

Smacq USB-3000 Series Multifunctional Data Acquisition Devices

Programming Guide

Rev. C

Smacq

www.smacq.com

www.smacq.cn

Statement

Copyright

©2018 Beijing Smacq Technology Co., Ltd. All rights reserved
No content of this manual may be reproduced, modified or abridged without prior consent and written permission.

Trademark information

Smacq is a registered trademark of Beijing Smacq Technology Co., Ltd.

The names of the other products and companies mentioned in this document are trademarks or trade names of their respective companies.

Other Disclaimer

- The information provided in this document may be modified and updated in subsequent versions without prior notice.
- Smacq does not provide any warranties, express or implied, for this document as well as the information inside it, including but not limited to implied warranties of the marketable nature of the product and its suitability for a particular purpose.
- There may be a chance that inaccurate descriptions or errors exist in this document. Smacq does not hold any liability for accidents because of the information and deductive functions provided in this manual, as well as the resulting loss of any chance because of the use of this document.
- Smacq reserves the rights to change product specifications, prices, and decide whether to stop the production.

Contact Us

If you have any questions or need assistance in using this product or this document, please contact us via:

Phone: (+86)10-52482802
E-Mail: service@smacq.com
Website: <http://www.smacq.com>
<http://www.smacq.cn>

CONTENT

STATEMENT	1
1. GETTING STARTED	5
1.1. PROGRAMMING LANGUAGE	5
2. FUNCTION DESCRIPTION	6
2.1. FUNCTION TYPE.....	6
2.2. BASIC OPERATING FUNCTIONS	9
<i>FindUSB3DAQ()</i>	9
<i>USB3OpenDevice()</i>	9
<i>USB3CloseDevice()</i>	9
<i>USB3GetDeviceSN()</i>	9
<i>USB3GetDeviceModel()</i>	9
2.3. ANALOG INPUT SETUP FUNCTION.....	11
<i>SetUSB3AiSampleRate()</i>	11
<i>SetUSB3AiSampleMode()</i>	11
<i>SetUSB3AiConnectType()</i>	11
<i>SetUSB3AiRange()</i>	11
<i>SetUSB3AiChanSel()</i>	12
<i>SetUSB3AiTrigSource()</i>	12
<i>SetUSB3AiConvSource()</i>	13
<i>SetUSB3AiPreTrigPoints()</i>	13
<i>SetUSB3AiOneShotPoints()</i>	13
<i>SetUSB3ClrAiFifo()</i>	14
2.4. DIGITAL I/O SETTING FUNCTIONS	15
<i>SetUSB3DiSampleRate()</i>	15
<i>SetUSB3DiSampleMode()</i>	15
<i>SetUSB3DiTrigSource()</i>	15
<i>SetUSB3DiConvSource()</i>	16
<i>SetUSB3DiPreTrigPoints()</i>	16
<i>SetUSB3DiOneShotPoints()</i>	16
<i>SetUSB3ClrDiFifo()</i>	17
<i>SetUSB3DoSampleRate()</i>	17
<i>SetUSB3DoSampleMode()</i>	17
<i>SetUSB3DoTrigSource()</i>	17
<i>SetUSB3DoConvSource()</i>	18
<i>SetUSB3DoCycle()</i>	19
<i>SetUSB3DoDataFifo()</i>	19
<i>SetUSB3ClrDoFifo()</i>	19
<i>SetUSB3DoWaveCtrl()</i>	19
<i>SetUSB3DoImmediately()</i>	20
2.5. ANALOG OUTPUT SETUP FUNCTIONS	21
<i>SetUSB3AoSampleRate()</i>	21

<i>SetUSB3AoSampleMode()</i>	21
<i>SetUSB3AoTrigSource()</i>	21
<i>SetUSB3AoConvSource()</i>	22
<i>SetUSB3AoCycle()</i>	23
<i>SetUSB3AoDataFifo()</i>	23
<i>SetUSB3ClrAoFifo()</i>	23
<i>SetUSB3AoSync()</i>	23
<i>SetUSB3AoImmediately()</i>	24
2.6. SETTING FUNCTION OF COUNTER	25
<i>SetCtRunMode()</i>	25
<i>SetCtSampleMode()</i>	25
<i>SetCtSampleRate()</i>	26
<i>SetCtTrigSource()</i>	26
<i>SetCtConvSource()</i>	27
<i>SetCtCountEdge()</i>	28
<i>SetCtValue()</i>	28
<i>SetCtOverflowValueEdgeCount()</i>	29
<i>SetCtReloadValueEdgeCount()</i>	29
<i>SetCtDirEdgeCount()</i>	30
<i>SetCtPulseWidthStartEdge()</i>	30
<i>SetCtInternalSrcPeriod()</i>	31
<i>SetCtSrcSel()</i>	31
<i>SetCtEncoderMode()</i>	31
<i>SetCtZPhase()</i>	32
<i>SetCtZValue()</i>	32
<i>SetCtZEnable()</i>	33
<i>SetCtSoftTrig()</i>	33
<i>SetCtClrFifo()</i>	33
2.7. TRIGGER SETUP FUNCTIONS	34
<i>SetUSB3AiSoftTrig()</i>	34
<i>SetUSB3DiSoftTrig()</i>	34
<i>SetUSB3DoSoftTrig()</i>	34
<i>SetUSB3AoSoftTrig()</i>	34
<i>SetUSB3GlobalSoftTrig()</i>	34
<i>SetUSB3ClrTrigger()</i>	35
<i>SetUSB3ClrAiTrigger()</i>	35
<i>SetUSB3ClrDiTrigger()</i>	35
<i>SetUSB3ClrDoTrigger()</i>	35
<i>SetUSB3ClrAoTrigger()</i>	35
<i>SetUSB3ClrGlobalSoftTrig()</i>	36
2.8. SYNCHRONIZATION SYSTEM SETUP FUNCTIONS	37
<i>SetUSB3ExtTrigOutSource()</i>	37
<i>SetUSB3ExtConvOutSource()</i>	37
2.9. DATA READ FUNCTION	38

<i>USB3GetAi()</i>	38
<i>USB3GetDi()</i>	38
<i>USB3GetCt()</i>	39
<i>GetLatestCtValue()</i>	39
3. ERROR CODES	40

1. Getting Started

User application can interact with USB-3000 series data acquisition device via functions provided by USB3000.dll, which is a standard dynamic link library. This manual will introduce all functions provided by USB3000.dll.

This guide covers all models of USB-3xxx data acquisition device. If the acquisition device you purchase does not support certain functions, then in the programming, you will not be able to use these functions. Only the specific models can use the corresponding functions.

1.1. Programming language

USB-3000.dll is a standard dynamic link library that supports Windows OS. In principle, all development languages that can invoke the standard dynamic link library are supported. We provided example codes for languages including VC ++, VB, C#, LabVIEW, and MATLAB.

This manual is based on C language functions when introducing each function provided by USB-3000.dll.



Attention When programming in other languages, you need to pay attention to the data type difference from C language. In different programming languages, it is possible when data type names are the same, the corresponding data length is different. When programming, the substitution parameters should be based on the data length.

2. Function Description

This chapter introduces the type and details of all functions provided by USB3000.dll.

2.1. Function type

The types of functions provided by USB3000.dll are listed below.

- Basic operating functions

Function name	Description
FindUSB3DAQ()	Query the number of connected acquisition devices
USB3OpenDevice()	Turn on the device
USB3CloseDevice()	Turn off the device
USB3GetDeviceSN()	Read the acquisition device serial number
USB3GetDeviceModel()	Read the acquisition device model

- Analog input setup function

Function name	Description
SetUSB3AiSampleRate()	Set analog input sampling rate
SetUSB3AiSampleMode()	Set up analog input acquisition mode
SetUSB3AiConnectType()	Set up analog input wiring mode
SetUSB3AiRange()	Set up analog input range
SetUSB3AiChanSel()	Set the channel you want to select for analog input
SetUSB3AiTrigSource()	Set up an analog input trigger source
SetUSB3AiConvSource()	Set up analog input sampling clock source
SetUSB3AiPreTrigPoints()	Set analog input pre-trigger points
SetUSB3AiOneShotPoints()	Set up analog input collection points when in OneShot mode
SetUSB3ClrAiFifo()	Empty analog input FIFO cache

- Digital I/O setting functions

Function name	Description
SetUSB3DiSampleRate()	Set the digital input sampling rate
SetUSB3DiSampleMode()	Set up digital input acquisition mode
SetUSB3DiTrigSource()	Set up a digital input trigger source
SetUSB3DiConvSource()	Set up a digital input sampling clock source
SetUSB3DiPreTrigPoints()	Set up digital input pre-trigger points
SetUSB3DiOneShotPoints()	Set up digital input collection points when in OneShot mode

SetUSB3ClrDiFifo()	Empty digital input FIFO cache
SetUSB3DoSampleRate()	Set digital output sampling rate
SetUSB3DoSampleMode()	Set up digital output mode
SetUSB3DoTrigSource()	Set up a digital output trigger source
SetUSB3DoConvSource()	Set up a digital output sampling clock source
SetUSB3DoCycle()	Set waveform output loops
SetUSB3DoDataFifo()	Set waveform output FIFO caching data
SetUSB3ClrDoFifo()	Empty waveform output FIFO cache
SetUSB3DoWaveCtrl()	Channel control on whether to output waveform
SetUSB3DoImmediately()	Set digital output to output immediately

- Analog output setup function

Function name	Description
SetUSB3AoSampleRate()	Set analog output sampling rate
SetUSB3AoSampleMode()	Set analog output mode
SetUSB3AoTrigSource()	Set up an analog output trigger source
SetUSB3AoConvSource()	Set up analog output sampling clock source
SetUSB3AoCycle()	Set analog waveform output loops
SetUSB3AoDataFif ()	Set up analog waveform output FIFO cache data
SetUSB3ClrAoFifo()	Empty analog waveform output FIFO cache
SetUSB3AoSync()	Set up analog waveform output synchronization channel
SetUSB3AoImmediately()	Set the analog output voltage value to output immediately

- Trigger Setup function

Function name	Description
SetUSB3AiSoftTrig()	Analog input software triggers
SetUSB3DiSoftTrig()	Digital input software triggers
SetUSB3DoSoftTrig()	Digital output software triggers
SetUSB3AoSoftTrig()	Analog output software triggers
SetUSB3GlobalSoftTrig()	Global software triggers
SetUSB3ClrTrigger()	Empty all trigger states
SetUSB3ClrAiTrigger()	Empty analog input trigger state
SetUSB3ClrDiTrigger()	Empty digital input trigger status
SetUSB3ClrDoTrigger()	Empty digital output trigger state
SetUSB3ClrAoTrigger()	Empty analog output trigger state
SetUSB3ClrGlobalSoftTrig()	Empty global software Trigger state

- Synchronize system Setup Functions

Function name	Description
SetUSB3ExtTrigOutSource()	Set up an external trigger output signal source
SetUSB3ExtConvOutSource()	Set up an external clock output signal source

- Data Read function

Function name	Description
USB3GetAi()	Read analog input data
USB3GetDi()	Read digital input data

2.2. Basic operating functions

FindUSB3DAQ()

`int _stdcall FindUSB3DAQ();`

Queries the number of acquisition devices that are connected to the computer.

Parameters	Note
Return value	Number of acquisition devices connected to the computer.

USB3OpenDevice()

`int _stdcall USB3OpenDevice(int DevIndex);`

Turn on the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

USB3CloseDevice()

`int _stdcall USB3CloseDevice(int DevIndex);`

Turn off the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

USB3GetDeviceSN()

`int _stdcall USB3GetDeviceSN(int DevIndex, char *SN);`

Load the serial number of the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
*SN	Byte array pointer, used to save the device serial number.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

USB3GetDeviceModel()

`int _stdcall USB3GetDeviceModel(int DevIndex, char *Model);`

Reads the model number of the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.

*Model	Byte array pointer, used to save the device model.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

2.3. Analog input setup function

SetUSB3AiSampleRate()

`int _stdcall SetUSB3AiSampleRate(int DevIndex, unsigned int SamplePeriod);`

Set the simulated input sampling rate for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
SamplePeriod	Sampling period, in ns. The sampling period must be in steps of 10ns.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AiSampleMode()

`int _stdcall SetUSB3AiSampleMode(int DevIndex, unsigned char AiSampleMode);`

Sets the analog input acquisition mode for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
AiSampleMode	Analog input acquisition mode. When set to 0, it means continuous acquisition; When set to 1, it means limited number of acquisitions, i.e. OneShot mode.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AiConnectType()

`int _stdcall SetUSB3AiConnectType(int DevIndex, unsigned char AiConnectType);`

Sets the analog input wiring method for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
AiConnectType	Analog input wiring mode. When set at 0, it means DIFF input wiring; When set to 1, it means NRSE input wiring.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AiRange()

`int _stdcall SetUSB3AiRange(int DevIndex, unsigned char Chan, float AiRange);`

Sets the range for specified device, or specified analog input channel.

Parameters	Note
------------	------

DevIndex	Device index, counting from 0.
Chan	The number of the channel to set.
AiRange	The range for analog input. When set to 10.24, it means analog input range as $\pm 10.24V$; When set to 5.12, it means analog input range as $\pm 5.12V$; When set to 2.56, it means analog input range as $\pm 2.56V$; When set to 1.28, it means analog input range as $\pm 1.28V$; When set to 0.64, it means analog input range as $\pm 0.64V$.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AiChanSel()

`int _stdcall SetUSB3AiChanSel(int DevIndex, unsigned char Chan, unsigned char SALT);`

Sets whether to enable the specified analog input channel.

Parameters	Note
DevIndex	Device index, counting from 0.
Chan	The number of the channel to set.
Sel	Analog input channel switch. When set to 1, it means to enable the channel; When set to 0, it means to disable the channel.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AiTrigSource()

`int _stdcall SetUSB3AiTrigSource(int DevIndex, unsigned char AiTrigSource);`

Sets the analog input trigger source for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
AiTrigSource	Analog input trigger source. When set to 0, it means AiSoftTrig analog input software triggers; When set to 1, it means Ext_Trig_In external trigger; When set to 2, it means Din_0 rise edge trigger; When set to 3, it means Din_1 rise edge trigger; When set to 4, it means Din_2 rise edge trigger; When set to 5, it means Din_3 rise edge trigger; The setting of 6 is reserved; When set to 7, it means triggering together with Di acquisition; When set to 8, it means triggering together with Do acquisition; When set to 9, it means triggering together with Ao_0;

	When set to 10, it means triggering together with Ao_1; When set to 11, it means triggering together with Ao_2; When set to 12, it means triggering together with Ao_3; When set to 255, it means GlobalSoftTrig global software trigger.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AiConvSource()

`int _stdcall SetUSB3AiConvSource(int DevIndex, unsigned char AiConvSource);`

Set the analog input sampling clock source for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
AiConvSource	The sampled clock source for the analog input. When set to 0, it means Ai internal sampling clock source, which is determined by function setting of SetUSB3AiSampleRate(). When set to 1, it means Conv_In external sampling clock input; The setting of 2 is reserved; When set to 3, it means Di sampling clock source; When set to 4, it means Do sampling clock source; When set to 5, it means Ao_0 sampling clock source; When set to 6, it means Ao_1 sampling clock source; When set to 7, it means Ao_2 sampling clock source; When set to 8, it means Ao_3 sampling clock source.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AiPreTrigPoints()

`int _stdcall SetUSB3AiPreTrigPoints(int DevIndex, unsigned int AiPreTrigPoints);`

Set the specified device's analog input pre-trigger points.

Parameters	Note
DevIndex	Device index, counting from 0.
AiPreTrigPoints	Each selected analog input pre-trigger points. When setting this parameter, be aware the total selected channel points cannot exceed 4095.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AiOneShotPoints()

`int _stdcall SetUSB3AiOneShotPoints(int DevIndex, unsigned int AiOneShotPoints);`

Set the number of analog input acquisition points for the specified device in OneShot mode.

Parameters	Note
DevIndex	Device index, counting from 0.
AiOneShotPoints	The number of analog input acquisition points in OneShot mode.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3ClrAiFifo()

`int _stdcall SetUSB3ClrAiFifo(int DevIndex);`

Empty analog input FIFO cache for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

2.4. Digital I/O setting functions

SetUSB3DiSampleRate()

`int _stdcall SetUSB3DiSampleRate(int DevIndex, unsigned int SamplePeriod);`

Sets the digital input sampling rate for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
SamplePeriod	Sampling period, in ns. The sampling period must be in steps of 10ns.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DiSampleMode()

`int _stdcall SetUSB3DiSampleMode(int DevIndex, unsigned char DiSampleMode);`

Sets the digital input acquisition mode for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
DiSampleMode	Digital input acquisition mode. When set to 0, it means continuous acquisition; When set to 1, it means limited number acquisition, i.e. OneShot mode.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DiTrigSource()

`int _stdcall SetUSB3DiTrigSource(int DevIndex, unsigned char DiTrigSource);`

Sets the digital input trigger source for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
DiTrigSource	Digital input trigger source. When set to 0, it means digital input software trigger DiSoftTrig; When set to 1, it means external trigger Ext_Trig_In; When set to 2, it means rise edge trigger Din_0; When set to 3, it means rise edge trigger Din_1; When set to 4, it means rise edge trigger Din_2; When set to 5, it means rise edge trigger Din_3; When set to 6, it means triggering together with Ai acquisition; The setting of 7 is reserved. When set to 8, it means triggering together with Do acquisition; When set to 9, it means triggering together with Ao_0;

	When set to 10, it means triggering together with Ao_1; When set to 11, it means triggering together with Ao_2; When set to 12, it means triggering together with Ao_3; When set to 255, it means global software trigger GlobalSoftTrig.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DiConvSource()

`int _stdcall SetUSB3DiConvSource(int DevIndex, unsigned char DiConvSource);`

Sets the digital input sampling clock source for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
DiConvSource	DI sampling clock source. When set to 0, it means Di internal sampling clock source, with frequency determined by the function setting of SetUSB3DiSampleRate(). When set to 1, it means Conv_In external sampling clock input; When set to 2, it means Ai sampling clock source; The setting of 3 is reserved; When set to 4, it means Do sampling clock source; When set to 5, it means Ao_0 sampling clock source; When set to 6, it means Ao_1 sampling clock source; When set to 7, it means Ao_2 sampling clock source; When set to 8, it means Ao_3 sampling clock source.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DiPreTrigPoints()

`int _stdcall SetUSB3DiPreTrigPoints(int DevIndex, unsigned int DiPreTrigPoints);`

Set the digital input pre-trigger number for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
DiPreTrigPoints	Digital input pre-trigger points. When setting this parameter, you should be aware that the total number of points cannot exceed 2047.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DiOneShotPoints()

`int _stdcall SetUSB3DiOneShotPoints(int DevIndex, unsigned int DiOneShotPoints);`

Set the number of digital input acquisition points for the specified device in OneShot mode.

Parameters	Note
DevIndex	Device index, counting from 0.
DiOneShotPoints	The number of digital input acquisition points in OneShot mode.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3ClrDiFifo()

`int _stdcall SetUSB3ClrDiFifo(int DevIndex);`

Empty the specified device's DI FIFO cache.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DoSampleRate()

`int _stdcall SetUSB3DoSampleRate(int DevIndex, unsigned int SamplePeriod);`

Set the digital output sampling rate for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
SamplePeriod	Sampling period, in ns. The sampling period must be in steps of 10ns.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DoSampleMode()

`int _stdcall SetUSB3DoSampleMode(int DevIndex, unsigned char DoSampleMode);`

Set the digital input acquisition mode for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
DoSampleMode	DO acquisition mode. When set to 0, it means continuous non-loop output to computer buffer; When set to 1, it means limited number/infinite loop output of hardware FIFO, with the number of cycles determined by the function setting of SetUSB3DoCycle().
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DoTrigSource()

`int _stdcall SetUSB3DoTrigSource(int DevIndex, unsigned char DoTrigSource);`

Set the digital output trigger source for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
DoTrigSource	<p>Digital output trigger source.</p> <p>When set to 0, it means digital output software trigger DoSoftTrig;</p> <p>When set to 1, it means external trigger Ext_Trig_In;</p> <p>When set to 2, it means rise edge trigger Din_0;</p> <p>When set to 3, it means rise edge trigger Din_1;</p> <p>When set to 4, it means rise edge trigger Din_2;</p> <p>When set to 5, it means rise edge trigger Din_3;</p> <p>When set to 6, it means triggering together with Ai acquisition;</p> <p>When set to 7, it means triggering together with Di acquisition;</p> <p>The setting of 8 is reserved;</p> <p>When set to 9, it means triggering together with Ao_0;</p> <p>When set to 10, it means triggering together with Ao_1;</p> <p>When set to 11, it means triggering together with Ao_2;</p> <p>When set to 12, it means triggering together with Ao_3;</p> <p>When set to 255, it means global software trigger GlobalSoftTrig.</p>
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DoConvSource()

`int _stdcall SetUSB3DoConvSource(int DevIndex, unsigned char DoConvSource);`

Set the digital output sampling clock source for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
DoConvSource	<p>Digital output sampling clock source.</p> <p>When set to 0, it means Do internal sampling clock source, with frequency determined by the function setting of SetUSB3DoSampleRate().</p> <p>When set to 1, it means Conv_In external sampling clock input;</p> <p>When set to 2, it means Ai sampling clock source;</p> <p>When set to 3, it means Di sampling clock source;</p> <p>The setting of 4 is reserved;</p> <p>When set to 5, it means Ao_0 sampling clock source;</p> <p>When set to 6, it means Ao_1 sampling clock source;</p> <p>When set to 7, it means Ao_2 sampling clock source;</p> <p>When set to 8, it means Ao_3 sampling clock source.</p>
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DoCycle()

`int _stdcall SetUSB3DoCycle(int DevIndex, unsigned int DoCycle);`

Set the number of digital waveform output loops for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
DoCycle	Number of digital waveform output loops. When set to 0, it means infinite loop output; When set to 1, it means 1 time output; When set to 2, it means 2 times output; and so on.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DoDataFifo()

`int _stdcall SetUSB3DoDataFifo(int DevIndex, unsigned int Value[], unsigned int only);`

Set the digital waveform output FIFO cache data for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Value[]	Array of numeric waveforms.
Len	The analog waveform length to download to hardware FIFO.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3ClrDoFifo()

`int _stdcall SetUSB3ClrDoFifo(int DevIndex);`

Empty the digital waveform output FIFO cache for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DoWaveCtrl()

`int _stdcall SetUSB3DoWaveCtrl(int DevIndex, unsigned int Chan);`

Set the channel control to output waveforms for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Chan	Channel selection. Each bit represents a channel. When set to 0x01, it means only channel Do_0 will output waveforms; When set to 0x03, it means both Do_0 and Do_1 will output

	<p>waveforms;</p> <p>When set to 0x0f, it means Do_0, Do_1, Do_2, and Do_3 will all output waveforms;</p> <p>and so on.</p>
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DoImmediately()

`int _stdcall SetUSB3DoImmediately(int DevIndex, unsigned int Chan, unsigned int Value);`

Set the digital output immediate output state of the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Chan	<p>Channel selection.</p> <p>The setting 0~3 corresponds to Do_0 ~ Do_3;</p> <p>The setting of 0xff means DO_0 ~ Do_3 to be set altogether.</p>
Value	<p>Digital output Do state.</p> <p>When Chan is set to 0~3, Value=1 means high level; Value=0 means low level.</p> <p>When Chan is set to 0xff, the lower 4 bit of Value corresponds to Do_0 ~ Do_3.</p>
Return value	Error code. 0 means no error. For others, refer to error code chapter.

2.5. Analog output setup functions

SetUSB3AoSampleRate()

`int _stdcall SetUSB3AoSampleRate(int DevIndex, unsigned char Chan, unsigned int SamplePeriod);`

Set the sampling rate for the specified analog output channel on the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Chan	Channel number that needs to be set. The setting of 0~3 corresponds to AO_0 ~AO_3
SamplePeriod	Sampling period, in ns. The sampling period must be in steps of 10ns.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AoSampleMode()

`int _stdcall SetUSB3AoSampleMode(int DevIndex, unsigned char Chan, unsigned char AoSampleMode);`

Set the output mode for the specified analog output channel on the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Chan	The number of the channel to set. The setting of 0~3 corresponds to AO_0 ~ Ao_3
AoSampleMode	Analog output acquisition mode. When set to 0, it means continuous non-loop output mode for the computer buffer; When set to 1, it means limited number/infinite output mode, with the loop number determined by the function setting of SetUSB3DoCycle().
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AoTrigSource()

`int _stdcall SetUSB3AoTrigSource(int DevIndex, unsigned char Chan, unsigned char AoTrigSource);`

Set the trigger source for the specified analog output channel on the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Chan	The channel that needs to be set.

AoTrigSource	<p>Analog output trigger source.</p> <p>When set to 0, it means analog output software trigger AoSoftTrig;</p> <p>When set to 1, it means external trigger Ext_Trig_In;</p> <p>When set to 2, it means rise edge trigger Din_0;</p> <p>When set to 3, it means rise edge trigger Din_1;</p> <p>When set to 4, it means rise edge trigger Din_2;</p> <p>When set to 5, it means rise edge trigger Din_3;</p> <p>When set to 6, it means triggering together with Ai acquisition;</p> <p>When set to 7, it means triggering together with Di acquisition;</p> <p>When set to 8, it means triggering together with Do output;</p> <p>When set to 9, it means triggering together with Ao_0;</p> <p>When set to 10, it means triggering together with Ao_1;</p> <p>When set to 11, it means triggering together with Ao_2;</p> <p>When set to 12, it means triggering together with Ao_3;</p> <p>When set to 255, it means global software trigger GlobalSoftTrig.</p>
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AoConvSource()

```
int _stdcall SetUSB3AoConvSource(int DevIndex, unsigned char Chan, unsigned char
AoConvSource);
```

Set the sampling clock source for the specified analog output channel on the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Chan	The channel that needs to be set.
AoConvSource	<p>Analog output sampling clock source.</p> <p>When set to 0, it means Ao internal sampling clock source, with frequency determined by the function setting of SetUSB3DoSampleRate().</p> <p>When set to 1, it means Conv_In external sampling clock input;</p> <p>When set to 2, it means Ai sampling clock source;</p> <p>When set to 3, it means Di sampling clock source;</p> <p>When set to 4, it means Do sampling clock source;</p> <p>When set to 5, it means Ao_0 sampling clock source;</p> <p>When set to 6, it means Ao_1 sampling clock source;</p> <p>When set to 7, it means Ao_2 sampling clock source;</p> <p>When set to 8, it means Ao_3 sampling clock source.</p>
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AoCycle()

`int _stdcall SetUSB3AoCycle(int DevIndex, unsigned char Chan, unsigned int AoCycle);`

Set the number of analog waveform output loops for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Chan	The channel that needs to be set.
AoCycle	The number of analog waveform output loops. When set to 0, it means infinite loop output; When set to 1, it means 1 time output; When set to 2, it means 2 times output; and so on.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AoDataFifo()

`int _stdcall SetUSB3AoDataFifo(int DevIndex, unsigned char Chan, float *Voltage, unsigned int only);`

Set analog waveform output FIFO cache data for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Chan	The number of the channel to set.
*Voltage	Analog waveform array pointer.
Len	The analog waveform length to download to hardware FIFO.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3ClrAoFifo()

`int _stdcall SetUSB3ClrAoFifo(int DevIndex, unsigned char Chan);`

Empty analog waveform output FIFO cache for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Chan	The number of the channel that needs to be emptied.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AoSync()

`int _stdcall SetUSB3AoSync(int DevIndex, unsigned char Chans);`

Set the analog waveform output synchronization channel for the specified device.

Parameters	Note
------------	------

DevIndex	Device index, counting from 0.
Chans	The channel that needs to output synchronization. Each bit represents a channel. When set to 0x03, it means channel Ao_0 and Ao_1 to output synchronization; When set to 0x07, it means Ao_0, Ao_1 and Ao_2 to output synchronization; When set to 0x0f, it means Ao_0, Ao_1, Ao_2, and Ao_3 to output synchronization; and so on.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AoImmediately()

`int _stdcall SetUSB3AoImmediately(int DevIndex, unsigned char Chan, float Voltage);`

Set the specified analog output channel of the specified device to output the voltage immediately.

Parameters	Note
DevIndex	Device index, counting from 0.
Chan	The number of the channel to set.
Voltage	The voltage that needs to be set. Can only be in the range of -10 ~ 10V.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

2.6. Setting Function of Counter

SetCtRunMode()

`int _stdcall SetCtRunMode(int DevIndex, unsigned char Chan, unsigned char CtRunMode)` Set the counter operation mode of the specified device.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
CtRunMode	Working mode of counter. 0: Counting edge 1: Pulse width measurement 2: Pulse measurement 3: Reserve 4: Frequency measurement 5: Quadrature encoder measurement
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtSampleMode()

`int _stdcall SetCtSampleMode(int DevIndex, unsigned char Chan, unsigned char CtSampleMode)`
Set the sampling method of the counter.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
CtSampleMode	Sampling method of counter. 0: Single spot measurement 1: Sampling clock buffer continuous acquisition 2: Implicit buffer acquisition
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtSampleRate()

`int _stdcall SetCtSampleRate(int DevIndex, unsigned char Chan, unsigned int SamplePeriod)`

Set the sampling period when the counter sampling clock buffers acquisition.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
SamplePeriod	Sampling period, in ns, set in 10ns steps.
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtTrigSource()

`int _stdcall SetCtTrigSource(int DevIndex, unsigned char Chan, unsigned char CtTrigSource)`

Set the trigger source of the counter.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
CtTrigSource	Trigger source. Setting 0 means that the counter software triggers; Setting 1 represents external trigger of Ext_Trig_In; Setting 2 represents the rising edge trigger of Din_0; Setting 3 represents the rising edge trigger of Din_1; Setting 4 represents the rising edge trigger of Din_2; Setting 5 represents the rising edge trigger of Din_3; Setting 6 means triggering along with Ai acquisition; Setting 7 means triggering

CtTrigSource	<p>along with Di acquisition; Setting 8 means triggering along with Do output; Setting 9 means triggering along with Ao_0; Setting 10 means triggering along with Ao_1; Setting 11 means triggering along with Ao_2; Setting 12 means triggering along with Ao_3;</p> <p>Setting 255 represents GlobalSoftTrig global software trigger.</p>
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtConvSource()

`int _stdcall SetCtConvSource(int DevIndex, unsigned char Chan, unsigned char CtConvSource)`

Set the counter sampling clock source.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
CtConvSource	<p>Sampling clock source of counter.</p> <p>Setting 0 represents the internal sampling clock source of the counter, and the sampling clock frequency is set by SetCtSampleRate () function.</p> <p>1 setting represents Conv_In external sampling clock input;</p>

	<p>2. Set the clock source representing Ai sampling; 3. Set the clock source representing Di sampling; 4. Set the clock source representing Do sampling;</p> <p>5 sets the sampling clock source for Ao_0; 6 set the sampling clock source for Ao_1; 7 set the sampling clock source for Ao_2; Setting 8 represents the sampling clock source of Ao_3.</p>
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtCountEdge()

`int _stdcall SetCtCountEdge(int DevIndex, unsigned char Chan, unsigned char CtCountEdge)`

When setting the counter counting edge, it is to count the rising edge or falling edge of the finished Src signal.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
CtCountEdge	Select the counting edge. 1: Rising edge 2: Falling edge
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtValue()

`int _stdcall SetCtValue(int DevIndex, unsigned char Chan, unsigned int Value)`

`int _stdcall SetCtValue(int DevIndex, unsigned char Chan, unsigned int Value)` Set the current count value of the counter.

Parameter	Description
-----------	-------------

DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
Value	Current count value.
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtOverflowValueEdgeCount()

`int _stdcall SetCtOverflowValueEdgeCount(int DevIndex, unsigned char Chan, unsigned int OverflowValue)`

Set the overflow value when the counter counting edge.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
OverflowValue	Overflow value.
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtFrontPartValueEdgeCount()

`int _stdcall SetCtFrontPartValueEdgeCount(int DevIndex, unsigned char Chan, unsigned int FrontPartValue)`

Set the count median when counting edge of the counter.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
FrontPartValue	Median count.
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtReloadValueEdgeCount()

`int _stdcall SetCtReloadValueEdgeCount(int DevIndex, unsigned char Chan, unsigned int ReloadValue)`

When setting the counter counting edge, the default value after overflow.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
ReloadValue	Default value after overflow.
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtDirEdgeCount()

`int _stdcall SetCtDirEdgeCount(int DevIndex, unsigned char Chan, unsigned char Dir)`

Set the counting direction of counter counting edge, increase or decrease.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
Dir	Counting direction. 0: Decrease. 1: Increase.
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtPulseWidthStartEdge()

`int _stdcall SetCtPulseWidthStartEdge(int DevIndex, unsigned char Chan, unsigned char StartEdge)`

Set the start edge of counter pulse width measurement. If the set start edge is the rising edge, it means that the positive pulse width is measured; If the setting start edge is the falling edge, it means that the negative pulse width is measured.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
StartEdge	Start edge. 0: Falling edge. 1: Rising edge.
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtInternalSrcPeriod()

`int _stdcall SetCtInternalSrcPeriod(int DevIndex, unsigned char Chan, unsigned int Period)`

Set the internal Src signal period of the counter.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
Period	Internal Src signal period, in ns, set in steps of 10ns.
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtSrcSel()

`int _stdcall SetCtSrcSel(int DevIndex, unsigned char Chan, unsigned char SrcSel)`

Select whether the Src signal source of the counter is internal or external.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
SrcSel	Select Src signal source of counter. 0: Internal Src signal. 1: Src signal on external terminal.
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtEncoderMode()

`int _stdcall SetCtEncoderMode(int DevIndex, unsigned char Chan, unsigned char EncoderMode)`

Set the type of encoder when the counter measures the encoder.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.

Encoder Mode	Encoder type.
	0: Quadrature encoder, X1 1: Quadrature encoder, X2 2: Quadrature encoder, X4 3: Two pulse encoder
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtZPhase()

`int _stdcall SetCtZPhase(int DevIndex, unsigned char Chan, unsigned char ALev, unsigned char BLev)`

Set the starting condition of Z index when the counter measures the encoder.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
ALev	The level of A signal.
BLev	The level of B signal.
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtZValue()

`int _stdcall SetCtZValue(int DevIndex, unsigned char Chan, unsigned int Value)`

When setting the counter to measure the encoder, the measurement value initialized after the Z index is started.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
Value	Measured value initialized after Z index is started.
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtZEnable()

`int _stdcall SetCtZEnable(int DevIndex, unsigned char Chan, unsigned char Enable)`

The z index is enabled when the counter measurement encoder is set.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
Enable	The z index is enabled. 0: Close. 1: Open.
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtSoftTrig()

`int _stdcall SetCtSoftTrig(int DevIndex, unsigned char Chan)`

Set the software trigger of the counter.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
Return value	Error code, 0 means no error, please refer to the error code section for others.

SetCtClrFifo()

`int _stdcall SetCtClrFifo(int DevIndex, unsigned char Chan)`

Clear the Fifo of the counter.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	The channel number to be set.
Return value	Error code, 0 means no error, please refer to the error code section for others.

2.7. Trigger setup functions

SetUSB3AiSoftTrig()

`int _stdcall SetUSB3AiSoftTrig(int DevIndex);`

Set the analog input software trigger for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DiSoftTrig()

`int _stdcall SetUSB3DiSoftTrig(int DevIndex);`

Set the digital input software trigger for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3DoSoftTrig()

`int _stdcall SetUSB3DoSoftTrig(int DevIndex);`

Set the digital output software trigger for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3AoSoftTrig()

`int _stdcall SetUSB3AoSoftTrig(int DevIndex, unsigned char Chan);`

Set the software trigger for the specified analog output channel for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Chan	The number of analog output channel that needs to be triggered by the software.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3GlobalSoftTrig()

`int _stdcall SetUSB3GlobalSoftTrig(int DevIndex);`

Set the global software trigger for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3ClrTrigger()

`int _stdcall SetUSB3ClrTrigger(int DevIndex);`

Clear all trigger flags for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3ClrAiTrigger()

`int _stdcall SetUSB3ClrAiTrigger(int DevIndex);`

Clear the analog input trigger flag for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3ClrDiTrigger()

`int _stdcall SetUSB3ClrDiTrigger(int DevIndex);`

Clear the digital input trigger flag for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3ClrDoTrigger()

`int _stdcall SetUSB3ClrDoTrigger(int DevIndex);`

Clear the digital output trigger flag for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3ClrAoTrigger()

`int _stdcall SetUSB3ClrAoTrigger(int DevIndex, unsigned char Chan);`

Clears the trigger flag for the specified analog output channel for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Chan	The number of the analog output channel that needs to be cleared of the trigger flag.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3ClrGlobalSoftTrig()

`int _stdcall SetUSB3ClrGlobalSoftTrig(int DevIndex);`

Clear the global software trigger flag for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

2.8. Synchronization system setup functions

SetUSB3ExtTrigOutSource()

`int _stdcall SetUSB3ExtTrigOutSource(int DevIndex, unsigned char Source);`

Set signal source of external trigger output port for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Source	The output clock source for external trigger output ports. When set to 0, it means analog input trigger signal; When set to 1, it means digital input trigger signal; When set to 2, it means digital output trigger signal; When set to 3, it means analog output Ao_0 trigger signal; When set to 4, it means analog output Ao_1 trigger signal; When set to 5, it means analog output Ao_2 trigger signal; When set to 6, it means analog output Ao_3 trigger signal.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB3ExtConvOutSource()

`int _stdcall SetUSB3ExtConvOutSource(int DevIndex, unsigned char Source);`

Set the clock source from sampling clock output for the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Source	Clock source from sampling clock output: The setting of 0 is reserved; The setting of 1 is reserved; When set to 2, it means analog input sampling clock; When set to 3, it means digital input sampling clock; When set to 4, it means digital output sampling clock; When set to 5, it means analog output Ao_0 sampling clock; When set to 6, it means analog output Ao_1 sampling clock; When set to 7, it means analog output Ao_2 sampling clock; When set to 8, it means analog output Ao_3 sampling clock.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

2.9. Data read function

USB3GetAi()

`int _stdcall USB3GetAi(int DevIndex, unsigned long Points, float To long TimeOut);`

Load the analog input data collected by the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Points	The number of points per channel that need to be loaded.
*Ai	An array pointer used to store the acquired analog input data.
TimeOut	Timeout, in ms. After the set timeout period is reached, if the acquisition device has not collected enough points of data, the function exits and returns error code -7.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

USB3GetDi()

`int _stdcall USB3GetDi(int DevIndex, unsigned long Points, unsigned char Of long TimeOut);`

Load the digital input data collected by the specified device.

Parameters	Note
DevIndex	Device index, counting from 0.
Points	The number of points per channel that need to be loaded.
*Di	An array pointer used to store the acquired digital input data.
TimeOut	Timeout, in ms. After the set timeout period is reached, if the acquisition device has not collected enough points of data, the function exits and returns error code -7.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

USB3GetCt()

`int _stdcall USB3GetCt(int DevIndex, unsigned char Chan, unsigned long Points, unsigned int *Ct, long TimeOut)`

Read the measured value of the specified counter channel.

	If the counter is set to pulse measurement, the upper 16-bit of this data represents positive pulse width and the lower 16-bit represents negative pulse width.
TimeOut	Timeout time, in ms. When the set timeout is reached, if the acquisition device has not acquired enough data of Points, the function exits and returns error code -7.
Return value	Error code, 0 means no error, please refer to the error code section for others.

GetLatestCtValue()

`int _stdcall GetLatestCtValue(int DevIndex, unsigned char Chan, unsigned int *Value, long TimeOut)`

Read the last measured value of the counter.

Parameter	Description
DevIndex	Device index, counting from 0.
Chan	Specified counter channel
*Value	A pointer used to store the last measured value of the counter. If the counter is set to pulse measurement, the upper 16-bit of this data represents positive pulse width and the lower 16-bit represents negative pulse width.
TimeOut	Timeout time, in ms. When the set timeout is reached, if the acquisition device has not acquired enough data of Points, the function exits and returns error code -7.
Return value	Error code, 0 means no error, please refer to the error code section for others.

3. Error codes

Error code	Note
-1	NO_USBDAQ Cannot find USB series DAQ on your computer. Make sure the driver is correctly installed.
-2	DevIndex_Overflow DevIndex parameters overflow.
-3	Bad _ Firmware Cannot recognize the DAQ card. Try re-plugging the DAQ card.
-4	USBDAQ_Closed The DAQ card is shut down. The function Open_Device() needs to be called correctly.
-5	Transfer_Data_Fail Abnormal communication between the computer and DAQ card. Try re-plugging the DAQ card.
-6	No_Enough_Memory The computer does not have enough free memory.
-7	Time_Out Function timeout.
-8	Undefined, reserved.
-9	ChanIndex_Overflow Channel index overflow.
-10	Undefined_AiRange Undefined analog input range setting.
-11	Undefined_SamplePeriod Undefined sampling period setting.
-12	Undefined_AiConnectType Undefined analog input connection setting.
-13	Undefined_AiSampleMode Undefined analog input acquisition mode.
-14	Undefined_WaveLen Undefined waveform length.