



FUNCTION REFERENCE

DLL for Monochromator 7ISW Series

Table of Contents

Properties	3
1. Port	3
2. ConnectStatus	3
3. Busy	3
4. Language	3
5. CurrentSpeed	3
6. SpectrometerType	4
7. SpectrometerNo	4
8. CurrentWaveLen	4
9. CurrentGrating	4
10. GratingCount	4
11. GratingTotalStep	4
12. DoubleOutType	5
13. DoubleOutEnable	5
14. DoubleOutCurrentOut	5
15. DoubleOutSwitchStatus	5
16. DoubleOutSwitchWaveLen	5
17. DoubleOutStartStep	6
18. DoubleOutCurrentStep	6
19. FilterStatus	6
20. FilterCount	6
21. FilterUseCount	6
22. FilterZeroStep	7
23. FilterTotalStep	7
Functions	7
1. ConnectPort	7
2. ClosePort	7
3. GetGratingExist	8
4. GetGratingZeroStep	8
5. GetGratingReticleCount	8
6. GetGratingEmendation	9
7. GetGratingBlazeWaveLen	9
8. GetFilterWorkWaveLen	9
9. GetMaxEffiWaveLen	10
10. RunToWaveLen	10
11. RunToStep	10
12. RunToZero	11
13. RunDoubleOutStep	11
14. StopRun	11
15. StepToWaveLength	12
16. SetGratingSwitchParam	12

17.	SetDoubleOutEnable.....	12
18.	SetDoubleOutCurrentOut.....	13
19.	SetDoubleOutSwitchStatus.....	13
20.	SetDoubleOutSwitchWaveLen.....	13
21.	SetFilterStatus	14
22.	SetFilterWorkParam	14
23.	SaveAllParameter.....	14
24.	OpenAllParameter.....	15

Properties

1. Port

Definition: short Port

Description: Read only. The port number connected successfully. It can be changed by function ConnectPort.

C# Code:

```
short sPort;  
sPort = ISWU1.Port;
```

2. ConnectStatus

Definition: bool ConnectStatus

Description: Read only. The connection status between the monochromator and computer. It can be changed by the function ConnectPort and ClosePort.

C# Code:

```
bool bConnect;  
bConnect = ISWU1.ConnectStatus;
```

3. Busy

Definition: bool Busy

Description: Read only. Get the busy status of the monochromator.

C# Code:

```
bool bBusy;  
bBusy = ISWU1.Busy;
```

4. Language

Definition: short Language

Description: Read and write. Get or set the current language.

C# Code:

```
short sLanguage;  
sLanguage = ISWU1.Language;
```

5. CurrentSpeed

Definition: short CurrentSpeed

Description: Read and write. Get or set the current speed value.

C# Code:

```
short sCurrentSpeed;  
sCurrentSpeed = ISWU1.CurrentSpeed;
```

6. SpectrometerType

Definition: string SpectrometerType

Description: Read only. Get the part number of the monochromator.

C# Code:

```
string sSpectrometerType;  
sSpectrometerType = ISWU1.SpectrometerType;
```

7. SpectrometerNo

Definition: string SpectrometerNo

Description: Read only. Get the serial number of the monochromator.

C# Code:

```
string sSpectrometerNo;  
sSpectrometerNo = ISWU1.SpectrometerNo;
```

8. CurrentWaveLen

Definition: double CurrentWaveLen

Description: Read only. Get the current wavelength.

C# Code:

```
double dCurrentWaveLen;  
dCurrentWaveLen = ISWU1.CurrentWaveLen;
```

9. CurrentGrating

Definition: short CurrentGrating

Description: Read and write. Get or change the number of the current grating. 1 is the NO.1 grating, and so on.

C# Code:

```
short sCurrentGrating;  
sCurrentGrating = ISWU1.CurrentGrating;
```

10. GratingCount

Definition: short GratingCount

Description: Read only. Get the number of the gratings.

C# Code:

```
short sGratingCount;  
sGratingCount = ISWU1.GratingCount;
```

11. GratingTotalStep

Definition: long GratingTotalStep

Description: Read only. Get the total step of the grating.

C# Code:

```
long IGratingTotalStep;  
IGratingTotalStep = ISWU1.GratingTotalStep;
```

12. DoubleOutType

Definition: short DoubleOutType

Description: Read only. Get the type of output ports. 0: single output port. 1: two motorized output ports. 2: two manual output ports.

C# Code:

```
short sDoubleOutType;  
sDoubleOutType = ISWU1.DoubleOutType;
```

13. DoubleOutEnable

Definition: short DoubleOutEnable

Description: Read only. Get the status of the motorized dual output port enabled. 0: unenabled, 1: enabled.

C# Code:

```
short sDoubleOutEnable;  
sDoubleOutEnable = ISWU1.DoubleOutEnable;
```

14. DoubleOutCurrentOut

Definition: short DoubleOutCurrentOut

Description: Read only. Get the current number of output ports. 0: NO.0 output port, 1: NO.1 output port.

C# Code:

```
short sDoubleOutCurrentOut;  
sDoubleOutCurrentOut = ISWU1.DoubleOutCurrentOut;
```

15. DoubleOutSwitchStatus

Definition: short DoubleOutSwitchStatus

Description: Read only. Get the status of automatic dual output port switching. 0: not automatic switching, 1: automatic switching.

C# Code:

```
short sDoubleOutSwitchStatus;  
sDoubleOutSwitchStatus = ISWU1.DoubleOutSwitchStatus;
```

16. DoubleOutSwitchWaveLen

Definition: double DoubleOutSwitchWaveLen

Description: Read only. Get the wavelength when switching the dual output port.

C# Code:

```
double dDoubleOutSwitchWaveLen;  
dDoubleOutSwitchWaveLen = ISWU1.DoubleOutSwitchWaveLen;
```

17. DoubleOutStartStep

Definition: long DoubleOutStartStep

Description: Read only. Get the starting position of the NO.1 output.

C# Code:

```
long lDoubleOutStartStep;  
lDoubleOutStartStep = ISWU1.DoubleOutStartStep;
```

18. DoubleOutCurrentStep

Definition: long DoubleOutCurrentStep

Description: Read only. Get the current position of the dual output.

C# Code:

```
long lDoubleOutCurrentStep;  
lDoubleOutCurrentStep = ISWU1.DoubleOutStartStep;
```

19. FilterStatus

Definition: short FilterStatus

Description: Read only. Get the status of the filter wheel. 2: no set, 0: unenabled, 1: enabled.

C# Code:

```
short sFilterStatus;  
sFilterStatus = ISWU1.FilterStatus;
```

20. FilterCount

Definition: short FilterCount

Description: Read only. Get the number of filters available.

C# Code:

```
short sFilterCount;  
sFilterCount = ISWU1.FilterCount;
```

21. FilterUseCount

Definition: short FilterUseCount

Description: Read only. Get the actual number of filters used. It cannot be greater than the number of filters available for installation.

C# Code:

```
short sFilterUseCount;  
sFilterUseCount = ISWU1.FilterUseCount;
```

22. FilterZeroStep

Definition: int FilterZeroStep

Description: Read only. Get the zero position of the filter wheels.

C# Code:

```
int iFilterZeroStep;  
iFilterZeroStep = ISWU1.FilterZeroStep;
```

23. FilterTotalStep

Definition: long FilterTotalStep

Description: Read only. Get the total steps of the filter wheels.

C# Code:

```
long lFilterTotalStep;  
lFilterTotalStep = ISWU1.FilterTotalStep;
```

Functions

1. ConnectPort

Definition: void ConnectPort(short sPort)

Description: Connect the monochromator with the specified serial port number and read the parameters.

Parameters:

sPort: serial port number

Return Value:

None

C# Code:

```
short sPort=3;  
ISWU1.ConnectPort(sPort);    //connect the monochromator using COM3 serial port
```

2. ClosePort

Definition: void ClosePort()

Description: Close the connection.

Parameters:

None

Return Value:

None

C# Code:

```
ISWU1.ClosePort();    //close the connection
```


3. GetGratingExist

Definition: short GetGratingExist(short sIndex, ref bool bGratingExist)

Description: Get whether the specified grating exists.

Parameters:

sIndex: grating number (from 1 to total grating number)

bGratingExist: return whether the specified grating exists.

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
bool bExist=false;  
iRet = ISWU1.GetGratingExist(1, ref bExist); //get whether the NO.1 grating exists
```

4. GetGratingZeroStep

Definition: short GetGratingZeroStep(short sIndex, ref long lZeroStep)

Description: Get the zero position of the specified grating.

Parameters:

sIndex: grating number (from 1 to total grating number)

lZeroStep: return the zero position of the specified grating

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
long lZeroStep =0;  
iRet = ISWU1.GetGratingZeroStep(1, ref lZeroStep); //get the zero position of the NO.1  
grating
```

5. GetGratingReticleCount

Definition: short GetGratingReticleCount(short sIndex, ref int iReticleCount)

Description: Get the number of grating lines of the specified grating.

Parameters:

sIndex: grating number (from 1 to total grating number)

iReticleCount: return the number of grating lines of the specified grating

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
int iReticleCount =0;  
iRet = ISWU1.GetGratingReticleCount(1, ref iReticleCount);    //get the number of grating  
lines of NO.1 grating
```

6. GetGratingEmendation

Definition: short GetGratingEmendation(short sIndex, ref double dEmendation)

Description: Get the calibration factor of the specified grating.

Parameters:

sIndex: grating number (from 1 to total grating number)

dEmendation: return the calibration factor of the specified grating

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
double dEmendation =0;  
iRet = ISWU1.GetGratingEmendation(1, ref dEmendation);    //get the calibration factor of the  
NO.1 grating
```

7. GetGratingBlazeWaveLen

Definition: short GetGratingBlazeWaveLen(short sIndex, ref int iBlazeWaveLen)

Description: Get the blaze wavelength of the specified grating.

Parameters:

sIndex: grating number (from 1 to total grating number)

iBlazeWaveLen: return the blaze wavelength of the specified grating

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
int iBlazeWaveLen =0;  
iRet = ISWU1.GetGratingBlazeWaveLen(1, ref iBlazeWaveLen);    //get the blaze wavelength of  
the NO.1 grating
```

8. GetFilterWorkWaveLen

Definition: short GetFilterWorkWaveLen(short sIndex, ref double dWorkWavelen)

Description: Get the wavelength of the specified filter.

Parameters:

sIndex: filter number (from 1 to actual filter numbers **FilterUseCount**)

dWorkWavelen: return the wavelength of the specified filter

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
double dWorkWavelen =0;  
iRet = ISWU1.GetFilterWorkWavelen(1, ref dWorkWavelen);    //get the wavelength of the  
specified filter
```

9. GetMaxEffiWaveLen

Definition: short GetMaxEffiWaveLen(short sGrating, ref double dMaxEffiWavelen)

Description: Get the maximum effective wavelength of the specified grating.

Parameters:

sGrating: grating number (from 1 to total grating number)

dMaxEffiWavelen: return the maximum effective wavelength of the specified grating

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
double dMaxEffiWavelen =0;  
iRet = ISWU1.GetMaxEffiWaveLen(3, ref dMaxEffiWavelen);    //get the maximum effective  
wavelength of the NO.3 grating
```

10. RunToWaveLen

Definition: short RunToWaveLen(double dWaveLen)

Description: Move to the specified wavelength

Parameters:

dWaveLen: the wavelength that will move to

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
iRet = ISWU1.RunToWaveLen(100);    //move the current grating to 10nm
```

11. RunToStep

Definition: short RunToStep(int iStep)

Description: Move to the specified steps

Parameters:

iStep: the steps that will move to

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
iRet = ISWU1.RunToStep(1000);    //move the current grating to 1000 steps
```

12. RunToZero

Definition: short RunToZero()

Description: Return to mechanical zero position.

Parameters:

None

Return Value:

-1: false, 1: true

C# Code:

```
ISWU1.RunToZero();    //return to mechanical zero position
```

13. RunDoubleOutStep

Definition: short RunDoubleOutStep(long IStep)

Description: Move the dual output ports to the specified steps

Parameters:

IStep: the steps that will move to

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
iRet = ISWU1.RunDoubleOutStep(1000);    //move the dual output ports to 1000 steps
```

14. StopRun

Definition: short StopRun()

Description: Stop the gratings running (Do not use this function when the monochromator is homing or switching the gratings.)

Parameters:

None

Return Value:

-1: false, 1: true

C# Code:

```
ISWU1.StopRun();    //stop running
```

15. StepToWaveLength

Definition: short StepToWaveLength(short sGrating, long lStep, ref double dWaveLen)

Description: Convert the specified steps to the wavelength.

Parameters:

sGrating: grating number (from 1 to total grating number)

lStep: the steps to convert

dWaveLen: return the converted wavelength

Return Value:

-1: false, 1: true

C# Code:

```
short mRet;  
double mWaveLen=0;  
mRet =ISWU1. StepToWaveLength (1,1000,ref mWaveLen);    //return the corresponding  
wavelength of 1000 steps of NO.1 grating.
```

16. SetGratingSwitchParam

Definition: short SetGratingSwitchParam(short sStartPosiMode, double[] dSwitchWaveLen, short sStartGratingID, double dStartWaveLen)

Description: Set the initial position mode, grating, and wavelength, as well as the switching wavelengths for each grating.

Parameters:

sStartPosiMode: initial position mode (0: the wavelength before the last shutdown, 1: the specified grating and wavelength)

dSwitchWaveLen: move to this wavelength after switching gratings. It's an array list containing the switching wavelengths for all gratings.

sStartGratingID: initial grating number (1,2, or 3)

dStartWaveLen: initial wavelength

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
double[] dSwitchWaveLen=new double[3];  
dSwitchWaveLen[0]=0; dSwitchWaveLen[1]=500; dSwitchWaveLen[2]=1000;  
iRet =ISWU1.SetGratingSwitchParam(0, dSwitchWaveLen,1,0);    //set the grating to move  
to the wavelength of the last shutdown, set the position of 0nm after switching to NO.1 grating,  
500nm after switching to NO.2 grating, and 1000nm after switching to NO.3 grating.
```

```
iRet =ISWU1.SetGratingSwitchParam(1, dSwitchWaveLen,1,0);    //set the grating to move  
to the 0nm wavelength of NO.1 grating, set the position of 0nm after switching to NO.1 grating,  
500nm after switching to NO.2 grating, and 1000nm after switching to NO.3 grating.
```

17. SetDoubleOutEnable

Definition: short SetDoubleOutEnable(short sEnable)

Description: Set whether to enable the motorized dual output ports.

Parameters:

sEnable: 0: disable, 1: enable

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
iRet = ISWU1.SetDoubleOutEnable(1);    //set to enable the motorized dual output ports
```

18. SetDoubleOutCurrentOut

Definition: short SetDoubleOutCurrentOut(short sOut)

Description: Switch to the specified output port.

Parameters:

sOut: 0: NO.0 output port, 1: NO.1 output port

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
iRet = ISWU1.SetDoubleOutCurrentOut(0);    //switch to NO.0 output port
```

19. SetDoubleOutSwitchStatus

Definition: short SetDoubleOutSwitchStatus(short sStatus)

Description: Set whether automatic switching of the dual output ports.

Parameters:

sStatus: 0: not automatic switching, 1: automatic switching

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
iRet = ISWU1.SetDoubleOutSwitchStatus(1);    //set automatic switching
```

20. SetDoubleOutSwitchWaveLen

Definition: short SetDoubleOutSwitchWaveLen(double dWaveLen)

Description: Set the switching wavelength of dual output ports.

Parameters:

dWaveLen: switching wavelength

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;
```

```
iRet =ISWU1.SetDoubleOutSwitchWaveLen(100);    //switching the output ports when the  
grating moves to 100nm
```

21. SetFilterStatus

Definition: short SetFilterStatus(short sStatus)

Description: Set whether to enable the filter wheels.

Parameters:

sStatus: 0: disable, 1: enable

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
iRet =ISWU1.SetFilterStatus(1);    //enable the filter wheels
```

22. SetFilterWorkParam

Definition: short SetFilterWorkParam(short sFilterUseCount,double[] dFilterWorkWaveLen)

Description: Set the number of filter wheels used and the working wavelength for each filter.

The wavelengths must be ordered from smallest to largest.

Parameters:

sFilterUseCount: the quantity of filter wheels used, should be less than the installation quantity.

dFilterWorkWaveLen: the working wavelengths for each filter

Return Value:

-1: false, 1: true

C# Code:

```
short iRet;  
double[]dWaveLen =new double[6];  
dWaveLen [1]=40;           //use NO.1 filter when wavelength exceeds 40nm  
dWaveLen [2]=50;           //use NO.2 filter when wavelength exceeds 50nm  
dWaveLen [3]=60;           //use NO.3 filter when wavelength exceeds 60nm  
iRet =ISWU1.SetFilterWorkParam(3, dWaveLen);    //set to use 3 filters and the working  
wavelengths of each filter
```

23. SaveAllParameter

Definition: short SaveAllParameter(string sFilePath)

Description: Save all the parameters to the specified file.

Parameters:

sFilePath: the file's path and name.

Return Value:

-1: false, 1: true

C# Code:

```
short sSaveAllParameter;
```

```
sSaveAllParameter=ISWU1.SaveAllParameter ("C:\\1.txt");    //save all the parameters to  
C:\\1.txt
```

24. OpenAllParameter

Definition: short OpenAllParameter(string sFilePath)

Description: Read all the parameters from the specified file and set them to the monochromator.

Parameters:

sFilePath: the file's path and name.

Return Value:

-1: false, 1: true

C# Code:

```
short sOpenAllParameter;  
sOpenAllParameter=ISWU1.OpenAllParameter ("C:\\1.txt");    //read all the parameters  
from C:\\1.txt and set them to the monochromator
```