OL 750-420 Software Development Kit for the OL Series 750 Automated Spectroradiometric Measurement System

(.NET DLLs)

Manual No: M000274
Revision: F
Date: June 2015

Gooch & Housego

4632 36th Street, Orlando, Florida 32811 USA
T +1 407 422 3171 F +1 407 648 5412
Visit our website at www.ghinstruments.com

GOOCH & HOUSEGO
SOFTWARE LICENSE AGREEMENT

LICENSE. You may not use, copy, modify, or transfer the program, or any copy, modification or merged portion, in whole or in part, except as expressly provided for in this license. You may:

a. Use the program on a single machine.

b. Copy the program into any machine readable or printed form for backup purposes.

c. Transfer the program and license to another party if the other party agrees to accept the terms and conditions of this Agreement. If you transfer the program, you must at the same time either transfer all copies, whether in printed or machine-readable form to the same party, or destroy any copies not transferred. This includes all modifications and portions of the program contained or merged into other programs. If you transfer possession of any copy, modification or merged portion of the program to another party, your license is automatically terminated.

OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Gooch & Housego or its suppliers. The Software is copyrighted and protected by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Gooch & Housego may make changes to the Software, or to items referenced therein, at any time without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Gooch and Housego grants no express or implied right under Gooch & Housego patents, copyrights, trademarks, or other intellectual property rights. You may transfer the Software only if the recipient agrees to be fully bound by these terms and if you retain no copies of the Software.

LIMITED MEDIA WARRANTY. If the Software has been delivered by Gooch & Housego on physical media, Gooch & Housego warrants the media to be free from material physical defects for a period of ninety days after delivery by Gooch & Housego. If such a defect is found, return the media to Gooch & Housego for replacement or alternate delivery of the Software as Gooch & Housego may select.

EXCLUSION OF OTHER WARRANTIES. Except as provided above, the Software is provided “as is” without any express or implied warranty of any kind including warranties of merchantability, noninfringement, or fitness for a particular purpose**.** Gooch & Housego does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the Software.

LIMITATION OF LIABILITY. In no event shall Gooch & Housego or its suppliers be liable for any damages whatsoever, (including, without limitation, lost profits, business interruption, or lost information) arising out of the use of or inability to use the Software, even if Gooch & Housego has been advised of the possibility of such damages. Some jurisdictions prohibit exclusion or limitation of liability for implied warranties or consequential or incidental damages, so the above limitation may not apply to you. You may also have other legal rights that vary from jurisdiction to jurisdiction.

EXPORT. You agree that all Software products (or portions thereof) and technical data delivered under this Agreement will be exported, re-exported or transferred exclusively as authorized and as permitted by the laws and regulations of the U.S. government. Prohibited exports include but are not limited to: the export, re-export, or transfer of the Software product or technical data to any prohibited entities or destinations subject to the US government’s current list of restricted or embargoed countries or entities, any parties currently listed on the US government’s Denied Parties or Specially Designated National List, and any proliferation activities prohibited by the US government such as chemical, biological, nuclear or missile technology.

OL 750-420
Software Development Kit
for the
OL Series 750
Automated Spectroradiometric Measurement System

Table of Contents

OL 750-420
Software Development Kit
for the
OL Series 750
Automated Spectroradiometric Measurement System

The OL 750 Software Development Kit (SDK) consists principally of a .NET DLL library with methods and properties to fully access the capabilities of your OL 750 Controller. The SDK contains an example application program including project files and source code. The project was developed using Microsoft Visual Studio 2010, written using C#. However, you can develop your application using various suitable languages and development environments. If the platform allows linking to .NET or COM libraries you can develop a custom application that makes use of the SDK library. Example platforms include LabVIEW, Excel VBA, and Visual Basic.

OL 750-420 INSTALLATION

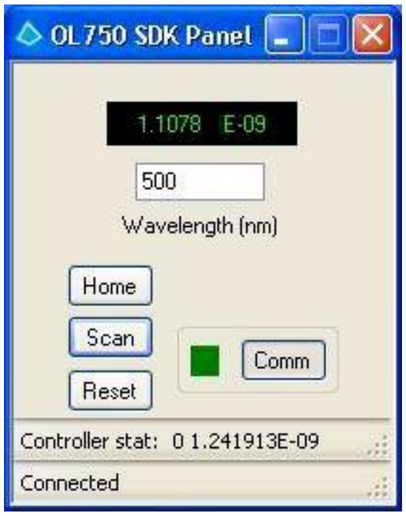
To install the SDK, run the **setup.exe** program located on the OL 750-420 Software and follow the install wizard prompts. Upon completion of the installation the OL 750 will be registered for COM inter-operability. This allows compatibility with older technologies, such as Visual Basic 6.0.

EXAMPLE APPLICATION

The example application was created in Microsoft Visual Studio 2010. The source project and C# code is included in this SDK. You can use the source as a basis for creating your own custom application. This example can perform many OL 750 Controller operations, including scans.

Communications Setup

1. Right click in the OL750 SDK Panel and select **Setup**.
2. Select Connection type parameters, such as Serial, COM port 1.
3. Click **OK** to exit back to the initial screen.
4. Click the **Comm** button to toggle communications.
5. The connection status is displayed on the lower of two status bars. The controller status is displayed on the upper bar. Refer to the *The Query Commands/Properties/Status* section to interrupt status code values.
6. Place a destination wavelength in the text box and press the Enter key. The monochromator moves to that wavelength and returns the signal value in the black box.
7. Click the **Home** button to home the monochromator.

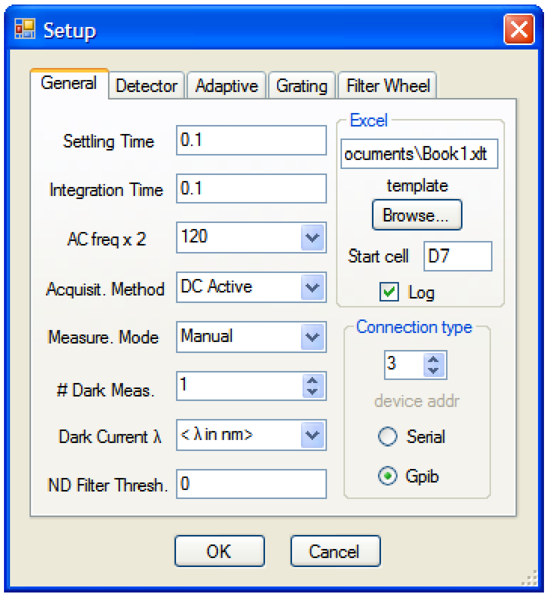


Controller stat: 0 1.241913E-09

Connected

Controller Setup

To access Setup, right click in the OL750 SDK Panel. Setup consists of a series of tabs containing numerous boxes to setup the controller *(Refer to the OL Series 750 Automated Spectroradiometric Measurement System user's manual for further information)*. Each tab contains items that become a parameter in the library functions described below.



Settling Time	0.1
Integration Time	0.1
AC freq x 2	120
Acquisit. Method	DC Active
Measure. Mode	Manual
# Dark Meas.	1
Dark Current λ	< λ in nm>
ND Filter Thresh.	0

Excel: ocuments\Book1.xlt template
Start cell: D7
✓ Log
Connection type: 3 device addr
Serial [✓ Gpib]

General and Detector Tab - Contain parameters for the SendDetectorAndGeneral function.

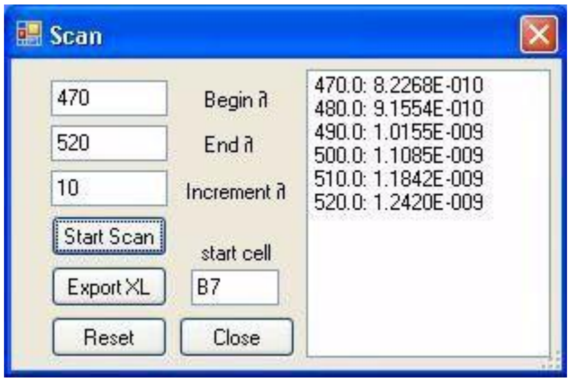
Adaptive Tab - Contains integration and settling times used in adaptive mode.

Grating Tab - Contains parameters for three gratings that are used in the SendGrating function.

Filter Wheel Tab - Wavelength entries are used in the SetupFilterWheel function.

Performing a Scan

1. Click the **Scan** button to display the scan dialog.
2. Enter the Begin, End, and Increment wavelengths to setup your scan.
3. Optionally click the **Export XL** button to log to the Excel worksheet previously set in the Setup/General page.
4. The start cell can be changed prior to each scan.
5. To send a reset command to the controller and abort pending commands click the **Reset** button.
6. Click the **Start Scan** button to execute the scan. Results are logged to the Scan dialog's list box and Excel worksheet.



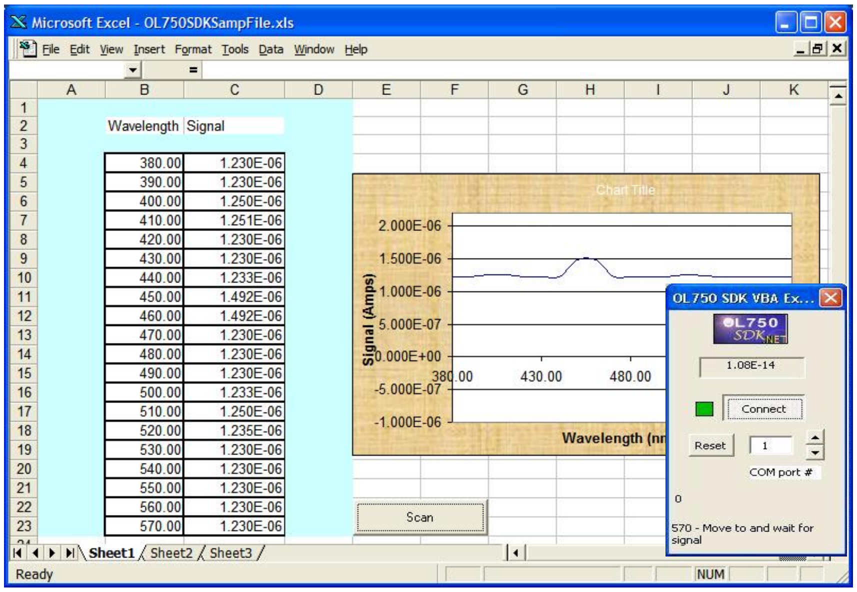
Begin λ	470
End λ	520
Increment λ	10
start cell	B7

```
470.0: 8.2268E-010
480.0: 9.1554E-010
490.0: 1.0155E-009
500.0: 1.1085E-009
510.0: 1.1842E-009
520.0: 1.2420E-009
```

EXCEL EXAMPLE

The spreadsheet OL750SDKSampFile.xls included in this SDK illustrates how to use the library DLL via COM. Upon installation the DLL is registered as a type library and can be accessed as a Visual Basic Reference. By opening Excel's Visual Basic editor the VBA code that runs, logs, and plots a simple scan can be inspected. Use this code as a starting point to create your own custom application.

Click the **Scan** button on the excel spreadsheet to activate the OL750 SDK Userform. Set the COM port value and click the **Connect** button to run a 20 point scan as written in the VBA code.



Wavelength	Signal
380.00	1.230E-06
390.00	1.230E-06
400.00	1.250E-06
410.00	1.251E-06
420.00	1.230E-06
430.00	1.230E-06
440.00	1.233E-06
450.00	1.492E-06
460.00	1.492E-06
470.00	1.230E-06
480.00	1.230E-06
490.00	1.230E-06
500.00	1.233E-06
510.00	1.250E-06
520.00	1.235E-06
530.00	1.230E-06
540.00	1.230E-06
550.00	1.230E-06
560.00	1.230E-06
570.00	1.230E-06

1.08E-14
COM port #: 1
0
570 - Move to and wait for signal

OL 750-420 COMMANDS

The OL 750-420 commands are subdivided into the following five categories:

- Communication Commands
- Setup Commands
- Manual Mode Commands
- Query Commands
- Miscellaneous Commands

Communication Commands

GpibOpen

int GpibOpen(int boardNum, int address)

Return Value

0 – Success
1 – Fail to connect
29 – Invalid Board
30 – Invalid Address

Parameters

BoardNum

The Board parameter specifies the address of the GPIB board in the user's host computer. This value is typically 0 but can be any number from 0 – 31.

Address

The Address parameter specifies the GPIB address of the instrument. Please note that the address specified must match the address of the OL 750 dipswitches.

Remarks

Opens the specified GPIB address to the OL 750 Controller. The open command must be called before any other commands can be successfully sent. If your instrument is communicating by RS232 (serial), please see the SerialOpen command.

SerialOpen

int SerialOpen(int comPort)

Return Value

0 – Success
1 – Fail to connect
31 – Invalid port number

Parameters

ComPort

The communications port in which the host computer is communicating with the OL 750 Controller, e.g., COM1.

Remarks

Opens the specified serial port to the OL 750 Controller. The open command must be called before any other commands can be successfully sent. If your instrument is communicating by GPIB, please see the GPIBOpen command.

CloseComm

void CloseComm()

Return Value

None.

Parameters

None.

Remarks

Closes any open communications. Call this function before exiting your custom application to adequately free up resources.

Setup Commands

The setup commands provide the information to the controller needed to initialize the system. These commands should be sent before each measurement.

Reset

int Reset

Return Value

0 – Success
1 – OL 750 Not connected

Parameters

None.

Remarks

Sends a soft reset to the OL 750 Controller. It aborts any command that is in progress. This command should be sent before any setup information is sent and after a scan is complete.

SendGratingAndAutoSlitSetupInfo

int SendGratingAndAutoSlitSetupInfo(double[] gratingLowerW,double[] gratingUpperW, double[]gratingGrooves,double[] gratingSkew,int[] gratingEntrance, int[] gratingExit, int entranceHomeOffset, int exitHomeOffset)

Return Value

0 – Success
1 – OL 750 Not connected
4 – Arrays are NULL
5 – Invalid array lengths
6 – Grating wavelengths invalid
7 – Grating entrance slit indices are invalid
8 – Grating exit slit indices are invalid

Parameters

GratingLowerW

The lower effective wavelength for a grating. The wavelength should be specified in nanometers.

GratingUpperW

The upper effective wavelength for a grating. The wavelength should be specified in nanometers.

GratingGrooves

The grooves per millimeter for a grating. Please consult your grating documentation for this value.

GratingSkew

The skew angle for this grating. This value is usually provided by Gooch & Housego or can be generated by performing a grating calibration.

GratingEntrance

Automated entrance slit index is the slit index to be used when the corresponding grating is selected. This value can be a number from 0 to 11. This value should be set to a –1 if automated slits are not installed on the OL 750 system.

GratingExit

Automated exit slit index is the slit index to be used when the corresponding grating is selected. This value can be a number from 0 to 11. This value should be set to a –1 if automated slits are not installed on the OL 750 system.

EntranceHomeOffset

Entrance home offset is the offset needed to perfectly home the automated entrance slit system. This value should be provided by Gooch & Housego. If you do not have an automated slit system, enter a 0 for this parameter.

ExitHomeOffset

Exit home offset is the offset needed to perfectly home the automated exit slit system. This value should be provided by Gooch & Housego. If you do not have an automated slit system, enter a 0 for this parameter.

Remarks

Each array must contain three elements, one for each of three gratings. Sets up the three gratings for the OL 750 Monochromator. Also sets up the optional automated slits device. This information is provided by Gooch & Housego when the system shipped.

SendFilterWheelSetupInfo

int SendFilterWheelSetupInfo(double[] cutOnWavelength)

Return Value

0 – Success
1 – OL 750 Not connected
4 – Arrays are NULL;
5 – Invalid array length;
9 – Wavelengths are not in ascending order

Parameters

CutonWavelength

An 11element array. Parameters are the cut-on wavelengths for each of up to 11 filter positions specified in nanometers. The cut-on wavelength is the wavelength in which the indicated filter position becomes active. Enter a 99999.95 for an unused filter. The filters must have ascending cut-on wavelengths.

Remarks

This function sets up the filter cut-on wavelengths for each of 11 filter positions.

SendAdaptiveSettlingIntegrationTime

int SendAdaptiveSettlingIntegrationTime(double[] settlingTime, double[] integrationTime)

Return Value

0 – Success
1 – OL 750 Not connected
4 – Arrays are NULL
5 – Invalid array lengths

Parameters

SettlingTime

An 11 element array. These values specify the settling time in seconds for the gain index equal to the settlingTime array index. settlingTime[10], the eleventh item, is for a DC high sensitivity detector.

IntegrationTime

An 11-element array. These values specify the integration time in seconds for the gain index equal to the integrationTime array index. integrationTime[10], the eleventh item is for a DC high sensitivity detector.

Remarks

This function is used to set the settling and integration times for adaptive mode signal acquisition.

SendDetectorAndGeneralInfo

int SendDetectorAndGeneralInfo(double settlingTime, double integrationTime, double acFrequency,
int acqMethod, int measMode,int detector, double chopperFrequency, double fluxOverload,
double pmtHighVoltage, int detectorPortAndChannel, int connectAorA1, int connectB, int connectA2,
int connectA3, int darkCurrentGain, int dsm, double gainMultiplier, int gainIndex,int gainCount,
int darkCount, double darkWavelength, double ndFilterThresh)

Return Value

0 – Success
1 – OL 750 Not connected
22 – settling time invalid
23 – integration time invalid
24 – invalid acquisition mode
25 – invalid measurement mode
21 – invalid PMT voltage
26 – invalid detector port index value

Parameters

SettlingTime

Settling time in seconds, used when making measurement. This parameter is specified in seconds and should be set to –1.00 if measurements are to be made using adaptive mode.

IntegrationTime

Integration time in seconds, used when acquiring data during measurements. This parameter is specified in seconds and should be set to –1.00 if measurements are to be made using adaptive mode.

ACFrequency

The desired AC frequency in Hertz, times two. Should be set to 100 for 50 Hertz, or 120 for 60 Hertz AC power line frequency.

AcqMethod

Acquisition method is one of the following.
0 for DCACTIVE
1 for DCHIGHSENSITIVITY
2 for ACACTIVE
3 for PULSEACTIVE
4 for PHOTONACTIVE
5 for CCD_MEASURE
6 for DSM1F_MEASURE

MeasMode

This parameter specifies how the instrument will attempt to take measurements. Set this value to a 0 if you are performing scans with the GoToWavelengthAndReturnSignal function. When performing normal scans, including the use of dark currents, use this value. Set the value to 1 if you are attempting manual control of the instrument. This is typically used with the function prefixed by Manual. Use this value if you are going to be performing custom movements of the monochromator and are querying the signal with the GetSignal function.

Detector

Specifies the current detector in use. See remarks below for a listing of detector numbers.

ChopperFrequency

Specifies the chopper frequency in Hertz. Enter a 0.00 if in DC mode or the chopper is not installed.

FluxOverload

Specifies the PMT flux overload value. Flux overload is the signal value at which the PMT voltage will be automatically shut down to prevent exposure damage to the PMT detector.

PMTHighVoltage

Specifies the PMT high voltage value in Volts. Enter a value of 99999.95 if no PMT detector is attached.

DetectorPortAndChannel

A byte that sets the port number, detector channel, and optional mux. Examples:

0x20 - Use detector port 1, channel A, no mux.
0x41 - Use detector port 2, channel B, no mux.
0x4A - Use detector port 2, channel A1, with mux.
0x01 - No slide installed, channel B, no mux.

ConnectAorA1

The detector that is presently connected to channel A (or channel A1 for the multiplexer). See remarks below for detector numbers. Enter a –1 if no detector or DSM (detector support module) is installed.

ConnectB

The detector that is presently connected to channel B. See remarks below for detector numbers. Enter a –1 if no detector or DSM (detector support module) is installed.

ConnectA2

The detector that is presently connected to channel A2 of the multiplexer. See remarks below for detector numbers. Enter a –1 if no detector or DSM (detector support module) is installed.

ConnectA3

The detector that is presently connected to channel A3 of the multiplexer. See remarks below for detector numbers. Enter a –1 if no detector or DSM (detector support module) is installed.

DarkCurrentGain

A gain index. This normally should be a –1. Specifies the gain level to be used during dark current acquisitions. Enter a –1 for autoranging of the gain. Gains indices range from 0 to 10.

DSM

Specifies the detector support module (DSM) for this detector.

0 for DSM1A
1 for DSM1D
2 for DSM1F
3 for DSM2
4 for DSM3
5 for Not Used
6 for DSM1DH
7 for DSMCCD

GainMultiplier

Specifies the factor that can be used to adjust for additional signal amplification should it be present on the detector. This is also called the (front end) gain correction factor by Gooch & Housego.

GainIndex

The gain index for the first valid gain range used for this detector/DSM. Any gain proceeding this gain index will be ignored.

GainCount

Specifies the number of valid gain indexes including the first gain index. For instance, if you had 8 gains and only wanted to use the last 4, you would set gainIndex to 3 (for the 4th gain) and gainCount to 4 (total number of gains to be used).

DarkCount

Specifies the number of dark current acquisitions to average for the resultant dark current. Enter a 0 if no dark current reading is desired. Note that dark currents are taken at the integration time so multiple dark current acquisitions are each subject to that integration time.

DarkWavelength

Specifies the wavelength at which the dark current is to be measured. To measure the dark current in the shutter position, enter 99999.95. Otherwise, enter the desired wavelength.

NDFilterThresh

Specifies the filter activation threshold for NVIS measurements. Enter 1.0E30 or greater.

Remarks

Sends detector and measurement parameters to theOL 750 Controller.
Use this list to obtain a detector parameter value:

Number	Detector
-1	NO_DETECTOR_OR_DSM
0	DH300 Silicon (AC)
1	DH301 Silicon (DC)
2	DH302 Dual Si/PbS (AC) (Si)
3	DH304 Dual Si/Ge (AC) (Si)
4	DH310 PMT (AC)
5	DH311 PMT (DC)
6	DH320 Ge (AC)
7	DH321 Ge (DC)
8	DH330 InGaAs (AC)
9	DH331 InGaAs (DC)
10	DH340 PbS (AC)
11	DH350 PbSe (AC)
12	DH360 InSb (AC)
13	DH370 HgCdTe (AC)
14	DH380 Pyroelectric (AC)
15	DH302B Dual Si/PbS (AC) (PbS)
16	DH304B Dual Si/Ge (AC) (Ge)
17	DH375 HgCdTe (AC) (Extended)
18	UNUSED
19	DH315 The PMT photon counter.
20	DH300PI Pulse Integration (Si)
21	DH310PI Pulse Integration (PMT)
22	DH320PI Pulse Integration (Ge)
23	DH330PI Pulse Integration (InGaAs)
24	UNUSED
25	DH312 PMT (AC) (Solar Blind)
26	DH313 PMT (DC) (Solar Blind)
27	DHCCD1 CCD detector.
28	DH400 Si w/ DSM5
29	DH401 PMT w/ DSM5
30	UNUSED
31	UNUSED
32	UNUSED
33	DHEXT1 External signal source number one
34	DHEXT2 External signal source number two
35	DH306 Si - DSM4 (1A / 2) of Si/Pbs
36	DH306B PbS - DSM4 (1A / 2) of Si/Pbs
37	DH308 Si - DSM4 (1A / 2) of Si/PbSe
38	DH308B PbSe - DSM4 (1A / 2) of Si/PbSe
39	DHCUS1 User-customizable OLI detector #1
40	DHCUS2 User-customizable OLI detector #2
41	DHCUS3 User-customizable OLI detector #3
42	DHCUS4 User-customizable OLI detector #4
43	DH316 Solar Blind PMT Photon counter
44	DH318 GaAs AC detector DSM-1A
45	DH319 GaAs DC detector DSM-1D
46	DH332 Si -DSM5 (1A / 1A) of Si/InGaAs
47	DH332B InGaAs -DSM5 (1A / 1A) of Si/InGaAs
48	PHOTON_COUNTER Photon counter information
49	DH305 Dual Si/Ge (DC) (Si)
50	DH305B Dual Si/Ge (DC) (Ge)
51	DH600NVG OL600 Super High sensitivity DC SiPrepareGrating1SetupInfo

Manual Mode Commands

These commands provide fine access to the functionality of the OL 750 Controller. They are used when the SendDetectorAndGeneralInfo function is passed a 1 or a 2 for the measMode parameter. These functions are typically not to be used with scanning mode (measMode = 0). In order to use these functions ALL of the setup functions must be previously called.

ManualSetGratingPosition

int ManualSetGratingPosition(int grating)

Return Value

0 – Success
1 – OL 750 Not connected
3 – Invalid grating position

Parameters

Grating

Specifies the desired grating location. Can be a number from 0 to 2.

Remarks

This function causes the wavelength drive to move to the desired grating. The function will wait until the grating has stopped before returning control to the user.

ManualSetWavelengthPosition

int ManualSetWavelengthPosition(double wavelength,bool autoChangeFilter)

Return Value

0 – Success
1 – OL 750 Not connected
6 – Invalid wavelength position

Parameters

Wavelength

The desired wavelength position in nanometers.

AutoChangeFilter

Specifies whether the user wishes to automatically move the filter wheel to the correct filter or not (based upon the cut-on wavelengths specified in the PrepareFilterWheelSetupInfo function). Pass a 1 into this function to automatically move the filter wheel or pass a 0 to keep the filter wheel at its current position.

Remarks

This function changes the wavelength position of the monochromator. It also changes the filter position to the cut-on filter if the user so desires. This function will wait until the monochromator has stopped before returning control to the user.

ManualSetFilterWheelPosition

int ManualSetFilterWheelPosition(int grating)

Return Value

0 – Success
1 – OL 750 Not connected
3 – Invalid filter wheel position

Parameters

FilterPosition

Specifies the desired filter wheel position. Valid values are from 0 – 10.

Remarks

This function sets the filter wheel position to that specified by FilterPosition variable. This function will wait until the monochromator has stopped before returning control to the user.

ManualSetChopperPositionOpen

int ManualSetChopperPositionOpen( )

Return Value

0 – Success
1 – OL 750 Not connected

Parameters

None.

Remarks

This function sets the chopper position locked to the open position. This function will wait until the chopper has finished moving before returning control to the user.

Do not use this function if you do not have a chopper installed.

ManualSetChopperPositionClosed

int ManualSetChopperPositionClosed( )

Return Value

0 – Success
1 – OL 750 Not connected

Parameters

None.

Remarks

This function sets the chopper position locked to the closed position. This function will wait until the chopper has finished moving before returning control to the user.

Do not use this function if you do not have a chopper installed.

ManualSetChopperFrequency

int ManualSetChopperFrequency(double frequencyInHertz)

Return Value

0 – Success
1 – OL 750 Not connected
10 – Invalid frequency

Parameters

FrequencyInHertz

Specifies the new chopper frequency in Hertz. Valid range is from 10.0 to 500.0.

Remarks

This function is to change the chopper frequency. This function will wait until the chopper has finished moving before returning control to the user.

Do not use this function if you do not have a chopper installed.

ManualHomeWavelengthDrive

int ManualHomeWavelengthDrive()

Return Value

0 – Success
1 – OL 750 Not connected

Parameters

None.

Remarks

This function homes the wavelength drive. This will reset all monochromator counters and put the monochromator into a startup state. This function will wait until the wavelength drive has completely stopped before returning control to the user.

ManualHomeFilterWheelDrive

int ManualHomeFilterWheelDrive()

Return Value

0 – Success
1 – OL 750 Not connected

Parameters

None.

Remarks

This function homes the filter wheel drive. This will reset the filter wheel counters and put the filter wheel into a startup state. This function will wait until the filter wheel drive has completely stopped before returning control to the user

.

ManualHomeDetectorSlideDrive

int ManualHomeDetectorSlideDrive()

Return Value

0 – Success
1 – OL 750 Not connected

Parameters

None.

Remarks

This function homes the detector slide drive. This will reset the detector slide counters and put the detector slide into a startup state. This function will wait until the detector slide drive has completely stopped before returning control to the user.

Do not call this function if you do not have a detector slide installed.

ManualSetPMTHighVoltage

int ManualSetPMTHighVoltage(double voltagesInVolts)

Return Value

0 – Success
1 – OL 750 Not connected
21 – Invalid PMT high voltage

Parameters

VoltageInVolts

Specifies the desired PMT High Voltage in volts. The PMT detector voltage range is 200 to 1100.

Remarks

This function set the new PMT detector high voltage to the specified value.

Do not call this function if you are not using a PMT detector.

ManualSetSignalGain

int ManualSetSignalGain(int gainIndex)

Return Value

0 – Success
1 – OL 750 Not connected
11 – Invalid gain index

Parameters

GainIndex

Specifies the new desired signal gain. This gain value is an index and increases the detector gain by a factor of 10 for each index. The number of indexes are setup in the PrepareDetectorSetupInfo function. A value of –1 will engage autoranging.

Remarks

This function manually changes the detector gain. Specifying a –1 will engage the autoranging, which will attempt to find the best gain for the acquired signal.

ManualSetDetectorSlidePosition

int ManualSetDetectorSlidePosition(int detectorPort)

Return Value

0 – Success
1 – OL 750 Not connected
31 – Invalid detector slide index

Parameters

DetectorPort

Specifies the detector port to change to. Valid range is 0 to 2.

Remarks

This command is used to position the OL 750-630 detector slide. This function will wait until the detector slide drive has completely stopped before returning control to the user.

Do not call this function if you do not have a detector slide installed.

ManualSetIntegrationTime

int ManualSetIntegrationTime(double timeInSeconds)

Return Value

0 – Success
1 – OL 750 Not connected
15 – Invalid integration time

Parameters

TimeInSeconds

Specifies the new integration time in seconds for manual mode measurement. Valid range is 0.0 – 600.0 seconds.

Remarks

This function sets the manual mode integration time. To use this feature, you must have the MeasurementMode parameter of the PrepareGeneralMeasurementInfo function set to a 1 or a 2.

ManualSetAutoSlitsPositions

int ManualSetAutoSlitsPositions(short entrancePosition, short exitPosition, int entranceOffset, int exitOffset)

Return Value

0 – Success
1 – OL 750 Not connected
7 – Invalid entrance slit index
8 – Invalid exit slit index

Parameters

EntrancePosition

Specifies the new entrance slit position. Valid range is from 0 to 11.

ExitPosition

Specifies the new exit slit position. Valid range is from 0 to 11.

EntranceOffset

Specifies the entrance slit offset. This value is used to make minor adjustments for the setting of the home switch so that slit or aperture can be positioned in the center of the monochromator entrance optical path.

ExitOffset

Specifies the exit slit offset. This value is used to make minor adjustments for the setting of the home switch so that the slit or aperture can be positioned in the center of the monochromator exit optical path.

Remarks

This function allows the user to move the OL 750-635 and OL750-636 automated slits.

Do not call this function if you do not have the automated slits installed.

ManualSet75MAPositions

int ManualSet75MAPositions(double sampleAngleInDegrees, double detectorAngleInDegrees, double sampleOffsetInDegrees, double detectorOffsetInDegrees)

Return Value

0 – Success
1 – OL 750 Not connected
16 – Invalid angles

Parameters

SampleAngleInDegrees

Specifies the new angle for the sample table of the OL 750-75MA. Valid range is 0.0 to 90.0.

DetectorAngleInDegrees

Specifies the new angle for the detector arm of the OL 750-75MA. Valid range is 20.0 to 180.0.

SampleOffsetInDegrees

The sample table offset is used to adjust the home position of the sample table to allow for optimal optical alignment.

DetectorOffsetInDegrees

The detector arm offset is used to adjust the home position of the detector arm to allow for optimal optical alignment.

Remarks

This command is used to set the OL 750-75MA sample table and detector arm positions.

Do not call this command unless you have the OL 750-75MA installed on your system.

ManualPulseIntegrationTrigger

void ManualPulseIntegrationTrigger()

Return Value

None.

Parameters

None.

Remarks

Use this command to start a pulse integrator measurement sequence when the pulse integrator is in the Keyboard or External sychronization mode. See SendPulseIntegratorSetupInfo for setup parameters of the pulse integrator.

Do not call this function unless you have the Pulse Integrator installed.

Note: This command should not be set unless the Bit 29 or Bit 39 of the status is set. To retrieve the status, call the GetStatus query function.

ManualSetNDFilterWheelPosition

int ManualSetNDFilterWheelPosition(short filterPosition)

Return Value

0 – Success
1 – OL 750 Not connected
14 – Invalid filter position

Parameters

FilterPosition

Specifies the new ND filter wheel position. Valid range is 0 to 10.

Remarks

This command is used to position the ND filter wheel.

Do not call this command if you do not have a ND filter wheel installed.

ManualSetAutoLightSource

int ManualSetAutoLightSource(short position)

Return Value

0 – Success
1 – OL 750 Not connected
13 – Invalid position value

Parameters

Position

Specifies the new light source position. Specify a 0 for Position A and a 1 for Position B.

Remarks

This command is used to position the OL 750-20 Automated Light Source Attachment for the OL 750.

Do not call this command if you do not have an OL 750-20 Automated Light Source Attachment.

Query Commands

The query commands provide vital information to the user during either manual mode operation or scans. Some of these functions return information quickly while others can take many minutes to perform the results. The following is a list of these commands:

GetWavelengthPosition – Gets the present wavelength position of the monochromator.

GetFilterWheelPosition – Gets the present filter wheel position.

GetNDFilterWheelPosition – Get the present ND filter wheel position.

GetChopperFrequency – Get the present chopper frequency.

PerformDarkCurrent – Returns the dark current (may take a while).

GoToWavelengthAndReturnSignal – Returns signal at a wavelength (may take a while).

PerformGratingCalibration – Returns a grating skew angle (may take many minutes).

Properties

Status– Retrieves the current system status.

IntegratedSignal – Retrieves the manual mode signal.

ThreePerSecSignal – Retrieves the 0.33 integration time signal. See also the EnableTimerY property, which must be set to allow updates of this value.

IsConnected – Provides regularly updated connection status of host to controller.

The GetWavelengthPosition, GetFilterWheelPosition, GetChopperFrequency, and GetNDFilterWheelPosition functions can be called at any time during scans or manual mode.

PerformDarkCurrent should be called after the setup function has been called and before GoToWavelengthAndReturnSignal commands are called. This command is for scan mode and may take a few seconds depending on the drives current position, the desired wavelength for the dark current, and the integration time specified for the gain range.

GetWavelengthPosition

int GetWavelengthPosition(ref double wavelengthPosition)

Return Value

0 – Success
1 – OL 750 Not connected

Parameters

None.

Remarks

This function can be called anytime after initial setup information is sent down to the OL 750 Controller. This function can be used in manual mode or scans to give the user the current wavelength position of the monochromator.

GetFilterWheelPosition

short GetFilterWheelPosition()

Return Value

Returns the present filter wheel position of the monochromator. Can be a number from 0 to 10.

Parameters

None

Remarks

This function can be called anytime after initial setup information is sent down to the OL 750 Controller. This function can be used in manual mode or scans to give the user the current filter wheel position of the monochromator.
Specifies the PMT flux overload value. Flux overload is the signal value at which the PMT voltage will be automatically shut down to prevent exposure damage to the PMT detector.

GetNDFilterWheelPosition

int GetNDFilterWheelPosition(ref int filterWheelPosition,ref int NDFilterWheelPosition)

Return Value

0 – Success
1 – OL 750 Not connected

Parameters

filterWheelPosition

int reference value representing filter wheel position in nanometers

NDFilterWheelPosition

int reference value representing ND filter wheel position in nanometers

Remarks

This function can be called anytime after initial setup information is sent down to the OL 750 Controller. This function can be used in manual mode or scans to give the user the current ND filter wheel position of the monochromator.

Note: This function will return meaningless information if you do not have an ND filter wheel installed.

GetChopperFrequency

int GetChopperFrequency(ref double chopperFrequency)

Return Value

0 – Success
1 – OL 750 Not connected

Parameters

Double reference value representing chopper frequency in Hertz

Remarks

This function can be called anytime after initial setup information is sent down to the OL 750 Controller. This function can be used in manual mode or scans to give the user the current chopper frequency.

Note: This information will be meaningless if you do not have a chopper installed.

PerformDarkCurrent

int PerformDarkCurrent(int portNumber, ref double darkCurrent, ref int rangeLevelReturned)

Return Value

0 – Success
1 – OL 750 Not connected
12 – Port number invalid

Parameters

PortNumber

The port number in which to acquire the dark current measurement. Put a 0 for PORT A/A1, a 1 for PORT B, a 2 for PORT A2, and a 3 for PORT A3. Only put 2 or 3 if you have the multiplexer installed.

darkCurrent

Double reference value representing dark current value acquired.

RangeLevelReturned

Pass an int by reference to receive the range level for the signal acquired. Total system gain can be computed with this by raising 10^rangeLevelReturned.

Remarks

This function retrieves the dark current at a specified port. This function will move to the wavelength (or shutter) as defined in the SendDetectorAndGeneralInfo command if the monochromator is not already in position. This function will wait until the drive has stopped moving and the dark current is acquired before returning the signal to the user (this may take some time). The rangeLevelReturned is passed by reference (VisualBasic - ByRef).

GoToWavelengthAndReturnSignal

int GoToWavelengthAndReturnSignal(double wavelength, int portNumber, int gainIndex,
ref double lfData, ref int rangeLevelReturned)

Return Value

0 – Success
1 – OL 750 Not connected
11 – Invalid gain index
12 – Port number invalid

Parameters

Wavelength

The desired wavelength in which to retrieve signal. This wavelength is specified in nanometers.

PortNumber

The port number from which the reading should take place. Put a 0 for PORT A/A1, a 1 for PORT B, a 2 for PORT A2, and a 3 for PORT A3. Only put 2 or 3 if you have the multiplexer installed.

GainIndex

The desired gain index for this signal acquisition. Put a –1 for autoranging or 0 through 9 for a fixed gain.

lfData

Double reference value containing the data from the acquisition.

RangeLevelReturned

Pass a pointer to a long to receive the range level for the signal acquired. Total system gain can be computed with this by raising 10^RangeLevelReturned. Keep in mind that this is a pointer so the long must already be allocated.

Remarks

This function goes to a wavelength and returns a signal from a specified port with the specified gain. This function will move to the wavelength if the monochromator is not already in position. This function will wait until the drive has stopped moving and the signal has integrated before returning the signal to the user (this may take some time). Keep in mind that the rangeLevelReturned is a pointer to an already allocated long passed into this function. For Visual Basic users, this means using the ByRef call.

PerformGratingCalibration

int PerformGratingCalibration(int grating, ref double calibrationFactor)

Return Value

0 – Success
1 – OL 750 Not connected
3 – Invalid grating index

Parameters

Grating

The grating to calibrate. Valid values are 0, 1, and 2.

calibrationFactor

Reference double value containing the angular offset where the grating normal (0.0 nm) was found by the controller.

Remarks

This function automatically finds the grating normal position. This function can take quite a while to complete because of the need to do successive peak searches.

After this function is called, the setup info should be resent including this new skew angle. For a description of how the automatic grating calibration is performed, refer to the *Setup Menu/SETUP/Grating Setup* section of the OL 750 instrument manual.

The following five items are "properties," as contrasted with the above "methods." Values are obtained or set using assignment syntax. For example:

```
int myStat;
myStat= Status;  //gets the current controller status word to local variable myStat.
EnableTimerY = true;  //enables a software timer to periodically update signal values.
```

Status <property>

eOL750Status Status<get>

Return Value

Returns the status of the system as a decimal number. If the system status is nonzero, the controller is handling internal operations as follows:

Normal	= 0x00000000
HomingMonochrometer	= 0x00000001
PositioningMonochrometer	= 0x00000002
ShutteringSource	= 0x00000004
PositionFilterWheel	= 0x00000008
DarkCurrentMeasuring	= 0x00000010
LockInAmpCalibrating	= 0x00000020
SystemFluxOverloa	= 0x00000040
ChopperFrequencyPositionSetting	= 0x00000080
DetectorCooling	= 0x00000100
DetectorSlideMoving	= 0x00000200
WavelengthInvalid	= 0x00000400
ChangingGrating	= 0x00000800
InvalidControllerConfig	= 0x00004000
ControllerReset	= 0x00008000
EntranceSlitMoving	= 0x00010000
ExitSlitMoving	= 0x00020000
PMTDetectorStabilizing	= 0x00040000
PositionAutoMonochromatorRail	= 0x00080000
PositionOL750_20	= 0x00100000
PositionOL750_75MA_YawPitchRoll	= 0x00200000
PositionOL750_75MA_RollIncident	= 0x00400000
NDFilterNotInstalled	= 0x01000000
ReadyForTriggeredEvent	= 0x02000000
ReadyForExternalTriggerOrKey	= 0x04000000
PIRechargin	= 0x80000000

Parameters

None.

Remarks

This function retrieves the controller status. This bit mask is retrieved from the controller approximately three times a second. This function is for information purposes and is typically used by the pulse integrator to determine if the trigger is ready.

IntegratedSignal <property>

float IntegratedSignal <get>

Return Value

Returns the signal integrated with the ManualSetIntegrationTime command.

Parameters

None.

Remarks

This function returns the present signal integrated by the manual mode integration time. You should be in manual mode (SetDetectorAndGeneralMeasurementInfo function parameter measMode set as 1).

ThreePerSecSignal <property>

float ThreePerSecSignal <get>

Return Value

Returns the signal integrated at 0.33 seconds.

Parameters

None.

Remarks

This function grabs the signal integrated at 0.33 seconds. This signal is presented with the status information three times a second.

IsConnected <property>

bool IsConnected <get>

Return Value

Returns the connection status between computer and controller.

Parameters

None.

Remarks

This function determines whether the OL 750-C Controller has been connected.

Miscellaneous Commands

The miscellaneous commands are the commands that don't fit into any of the previous categories. They involve accessory equipment and special measurement parameters.

SendNDFilterWheelTransmittances – Needs to be called before each GoToWavelenthAndReturnSignal command to setup up the ND filter wheel transmittance value.

SetPostMonoAndFilterDelays – Should be sent after setup commands to establish the drive delays.

SendNDFilterWheelTransmittances

void SendNDFilterWheelTransmittances(double Wavelength, double TransmittanceForFilter1, double TransmittanceForFilter2, double TransmittanceForFilter3, double TransmittanceForFilter4, double TransmittanceForFilter5);

Return Value

None.

Parameters

Wavelength

The wavelength, in nanometers, in which these transmittance values apply.

TransmittanceForFilter#

Where # is the filter number. This is the transmittance value for the filter specified.

Remarks

This function sends down the transmittance values for the ND filters for the wavelength specified. This function should be called before each GoToWavelengthAndReturnSignal command call with the wavelength desired for acquisition. Not sending this information down to the controller could cause the controller to choose the wrong ND filter wheel position.

SetPostMonoAndFilterDelays

void SetPostMonoAndFilterDelays(double MonochromatorDelay, double FilterWheelDelay);

Return Value

None.

Parameters

MonochromatorDelay

Specifies the monochromator move delay in seconds. If you are unsure, use 0.0.

FilterWheelDelay

Specifies the filter wheel move delay in seconds. If you are unsure, use 2.0.

Remarks

The Post Monochromator Change Delay and Post Filter Wheel Change Delay are internal delays that are used by the OL 750 as a minimum mandatory delay following a monochromator move or a filter wheel move during signal acquisition. No signal integration will occur until the system response settling time delay and the post change delay have expired.

WARRANTY

Gooch & Housego warrants that all goods supplied will be of the kind described or in any specification and drawings approved by Gooch & Housego and will be free from defects in material and workmanship for one (1) year from the date of purchase by the original Purchaser. During this period, Gooch & Housego will at its option either repair or replace any goods that are found to be defective in material or workmanship, provided the goods are returned to Gooch & Housego in Orlando, Florida, with all shipping, insurance, and delivery charge prepaid (DDU Orlando, Florida). This Warranty does not extend to batteries, fuses, and glass phototubes (if any), or other items of limited durability, nor does this warranty cover damage to goods caused by leaky or otherwise defective batteries, by Purchaser use of improper batteries and by misuse or abuse, unauthorized alteration, excessive line voltage, excessive temperatures (above 160 degrees F), extreme environmental conditions (such as extremely dusty or wet environment), corrosive atmospheres or servicing by unauthorized personnel. The items returned shall only be accepted when accompanied by a written statement setting forth the nature and suspected cause of the alleged deficiencies. After the initial one (1) year warranty, repairs performed by Gooch & Housego are covered under a limited ninety (90) day warranty. The warranty repaired (and/or replacement) items will be returned DDU to customer. THERE ARE NO OTHER WARRANTIES, EXPRESS OR IMPLIED, (INCLUDING THE WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR IMPLIED WARRANTY OF MERCHANTABILITY) OTHER THAN THE WARRANTY SET FORTH HEREIN.