Omni-λActiveX control

（ZolixOmniSpec-V2.4.0.13-2011-07-07-YL-Liu）

Operation manual

Zolix Instruments Co., Ltd.

Thanks for using Zolix products ！

Please read this introduction carefully before using. And kindly please contact Zolix office in Beijing or local office if you have any questions and suggestions.

Directory

1. Description

ZolixOmniSpec is designed for omni-λserial monochromators no matter that they have the RS232 or USB communicating port.

Before using the ZolixOmniSpec, please set up the driver of the RS232 or USB port of the CCD system.

Use Process:

- SetUsbMode (TRUE or FALSE);   //TRUE->USB, FALSE->RS232.
- USB:GetUsbDevice()->SetUsbSerial()->Connect()->…
- RS232::SetCommPort()->Connect()->…

Note:

The Reload function should be called at least once before the Get members of the function group containing Reload are called. That is in order to renew the ActiveX control data. You can call all Reload functions through calling the initialize function.

2. CZolixOmniSpec

2.1 Characteristic

2.1.1 short Comport

RS232 communicating port number

2.1.2 CString USBSerials

USB communicating port serials number

2.1.3 BOOL USBMode

The mode of the USB communicating port, it is TRUE means USB mode and FALSE means RS232 mode.

2.1.4 BOOL GetIsOpen

Whether the port is opened or not, is connected or not.

2.1.5 CString GetSystemManuFacturer

The name of the manufacturer and the character length is less than 8.

2.1.6 CString GetSystemModel

The model of the instruments, include Omni-λ150, Omni-λ300, Omni-λ500, Omni-λ750, and SSM300.

2.1.7 CString GetSystemSerialsNumber

The license serial number and the character length is less than 10.

2.1.8 CString GetSystemFactoryTime

The leave factory date of the instrument

2.1.9 CString GetFirmwareVersion

The firmware version

2.1.10 double GetCurrentWave

Current wavelength

2.1.11 BOOL GetSynchroMode

Synchronization mode

TRUE: The ActiveX control wait returns after send the command.

FALSE: don’t wait returns.

2.2 Operation method

2.2.1 CZolixOmniSpec:: Open

BOOL Open ();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Call this function in order that the instrument connects with the computer through RS232 or USB communicating port.

2.2.2 CZolixOmniSpec:: Close

void Close ();

Return: none.

Explain: disconnect the instrument and the computer.

2.2.3 CZolixOmniSpec::Connect

BOOL Connect();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: None**.**

Explain: Call this function in order to that the instrument connects with the computer through RS232 or USB communication port**.**

2.2.4 CZolixOmniSpec::DisConnect

void DisConnect();

Rerurn: none.

Parameter:None.

Explain: Disconnects the instrument and the computer .

2.2.5 CZolixOmniSpec:: SearchZolixUSBDevice

long SearchZolixUSBDevice();

Return: The amount of equipments that have the USB port.

Parameter: None.

Explain: The amount of the USB devices that have been connected to the computer but not used. This function can be called together with GetZolixUSBSerials and SetUSBSerilsSet to set the USB connection objects for the Zolixdccd ActiveX.

2.2.6 CZolixOmniSpec:: GetZolixUSBSerial

CString GetZolixUSBSerial (long DeviceIndex);

Return: The identifier of the equipment which has the USB communicating port.

Parameters:

DeviceIndex: The number of the equipment and its range if from 0 to n-1. n is the return of function SearchZolixUSBDevice.

Explain: This function can be called together with SearchZolixUSBDeviced and SetUSBSerials to set the USB connection objects for the Zolixdccd ActiveX.

2.2.7 CZolixOmniSpec::GetCurrentGrating

short GetCurrentGrating();

Return: CurrentGrating Number, Maximum value is 3.

Explain: Get the number of current Grating, its range from1 to 3.

2.2.8 CZolixOmniSpec::SetCurrentGrating

BOOL SetCurrtentGrating(short CurrentGrating);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Paramerter:

CurrentGrating : New Grating Number，maximum value is 3.

Explain: Call this function to switch to a new grating, nGrating value depended on system parameters, the current maximum of 3. After calling this function, the spectrometer will be moved to the appropriate positioning of a grating location, the location of a user calls the function given SetInitWave() to set free.

2.2.9 CZolixOmniSpec::GratingHome

BOLL GratingHome();

Rerurn: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: None

Explain: Call this function for rester re mechanical reset, reset and run automatically after the user call SetInitWave() to set the initial positioning of the location, the default value is 1,the grating 0nm position.

2.2.10 CZolixOmniSpec:: MoveToWave

BOOL MoveToWave (double wavelength);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameters: Wavelength is the aim wavelength and its unit is nm.

Explain: Control the grating to move to the appointed position through calling this function. If you set the wavelength is out range, the return is FALSE.

2.2.11 CZolixOmniSpec:: MoveWave

BOOL MoveWave (double wavelength);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: wavelength is the change of wavelength and its unit is nm.

Explain: Control the grating to move a distance through calling this function.

If you set the wavelength is out range, the return is FALSE.

2.2.12 CZolixOmniSpec::RefreshCurrentWave

BOOL RefreshCurrentWave ();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Refresh the current wavelength. The properties of control corresponding CurrentWave.

2.2.13 CZolixOmniSpec:: Stop

BOOL Stop();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain:

Call this function to stop the grating moving after call MoveTo() or MoveRel(), But cannot stop the grating during reset or changing the grating.

2.2.14 CZolixOmniSpec:: SetSpeed

BOOL SetSpeed(LONG Index,LONG nSpeed)；

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

Index: grating number, maximum of 3.

nSpeed: is the new moving speed of the grating and its unit is nm/s.

Explain: Change the moving speed of the grating.

Note: The maximum speed range of the grating can get from calculating the grating parameters.

2.2.15 CZolixOmniSpec:: GetSpeed

BOOL GetSpeed(LONG Index, LONG* pnSpeed)；

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

Index grating number, maximum value is 3.

***pnSpeed** is the pointer at the moving speed of the grating and its unit is nm/s。

Explain: Get the moving speed of the grating through calling this function.

2.2.16 CZolixOmniSpec:: SetFilter

BOOL SetFilter(short nFilter);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: nFilter is the serial number of the filter and its value is from 1 to 6.

Explain: Select filter through calling this function. The motor running to the position of nFilter motor by controling the wheel. The account of filter wheel is fixed to 6.

2.2.17 CZolixOmniSpec:: GetFilter

BOOL GetFilter (short* pFilter);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: *pFilter is the pointer at the serial number of the filter, its value range from 1to 6.

Explain: Get the serial number of the filter through calling this function.

2.2.18 CZolixOmniSpec:: FilterHome

BOOL FilterHome ();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Reset the filter wheel.

2.2.19 CZolixOmniSpec:: SetExitPort

BOOL SetExitPort (short nPort);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nPort=0 indicate that the straight slit is the light exit.

nPort=1 indicate that the side slit is the light exit.

Explain: Set the spectrometer light path exit , the straight slit and the side slit.

2.2.20 CZolixOmniSpec:: GetExitPort

BOOL GetExitPort (short*pnPort);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: pnPort is the pointer at the light exit, which value is 0 or 1.

2.2.21 CZolixOmniSpec:: SetExitSidePos

BOOL SetExitSidePos (long nPos);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: nPos is the position of the reflect mirror and its unit is step.

Explain: Getting the position of the side path. Straight and side. Straight position can reset.

2.2.22 CZolixOmniSpec:: GetExitSidePos

BOOL GetExitSidePos (long* pnPos);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: ***pnPos**is the pointer at the position of the reflect mirror.

Explain: See setexitsidepos.

2.2.23 CZolixOmniSpec:: GetPortInput

BOOL GetPortInput(short* nValue);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: *nValue is the pointer at the port status and its value is from 0 to 255.

Explain: The user I/O port has 8bit. The higher four bits is output and lower four bits is input.

0 is the TTL lower level and 1 is TTL higher level.

D7	D6	D5	D4	D3	D2	D1	D0
OUT3	OUT2	OUT1	OUT0	IN3	IN2	IN1	IN0

2.2.24 CZolixOmniSpec:: SetPortOutput

BOOL SetPortOutput (short nValue);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: nValue is the port status and its value is from 0 to 7.

Explain: Set the voltage level of the output port through calling this function.

Example: SetPortOutput (2):

The output port will be the following:

				OUT3	OUT2	OUT1	OUT0
				0	0	1	0

2.2.25 CZolixOmniSpec:: ShowPortTest

void ShowPortTest();

Explain: To display the dialog box for input or output.

2.2.26 CZolixOmniSpec:: ReloadPeripheral

BOOL ReloadPeripheral();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Read parameters of the peripheral equipment from the instrument.

Call the Reload function at least once before call the get- function.

2.2.27 CZolixOmniSpec:: GetPeripheralCount

short GetPeripheralCount();

Return: The amount of peripheral equipments

Explain: Get the amount of peripheral equipments controlled by the monochromator, update the conctol data.

2.2.28 CZolixOmniSpec:: SetPeripheral

BOOL SetPeripheral(short Index, LPCTSTR PeripheralName);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

Index is the serial number of the peripheral equipment.

PeripheralName is the ab. name of the peripheral equipment.

Example: FLT (Filter wheel), TMX, LPA

Explain: Add external device.

Read parameters of the peripheral equipment from the instrument.

Call the Reload function at least once before call the get- function.

2.2.29 CZolixOmniSpec:: AddPeripheral

BOOL AddPeripheral (LPCTSTR PeripheralName);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: PeripheralName is the ab. name of the peripheral equipment.

Example: FLT (Filter wheel), TMX, LPA

Explain: Add new peripheral equipments

2.2.30 CZolixOmniSpec:: GetPeripheralName

CString GetPeripheralName (short Index);

Return: The name of the peripheral equipment.

Parameter: Index is the serial number of the peripheral equipment.

Explain: To get the peripheral equipment name.

See: SetPeripheral, GetPeripheralCount, ReloadPeripheral

2.2.31 CZolixOmniSpec:: SetPeripheralString

BOOL SetPeripheralString (LPCTSTR PeripheralString);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: PeripheralString is the string of the peripheral equipment.

Explain: SetPeripheralString.

2.2.32 CZolixOmniSpec:: MotorHome

BOOL MotorHome (short MotorIndex);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: MotorIndex is the number of the motor.

Explain: To reset the appointed motor through calling this function.

2.2.33 CZolixOmniSpec:: MoveMotorTo

BOOL MoveMotorTo (short MotorIndex, long steps);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

MotorIndex is the number of the motor and its value is from 3 to 9.

Steps is the new position of the appointed motor.

Note:

In Omni-λsystem, the motor for gratings is No. 1 motor and for the filter wheel is No. 2 motor. There are nine motors total. And the filter wheel and the grating have the special control function.

2.2.34 CZolixOmniSpec:: GetMotorCurPos

BOOL GetMotorCurPos (short MotorIndex, long* CurSteps);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

MotorIndex is the number of the motor and its value is from 3 to 9.

***CurSteps** is the pointer at the current step.

Explain: Get the current position of the appointed motor through calling this function.

2.2.35 CZolixOmniSpec:: SetMotorSpeed

BOOL SetMotorSpeed(short nMotorIndex, long nSpeed);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nMotorIndex is the number of the motor and its value is from 3 to 9.

nSpeed is the speed of the motor which value is from 0 to 2000.

Note:

The running direction of the motor cannot be stored in the monochromator, so you must set it every time.

2.2.36 CZolixOmniSpec:: GetMotorSpeed

BOOL GetMotorSpeed(short nMotorIndex, long* pnSpeed);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nMotorIndex is the number of the motor and its value is from 3 to 9.

***nSpeed** is the pointer at the speed of the motor which value is from 0 to 2000.

Note:

The running direction of the motor cannot be stored in the monochromator, so you must set it every time.

2.2.37 CZolixOmniSpec:: SetMotorHomeDirection

BOOL SetMotorHomeDirection(short nMotorIndex, short nDirection);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nMotorIndex is the number of the motor and its value is from 3 to 9.

nDirection is the direction of the motor running and its value is 0 or 1.

Note:

If you let nDirection =0 and the motor run clockwise, nDirection =1 means that the motor run anticlockwise.

If you let nDirection =1 and the motor run clockwise, nDirection =0 means that the motor run anticlockwise.

2.2.38 CZolixOmniSpec:: GetMotorHomeDirection

BOOL GetMotorHomeDirection(short nMotorIndex, short* pnDirection);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nMotorIndex is the number of the motor and its value is from 3 to 9.

***pnDirection** is the pointer at the direction of the motor running and its value is 0 or 1.

Note:

If you let nDirection =0 and the motor run clockwise, nDirection =1 means that the motor run anticlockwise.

If you let nDirection =1 and the motor run clockwise, nDirection =0 means that the motor

2.2.39 CZolixOmniSpec:: ShowSettingDialog

void ShowSettingDialog();

Explain: To display the Setting dialog box and contain some function of control.

2.2.40 CZolixOmniSpec:: SendCommand

BOOL SendCommand (LPCTSTR cmd);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: cmd is the command according to the firmware.

Explain: Control the monochromator by the command according to the firmware communication protocol through calling this function.

See: The communication protocol of the monochromator, **GetReceiveString.**

2.2.41 CZolixOmniSpec:: GetReceiveString

CString GetReceiveString();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Receive the returns from the monochromator according to the firmware communication protocol.

See: The communication protocol of the monochromator, **SendCommand.**

2.2.42 CZolixOmniSpec:: StoreData

BOOL StoreData(long nAddress, short byteData);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nAddress is the address in the EEPROM (from 0 to 1023).

byteData is the data which will be stored.

2.2.43 CZolixOmniSpec:: LoadData

BOOL LoadData (long nAddress, short*pbyteData);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nAddress is the address in the EEPROM (from 0 to 1023).

***byteData** is the pointer at the data which will be stored.

Explain: Read Information from EEPROM of hardware system.

2.2.44 CZolixOmniSpec:: Backup

BOOL Backup(void);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Backup parameters of the monochromator.

There are two memory blocks in the EEPROM. One is for current parameters store. The other is for backup or restore. When the instrument leave factory, the manufactory set parameters in the memory for current parameters store and back up them in the other memory block.

Parameters include:

No. of the grating turret	No. of the grating	0 nm position	Calibration	Lines	Glaze wavelength	Appointed position
1	1	m	x	k	m	x
1	2	m	x	k	m	x
1	3	m	x	k	m	x
2	1	m	x	k	m	x
2	2	m	x	k	m	x
2	3	m	x	k	m	x
3	1	m	x	k	m	x
3	2	m	x	k	m	x
3	3	m	x	k	m	x

2.2.45 CZolixOmniSpec:: Restore

BOOL Restore(void);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

2.2.46 CZolixOmniSpec:: Initialize

BOOL Initialize();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Initialize. All Reload- function will be called.

2.2.47 CZolixOmniSpec:: ReloadSystemInfor

BOOL ReloadSystemInfor();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Get the system information from the hardware including manufacturer, model, serial number, leave factory date.

2.2.48 CZolixOmniSpec:: SetSystemInfor

BOOL SetSystemInfor(LPCTSTR ManuFacturer, LPCTSTR Model, LPCTSTR SerialsNumber, LPCTSTR FactoryDate);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

ManuFacturer is the name of the manufacturer and its string length is less than 8.

Model is the model of the instrument such as Omni-λ150, Omni-λ300, Omni-λ500, Omni-λ750, SSM300.

SerialsNumber is the serial number of the instrument and its string length is less than 10.

FactoryDate is the leave factory date of the instrument.

2.2.49 CZolixOmniSpec:: ReloadTotalSteps

BOOL ReloadTotalSteps();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Reload total steps of the grating motor.

2.2.50 CZolixOmniSpec:: GetTotalSteps

long GetTotalSteps();

Return: Current total steps of the grating motor.

2.2.51 CZolixOmniSpec:: SetTotalSteps

BOOL SetTotalSteps(long TotalSteps);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

TotalSteps is total steps of the grating motor, such as 2304000 or 1152000.

Explain:

Set the new total steps.

Y= the grating total steps

X= total steps for motor running one circle

Z= total steps for grating turret running one circle

W=total steps for motor driver

Y= (X*Z*W)

Omni-λserial monochromators have two type total steps, one is 2304000, the other is 1152000.

Note:

If this parameter is changed, interrelated parameters must be changed also, such as calibrations, 0 nm position of the grating.

2.2.52 CZolixOmniSpec:: ReloadTurret

BOOL ReloadTurret();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Read the number of the grating turret from the hardware.

2.2.53 CZolixOmniSpec:: GetTurret

short GetTurret ();

Return: the number of the current grating turret.

Explain: Every instrument can mount three grating turrets and every turret can mount three gratings.

2.2.54 CZolixOmniSpec:: SetTurret

BOOL SetTurret(short Index);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter: Index is the number of the grating turret.

2.2.55 CZolixOmniSpec:: ReloadRomInfor

BOOL ReloadRomInfor();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Renew the information in the firmware.

2.2.56 CZolixOmniSpec:: GetZeroPos

BOOL GetZeroPos(short Grating, long* pZeroPos);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

Grating is the number of the grating.

***pZeroPos** is the pointer at the 0 nm position of the grating and its unit is step.

Explain: Get the 0nm position of every grating.

Default = total steps *（N-1）/N. N is the amount of gratings.

2.2.57 CZolixOmniSpec:: SetZeroPos

BOOL SetZeroPos(short Grating, long ZeroPos);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

Grating is the appointed grating.

ZeroPos is the position of the grating and its unit is step.

Explain: Set the grating motor position when the grating output wavelength is 0 nm. Default = total steps *（N-1）/N. N is the amount of gratings.

2.2.58 CZolixOmniSpec:: GetAdjustCoefficient

BOOL GetAdjustCoefficient (short Grating, double StandWave, double TestWave, double* Coeffiecient);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

Grating is the number of the grating.

StandWave is the standard wavelength.

TestWave is the supposed wavelength.

***Coeffiecient** is the pointer at the correction coefficient.

Note:

Make sure the position of the grating motor when wavelength is 0nm is correct before calling this function.

See: SetZeroPos，SetAdjustment

2.2.59 CZolixOmniSpec:: GetCoefficient

BOOL GetCoefficient (short Grating, double* pCoefficient);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

Grating is the number of the grating.

***pCoeffiecient** is the pointer at the correction coefficient.

Explain: Grating correction factor.

2.2.60 CZolixOmniSpec:: GetGratingWaveRangeMax

BOOL GetGratingWaveRangeMax(short Grating, double* pWaveMax);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

Grating is the number of the grating.

***pWaveMax** is the pointer at the max wavelength of the grating.

Explain: Get the max wavelength of the appointed grating and the min wavelength is 0nm all along.

2.2.61 CZolixOmniSpec:: SetAdjustment

BOOL SetAdjustment(short Grating, double Coefficient);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

Grating is the number of the grating.

Coefficient is the new calibration of the wavelength.

Explain:

Set the new calibration of the wavelength through calling this function and get new calibrations through calling function Adjusting().

See: SetZeroPos, Calibrate

2.2.62 CZolixOmniSpec:: Calibrate

BOOL Calibrate(short Grating, double StandWave, double CurrentWave);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

Grating is the number of the grating.

CurrentWave is current testing wavelength.

StandWave is the real wavelength.

Note:

Make sure the position of the grating motor when wavelength is 0nm is correct before calling this function. Set the right 0nm position through calling SetZeroPos() before calling this function.

See: SetZeroPos, SetAdjustment

2.2.63 CZolixOmniSpec:: ReloadInitWave

BOOL ReloadInitWave ();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Renew the ActiveX control characteristic **InitWave.**

The Reload function should be called at least once before the Get members of the function group containing Reload are called. That is in order to renew the ActiveX control data. You can call all Reload functions through calling the initialize function.

2.2.64 CZolixOmniSpec:: GetInitWave

BOOL GetInitWave(short Grating, double* InitWavePos);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

Grating is the number of the grating (from 1 to 3).

***InitWavePos** is the pointer at the appointed wavelength of the grating.

Explain: After changing the grating, the output wavelength of the appointed grating is the value that the pointer ***InitWavePos** point at.

2.2.65 CZolixOmniSpec:: SetInitWave

BOOL SetInitWave(short Grating, double InitWavePos);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

Grating is the number of the grating (from 1 to 3).

InitWavePos is the position of the appointed wavelength and its unit is nm.

Explain: Select the position of the initial wavelength of grating.Changing the grating.

2.2.66 CZolixOmniSpec:: ReloadGratingsParam

BOOL ReloadGratingsParam();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Read parameters of the grating from the instrument.

2.2.67 CZolixOmniSpec:: SetGratingParam

BOOL SetGratingParam(short nGrating, long nGroove, long nBlaze);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nGrating is the number of the grating (from 1 to 3).

nGroove is the line of the grating and its value is from 0 to 65535.

nBlaze is the blaze wavelength and its value is from 0 to 65535.

Explain: Setting the line of the grating and the blaze wavelength, after that the wavelength will get.

2.2.68 CZolixOmniSpec:: GetGratingParam

BOOL GetGratingParam(short nGrating, long* pGroove, long* pBlaze);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nGrating is the pointer at the number of the grating.

***pGroove** is the pointer at the line of the grating.

***pBlaze** is the pointer at the blaze wavelength.

Explain: Get grating parameter of spectrometer.

If return=TRUE. Storage information to desired location that pGroove and pBlaze point, grating number range from1 to 3.

2.2.69 CZolixOmniSpec:: WaveToSteps

BOOL WaveToSteps(short Grating, double wavelength, long* pSteps);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nGrating is the pointer at the number of the grating.

wavelength is the appointed wavelength.

***pSteps** is the pointer at the motor step which is according to the appointed wavelength.

Explain: Change the wavelength to the step.

2.2.70 CZolixOmniSpec:: ReloadCurrentGrating

BOOL ReloadCurrentGrating ();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Refresh the current grating serial number through calling this function.

2.2.71 CZolixOmniSpec:: ReloadPowerGrating

BOOL ReloadPowerGrating();

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Explain: Change the initial grating.

2.2.72 CZolixOmniSpec:: SetPowerGrating

BOOL SetPowerGrating(short GratingIndex);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

GratingIndex is the number of the grating.

Explain: GratingIndex =0 means that the grating is located on the last position when you close down the instrument; **GratingIndex** =1 means that the grating is located on the No. 1 grating position; **GratingIndex** =2 means that the grating is located on the No. 2 grating position; **GratingIndex** =3 means that the grating is located on the No. 3 grating position.

See: GetPowerGrating, SetZeroPos

2.2.73 CZolixOmniSpec:: GetPowerGrating

short GetPowerGrating();

Return:

Return = 0 means that the grating is located on the last position when you close down the instrument; Return =1 means that the grating is located on the No. 1 grating position; Return =2 means that the grating is located on the No. 2 grating position; Return =3 means that the grating is located on the No. 3 grating position.

See: SetPowerGrating, SetZeroPos

2.2.74 CZolixOmniSpec::AboutBox

void CZolixOmniSpec::AboutBox();

Explain: Display the version of this ActiveX control.

2.2.75 CZolixOmniSpec:: SlitHome

BOOL SlitHome(int nIndex);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nIndex is the serial number of slits and its value is from 1 to 4.

Explain: This function is for the slit home.

2.2.76 CZolixOmniSpec::SetSlitZeroPos

BOOL SetSlitZeroPos(int nIndex, long nPos);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nIndex is the serial number of slits and its value is from 1 to 4.

nPos is the zero position and its value is from -10 to 10μm.

Explain: This function is for setting the new zero position of the slit.

2.2.77 CZolixOmniSpec::GetSlitZeroPos

BOOL GetSlitZeroPos(int nIndex, long *pnPos);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nIndex is the serial number of slits and its value is from 1 to 4.

***pnPos** is the pointer at the zero position and its value is from -10 to 10μm.

Explain: This function is for setting the new zero position of the slit.

2.2.78 CZolixOmniSpec::SetSlitBandpass

BOOL SetSlitBandpass(int nIndex, double dblBandpass);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nIndex is the serial number of slits and its value is from 1 to 4.

dblBandpass is the current position of the slit and its unit is nm.

Explain: This function is for getting the current position of the slit.

2.2.79 CZolixOmniSpec::GetSlitBandpass

BOOL GetSlitBandpass(int nIndex, double *pdblBandpass);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nIndex is the serial number of slits and its value is from 1 to 4.

***pdblBandpass** is the pointer at the current position of the slit and its unit is nm.

Explain: This function is for getting the current position of the slit.

2.2.80 CZolixOmniSpec::SetSlitWidth

BOOL SetSlitWidth(int nIndex,long nWidth);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nIndex is the serial number of slits and its value is from 1 to 4.

nWidth is the width of the slit and its value is from 10 to 3000μm.

Explain: This function is for setting the width of the slit.

2.2.81 CZolixOmniSpec::GetSlitWidth

BOOL GetSlitWidth(int nIndex, long *pnWidth);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nIndex is the serial number of slits and its value is from 1 to 4.

***pnWidth** is the pointer at width of the slit and its value is from 10 to 3000μm.

Explain: This function is for setting the width of the slit.

2.2.82 CZolixOmniSpec::SetSlitType

BOOL SetSlitType(short nSlitType);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nSlitType:

SLIT	0
SR-SLIT	1
SLIT-I24	2

Explain: SLIT-I24, slit width value is 24000nm, SLIT and SR-SLIT value is 3000nm.

2.2.83 CZolixOmniSpec::GetSlitType

SHORT GetSlitType();

Return: nSlitType.

Parameter:

nSlitType:

SLIT	0
SR-SLIT	1
SLIT-I24	2

2.2.84 CZolixOmniSpec::SetEntrancePort

BOOL SetEntrancePort(short nport);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nPort: Entry number(/direction, straight or side).

Explain: Get entrance path of optical, straight or side.

2.2.85 CZolixOmniSpec::GetEntrancePort

BOOL GetEntrancePort(short* pnport);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

pnport: the pointer of entry number(/direction, straight or side).

Explain: Get entrance path of optical, straight or side.

2.2.86 CZolixOmniSpec::SetMotorTotalSteps

BOOL SetMotorTotal Steps(short nMotorIndex, long nTotalSteps);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nMotorIndex:2~9.

nTotalSteps: 0~2147483647.

Explain: This system returns “OK”+“↙” on success, if the input data is out the range, the system will return “E06”+“↙” on failure.

2.2.87 CZolixOmniSpec::GetMotorTotalSteps

BOOL GetMotorTotalSteps(short nMotorIndex, long FAR* pnTotalSteps);

Return: If succeed, return = TRUE. Otherwise, return = FALSE.

Parameter:

nMotorIndex:2~9.

pnTotalSteps: 0~2147483647.

Explain: The total steps of instruction, this command determines the total travel steps. This system returns “OK”+“↙” on success, if the input data is out the range, the system will return “E06”+“↙” on failure.

2.2.88 CZolixOmniSpec::SetSetupInfo

Void SetSetupInfo(BOOL bsetupinfo)

Return:

Parameter:

BOOL bsetupinfo

Automatically command the instrument parameters.

Explain: Set up true, the control to open the process to obtain the parameters of the instrument. if set up false, no parameters.

3. Example

3.1 The simple steps to establish connection

```
CZolixOmniSpec*   m_ pctrl;
SetMeter(CZolixOmniSpec*   p_ctrl)      //ActiveX Pointer
{
     m_pctrl=p_ctrl;
};
Connect()                                //basic connection
{
     if(!GetIsOpen())
```

```
        {
                Open();
        }
        BOOL bResult = FALSE;
        bResult = m_pctrl->Connect();
        m_pctrl->SetSynchroMode(TRUE);    //Synchronous mode(control and instrumentation)
        return bResult;
};
OnRS232()                                       //The connection with setting 1
{
        m_pctrl->SetCommPort(1);                //other numbers
        m_pctrl->SetUSBMode(FALSE);
        this->Connect();
};
OnUSB()                                         //The connection with setting 2
{
        Cstring str;
        int n = m_pctrl->SearchZolixUSBDevice ();
        if(n>0)
        {
                str = m_pctrl-> GetZolixUSBSerial (0);          //0---n-1
                m_pctrl-> SetUSBSerials(str);
                m_pctrl->SetUSBMode(TRUE);
                m_pctrl->SetUSBSerial(str);
                this->Connect();
        }
};
```

3.2 Other functions

Please call them directly, example：

```
Movto(double wavelength)                        //the changes of wavelength
{
        m_pctrl->MoveTo(wavelength);
};
MoveMotorTo(int nIndex,long stepto)             // the running of External motor devices
{
        m_pctrl->MoveMotorTo(nIndex, stepto);
```

};

3.3 Functions including Reload-

For some functions, before calling, the Reload-function must be called first, please attention.

For example:

```
m_pctrl->ReloadPeripheral();                    //Must call the Reload-first, or the result is wrong.
int n = m_pctrl->GetPeripheralCount();          //Must Reload the external equipments first.
CString str = m_pctrl->GetPeripheralName(0); //The range from 0-1.
```