USB-1601

Specs and Manual

Specs and Manual Version: V1.0.1

Revision Date: September 1, 2025

1. USB-1601 Specifications

1.1 Overview

The USB-1601 is a versatile, bus-powered USB DAQ module offering 16 analog input channels, 2 analog output channels, and programmable digital I/O. With sampling rates up to 250 kS/s and update rates of 1 MS/s, it's equipped for rapid data acquisition. Its ±2.5V/±5V/±10V input ranges, removable screw-down terminals, lockable USB cable, and support for digital/software triggering make it ideal for diverse measurement and automation tasks.

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

1.2 Main Features

- 340 ppm accuracy
- 16 bits resolution
- High-speed USB 2.0
- Bus-powered USB DAQ module
- 3 voltage ranges: ±2.5V/±5V/±10V
- 16-CH voltage input (single-end), 8-CH voltage input(differential) and 2-CH voltage output
- Up to 250 kS/s sampling rate for analog input
- Up to 1 MS/s update rate for analog output
- Programmable 16-CH digital input/output
- 2 general 32-bit timer/counter
- Removable screw-down terminal module
- Lockable USB cable for secure connectivity
- Digital/Software Trigger

1.3 Hardware Specifications

1.3.1 Analog Input Specifications

Number of Channels	16(RSE)/8(DIFF)
Input modes	RSE/DIFF
ADC Resolution	16 bits
Sampling Rate	250 kS/s
Coupling	DC
Input Ranges	±10V, ±5V, ±2.5V
Max working voltage	±12 V
Input Impedance	1 MΩ
Crosstalk(at 10 kHz)	-77 dB
Overvoltage protection	±12 V
Input current during overvoltage protection	±10 mA maximum/AI pin
Bandwidth	15 kHz,30 kHz,70 kHz
FIFO	40K Samples

Table 1 Analog Input Specifications

1.3.2 AI Accuracy

JY-1601 Basic Accuracy = ±(% Reading+% Range)												
Nominal Range (V)	24 Hour Tcal ±1°C			90 Days Tcal±5°C			Temperature Coefficients(/°C)			24 Hr Full Scale Accuracy(V)	90 Days Full Scale Accuracy(V)	Full Scale Accuracy(%)
2.5	0.005	+	0.024	0.018	+	0.026	0.0025	+	0.0005	0.0007	0.0011	0.044
5	0.005	+	0.017	0.017	+	0.019	0.0024	+	0.0004	0.0011	0.0018	0.036
10	0.005	+	0.014	0.018	+	0.016	0.0026	+	0.0005	0.0019	0.0034	0.034
Notes: Accuracy valid to 99% of full range The 90-day data is estimated by multiplying the 24-hour data by 2.5.												

Table 2 AI Accuracy

1.3.3 Analog Output Specifications

Number of channels	2
Output type	RSE
Resolution	16 bits
Maximum update rate(1 ch)	1 MS/s
Maximum update rate(2 ch)	500 KS/s
Output range	±10 V
Output coupling	DC
Output impedance	0.2 mΩ
Output current drive	±50 mA
Overdrive protection	±12 V
Overdrive current	±65 mA
Power-on state	±3.3mV
Power-on/off glitch	±1.2V for 20ms
Data transfers	USB 2.0 Type-c
Slew rate	20 V/µs
Noise	23 µVrms

FIFO	40K Samples

Table 3 Analog Output Specifications

1.3.4 AO Accuracy

JY-1601 Basic AO Accuracy = ±(% Output+% Range)						
Nominal Range (V)	24 Hour Tcal ±1°C	90 Days Tcal±5°C	Temperature Coefficients(/°C)	24 Hr Full Scale Accuracy (V)	90 Days Full Scale Accuracy (V)	Full Scale Accuracy(%)
10	0.006 + 0.013	0.016 + 0.015	0.0011 + 0.0005	0.0018	0.003	0.031
Notes: Accuracy valid to 95% of full range The 90-day data is estimated by multiplying the 24-hour data by 2.5.						

Table 4 AO Accuracy

1.3.5 Digital IO Specifications

Number of channels(DIO)	16 (DIO <0..15>)
Ground reference	D GND
Directional control	Independent control of each line
Initial state	Low-level input
Pull-down resistor	10 kΩ ±1%
Logic Signal Level	3.3 V

Table 5 Digital IO Specifications

1.3.6 Counter/Timer Specifications

Number of channels	2
Resolution	32
CI	EdgeCounting, FrequencyMeasure, PulseMeasure, Two-Edge Separation, Encoder

CO	Pulse, pulse train with dynamic updates, frequency division
Clock(Internal)	100 MHz
Base clock accuracy	50 ppm
FIFO	1 sample
Input	Gate, Source, Aux
Output	OUT

Table 6 Counter/Timer Specifications

1.3.7 Bus Interface

USB	USB 2.0 Hi-Speed or full-speed

Table 7 Bus Interface

1.3.8 Power

+5 V terminal as input	Maximum inrush current	400 mA
	No load typical current	198 mA at 5.0 V
	Typical current	270 mA at 5.0 V

Table 8 Power Requirement

1.3.9 Physical Characteristics

Dimensions (includes connectors)	100 mm × 68 mm × 28 mm
Weight	190.7g (6.7 oz)

Table 9 Physical Characteristics

1.3.10 Environment

Operating Environment

Termperature Range	0 °C to 45 °C
Humidity	10% RH to 90% RH, noncondensing

Table 10 Operating Environment

Storage Environment

Termperature Range	-20 °C to 70 °C
Humidity	10% RH to 90% RH, noncondensing

Table 11 Storage Environment

2. Order Informations

- USB-1601 (PN: JY1672834-01)

  16-ch(RSE)/8-ch(DIFF) AI (16-Bit, 250 kS/s), 2-ch AO (16-Bit, 1 MS/s), 16 DIO Multifunction USB DAQ

Table of Contents

3. Introduction

3.1 Overview

The USB-1601 is a multifunctional, Bus-powered USB 2.0 data acquisition (DAQ) device that provides 16 channels for analog input, 2 channels for analog output, and flexible digital I/O programming. Capable of sampling at speeds up to 250,000 samples per second and updating at a rate of 1 million samples per second, this DAQ module is designed for swift data collection. It features input voltage ranges of ±2.5V, ±5V, and ±10V, detachable screw-down terminals for secure connections, a secure lockable USB cable, and is compatible with various triggering methods including digital, and software triggers.

3.2 Abbreviations

- AI: Analog Input
- AO: Analog Output
- DI: Digital Input
- DO: Digital Output
- CI: Counter Input
- CO: Counter Output
- DAQ: Data AcQuisition
- ADC: Analog-to-Digital Conversion
- DAC: Digital-to-Analog Conversion
- PFI: Programmable Function Interface
- SE: Single-Ended
- PPM: Parts Per Million

3.3 JYPEDIA and Learn by Example

JYTEK has added **Learn by Example** in this manual. We provide many sample programs for this device. Please download the sample programs for this device. You can download a JYPEDIA excel file from our web www.jytek.com. Open JYPEDIA and search for JY USB-1601 in the driver sheet, select **JY USB-1601 Examples.zip**. In addition to the download information, JYPEDIA also has a lot of other valuable information, JYTEK highly recommend you use this file to obtain information from JYTEK

.

Drivers	Update Date	Category
JYUSB-1601 V1.0.0 Win.rar	2024/12/13	Driver
JYUSB-1601 V1.0.0 Examples.rar	2024/12/13	Example
JYUSB-1601 V1.0.0 C++Examples.rar	2024/12/13	Driver
JYUSB-1601 V1.0.0 Python.rar	2024/12/13	Driver
JYUSB-1601 V1.0.0 PythonExamples.rar	2024/12/13	Example

Figure 1 JYPEDIA Information

4. Hardware Specifications

4.1 Front Panel

4.2 Pin Definition

Function (Left)	Pin Number	Pin Number	Function (Right)
AGND	1	40	AGND
AI15(AI7-)	2	39	AI7(AI7+)
AI14(AI6-)	3	38	AI6(AI6+)
AI13(AI5-)	4	37	AI5(AI5+)
AI12(AI4-)	5	36	AI4(AI4+)
AI11(AI3-)	6	35	AI3(AI3+)
AI10(AI2-)	7	34	AI2(AI2+)
AI9(AI1-)	8	33	AI1(AI1+)
AI8(AI0-)	9	32	AI0(AI0+)
AO1	10	31	AO0
AGND	11	30	AGND
DIO15/AI_CLK_Out	12	29	DIO7/CI_Src1/A
DIO14/AO_CLK_Out	13	28	DIO6/CI_Gate1
DIO13/AI_CLK_In	14	27	DIO5/CI_Aux1/B
DIO12/AO_CLK_In	15	26	DIO4/CO1
DIO11	16	25	DIO3/CI_Src0/A
DIO10	17	24	DIO2/CI_Gate0
DIO9	18	23	DIO1/CI_Aux0/B
DIO8	19	22	DIO0/CO0

DGND	20	21	DGND

Table 12 Pin Definition

4.3 AI Bandwith

Range(V)	-3 dB Bandwidth
±2.5 V,±5 V,±10 V	15 kHz, 30 kHz, 70 kHz

Table 13 AI Bandwith

4.4 Accuracy

4.4.1 AI Accuracy

JY-1601 Basic Accuracy = ±(% Reading+% Range)												
Nominal Range (V)	24 Hour Tcal ±1°C			90 Days Tcal±5°C			Temperature Coefficients(/°C)			24 Hr Full Scale Accuracy(V)	90 Days Full Scale Accuracy(V)	Full Scale Accuracy(%)
2.5	0.005	+	0.024	0.018	+	0.026	0.0025	+	0.0005	0.0007	0.0011	0.044
5	0.005	+	0.017	0.017	+	0.019	0.0024	+	0.0004	0.0011	0.0018	0.036
10	0.005	+	0.014	0.018	+	0.016	0.0026	+	0.0005	0.0019	0.0034	0.034
Notes: Accuracy valid to 99% of full range The 90-day data is estimated by multiplying the 24-hour data by 2.5.												

Table 14 AI Accuracy

4.4.2 AO Accuracy

JY-1601 Basic AO Accuracy = ±(% Output+% Range)												
Nominal Range (V)	24 Hour Tcal ±1°C			90 Days Tcal±5°C			Temperature Coefficients(/°C)			24 Hr Full Scale Accuracy (V)	90 Days Full Scale Accuracy (V)	Full Scale Accuracy(%)
10	0.006	+	0.013	0.016	+	0.015	0.0011	+	0.0005	0.0018	0.003	0.031
Notes: Accuracy valid to 95% of full range The 90-day data is estimated by multiplying the 24-hour data by 2.5.												

Table 15 AO Accuracy

4.4.3 Clock Accuracy

Clock Accuracy	50 ppm

Table 16 Clock Accuracy

4.4.4 System Noise

Range(V)	Noise (μVrms)
±2.5 V	54
±5 V	88
±10 V	164

Table 17 System Noise

4.5 Connector

I/O connectors	Two 20-pin removable screw-down terminals
USB connector	USB 2.0 Type-c
Screw terminal wiring	16 AWG to 28 AWG
Torque for screw terminals	0.2 N · m to 0.25 N · m

Table 18 Connector Information

4.6 Crosstalk

Type	Range(V)	Crosstalk (dB at 10 kHz)
Adjacent	±2.5V	-79 dB
	±5V	-85 dB
	±10V	-91 dB
Non-adjacent	±2.5V	-77 dB
	±5V	-83 dB
	±10V	-89 dB

Table 19 Crosstalk

5. Software

5.1 System Requirements

JY USB-1601 boards can be used in a Windows or a Linux operating system.

Microsoft Windows: 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.

5.2 System Software

When using the JY USB-1601 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 JY USB-1601 be with .NET Framework 4.0 with Microsoft Visual Studio 2015. JYTEK relies on Microsoft to maintain the compatibility for the newer versions.

5.3 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.

5.4 USB-1601 Hardware Driver

After installing the required application development environment as described above, you need to install the JY USB-1601 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 JY USB-1601 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.

5.5 Install the SeeSharpTools from JYTEK

To efficiently and effectively use JY USB-1601 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 JY USB-1601 hardware. Please register and down load the latest SeeSharpTools from our website, www.jytek.com.

5.6 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.

6. Operating USB-1601

This chapter provides the operation guides for JY USB-1601, including AI, AO, DI, DO and Counter, etc.

JYTEK provides extensive examples, on-line help and documentation to assist you to operate the JY USB-1601 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.

6.1 Quick Start

After you have installed the driver software and the SeeSharpTools, you are ready to use Microsoft Visual Studio C# to operate the JY USB-1601 products.

If you are already familiar with Microsoft Visual Studio C#, the quickest way to use USB-1601 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 USB-1601.

6.2 Analog Input Operations

you need to configure AI channels and set up some parameters through USB-1601 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*): USB-1601 provides 3 acquisiontion modes, **Continuous**, **Finite**, **Single Point**, Record, which will be described in details in Section 6.2.1-6.2.4.

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

SamplesToAcquire: This parameter behaves differently in the different AI acquisition modes. In the continuous acquisition mode, *SamplesToAcuire* is the buffer size used in the AI acquisition task, please see Section 6.2.1; in the finite acquisition mode, it is the total number of samples to capture, please see Section 6.2.2.

Channel Count: how many channels you want to collect data. Users can choose the collection channel according to their actual needs

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.*

Learn by Example 6.2

- Connect the signal source's positive outputs to USB-1601 AI Ch0 and GND.
- Set a sinewave signal (f=10kHz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous**, set the following numbers as shown. This sample program will continuously acquire data from AI Ch0.

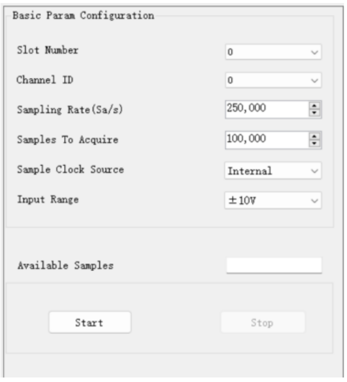

Figure 2 Analog Input Parameters

- *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.

- Acquision will start after clicking Start Button. The result is shown below.

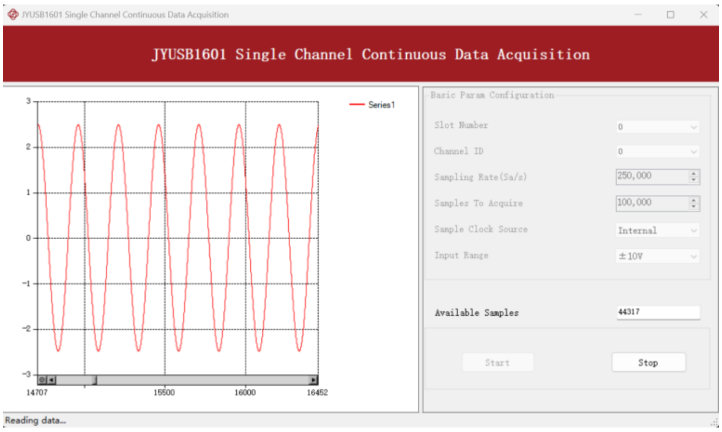

Figure 3 AI Acquisition Result

6.2.1 Continuous Acquisition

An AI acquisition task will acquire the data continuously until the task is stopped. The JYUSB-1601 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.

6.2.2 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.

6.2.3 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.

6.2.4 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.

6.3 Analog Input Terminal Type

The USB-1601 provide 2 analog input terminal types:

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

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

6.3.1 DIFF Mode

The following figures show the actual collected voltage range diamond plots in DIFF mode for three ranges:±10V**(**错误!未找到引用源。**)**, ±5V**(**错误!未找到引用源。**)**, and ±2.5V**(**错误!未找到引用源。**)**.The inputs at both ends Vin+ and Vin- must be within the range.

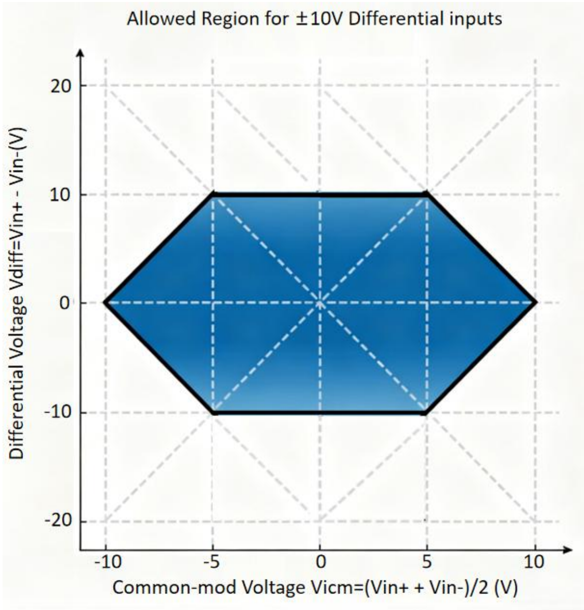

Figure 4 10V Range Diamond Plot

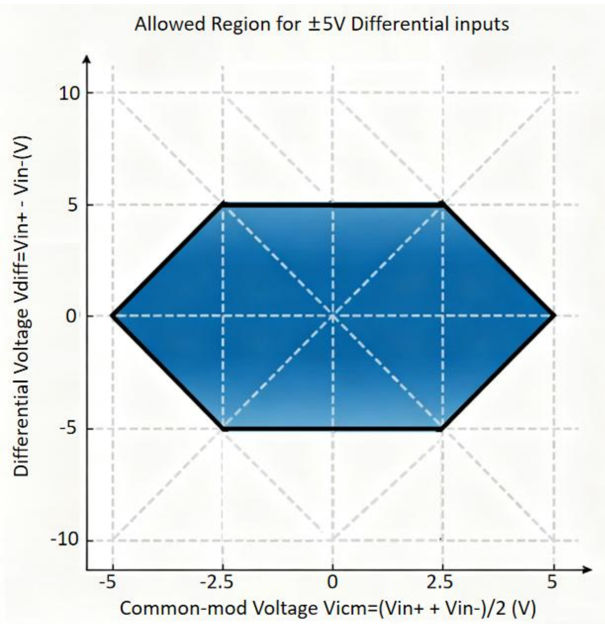

Figure 5 5V Range Diamond Plot

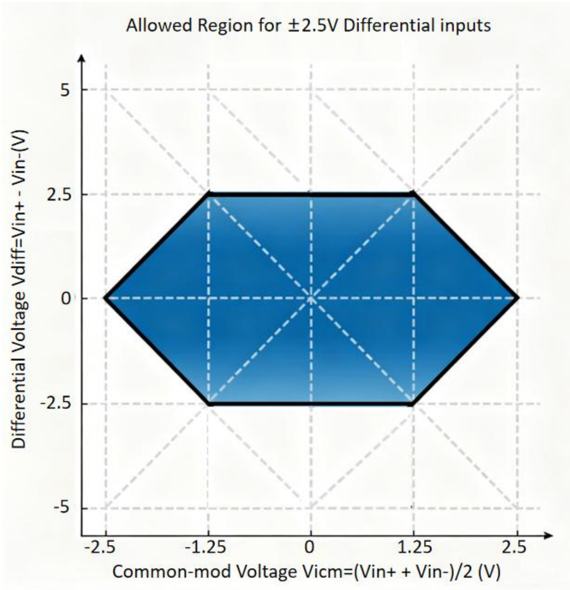

Figure 6 2.5V Range Diamond Plot

6.4 Trigger Source

There are 3 trigger types: Immediate trigger, Software trigger and Digital trigger. The trigger type is a property and set by driver software.

6.4.1 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 6.4.1

- Use the same program and connection as in Learn by example 6.2**.**

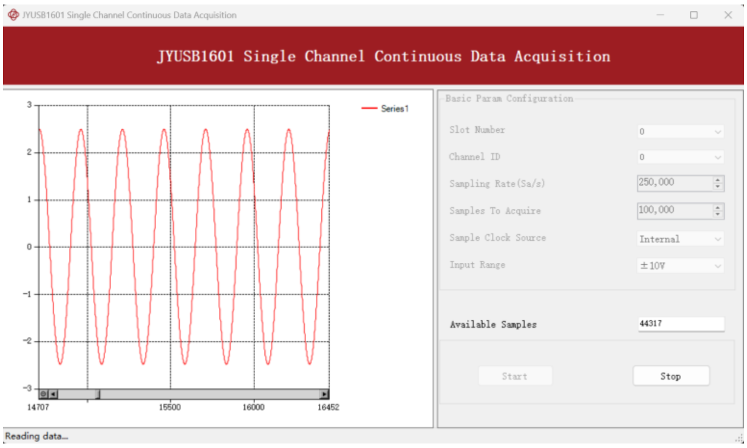

Figure 7 Immediate trigger Paraments

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

6.4.2 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 6.4.2

- Connect the signal source's positive terminal to JYUSB-1601 AI Ch0, the negative terminal to the ground
- Set a sinewave signal (f=10kHz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous Soft Trigger**, set the following numbers as shown.
- Click **Start** to run the task.

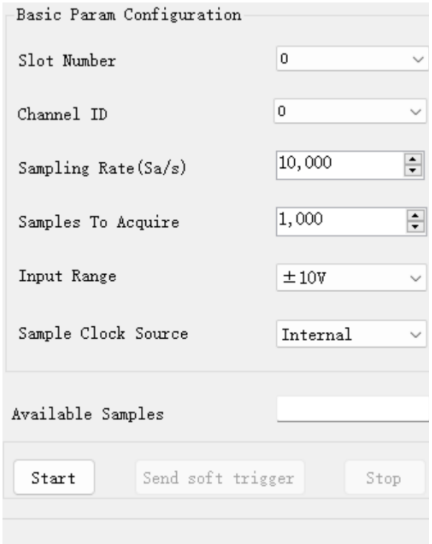

Figure 8 Software trigger Paraments

- 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:

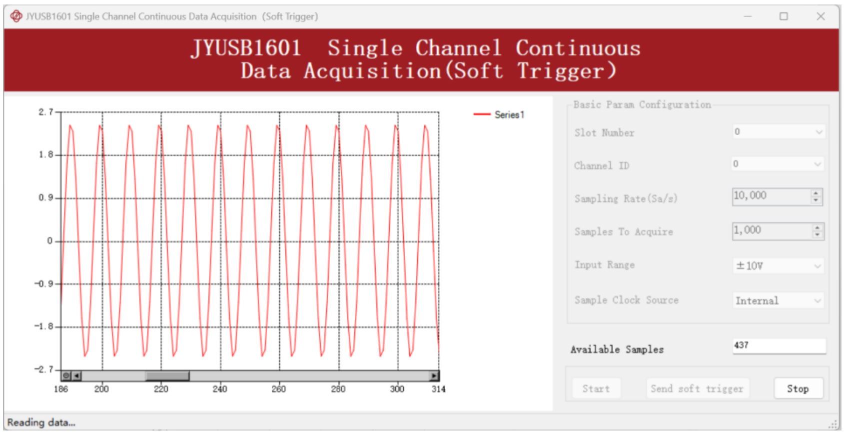

Figure 9 Software trigger Acquisition

6.4.3 Digital Trigger

JYUSB-1601 supports external digital trigger source from 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 10

Figure 10 External Digital Trigger

Learn by Example 6.4.3

- Connect the signal source positive terminals to JYUSB-1601 AI Ch0 and digital trigger source (DIO_0), negative terminals to the ground.
- Set a squarewave signal (f=10Hz, Vpp=5V).
- Open **Analog Input-->Winform AI Continuous Digital Trigger**, set the following numbers as shown.

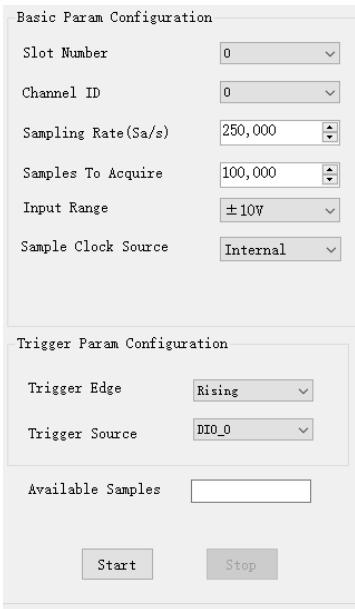

Figure 11 Digital Trigger Paraments

- **Trigger Source** must match the pin on JYUSB-1601.
- There are four **Trigger Edge**: **Rising**, **Falling, HighLevel and LowLevel**.

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

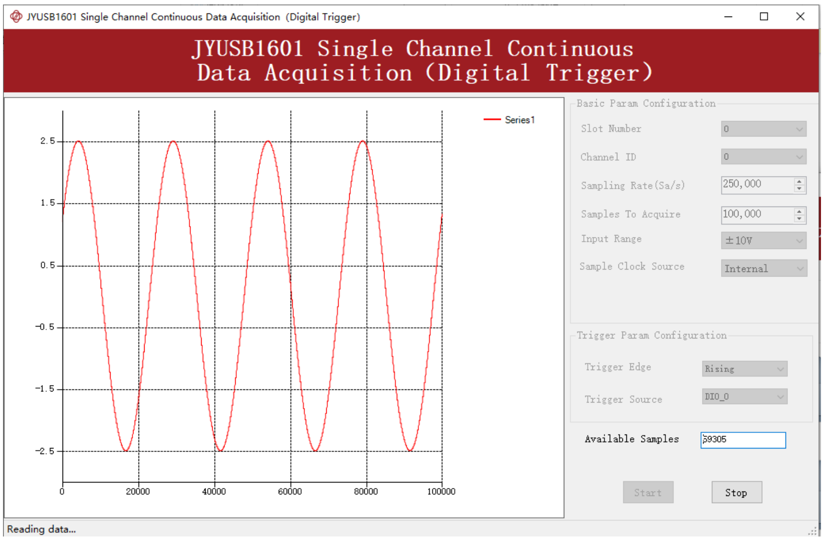

Figure 12 Digital Trigger Acquisition

➢ 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.

6.5 Analog Output Operations

The JYUSB-1601 AO provides 16-bit simultaneous outputs. The analog output has 4 modes of operation: Single, Finite, ContinuousWrapping, and ContinuousNoWrapping.

6.5.1 Single Output

The Single output mode achieves specific simulated voltage output at a single point through the Console

Learn by Example 6.5.1

■ Open **Analog Output--> Console AO Single Point**, upon execution, the console window will be launched.

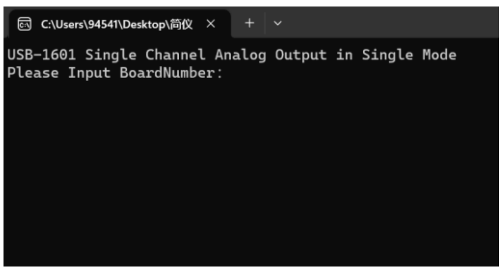

Figure 13 AO Single Point Console

- Based on the console prompts, enter the card number, output channel number, and output voltage value. Here, select to output 5V on AO0.

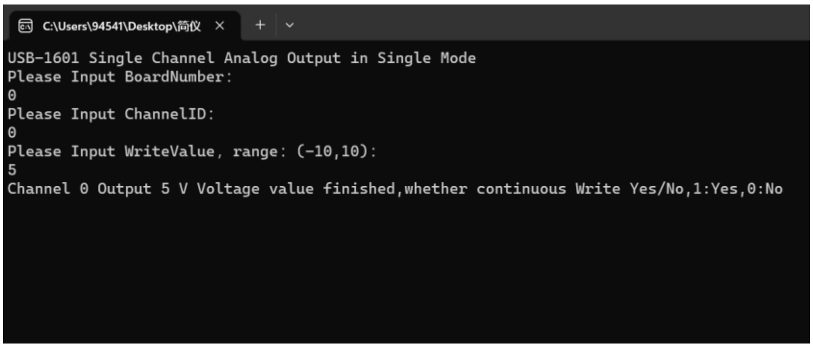

Figure 14 Console AO Single Point Result

- Using a multimeter to measure the voltage between AO0 and ground yields a voltage of 5V.

6.5.2 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 6.5.2

- Connect JYUSB-1601 AO Ch0 and ground to oscilloscope BNC connector.
- Open **Analog Output-->Winform AO Finite**, set the following numbers as shown:

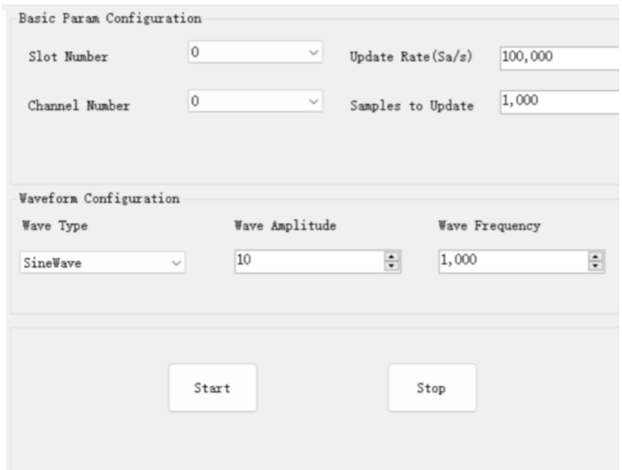

Figure 15 AO Finite Output Paraments

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

Figure 16 AO Finite Signal

- And the received signal is shown below.

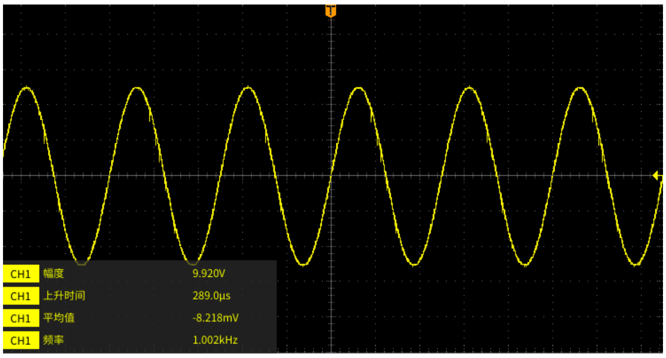

Figure 17 Oscilloscope Acquisition Signal

6.5.3 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 6.5.3

- Connect JYUSB-1601 AO Ch0 and ground to oscilloscope BNC connector.
- Open **Analog Output-->Winform AO Continuous NoWrapping**, set the following numbers as shown:

Figure 18 AO Continuous NoWrapping 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.

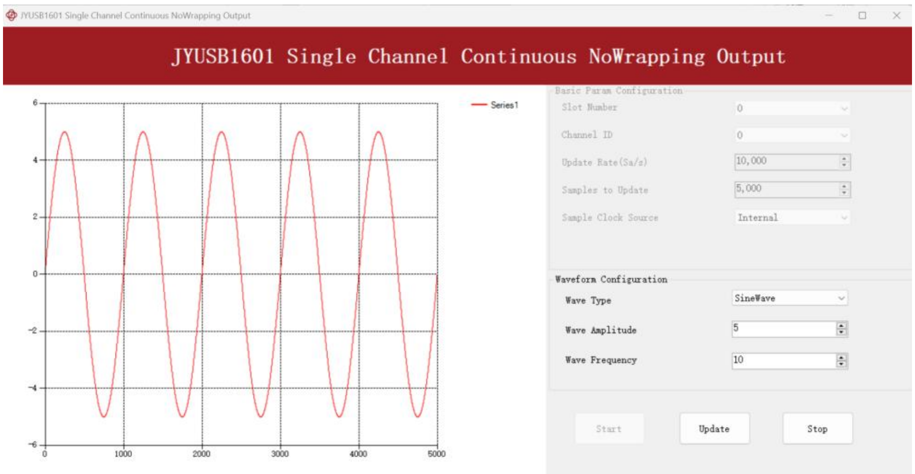

Figure 19 AO Continuous NoWrapping Signal

■ And the received signal is shown below.

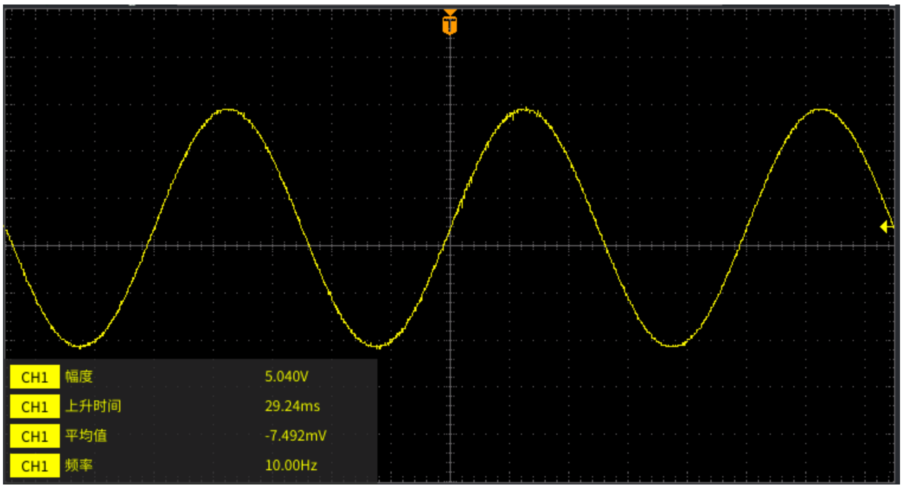

Figure 20 Oscilloscope Acquisition AO Sin Signal

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

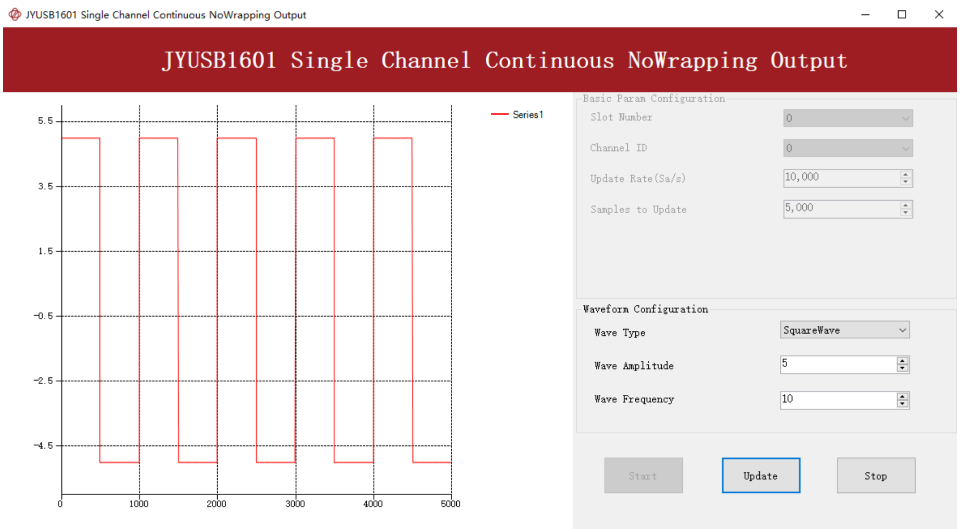

Figure 21 Update AO Square Signal

- And the received signal is shown below.

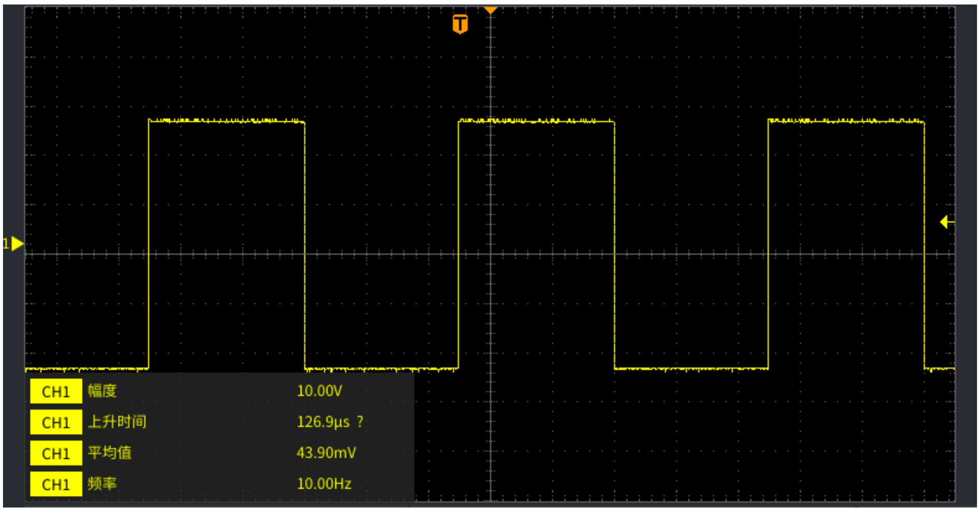

Figure 22 Oscilloscope Acquisition AO Square Signal

6.5.4 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 6.5.4

- Connect JYUSB-1601 AO Ch0 and ground to oscilloscope BNC connector.
- Open **Analog Output-->Winform AO Continuous Wrapping**, set the numbers as shown.

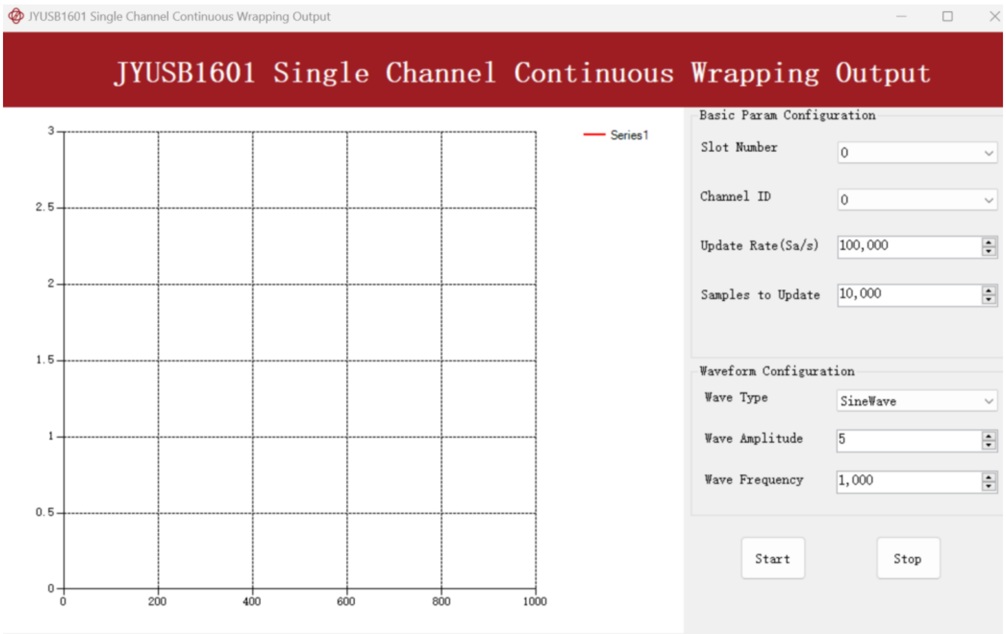

Figure 23 AO Continuous Wrapping Paraments

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

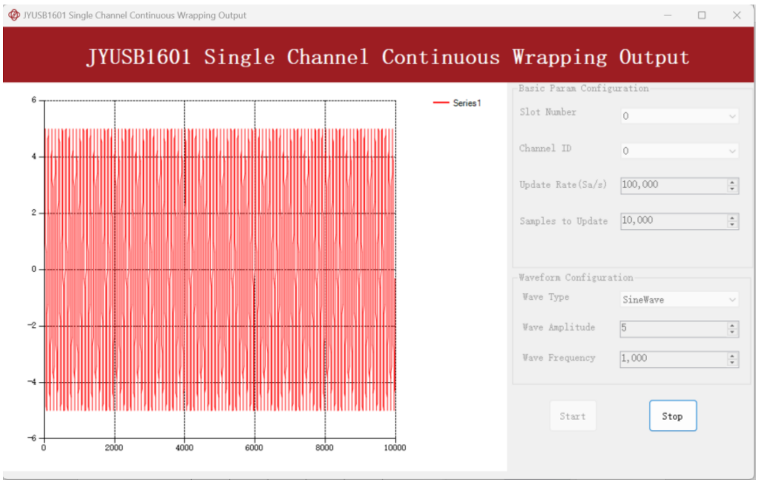

Figure 24 AO Continuous Wrapping Signal

- And the received signal is shown below.

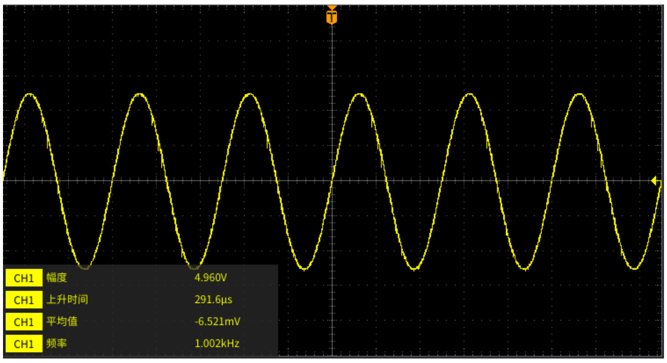

Figure 25 Oscilloscope Acquisition AO Signal

6.6 Digital I/O Operations

The JYUSB-1601 provides powerful programmable digital I/O functions. Programmable I/O supports static TTL. User can acess these I / O information through software polling.

Learn by Example 6.6

- In this example JYUSB-1601 outputs a digital signal by its DO function and reads it back by its DI function.
- Connect DIO Ch0 to Ch4 and Ch5. JYUSB-1601 sends a digital signal through Ch0 and reads the signal back from Ch4 and Ch5.
- Open the first program **Digital Output-->Winform DO SinglePoint.**
- Set Line 1 in High-Level positions, make sure all other lines are in Low-Level positions. Click **Start** to generate the High-Levels as shown.

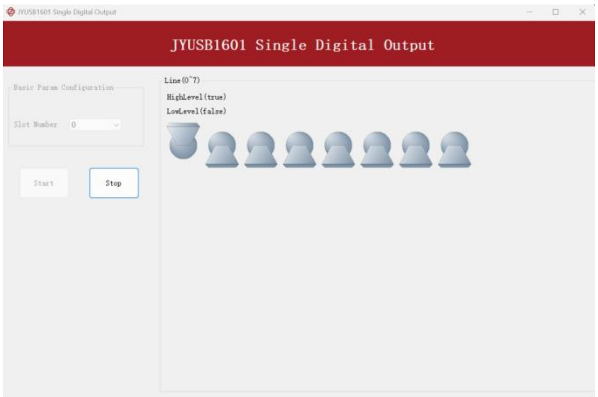

Figure 26 Single Digital Output

- Open the second program **Digital Input-->Winform DI SinglePoint**.
- Click **Check DI Status**. The result is shown below.

Figure 27 Single Digital Input

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

6.7 Counter Input Operations

6.7.1 Edge Counting

The counter counts the number of active edges of input signal.The count value is written to the register on each rising edge or falling edge of the signal to measure as shown in Figure 28.

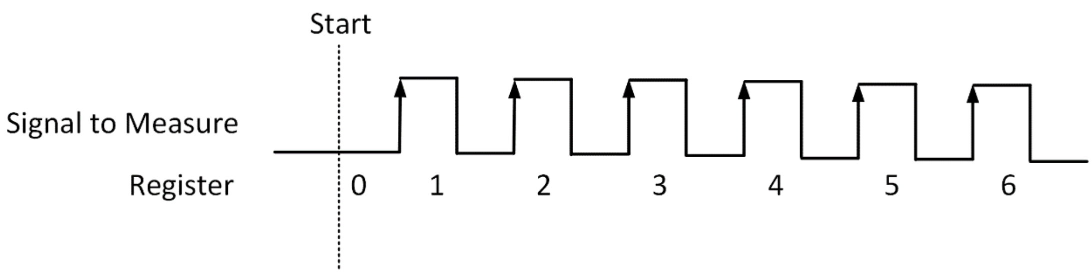

Figure 28 Simple Edge Counting in Single Mode

Learn by Examples 6.7.1

- Connect the signal source's positive terminal of a signal source to JYUSB-1601 SRC0, negative terminal to the ground.
- Set a squarewave signal (f=1kHz, Vpp=5V).
- Open **Counter Input-->Winform CI Single EdgeCounting**, set the following numbers as shown:

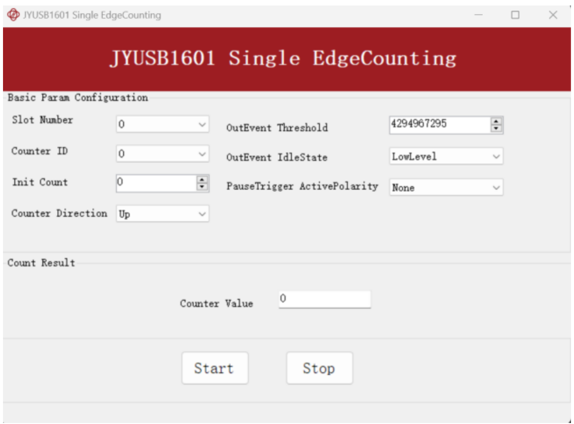

Figure 29 EdgeCounting for Single Mode

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

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

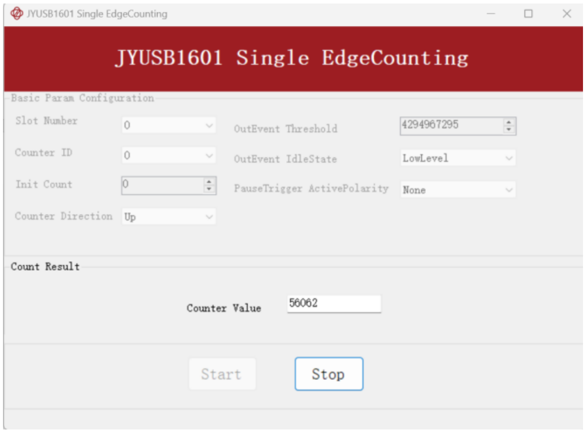

Figure 30 Edge Counting Result

6.7.2 Pulse Measurement

The counter measures the high-level and low-level duration of signal.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 31.

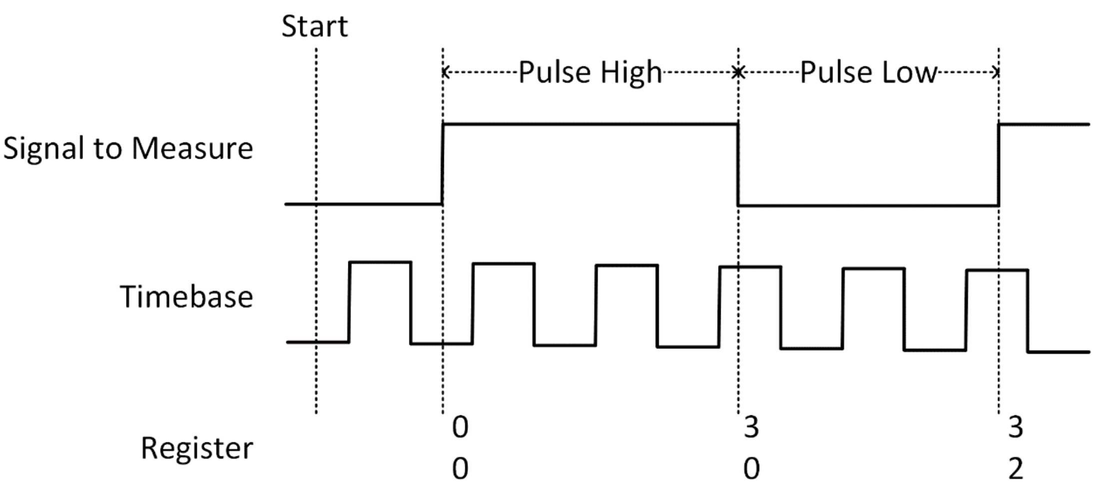

Figure 31 Pulse Measurement in Single Mode

Learn by Examples 6.7.2

- Connect the signal source's positive terminal of a signal source to JYUSB-1601 SRC0, negative terminal to the ground.
- Set a squarewave signal (f=1kHz, Duty Cycle=50%, Vpp=5V).
- Open **Counter Input-->Winform CI Single PulseMeasure**, set the following numbers as shown:

Figure 32 Pulse Measure for Single Mode

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

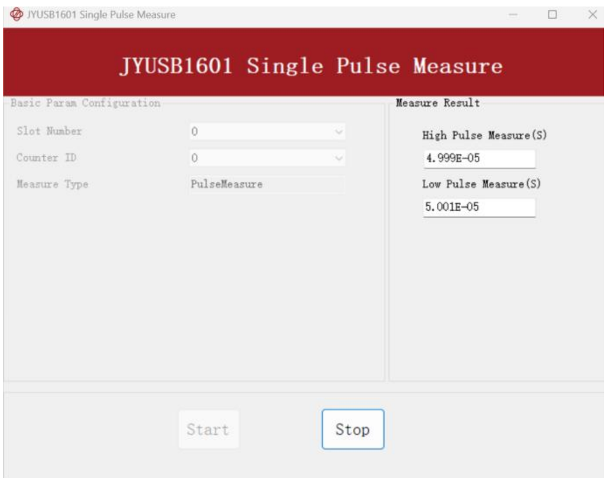

Figure 33 Pulse Measure Value for Single Mode

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

6.7.3 Frequency Measurement

The counter measures the frequency of signal to measure.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.

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

Learn by Examples 6.7.3

- Connect the signal source's positive terminal of a signal source to JYUSB-1601 SRC0, negative terminal to the ground.
- Set a squarewave signal (f=10kHz, Duty Cycle=50%, Vpp=5V).
- Open **Counter Input-->Winform CI Single Frequency Measure** and click **Start**. The result is shown below by **Frequency Measure (Hz)**:

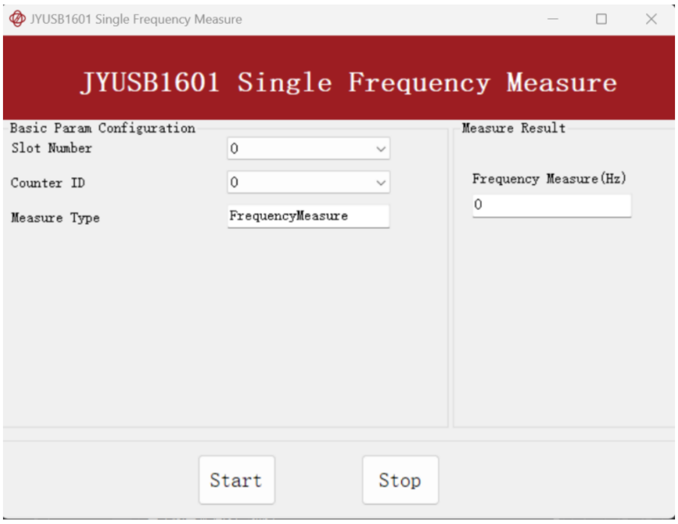

Figure 34 Frequency Measure for Single Mode

- ➢ The result matches the frequency set before.

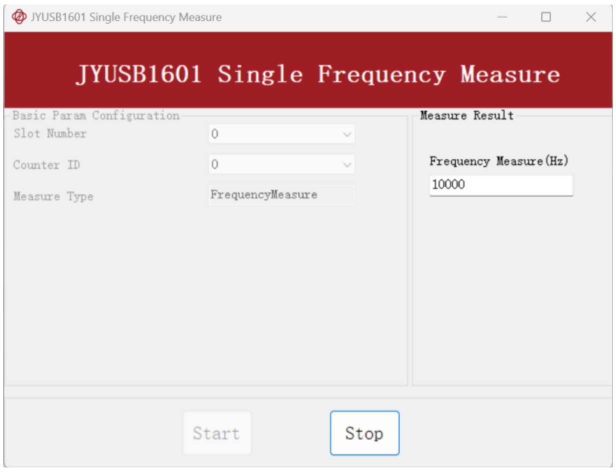

Figure 35 Frequency Measure Result

6.7.4 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.

Learn by Examples 6.7.4

- Connect the signal source's positive terminal of a signal source to JYUSB-1601 SRC0, negative terminal to the ground.
- Set a squarewave signal (f=10kHz, Duty Cycle=50%, Vpp=5V).
- Open **Counter Input-->Winform CI Single Period Measure** and click **Start**. The result is shown below by **Period Measure(S)**:

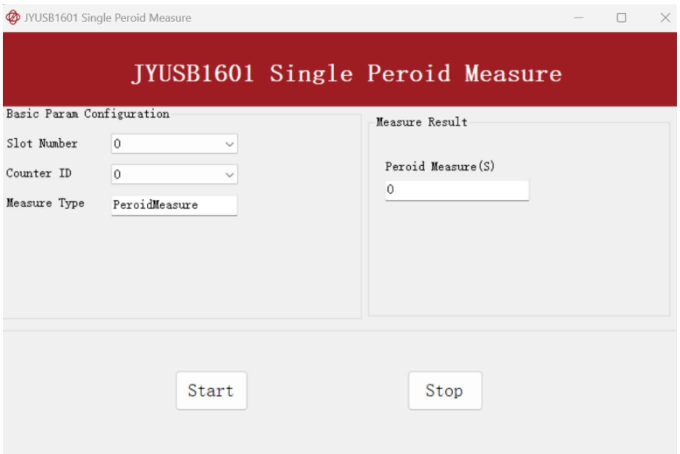

Figure 36 Peroid Measure for Single Mode

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

6.7.5 Two-Edge Separation

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

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 37.

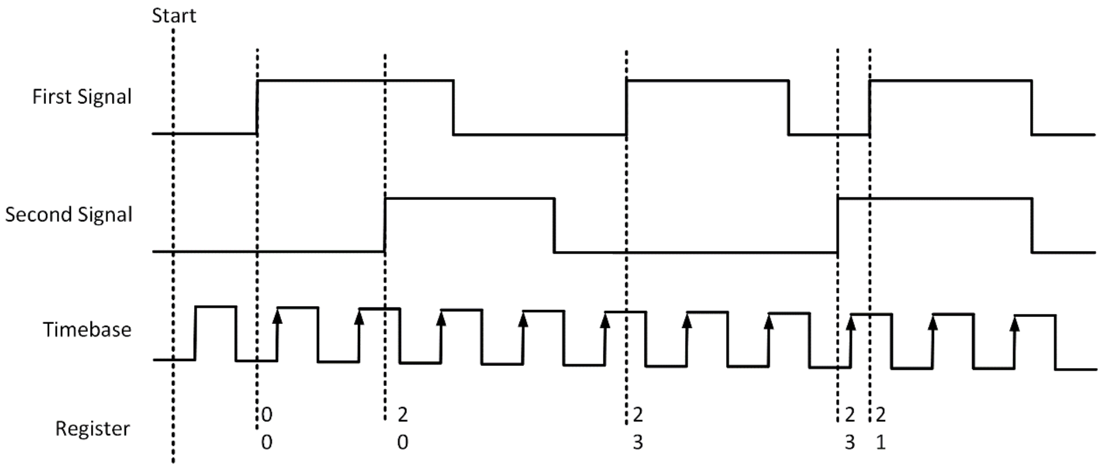

Figure 37 Two-Edge Separation in Single Mode

Learn by Examples 6.7.5

- Connect the 2 signal source's positive terminal of a signal source to JYUSB-1601 SRC0 and AUX, 2 negative terminal to the ground.
- Set a squarewave signal (f=1Hz, Phase=0°) and a squarewave signal (f=1Hz, Phase=135°).
- Open **Counter Input-->Winform CI Single TwoEdgeSeparation Measure** and click **Start**. The result is shown below by **First to Second(S)** and **Second to First(S)**, which represent the time difference between the rising edges of the two signals:

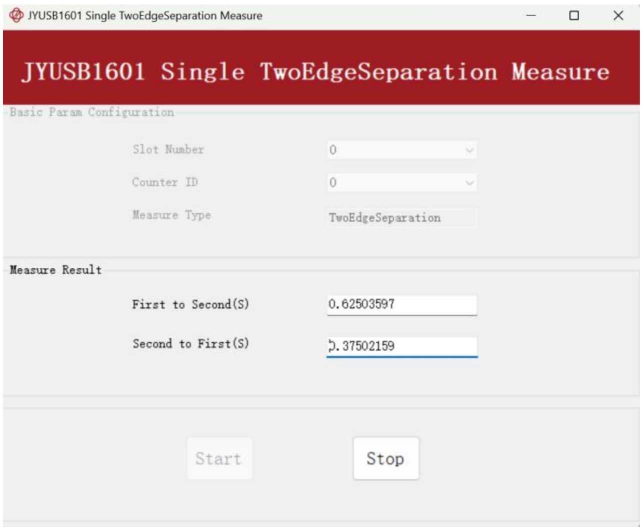

Figure 38 Two-EdgeSeparation Measure for Single Mode

- 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 s.
- 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 s.

6.7.6 Quadrature Encoder

The quadrature encoder includestwo encoding types: x2, and x4.

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 39.

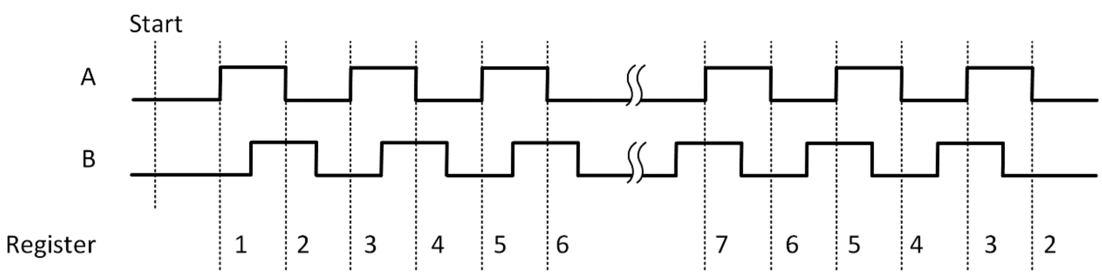

Figure 39 Quadrature Encoder x2 Mode

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 40.

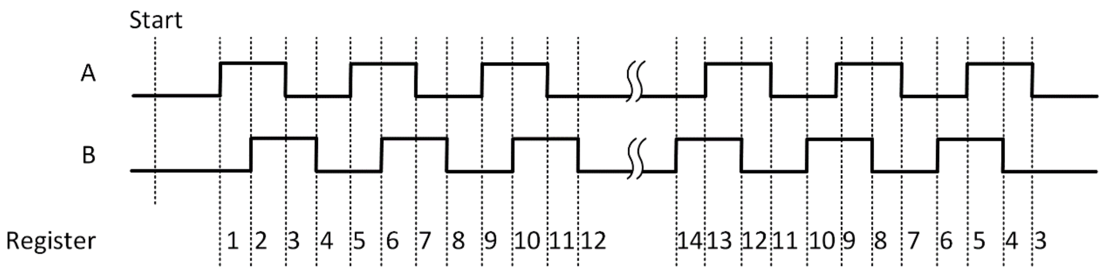

Figure 40 Quadrature Encoder x4 modeTake Encoding x2 mode as an example.

The count value is written to the register on each rising edge of the signal A, as shown in Figure 28.

Learn by Examples 6.7.6

- Connect the 2 signal source's positive terminal of a signal source to JYUSB-1601 SRC0 and AUX, 2 negative terminals to the ground.
- Set a squarewave signal (f=1Hz, Phase=0°) and a squarewave signal (f=1Hz, Phase=135°).
- Open **Counter Input--> Winform CI Single QuadEncoder** and click **Start**. The result is shown below by **CounterValue** according to the counting rules.

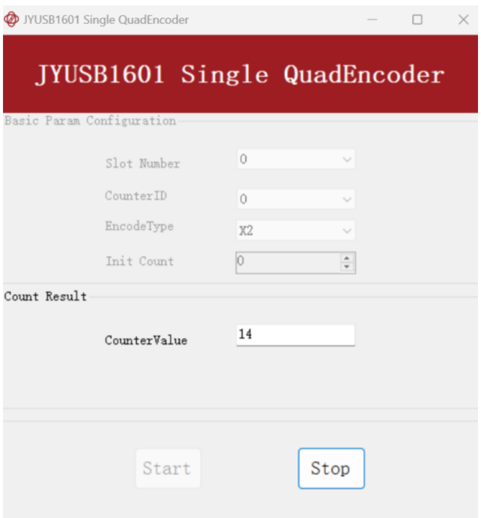

Figure 41 QuadEncoder for Single Mode

- The table in the sample program is a connection diagram for your convenience.
- *Encoding Type* is set by **Encode Type (x2, x4)**.
- When the *encode type* is changed from x2 tox4, you can see the rising speed of **CounterValue** is twice than x2Mode.

7. Calibration

JY USB-1601 Series boards are precalibrated before the shipment. We recommend you recalibrate JY USB-1601 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.

8. Using JY USB-1601 in Other Software

While JYTEK's default application platform is Visual Studio, the programming language is C#, we recognize there are other platforms that are either becoming very popular or have been widely used in the data acquisition applications. Among them are Python, C++ and LabVIEW. This chapter explains how you can use JY USB-1601 DAQ card using one of this software.

8.1 Python

JYTEK provides and supports a native Python driver for JY USB-1601 boards. There are many different versions of Python. JYTEK has only tested in CPython version 3.5.4. There is no guarantee that JYTEK python drivers will work correctly with other versions of Python.

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

8.2 C++

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

8.3 LabVIEW

LabVIEW is a software product from National Instruments. JYTEK does not support LabVIEW and will no longer provide LabVIEW interface to JY USB-1601 boards. Our third-party partners may have LabVIEW support to JY USB-1601 boards. We can recommend you if you want to convert your LabVIEW applications to C# based applications.

9. Appendix

9.1 Typical Measurement Error

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

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

9.2 System Noise

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

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

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

9.3 Temperature Drift

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

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

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

10.About JYTEK

10.1 JYTEK China

Founded in June, 2016, JYTEK China is a leading Chinese test & measurement company, providing complete software and hardware products for the test and measurement industry. The company has evolved from re-branding and reselling PXI(e) and DAQ products to a fully-fledged product company. The company offers complete lines of PXI, DAQ, USB products. More importantly, JYTEK has been promoting open-sourced based ecosystem and offers complete software products. Presently, JYTEK is focused on the Chinese market. Our Shanghai headquarters and production service center have regular stocks to ensure timely supply; we also have R&D centers in Xi'an and Chongqing. We also have highly trained direct technical sales representatives in Shanghai, Beijing, Tianjin, Xi'an, Chengdu, Nanjing, Wuhan, Guangdong, Haerbin, and Changchun. We also have many partners who provide system level support in various cities.

10.2 JYTEK Software Platform

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

10.3 JYTEK Warranty and Support Services

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

11.Statement

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

This manual provides the product review, quick start, some driver interface explanation for JYTEK JY USB-1601 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