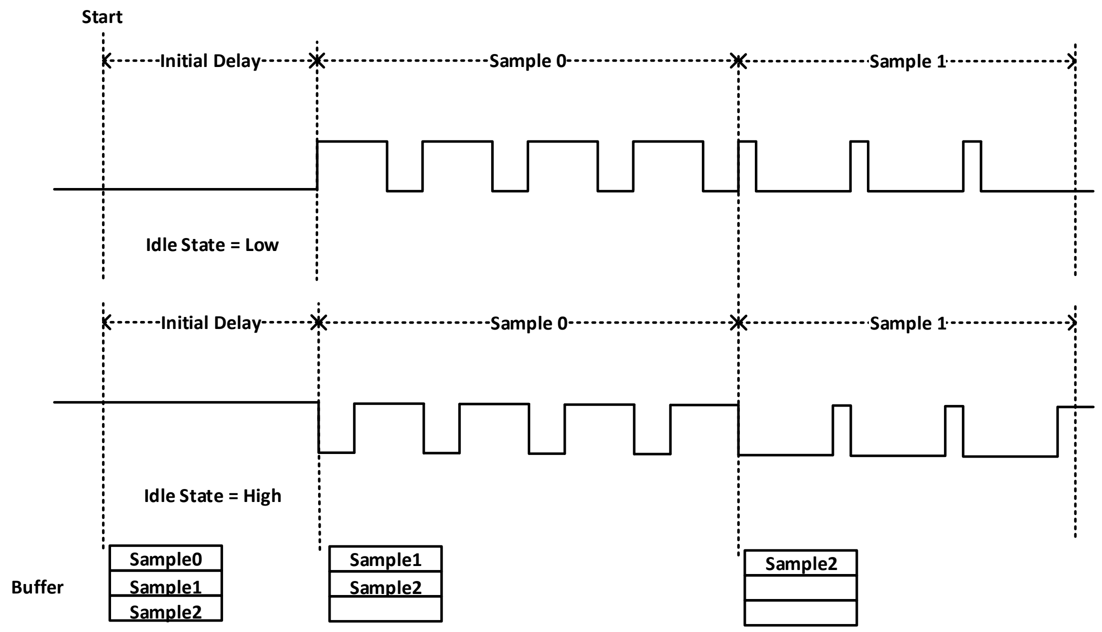


Figure 48 Buffered Pulse Sequence Generation

Each set of pulses can be configured in different ways:

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

Timebase

By default, the counter uses the onboard 200MHz timebase to generate pulses. Use the property JY5211COTask.Timebase to configure the timebase.

Please refer to chapter 4.4.3 for more information about timebase.

Terminals

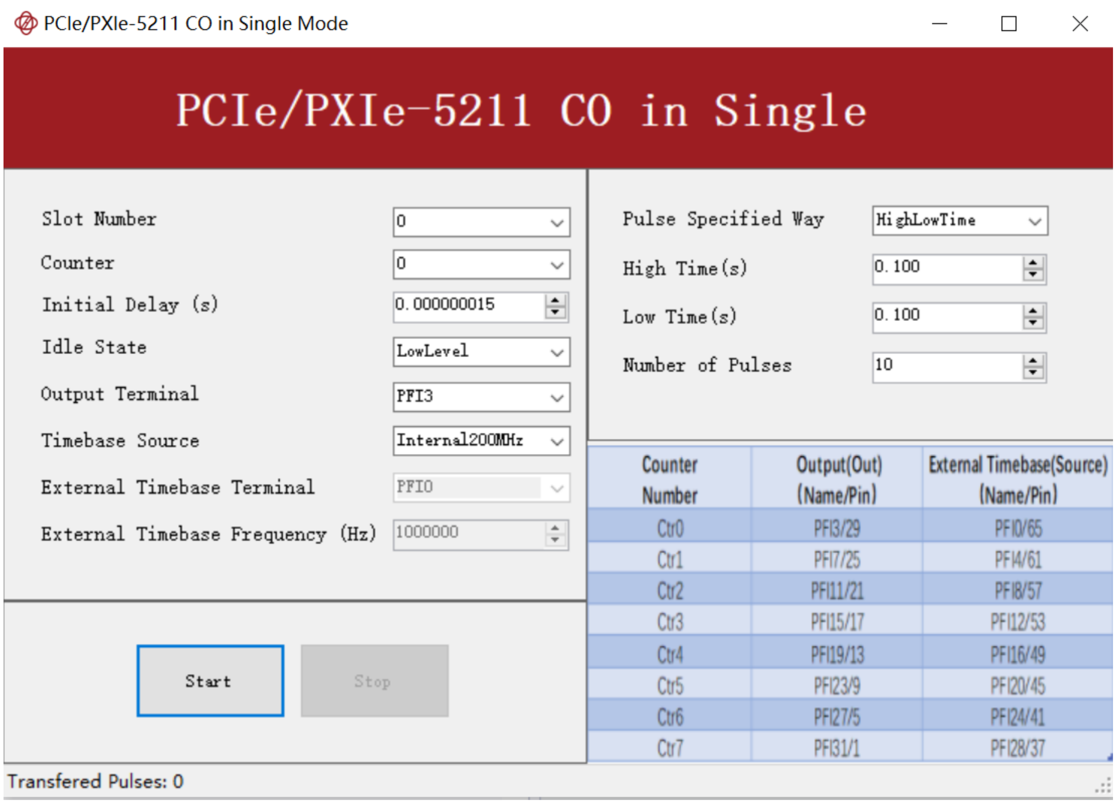
To change the terminal of signals instead of using its default value as shown in chapter 2.3, using following properties:

- JY5211COTask.OutputTerminal – Signal output terminal.

Learn by Examples

Single Mode

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



Slot Number	0
Counter	0
Initial Delay (s)	0.000000015
Idle State	LowLevel
Output Terminal	PFI3
Timebase Source	Internal200MHz
External Timebase Terminal	PFI0
External Timebase Frequency (Hz)	1000000
Pulse Specified Way	HighLowTime
High Time(s)	0.100
Low Time(s)	0.100
Number of Pulses	10

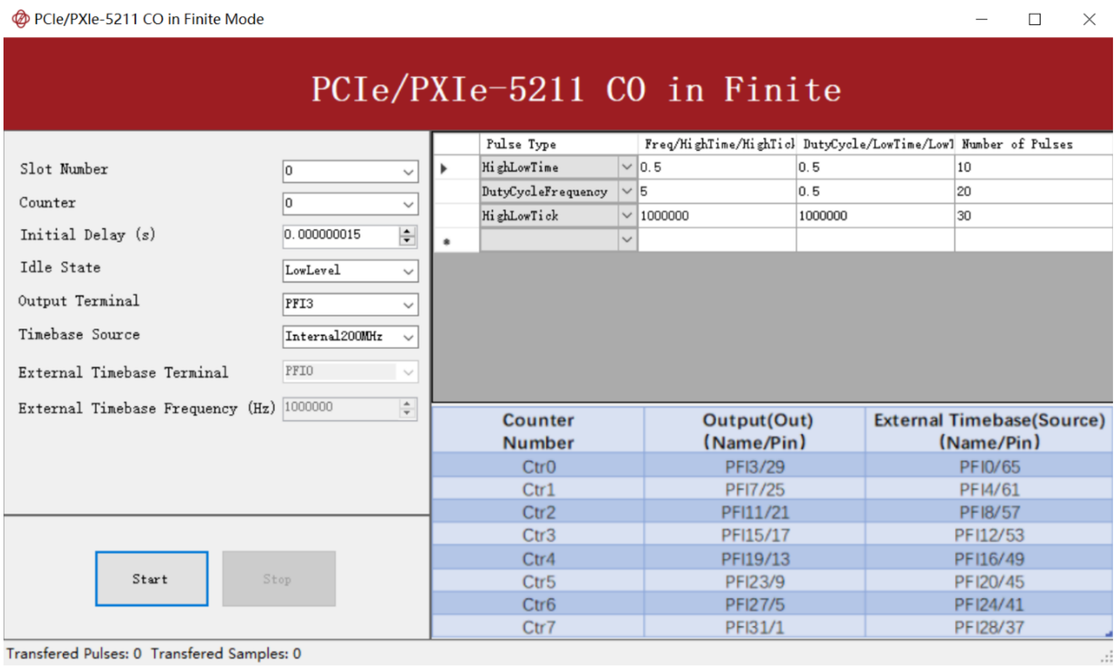
Counter Number	Output(Out) (Name/Pin)	External Timebase(Source) (Name/Pin)
Ctr0	PFI3/29	PFI0/65
Ctr1	PFI7/25	PFI4/61
Ctr2	PFI11/21	PFI8/57
Ctr3	PFI15/17	PFI12/53
Ctr4	PFI19/13	PFI16/49
Ctr5	PFI23/9	PFI20/45
Ctr6	PFI27/5	PFI24/41
Ctr7	PFI31/1	PFI28/37

Transfered Pulses: 0

Figure 49 CO In Single Mode

Finite Mode

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



PCIe/PXIe-5211 CO in Finite Mode

PCIe/PXIe-5211 CO in Finite

Slot Number	0
Counter	0
Initial Delay (s)	0.000000015
Idle State	LowLevel
Output Terminal	PFI3
Timebase Source	Internal200MHz
External Timebase Terminal	PFI0
External Timebase Frequency (Hz)	1000000

	Pulse Type	Freq/HighTime/HighTicl	DutyCycle/LowTime/Lowl	Number of Pulses
▶	HighLowTime	0.5	0.5	10
	DutyCycleFrequency	5	0.5	20
	HighLowTick	1000000	1000000	30
*				

Counter Number	Output(Out) (Name/Pin)	External Timebase(Source) (Name/Pin)
Ctr0	PFI3/29	PFI0/65
Ctr1	PFI7/25	PFI4/61
Ctr2	PFI11/21	PFI8/57
Ctr3	PFI15/17	PFI12/53
Ctr4	PFI19/13	PFI16/49
Ctr5	PFI23/9	PFI20/45
Ctr6	PFI27/5	PFI24/41
Ctr7	PFI31/1	PFI28/37

Transfered Pulses: 0 Transfered Samples: 0

Figure 50 CO In Finite Mode

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

4.4 Clocks

Figure 51 shows the structure of counter clock system.

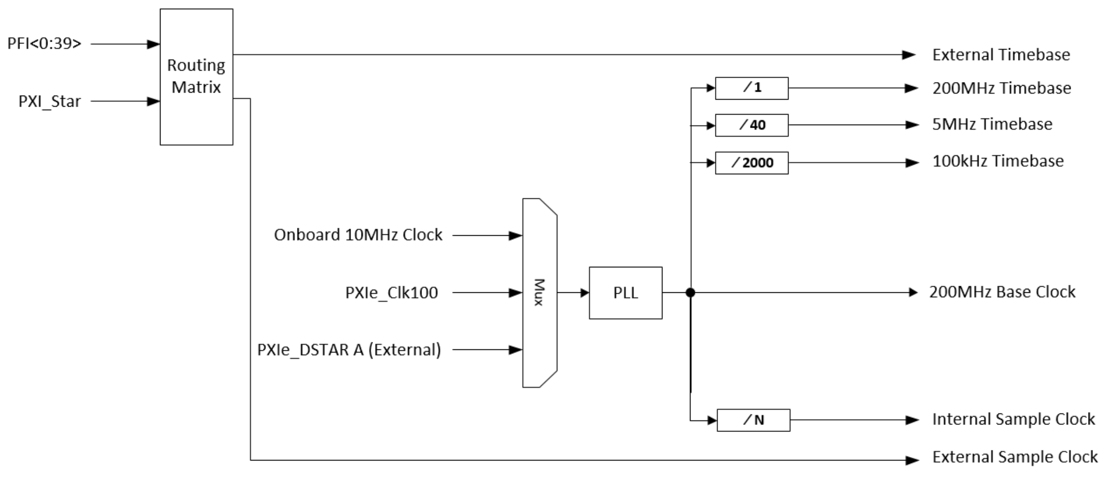


Figure 51 Clocks Diagram

4.4.1 PLL

PLL (Phase Locked Loop) is a phase-locked clock generator that can generate a clock signal of a specified frequency according to the selected reference clock source.

PCIe/PXIe–5211 Series boards support the following reference clock source:

1) Onboard 10MHz Clock

Using the on-board 10MHz (TXCO) as the PLL input source can help improve the PLL output clock performance, including improving clock accuracy, temperature stability, and phase noise.

2) PXIe_CLK100

The PXIe_CLK00 siganal is a 100MHz clock provided by the PXIe backplane for every peripheral slot. When using this clock, PXIe-5211 can provide multi-card synchronization.

3) External Reference Clock

An external reference clock is a clock provided by user through terminal PXIe-DSTARA. To use an external reference clock, the user needs to specify its frequency.

By defaut, onboard 10MHz Clock is selected as the reference clock source. To change the reference clock source, configure the device as follows:

Set property Task.Device.ReferenceClock.Source to target reference clock source. Task is user defined JY5211CITask or JY5211COTask.

If the target clock source is External:

- Set Task.Device.ReferenceClock.External.Terminal to dersired terminal.
- Set Task.Device.ReferenceClock.External.Frequency to the frequency of this clock source.

Call method Device.Commit() to activate the configuration.

Note:

1. Clock configuration are not allowed to change while any counter tasks are running.

Clock configuration is applied to all tasks (including DI, DO, CI, CO).

The PCIe/PXIe-5211 module must be powered off and restarted if the user submits the incorrect clock frequency by using external clock source (PXIe_DSTARA).

4.4.2 Sample Clock

For all counter measurement applications with buffered measurement, PCIe/PXIe-5211 provides 3 sample clock options as follows:

- Internal

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

To use the internal sample clock, configure as follows:

1. Set JY5211CITask.SampleClock.Source to CISampleClockSource.Internal
2. Set JY5211CITask.SampleClock.Internal.Rate to specify sample rate

- External

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

To use the external sample clock, configure as follows:

1. Set JY5211CITask.SampleClock.Source to CISampleClockSource.External
2. Set JY5211CITask.SampleClock.External.ExpectedRate to the rate of the external sample rate.

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

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

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

- Implicit

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

To use the implicit sample clock, configure as follows:

1. Set JY5211CITask.SampleClock.Source to CISampleClockSource.Implicit.
2. Set JY5211CITask.SampleClock.Implicit.ExpectedRate to desired sample rate. This property has the same effect as JY5211CITask.SampleClock.External.ExpectedRate.

4.4.3 Timebase

JY5211 provides four options for timebase source as follows:

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

4.5 Start Trigger

For all counter measurement and generation applications, the task starts running when the start trigger happens.

Start trigger has the following types:

Immediately

The task will start immediately after JY5211xxTask.Start() is called.

Software

After calling JY5211xxTask.Start () on the software, the task will not start until a software trigger is received.

Digital

An external digital trigger is generated when the external trigger source terminal detects a rising edge as shown in Figure 52.

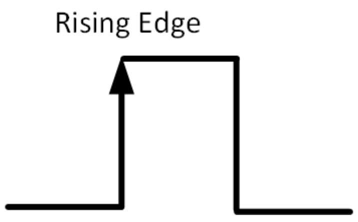


Figure 52 Rising Edge Digital Trigger

4.6 Logic Level

PCIe/PXIe-5211 module supports multiple logic levels as follows: 1.8 V, 2.5 V, 3.3 V and 5 V.

To change the logic level by the following properties:

1. Set property Device.LogicLevel to target logic level.

   Call method Device.Commit() to activate the configuration.

Note:

Logic Level configuration will be applied to all PFIs

4.7 Multi-Card Synchronization

PCIe/PXIe–5211 Series modules support master-slave synchronization mechanism to achieve multi-card synchronous acquisition.

Master-Slave Synchronization

It will use 3 signals, Reference clock, Sync Pulse and Triger to achieve data acqusistion simultaneously with multiple modules. First, the master module will notice all of slave modules by routing the trigger signal through PXI trigger bus, PXI_TRIG<0..7> when master module receives trigger. Second, we also need to make sure every module to start the acquisition task in the same time, therefore we could take advange of PXI system which can provide a synchronization pulse, PXIe_SYNC100 to coordinate with the acquisition task of multiple modules.

Third, every module must use the same reference clock to keep pace with each others and user can use PXIe_CLK100 which provides by PXI system as reference clock.

The timing diagram is shown as Figure 53.

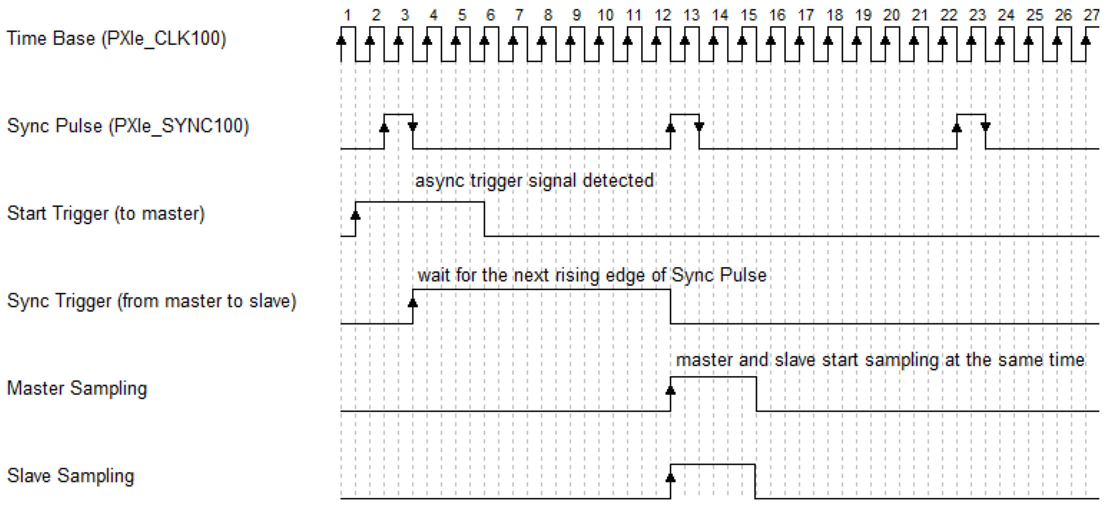

Time Base (PXIe_CLK100)
Sync Pulse (PXIe_SYNC100)
Start Trigger (to master)
Sync Trigger (from master to slave)
Master Sampling
Slave Sampling
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
async trigger signal detected
wait for the next rising edge of Sync Pulse
master and slave start sampling at the same time

Figure 53 Master-Slave Synchronization

To enable the multi-card synchronization, configure the board as follows:

1. Set each task on different card as Master or Slave. Only one master is allowed.
2. Set the reference clock source of master card and slave card to PXIe_CLK100. Refer to 4.4.1 for more information.
3. Route trigger signals of all tasks to the same signal terminal, trigger signal will be sent from the master card to all slave cards through this terminal.

4. Route Sync Pulse signals of all tasks to the same signal terminal, Sync Pulse signals will be sent from the master card to all slave cards through this terminal.
5. Start all slave tasks.
6. Start the master task (if digital trigger or software trigger is enabled, you need to wait for the trigger signal to arrive), all the tasks will start to work synchronously on a rising edge of the PXIe_SYNC100 signal.

4.8 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.

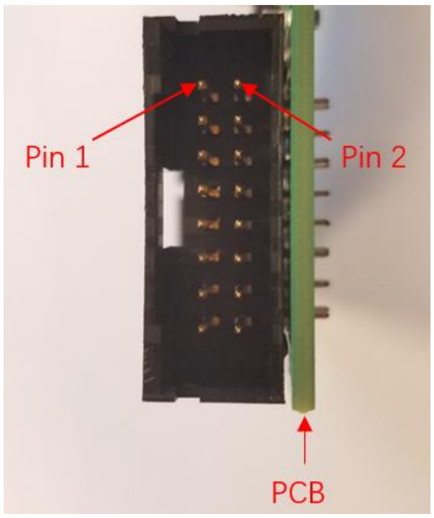


Figure 54 SSI Connector in PCIe-5211

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 7 SSI Connector Pin Assignment for PCIe-5211

4.9 DIP Switch in PCIe-5211

PCIe-5211 series modules have a DIP switch. The card number can be adjusted manually by turning the DIP switch, which is used to identify the module 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. Find the detail below.

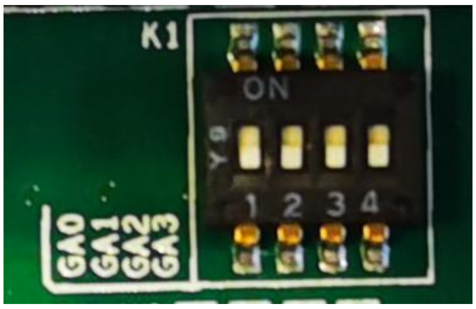


Figure 55 DIP Switch in PCIe-5211

	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 8 Relationship between switch position and slot number

5. Using PCIe/PXIe-5211 in Other Software

While JYTEK's defalt application platform is Visual Studio, the programming language is C#, we recognize there are other platforms that are either becoming very popular or have been widely used in the data acquisition applications. Among them are C++ etc. This chapter explains how you can use PCIe/PXIe-5211 DAQ card using one of this software.

5.1 C++

JYTEK internaly uses our C++drivers to design the C# drivers.   We recommend our customers to use C# drivers because C# platform deliver much better efficiency and performance in most situations. We also make our C++ drivers available. However, due to the limit of our resources, we do not actively support C++ drivers. If you want to be our partner to support C++ drivers, please contact us.

6. About JYTEK

6.1 JYTEK China

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

6.2 JYTEK Hardware Products

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

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

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

6.3 JYTEK Software Platform

JYTEK has developed a complete software platform, SeeSharp Platform, for the test and measurement applications. We leverage the open sources communities to provide the software tools. Our platform software is also open sourced and is free, thus lowering the cost of tests for our customers. We are the only domestic vendor to offer complete commercial software and hardware tools.

6.4 JYTEK Warranty and Support Services

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

7. 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 explaination for JYTEK PCIe/PXIe-5211 Series family of multi-function data acquisition boards. The manual is copyrighted by JYTEK.

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

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

Shanghai Jianyi Technology Co., Ltd.

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

Post Code: 201203

Tel: 021-5047 5899

Website: www.jytek.com