

**Smacq USB-5000 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

1. GETTING STARTED.....	4
1.1. PROGRAMMING LANGUAGE	4
2. FUNCTION DESCRIPTION	5
2.1. FUNCTION TYPE.....	5
2.2. BASIC OPERATING FUNCTIONS	8
<i>FindUSB5DAQ()</i>	8
<i>USB5OpenDevice()</i>	8
<i>USB5CloseDevice()</i>	8
<i>USB5GetDeviceSN()</i>	8
<i>USB5GetDeviceModel()</i>	8
2.3. ANALOG INPUT SETUP FUNCTION	10
<i>SetUSB5AiSampleRate()</i>	10
<i>SetUSB5AiSampleMode()</i>	10
<i>SetUSB5AiConnectType()</i>	10
<i>SetUSB5AiRange()</i>	10
<i>SetUSB5AiChanSel()</i>	11
<i>SetUSB5AiTrigSource()</i>	11
<i>SetUSB5AiConvSource()</i>	12
<i>SetUSB5AiPreTrigPoints()</i>	12
<i>SetUSB5AiOneShotPoints()</i>	12
<i>SetUSB5ClrAiFifo()</i>	13
2.4. DIGITAL I/O SETTING FUNCTIONS	14
<i>SetUSB5DiSampleRate()</i>	14
<i>SetUSB5DiSampleMode()</i>	14
<i>SetUSB5DiTrigSource()</i>	14
<i>SetUSB5DiConvSource()</i>	15
<i>SetUSB5DiPreTrigPoints()</i>	15
<i>SetUSB5DiOneShotPoints()</i>	15
<i>SetUSB5ClrDiFifo()</i>	16
<i>SetUSB5DoSampleRate()</i>	16
<i>SetUSB5DoSampleMode()</i>	16
<i>SetUSB5DoTrigSource()</i>	16
<i>SetUSB5DoConvSource()</i>	17
<i>SetUSB5DoCycle()</i>	18
<i>SetUSB5DoDataFifo()</i>	18
<i>SetUSB5ClrDoFifo()</i>	18
<i>SetUSB5DoWaveCtrl()</i>	18
<i>SetUSB5DoImmediately()</i>	19
2.5. ANALOG OUTPUT SETUP FUNCTIONS	20
<i>SetUSB5AoSampleRate()</i>	20
<i>SetUSB5AoSampleMode()</i>	20
<i>SetUSB5AoTrigSource()</i>	20

<i>SetUSB5AoConvSource()</i>	21
<i>SetUSB5AoCycle()</i>	22
<i>SetUSB5AoDataFifo()</i>	22
<i>SetUSB5ClrAoFifo()</i>	22
<i>SetUSB5AoSync()</i>	22
<i>SetUSB5AoImmediately()</i>	23
2.6. TRIGGER SETUP FUNCTIONS	24
<i>SetUSB5AiSoftTrig()</i>	24
<i>SetUSB5DiSoftTrig()</i>	24
<i>SetUSB5DoSoftTrig()</i>	24
<i>SetUSB5AoSoftTrig()</i>	24
<i>SetUSB5GlobalSoftTrig()</i>	24
<i>SetUSB5ClrTrigger()</i>	25
<i>SetUSB5ClrAiTrigger()</i>	25
<i>SetUSB5ClrDiTrigger()</i>	25
<i>SetUSB5ClrDoTrigger()</i>	25
<i>SetUSB5ClrAoTrigger()</i>	25
<i>SetUSB5ClrGlobalSoftTrig()</i>	26
2.7. SYNCHRONIZATION SYSTEM SETUP FUNCTIONS	27
<i>SetUSB5ExtTrigOutSource()</i>	27
<i>SetUSB5ExtConvOutSource()</i>	27
2.8. DATA READ FUNCTION	28
<i>USB5GetAi()</i>	28
<i>USB5GetDi()</i>	28
3. ERROR CODES	29

1. Getting Started

User application can interact with USB-5000 series data acquisition device via functions provided by USB5000.dll, which is a standard dynamic link library. This manual will introduce all functions provided by USB5000.dll.

This guide covers all models of USB-5xxx 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-5000.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-5000.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 USB5000.dll.

2.1. Function type

The types of functions provided by USB5000.dll are listed below.

- Basic operating functions

Function name	Description
FindUSB5DAQ()	Query the number of connected acquisition devices
USB5OpenDevice()	Turn on the device
USB5CloseDevice()	Turn off the device
USB5GetDeviceSN()	Read the acquisition device serial number
USB5GetDeviceModel()	Read the acquisition device model

- Analog input setup function

Function name	Description
SetUSB5AiSampleRate()	Set analog input sampling rate
SetUSB5AiSampleMode()	Set up analog input acquisition mode
SetUSB5AiConnectType()	Set up analog input wiring mode
SetUSB5AiRange()	Set up analog input range
SetUSB5AiChanSel()	Set the channel you want to select for analog input
SetUSB5AiTrigSource()	Set up an analog input trigger source
SetUSB5AiConvSource()	Set up analog input sampling clock source
SetUSB5AiPreTrigPoints()	Set analog input pre-trigger points
SetUSB5AiOneShotPoints()	Set up analog input collection points when in OneShot mode
SetUSB5ClrAiFifo()	Empty analog input FIFO cache

- Digital I/O setting functions

Function name	Description
SetUSB5DiSampleRate()	Set the digital input sampling rate
SetUSB5DiSampleMode()	Set up digital input acquisition mode
SetUSB5DiTrigSource()	Set up a digital input trigger source
SetUSB5DiConvSource()	Set up a digital input sampling clock source
SetUSB5DiPreTrigPoints()	Set up digital input pre-trigger points
SetUSB5DiOneShotPoints()	Set up digital input collection points when in OneShot mode

SetUSB5ClrDiFifo()	Empty digital input FIFO cache
SetUSB5DoSampleRate()	Set digital output sampling rate
SetUSB5DoSampleMode()	Set up digital output mode
SetUSB5DoTrigSource()	Set up a digital output trigger source
SetUSB5DoConvSource()	Set up a digital output sampling clock source
SetUSB5DoCycle()	Set waveform output loops
SetUSB5DoDataFifo()	Set waveform output FIFO caching data
SetUSB5ClrDoFifo()	Empty waveform output FIFO cache
SetUSB5DoWaveCtrl()	Channel control on whether to output waveform
SetUSB5DoImmediately()	Set digital output to output immediately

- Analog output setup function

Function name	Description
SetUSB5AoSampleRate()	Set analog output sampling rate
SetUSB5AoSampleMode()	Set analog output mode
SetUSB5AoTrigSource()	Set up an analog output trigger source
SetUSB5AoConvSource()	Set up analog output sampling clock source
SetUSB5AoCycle()	Set analog waveform output loops
SetUSB5AoDataFif ()	Set up analog waveform output FIFO cache data
SetUSB5ClrAoFifo()	Empty analog waveform output FIFO cache
SetUSB5AoSync()	Set up analog waveform output synchronization channel
SetUSB5AoImmediately()	Set the analog output voltage value to output immediately

- Trigger Setup function

Function name	Description
SetUSB5AiSoftTrig()	Analog input software triggers
SetUSB5DiSoftTrig()	Digital input software triggers
SetUSB5DoSoftTrig()	Digital output software triggers
SetUSB5AoSoftTrig()	Analog output software triggers
SetUSB5GlobalSoftTrig()	Global software triggers
SetUSB5ClrTrigger()	Empty all trigger states
SetUSB5ClrAiTrigger()	Empty analog input trigger state
SetUSB5ClrDiTrigger()	Empty digital input trigger status
SetUSB5ClrDoTrigger()	Empty digital output trigger state
SetUSB5ClrAoTrigger()	Empty analog output trigger state
SetUSB5ClrGlobalSoftTrig()	Empty global software Trigger state

- Synchronize system Setup Functions

Function name	Description
SetUSB5ExtTrigOutSource()	Set up an external trigger output signal source
SetUSB5ExtConvOutSource()	Set up an external clock output signal source

- Data Read function

Function name	Description
USB5GetAi()	Read analog input data
USB5GetDi()	Read digital input data

2.2. Basic operating functions

FindUSB5DAQ()

`int _stdcall FindUSB5DAQ();`

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

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

USB5OpenDevice()

`int _stdcall USB5OpenDevice(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.

USB5CloseDevice()

`int _stdcall USB5CloseDevice(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.

USB5GetDeviceSN()

`int _stdcall USB5GetDeviceSN(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.

USB5GetDeviceModel()

`int _stdcall USB5GetDeviceModel(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

SetUSB5AiSampleRate()

`int _stdcall SetUSB5AiSampleRate(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.

SetUSB5AiSampleMode()

`int _stdcall SetUSB5AiSampleMode(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.

SetUSB5AiConnectType()

`int _stdcall SetUSB5AiConnectType(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.

SetUSB5AiRange()

`int _stdcall SetUSB5AiRange(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.

SetUSB5AiChanSel()

`int _stdcall SetUSB5AiChanSel(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.

SetUSB5AiTrigSource()

`int _stdcall SetUSB5AiTrigSource(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.

SetUSB5AiConvSource()

`int _stdcall SetUSB5AiConvSource(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 SetUSB5AiSampleRate(). 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.

SetUSB5AiPreTrigPoints()

`int _stdcall SetUSB5AiPreTrigPoints(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.

SetUSB5AiOneShotPoints()

`int _stdcall SetUSB5AiOneShotPoints(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.

SetUSB5ClrAiFifo()

`int _stdcall SetUSB5ClrAiFifo(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

SetUSB5DiSampleRate()

`int _stdcall SetUSB5DiSampleRate(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.

SetUSB5DiSampleMode()

`int _stdcall SetUSB5DiSampleMode(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.

SetUSB5DiTrigSource()

`int _stdcall SetUSB5DiTrigSource(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.

SetUSB5DiConvSource()

`int _stdcall SetUSB5DiConvSource(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 SetUSB5DiSampleRate(). 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.

SetUSB5DiPreTrigPoints()

`int _stdcall SetUSB5DiPreTrigPoints(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.

SetUSB5DiOneShotPoints()

`int _stdcall SetUSB5DiOneShotPoints(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.

SetUSB5ClrDiFifo()

`int _stdcall SetUSB5ClrDiFifo(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.

SetUSB5DoSampleRate()

`int _stdcall SetUSB5DoSampleRate(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.

SetUSB5DoSampleMode()

`int _stdcall SetUSB5DoSampleMode(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 SetUSB5DoCycle().
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB5DoTrigSource()

`int _stdcall SetUSB5DoTrigSource(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.

SetUSB5DoConvSource()

`int _stdcall SetUSB5DoConvSource(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 SetUSB5DoSampleRate().</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.

SetUSB5DoCycle()

`int _stdcall SetUSB5DoCycle(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.

SetUSB5DoDataFifo()

`int _stdcall SetUSB5DoDataFifo(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.

SetUSB5ClrDoFifo()

`int _stdcall SetUSB5ClrDoFifo(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.

SetUSB5DoWaveCtrl()

`int _stdcall SetUSB5DoWaveCtrl(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

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

SetUSB5DoImmediately()

`int _stdcall SetUSB5DoImmediately(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	Channel selection. The setting 0~3 corresponds to Do_0 ~ Do_3; The setting of 0xff means DO_0 ~ Do_3 to be set altogether.
Value	Digital output Do state. When Chan is set to 0~3, Value=1 means high level; Value=0 means low level. When Chan is set to 0xff, the lower 4 bit of Value corresponds to Do_0 ~ Do_3.
Return value	Error code. 0 means no error. For others, refer to error code chapter.

2.5. Analog output setup functions

SetUSB5AoSampleRate()

`int _stdcall SetUSB5AoSampleRate(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.

SetUSB5AoSampleMode()

`int _stdcall SetUSB5AoSampleMode(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 SetUSB5DoCycle().
Return value	Error code. 0 means no error. For others, refer to error code chapter.

SetUSB5AoTrigSource()

`int _stdcall SetUSB5AoTrigSource(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.

SetUSB5AoConvSource()

`int _stdcall SetUSB5AoConvSource(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 SetUSB5DoSampleRate().</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.

SetUSB5AoCycle()

`int _stdcall SetUSB5AoCycle(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.

SetUSB5AoDataFifo()

`int _stdcall SetUSB5AoDataFifo(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.

SetUSB5ClrAoFifo()

`int _stdcall SetUSB5ClrAoFifo(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.

SetUSB5AoSync()

`int _stdcall SetUSB5AoSync(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.

SetUSB5AoImmediately()

`int _stdcall SetUSB5AoImmediately(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. Trigger setup functions

SetUSB5AiSoftTrig()

`int _stdcall SetUSB5AiSoftTrig(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.

SetUSB5DiSoftTrig()

`int _stdcall SetUSB5DiSoftTrig(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.

SetUSB5DoSoftTrig()

`int _stdcall SetUSB5DoSoftTrig(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.

SetUSB5AoSoftTrig()

`int _stdcall SetUSB5AoSoftTrig(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.

SetUSB5GlobalSoftTrig()

`int _stdcall SetUSB5GlobalSoftTrig(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.

SetUSB5ClrTrigger()

`int _stdcall SetUSB5ClrTrigger(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.

SetUSB5ClrAiTrigger()

`int _stdcall SetUSB5ClrAiTrigger(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.

SetUSB5ClrDiTrigger()

`int _stdcall SetUSB5ClrDiTrigger(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.

SetUSB5ClrDoTrigger()

`int _stdcall SetUSB5ClrDoTrigger(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.

SetUSB5ClrAoTrigger()

`int _stdcall SetUSB5ClrAoTrigger(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.

SetUSB5ClrGlobalSoftTrig()

`int _stdcall SetUSB5ClrGlobalSoftTrig(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.

2.7. Synchronization system setup functions

SetUSB5ExtTrigOutSource()

`int _stdcall SetUSB5ExtTrigOutSource(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.

SetUSB5ExtConvOutSource()

`int _stdcall SetUSB5ExtConvOutSource(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.

2.8. Data read function

USB5GetAi()

`int _stdcall USB5GetAi(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.

USB5GetDi()

`int _stdcall USB5GetDi(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.

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.