Instruments

MountainSource-Hyperchromator

Key Features

- Different configurations available
- Fast optics, up to f/1.5 for highest throughput
- Homogenous output distribution due to a proprietary design
- Etendue-matched to ISTEQ XWS-30
- Broad tunable range from DUV to NIR
- No input slit necessary
- Built-In shutter
- Easy to use Software, Windows GUI, LabView on request

MountainSource-Hyperchromator
Operation manual

Version 05

Mountain Photonics GmbH
Albert-Einstein-Str. 18
D-86899 Landsberg am Lech
Germany

Tel: 08191-985199-33
Fax: 08191-985199-99
info@mphotonics.de
https://www.mphotonics.de

Table of Contents

WARNING

This unit emits ultraviolet (UV) radiation that is harmful to humans. Avoid exposure to the direct or reflected output beam. Make certain that the appropriate output beam shields and optics are in place prior to energizing the unit. All interlocks must be satisfied prior to operation; failure to do so may lead to hazardous conditions.

CAUTION

The XWS-30 emits dangerous levels of UV radiation. This radiation is emitted at the Optical Output Port of the Hyperchromator. Even short exposures to skin or eyes may cause burns. Ensure that only authorized personnel are in the vicinity of source during operation. Personnel in vicinity of operating source should wear protective eyewear, clothing, and gloves. Lighted UV warning lights and signs posted on doors to lab areas may help prevent accidental exposure.

WARNING

The XWS-30 controller utilizes an internal Class 4 IR laser capable of causing severe injury to eyes or skin. Do not open or attempt to service this unit. Contact ISTEQ regarding any problems with the unit.

WARNING

The XWS-30 emits dangerous levels of UV radiation. Do not unmount the XWS-30 source from the Hyperchromator during operation. Mounting and unmounting should only be done with the XWS-30 turned off. Power on the XWS-30 only when mounted correctly to the Hyperchromator.

1.1 General precautions

The output beam from the Hyperchromator should be blocked when not in use with an electronic shutter or other appropriate beam blocking device. Due to the possibility of generating ozone when ambient oxygen is exposed to short wavelength light, the beam should always be enclosed in an appropriate beam pipe, tube, or enclosed space.

The Hyperchromator must also be cabled correctly and connected to a socket with a protective earth ground prior to operation.

Refer to the Installation section of this manual for details of the facilities connections.

There are no user-serviceable parts inside the Hyperchromator. For any problems encountered during operation, please contact Mountain Photonics for assistance.

If there is a component failure, do not attempt to open the Hyperchromator, the Power Supply Controller or Lamp House enclosure of the XWS-30. Dangerous invisible infrared laser beams and hazardous voltages exist inside the units.

For continuous operation, the light source manufacturer, Isteq, recommends purging the light source with nitrogen. For more information, please refer to the XWS-30 user manual.

Refer to the user manual of the XWS-30 for further precautions and all matters concerning the light source.

CAUTION

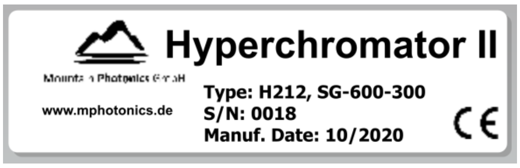
Use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure.

1.2 Labels and safety notification

The following safety labels appear on the product. The picture below shows the location of each label on the Hyperchromator. Please refer to the user manual of the XWS-30 for labels on the attached light source.

UV hazard warning label: indicates hazardous levels of UV light are present.

Optical radiation warning label: indicates strong optical radiation.



Manufacturer’s identification label:
Gives the manufacturer’s name and address, the model type, serial number, date of manufacture of the equipment and the label that show CE conformity.

1.3 Conformity

The Hyperchromator meets the requirements of EC Directives LVD(2014/35/EU) and EMC(2014/30/EU); it is marked with the CE mark. The product is designated as Class 1 during all normal operation.

2. System description

The Hyperchromator is a high throughput monochromator designed for the ISTEQ XWS-30 light source. Due to its extremely high radiance, the XWS-30 is especially well suited for generating monochromatic light in the wavelength range 220 nm – 2200 nm (UV/VIS/NIR). Bandwidths of 1 nm to 10 nm are possible.
The light is collected directly from the plasma of the lamp with an aperture of up to f/1.5 without using an additional entrance slit. This makes this tunable light source very efficient.
The output side has been designed with a very flexible opto-mechanical interface. This allows for a multitude of illumination or light coupling options using standard catalog components, rendering the integration of the Hyperchromator into your setup hassle free and straight-forward. Possible configurations include fiber coupling, collimated or free-beam output.
The wavelength is selected via USB interface from a PC or laptop. An easy-to-use software is provided.

General specifications:

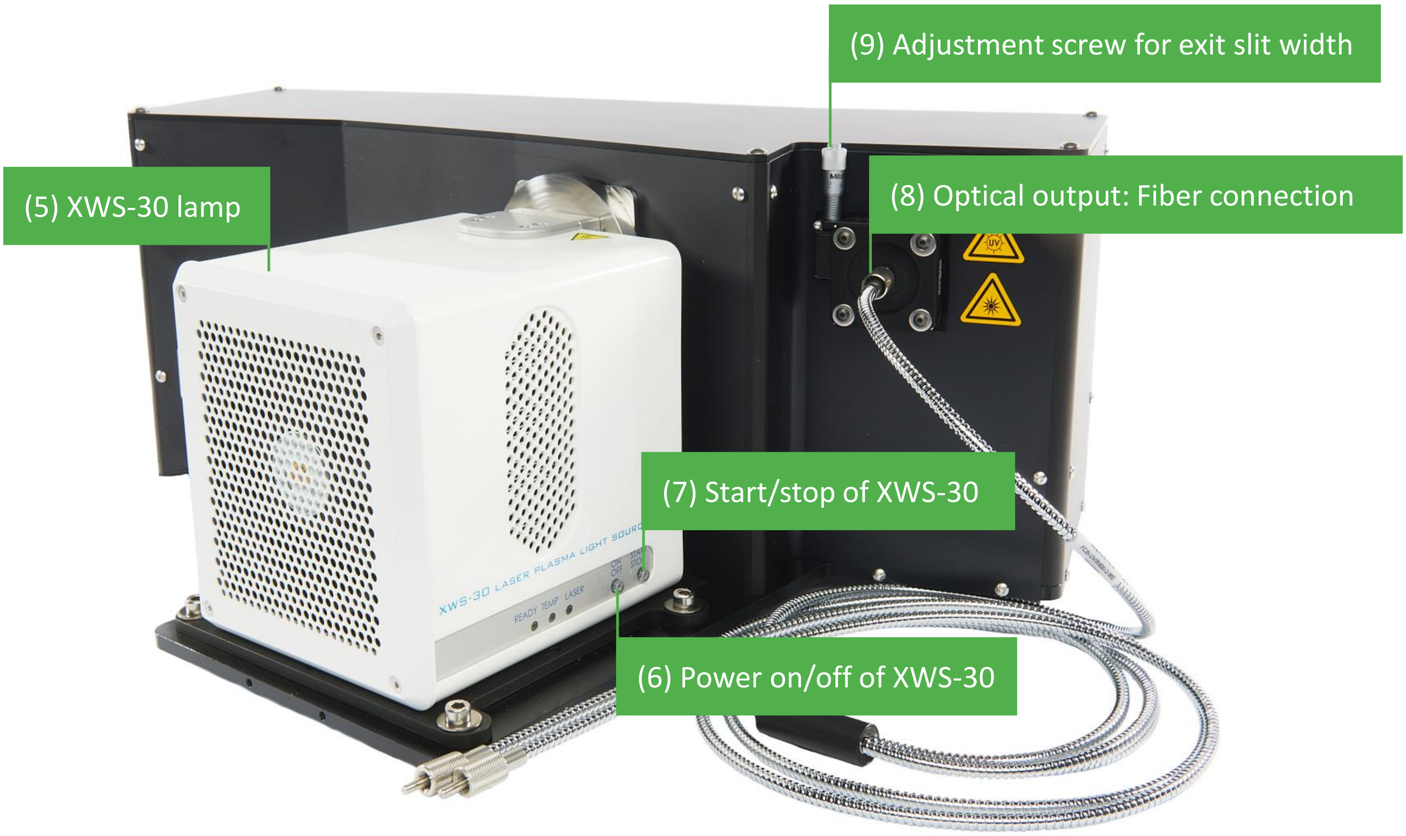
Optical input	ISTEQ XWS-30 light source, directly coupled (optionally many other light sources)
Optical output	Fused silica fiber, SMA or FC, 100-600 µm core or free beam output with adjustable slit or various collimator options. Spectral power monitoring on request.
Wavelength range	190 – 2200 nm*
Aperture	f/1.5
Bandwidth	1 – 20 nm FWHM*
Output power	Up to 800 µW (grating at blaze wavelength, 6 nm bandwidth and 400 µm fiber)
Reproducibility	Typ. 0.1 nm
Scanning speed	40 – 100 nm/s*
Control interface	USB/RS-232, LabVIEW™-based GUI, various external control options
Dimensions and weight	46,3 x 35,3 x 18,7 cm; 13 kg

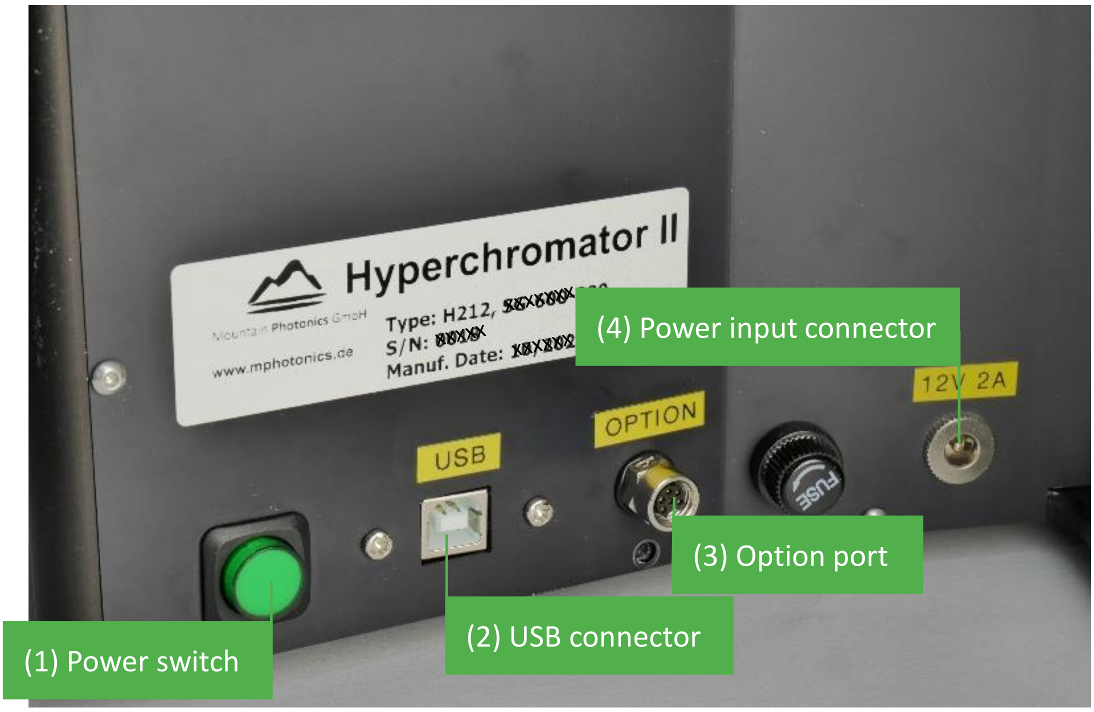
*depends on choice of grating and other requirements.

Delivery Items:

- Hyperchromator single grating SG or dual grating DG main body with XWS-30 attached
- Power Supply 12V/2.5A for Hyperchromator
- USB cable for Hyperchromator
- Interlock plug for XWS-30
- Power supply for XWS-30
- USB box and cable to connect the XWS-30 to a PC or laptop
- USB stick with software and manual

Hyperchromator with XWS-30





The following table provides a description of the system components and controls shown in the figure above.

(1)	Power switch	Power on/off of the Hyperchromator. The button shows a green light then the Hyperchromator is switched on.
(2)	USB connector	Interface to PC/laptop
(3)	Option port	Provides access to I/O ports for external interlock and others, see Appendix A.
(4)	Power input connector	Connects to the delivered power supply 12 V/2.5 A
(5)	XWS-30 lamp	The XWS-30 light emitting source (refer to user manual of the XWS-30)
(6)	Power on/off of XWS-30	To turn on the lamp first press this button for 2 sec., then proceed with (3).
(7)	Start/stop of XWS-30	After the lamp has been activated with (2) press this button to turn on the lamp.
(8)	Optical output	Here the monochromatic light exits the Hyperchromator. The optical output is designed for compatibility to common catalogue components. Standard configurations are fiber coupling, free-beam output with (adjustable) slit or collimators with different focal lengths. Fiber coupling can be combined with a manual or motorized slit to adjust the bandwith.
(9)	Adjustment screw for slit width	Adjustment screws for adjusting the slit width of the exit

3. Installation

3.1 Unpacking

Upon arrival, start by inspecting all parts of the system for completeness and any damage incurred in shipping. The Hyperchromator shipping box should contain:

- Hyperchromator SG or DG main body equipped with XWS-30 attached
- Power Supply 12V/2.5A for Hyperchromator
- USB cable for Hyperchromator
- Interlock plug for XWS-30
- Power supply for XWS-30
- USB box and cable to connect the XWS-30 with PC or laptop
- User manual of ISTEQ XWS-30
- USB stick with software and operation manual

Use care when unpacking to avoid damaging the armored fiber optic cable of the XWS-30.
If any part is missing or appears damaged, contact Mountain Photonics immediately.
Do not attempt to substitute any parts.

3.2 Connections

3.2.1 Electrical Power

For connection of the XWS-30 Power supply controller refer to the user manual of the XWS-30.
You also must install the interlock plug to make the lamp work (see picture).

The Hyperchromator needs 12 VDC at 2.5 A minimum.
Connect the power supply to the Hyperchromator via the power input connector (4).

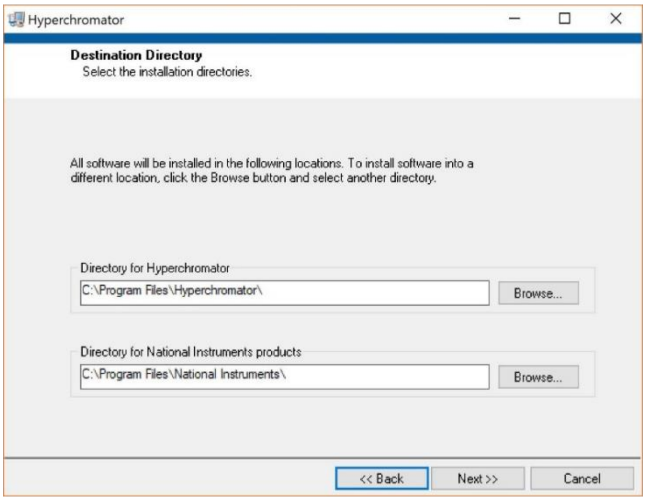
3.2.2 Control via PC/Laptop

Connect the Hyperchromator via the USB cable with a USB port of your PC or Laptop.

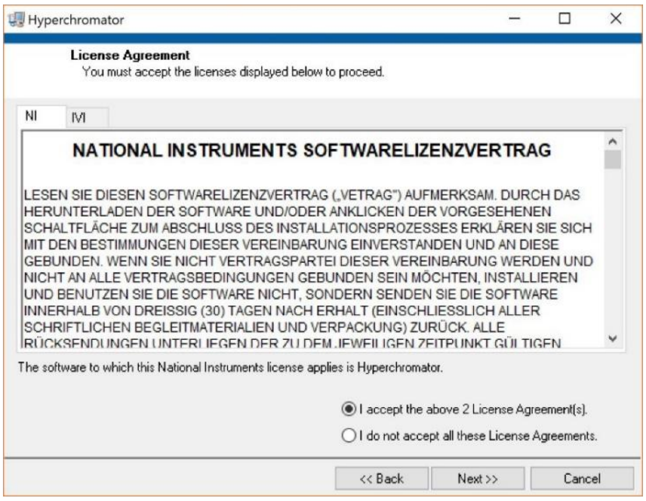
4. Software installation

Before installation make sure that the Hyperchromator is not connected to your computer.
Plug the USB stick that came with your Hyperchromator into your computer. You will find an executable named “setup.exe” on the USB stick. Start setup.exe and follow the instructions.

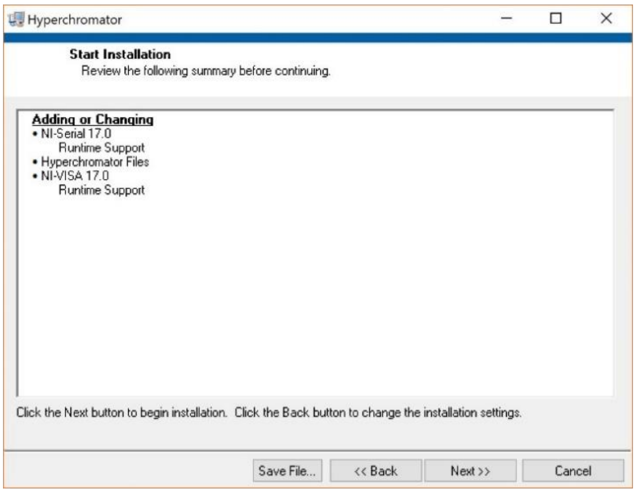
1. The installer will ask you for the destination of the installation. It is recommended to stay with the default directories and just click on “Next”.



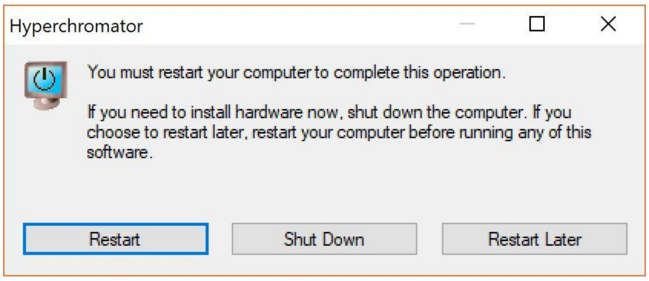
2. Select “I accept the above 2 License Agreement(s)” and click “Next”.



3. Start the installation process by clicking “Next”. This step can take some time to finish. After the installation is finished a window “Installation completed” will open. Just click “Next” here.



4. After finishing the installation, a window will pop up and ask you to restart your computer. Make sure to close all your applications and save your data. Then click “Restart”



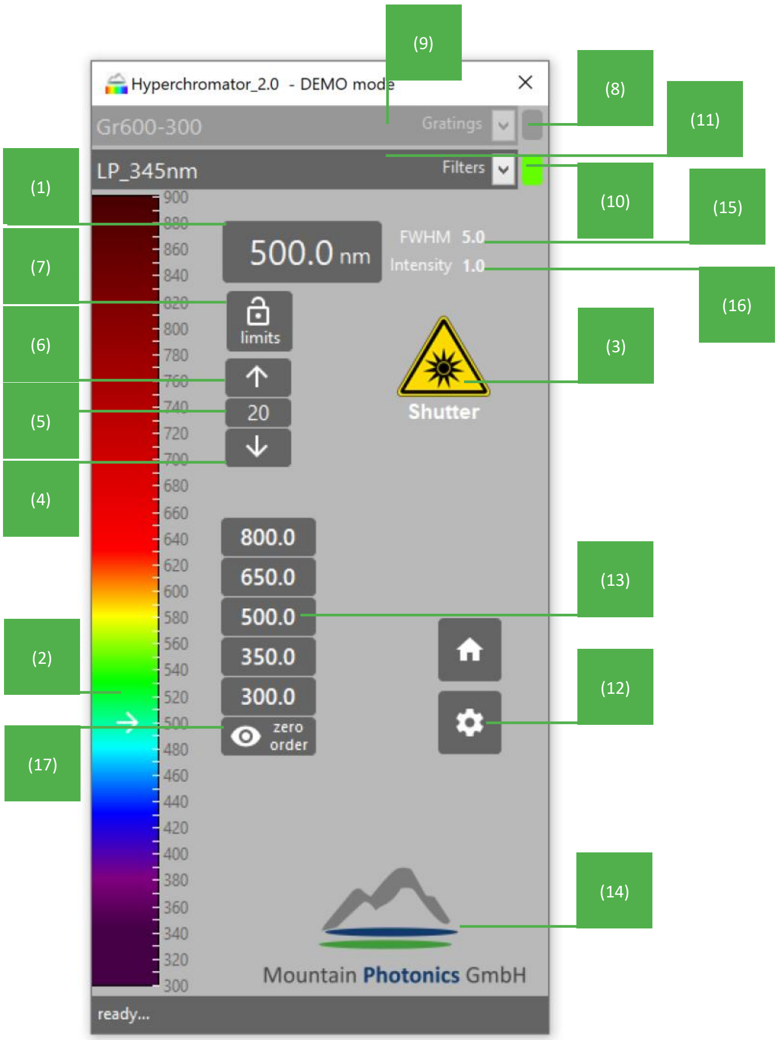
Before starting the Hyperchromator control software, connect the Hyperchromator with your computer via the USB-to-serial cable and power it on.
To start the Hyperchromator software, start the executable “Monochromator.exe” in the destination directory of the Hyperchromator installation. You will find a shortcut on your desktop.
When first starting the Hyperchromator software, a window of the windows defender might pop up and ask if you want to allow the Hyperchromator access your network. If you want to remote control the Hyperchromator via tcp/ip, you must allow this access.

5. Using the software

The Software of the Hyperchromator lets you control the wavelength of the output light. In addition, you can control a shutter that blocks the light right behind the lamp to switch on/off the output light.
The Hyperchromator has a motorized filter wheel with 4 positions right before the output port. This filter wheel is equipped with order sorting filters. These filters make sure, that light from higher orders is blocked. The filter wheel will select the right filter for the selected wavelength when the auto-mode is active. Note that you might have a fraction from higher orders in the output light when the filter wheel is not in auto-mode.

5.1 Main user interface

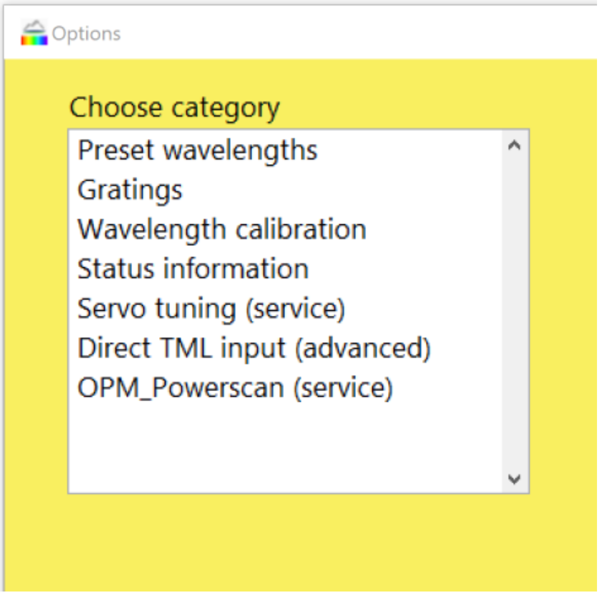
The picture below shows the GUI of the software with numbered controls. The following table explains their functions.



(1)	Wavelength	Here you can enter a wavelength with your keyboard. The Hyperchromator will switch to this wavelength.
(2)	Wavelength dial	With the wavelength dial you choose the wavelength with the mouse by clicking and holding the left mouse button on the dial and moving it to a new position.
(3)	Shutter	By clicking on the shutter symbol, you can toggle between open and closed shutter. The shutter is open when the symbol is yellow and closed when grey.
(4)	Step minus	By clicking on the arrow, the wavelength will be decreased by the number (5) shows in nanometers.
(5)	Step size	Size of the step taken when clicking on (4) and (6) in nanometers.
(6)	Step plus	By clicking on the arrow, the wavelength will be increased by the number (5) shows in nanometers.
(7)	Limits	Wavelength range limits defined in the settings can be switched on and off. When switched off, the user can set a wavelength outside of the defined range of the Hyperchromator.
(8)	Auto mode for gratings	When green, the Hyperchromator will choose the grating for the actual wavelength according to the configuration in the settings (13).
(9)	Gratings	Shows the actual grating that is in the beam path. You can choose a grating from the drop-down menu if you have a dual grating Hyperchromator. Note that you might not get the maximal possible output power when not choosing the right grating.
(10)	Filter wheel auto-mode	When green, the filter wheel is in auto-mode and will choose the correct order sorting filter for the actual wavelength.
(11)	Active filter	Shows the actual filter in the filter wheel that is in the beam path of the output light. You can choose a filter form the drop-down menu. Note that you might have a fraction from higher orders in the output light when you don't select the proper order sorting filter.
(12)	Settings	Here you can set the pre-set wavelength, change the calibration and configure the gratings. See the following sections.
(13)	Pre-set wavelength	Offers 5 pre-set wavelengths that can be defined by the user in the settings (12). When clicking on the pre-set wavelength the Hyperchromator will go directly to the defined wavelength.
(14)	About	Shows general information about the software like software version.
(15)	FWHM	Shows the current bandwidth of the spectral peak of the output light.
(16)	Intensity	Shows the intensity that has been stored in the calibration table.
(17)	Zero order	The grating inside the Hyperchromator will go to the zero order position where the entire spectrum of the lamp will be present at the output port.

5.2 Settings

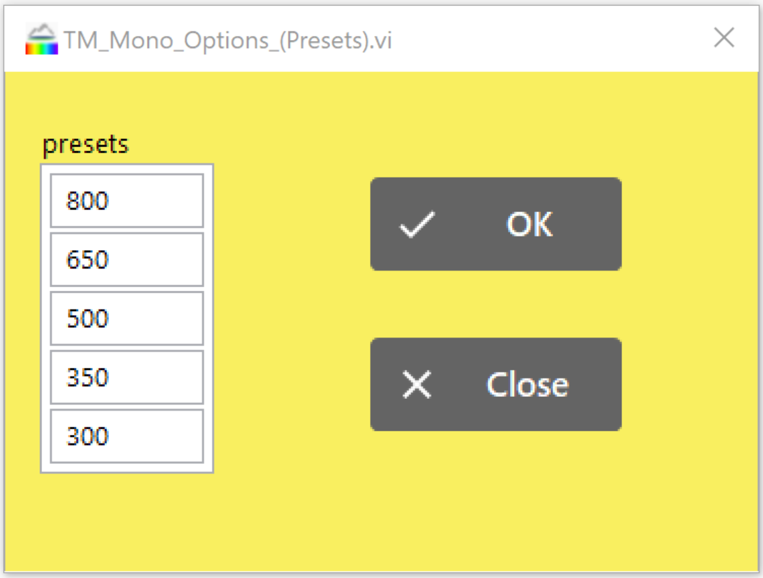
When clicking on the Settings button (12) the window shown below will be opened. Click on an item in the list to open the corresponding window. The options “Status information” and following are not recommended to be used by users and are for service only.



The settings will refer to the grating that is active and shown in the “gratings” drop down menu (10). Choose the grating that you want to calibrate before you go to settings (12).

5.2.1 Preset wavelengths

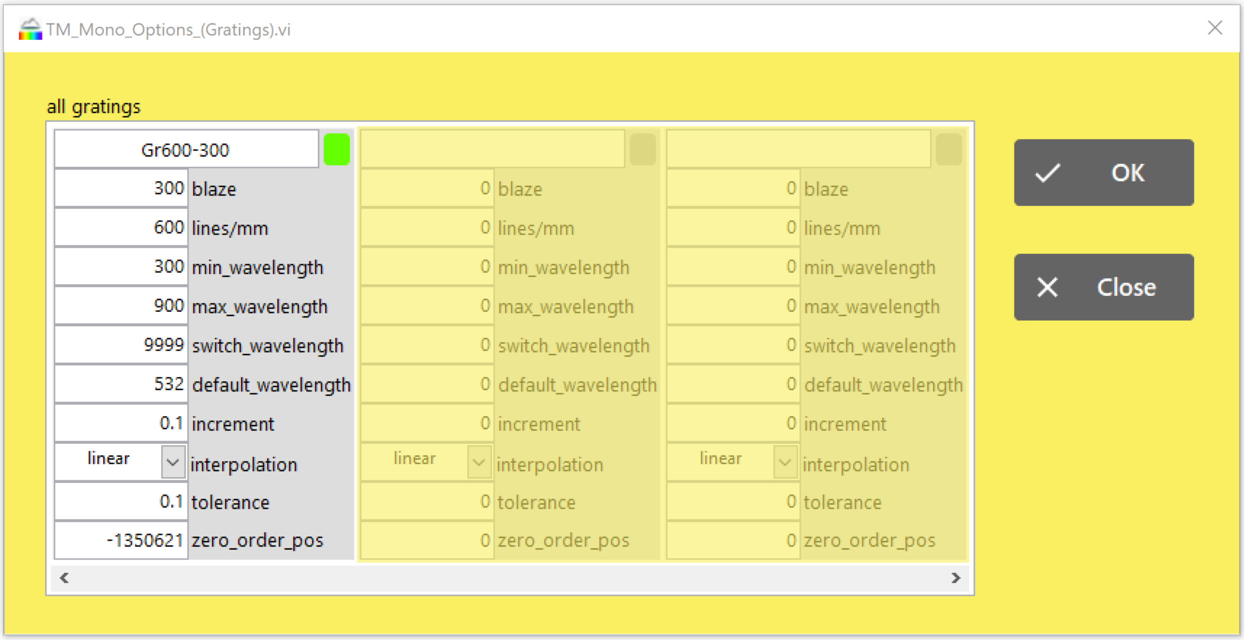
Here you can set your preset wavelengths that are shown in the main window (13)



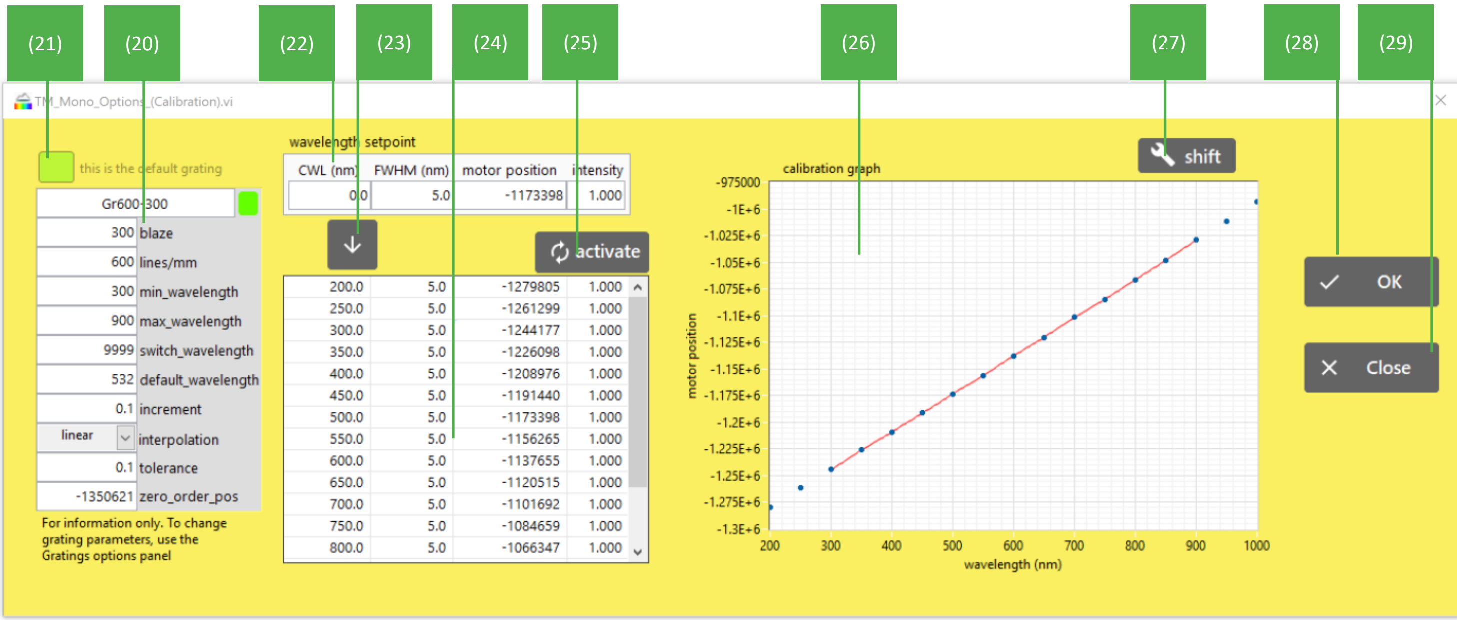
5.2.2 Gratings

Here you can change the settings of your gratings.

- **Name** for the grating to appear in the “gratings” drop down menu (10).
- **blaze:** The blaze wavelength of the grating.
- **lines/mm:** Lines per mm of the grating. This value determines the bandwidth of the output light.
- **min_wavelength:** When “limits” (7) are activated, the user cannot choose a wavelength below this value.
- **max_wavelength:** When “limits” (7) are activated, the user cannot choose a wavelength above this value.
- **switch_wavelength:** When “auto mode” for the gratings (9) is activated, the Hyperchromator will automatically switch from one grating to next when coming across this wavelength
- **default_wavelength:** The wavelength that is selected right after starting the software.
- **increment:** The smallest step size in which the user can change the wavelength.
- **interpolation:** The method used to interpolate the values between the calibration points in the following table. “linear” is recommended.
- **tolerance:** When the Hyperchromator is within “tolerance” of the set wavelength (1), it will consider the actual motor position as “on target” and stop the motor.
- **zero_order_pos:** The motor position for the zero order of the grating.



5.2.3 Wavelength calibration



(20)	General grating settings	**label:** Name for the grating to appear in the "gratings" drop down menu (10). **blaze:** The blaze wavelength of the grating. **lines/mm:** Lines per mm of the grating. This value determines the bandwidth of the output light. **min_wavelength:** When "limits" (7) are activated, the user cannot choose a wavelength below this value. **max_wavelength:** When "limits" (7) are activated, the user cannot choose a wavelength above this value. **switch_wavelength:** When "auto mode" for the gratings (9) is activated, the Hyperchromator will automatically switch from one grating to next when coming across this wavelength **default_wavelength:** The wavelength that is selected right after starting the software. **increment:** The smallest step size in which the user can change the wavelength. **interpolation:** The method used to interpolate the values between the calibration points in table (24). "linear" is recommended. **tolerance:** When the Hyperchromator is within "tolerance" of the set wavelength (1), it will consider the actual motor position as "on target" and stop the motor. **zero_order_pos:** The motor position for the zero order of the grating.
(21)	Default grating	If this box is checked, the actual grating will be selected when the software is started. To change the default grating go to "Gratings" in the item list.

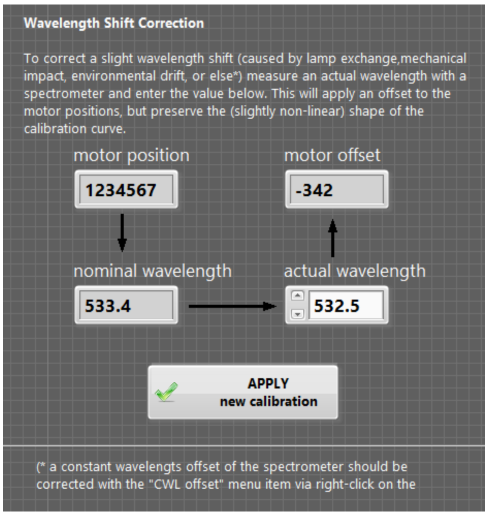
(22)	Calibration point	In this section you can generate a new calibration point for the calibration table (24). To add the new point for the calibration table, click on (23). CWL: Calibration wavelength. Type in the wavelength that you want to assign to the current position of the grating. Typically, this value will be given by a spectrometer connected to the output port of the Hyperchromator. FWHM: Type in bandwidth of the output light. This value depends on the slit width or fiber diameter and is for information only. motor pos: The actual position in counts of the motor that rotates the grating. You can rotate the grating directly from here by typing in a number, by using the up/down arrows on the left, or by positioning the cursor on a digit and incrementing/decrementing with the up/down arrow keys off the keyboard or by using the scroll-wheel of the mouse. Intensity: An additional value which can be set to give the user an intensity value for the chosen wavelength.
(23)	Enter calibration point	By clicking on this arrow, you add a new calibration point to the calibration table (24), given by the values above (22).
(24)	Calibration table	This table defines the calibration of the grating. Each point is a set of a wavelength (CWL), bandwidth (FWHM) and a motor position (motor pos). The table must contain at least 2 entries. By **right-clicking** on the table area, a pop-up menu opens that let you select various editing options.
(25)	Activate calibration	Clicking on this button will activate the actual calibration table. The Hyperchromator will then set the wavelength according to the actual calibration table.
(26)	Calibration graph	Shows a graph of the calibration defined in the calibration table (24). It should show a smooth curve close to a straight line when calibration is correct (a very slight curvature is expected)
(27)	Shift	See section 6.3.1
(28)	Ok	Will close the calibration window and save all changes. You will be prompted to „Overwrite…“ for safety.
(29)	Close	Will close the calibration window without saving the changes.

5.3 Calibration

In this section you learn how to calibrate the wavelength of the Hyperchromator. Each grating has a calibration table (24). In the first column you find the calibration wavelength (CWL), in the second the bandwidth of the output light and in the third the motor position which gives the angle of the diffractive grating inside the Hyperchromator. The table should have entries which cover the used wavelength range. There should be an entry at least every 50 nm to get an accuracy of about +/- 1 nm.
To measure the wavelength and bandwidth of the output light of the Hyperchromator connect the fiber that you use for your application to a spectrometer. Make sure to reduce the intensity of the light so it fits to your spectrometer. A good way to do this is to use a second fiber that goes into the spectrometer and connect is with the output fiber of the Hyperchromator via a coupler with very low efficiency.

5.3.1 Correct an existing calibration table

If you measure a constant offset between the nominal wavelength that the GUI shows you and the actual measured wavelength, you can use an integrated algorithm to correct this mismatch. To do so right click on the calibration table and then choose “WL shift correction”. A window will appear where you can enter the actual wavelength that your spectrometer gives you for the output light of the Hyperchromator (see figure below). The software will calculate the offset of the motor position to correct the mismatch.



When you click on “APPLY new calibration” the calibration table will be changed accordingly.

To save your new calibration table click on the OK button (28).

When you have a mismatch between measured and nominal wavelength only in a certain wavelength range you can correct it by adding a new entry to the calibration table and eventually delete a previous entry from the table. To add a new entry, move the grating of the Hyperchromator so you measure the desired wavelength. This can be done with the Front-GUI or by directly type in a motor position. You can also change the motor position in small steps by left clicking on the motor position.

Then put the cursor to the digit that you want to increase or decrease with the left/right cursor keys. Use the up/down cursor keys on your keyboard to count up or down the chosen digit. Now type in the measured wavelength and bandwidth into the field CWL and FWHM (22) and click on the arrow (23) to finally add the entry to the table.

To delete an entry right click on its row and chose “Delete row”.

Don’t forget to save your new calibration table by clicking on the OK button (28).

5.3.2 Wavelength calibration from scratch

To make a new calibration table first delete the existing table by right clicking on the table and choosing “Clear All”. Move the grating to the position where you measure the wavelength that you want to start with like described in the section above. Enter the measured wavelength and bandwidth into the field CWL and FWHM (22) and click on the green arrow (23).
Move to the next wavelength and generate the next entry in the same way. Keep steps below 50 nm.
Don’t forget to save your new calibration table by clicking on the OK button (28).
Check that you calibrate the first order of the grating. There should be no output light with higher wavelength.
The table below gives you the rough number of motor steps for a wavelength change of 1 nm.

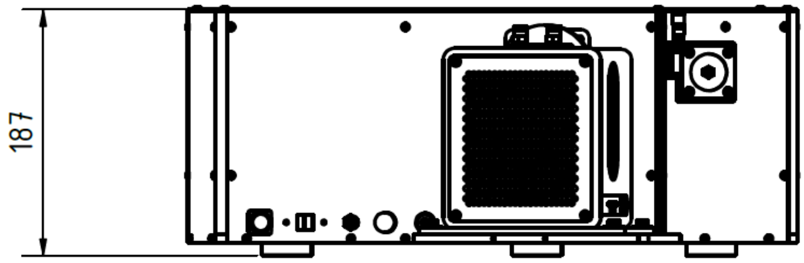
Grating lines per mm [l/mm]	Motor steps for 1 nm
300	175
600	350
1200	700

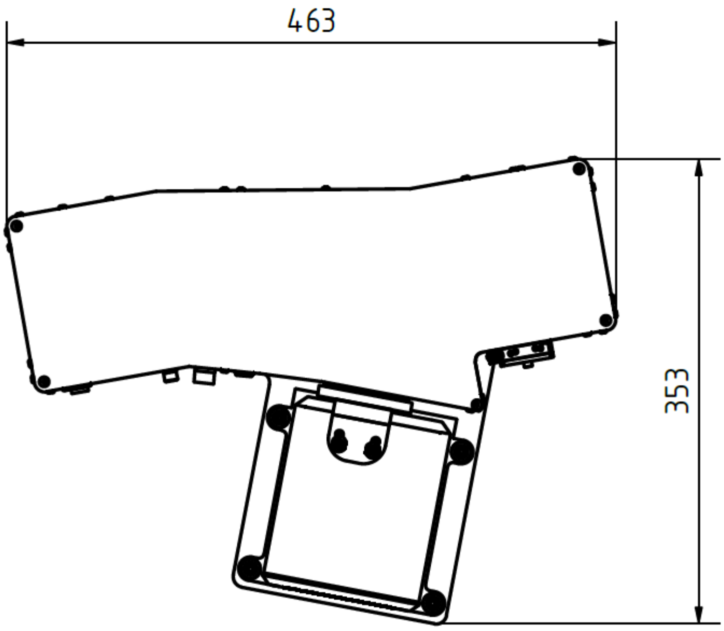
5.3.3 Calibration with high orders

When your spectrometer can only measure for instance the visible range from 400 to 800 nm, you still can calibrate wavelength above 800 nm by using the higher orders of the grating. To do so, switch off the auto filter option (11) and put the filter (12) to the empty position. When you go from i.e. 800 nm you will see a peak at 400 nm too. You can use this second order which has half the wavelength to calibrate from 800 nm to 1200 nm. You can use the third and fourth order in the same way for higher wavelengths.

6. Dimensions

Find below the dimensions of the Hyperchromator. Units are millimeters [mm].





7. Appendix A: High Speed Shutter (optional)

The optional High Speed Shutter (HSS) uses an advanced bi-stable iris diaphragm and allows for a faster opening or closing of the emission in comparison to the standard shutter. This may be useful for synchronization with an external device, e.g. for a camera. Additionally, it allows for precise dosimetry for measurements or application of actinic light. This is accomplished by a programmable precision timer independent of the host PC. The HSS is implemented on the exit side of the Hyperchromator.

The HSS may be operated and programmed from the main program (GUI). However, to be able to access the controls, it must be enabled. This can be done in the calibration window. On delivery, the HSS is enabled and usually will be opened after starting up the instrument.

While there are basic controls provided in the GUI to open or close the shutter, and also to open it for a programmed time, more importantly it may be electrically interfaced to enable precise hardware synchronization. To this end, the OPTION port of the Hyperchromator with HSS provides a TTL trigger input line (TrgIN), see below.

There are basically 2 modes of operation, each with 2 signal polarities:

a) Timed operation: the shutter will open for a pre-programmed time and close afterwards. The event is triggered by an edge on TrgIN. The direction of the edge can be programmed to active_high or active_low. In the active high case, the input line TrgIN is in high impedance state and waits for a transition from 0 to +5 V. It has shown that the input is sensitive to ESD stray signals, so it is recommended to pull-down TrgIn to GND with a 1 kOhm resistor. Similarly, in the active_low state, the input TrgIn is pulled up (internally via 10 kOhm) to +5 V, and TrgIN waits for being connected to ground GND. It may be useful to externally pull-up the input via a 1 kOhm resistor to +5 V.

b) Direct operation: the shutter remains open as long as the signal on TrgIN is either high (+5 V) for active high mode or low (GND) for active_low mode.

The state of the HSS may be monitored on the option connector via a built-in Hall sensor. This sensor operates at ca. 80 % open position of the diaphragm and provides software-independent signal. See chapter “timing details”.

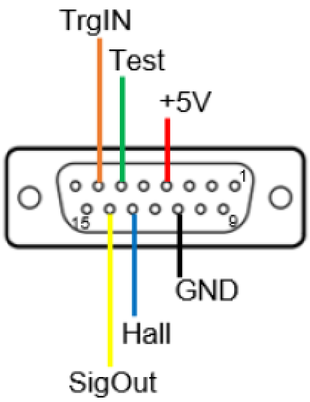
For testing of the connections, a test pin is also provided, which allows one to emulate the connection of a trigger signal. The test pin is actuated from the calibration window only.

7.1 Connecting the HSS

Electrical connection of the shutter is done via the option port.

On the Hyperchromator, this is a female 15-pol D-SUB connector next to the RS-232 connector:

Pin on option port 15-pol D-SUB	Extension wire cable color	Function
11	Black	GND, ground
7	Orange	TrgIN, trigger input, TTL
13	Blue	Hall sensor output, TTL
14	Yellow	Signal monitor, servicing only
6	Green	Test pin (Output)
4	Red	+5 V, 100 mA



A bifurcated cable for the option port has been provided as a starting point giving access to the control lines of the HSS. The option port is also being used for switching on the XWS-30 light source as a slave device by sensing the Hyperchromator´s +5 V vs GND. Should you require to modify the cable, please keep these 2 wires.

Note that the TrgIN (orange wire) has been pulled-down to GND (black wire) with a 1 kOhm resistor for immunity vs. stray signals. This is correct for active_high mode, see above. For active_low mode, the TrgIN may be pulled-up with this resistor to +5 V, if required (there is an internal 10 kOhm pullup already in place in this mode). Should you setup the HSS to active_low mode, please modify accordingly. If TrgIN and GND is connected to an external circuit (TTL levels), no pull-up/pull-down resistor is required.

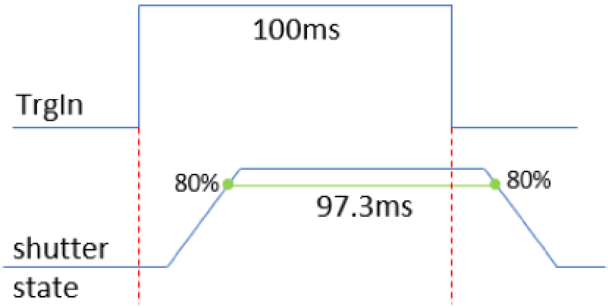
With wiring as described above, you can trigger the function of the HSS by connecting TrgIN to +5 V (red wire) in active_high mode, or by connecting TrgIN to GND (black wire) in active-low mode. Note however, that if you simply use a wire to test it, you will usually trigger the HSS multiple times each time the wire touches ("contact bounce").

7.2 Timing details

The total opening time of the shutter is in the range of 10 ms, about half of this is the actual transfer time of the shutter blades. Therefore, the minimum exposure time may not be less than ca. 20 ms. Similarly, some delay is required between subsequent calls to the shutter. Otherwise the HSS may lose the closing/opening pulse and remain open/closed. Therefore, the minimum programmable open time has been lower-limited to 20 ms (via the GUI only).

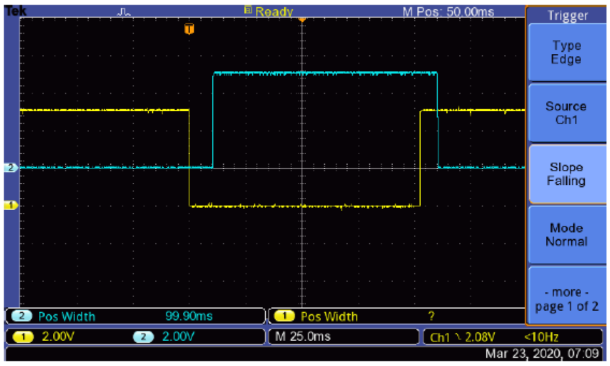
The reproducibility of the open time typically is in the range of +/-100 µs, corresponding to 0.1 % variation on a 100 ms open interval.

As mentioned, the Hall sensor monitor switches state at ca. 80% open position (see green points below). Thus, some hysteresis exists regarding the "real" opening time. As the diaphragm blades are closing the area of the aperture in a non-linear way, and also the illumination of the aperture is not homogeneous, the actual light emission dose may not be properly reflected neither by the programmed or directly triggered opening time, nor the Hall sensor.



This effect will cause the dose of light emitted to show a certain deviation from linearity. For example, a 200 ms emission would have a slightly more than double dose of light. This depends also -to a small amount- on the fiber diameter connected to the instrument. Thus, for the highest precision required it would be advised to calibrate the exposure time by independent means. This may, for example, be a photodetector with fast and linear response and an oscilloscope.
For the timed mode of operation, we have implemented a TIME_OFFSET which allows one to compensate for the hysteresis of the Hall sensor or to achieve a precisely linear response for different exposure times. TIME_OFFSET, which is in µs, is accessible via the calibration window.

Timing details: after issuing a timed opening (100 ms), the Hall sensor (blue) shows a delay of ca. 10 ms between the active-low trigger signal (yellow) due to the transfer time of the diaphragm. The oscilloscope reads 99.90 +/- 0.1 ms for the 80 % open state condition. This has been achieved by providing a device-specific TIME_OFFSET of 2700 µs.



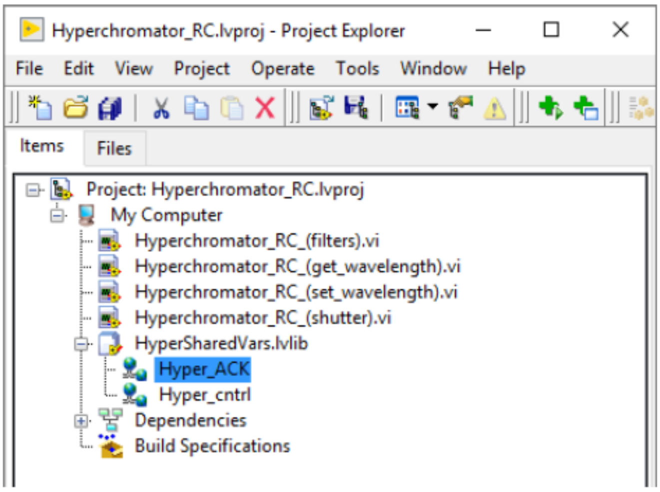
8 Appendix B: Remote control (RC) from LabVIEW

Hyperchromator remote control from another LabVIEW (LV) application or VI may be established using the NI Shared Variable Engine (SVE), in particular

- communication between the running Hyperchromator GUI and another LV application or development system in the LAN or on the same machine
- featuring high-level commands, e.g. for setting the wavelength
- uses NI Shared Variables (network published)
- as this is TCP/IP-based, it employs Windows' TCP stack, thus is monitored by the firewall. Therefore, you may need to allow access through the firewall. If the (Windows) firewall window pops up on first start, make sure to have your network area (Domain(Work)/Private/Public) properly selected. For simplicity, select all network areas. Then "allow access".

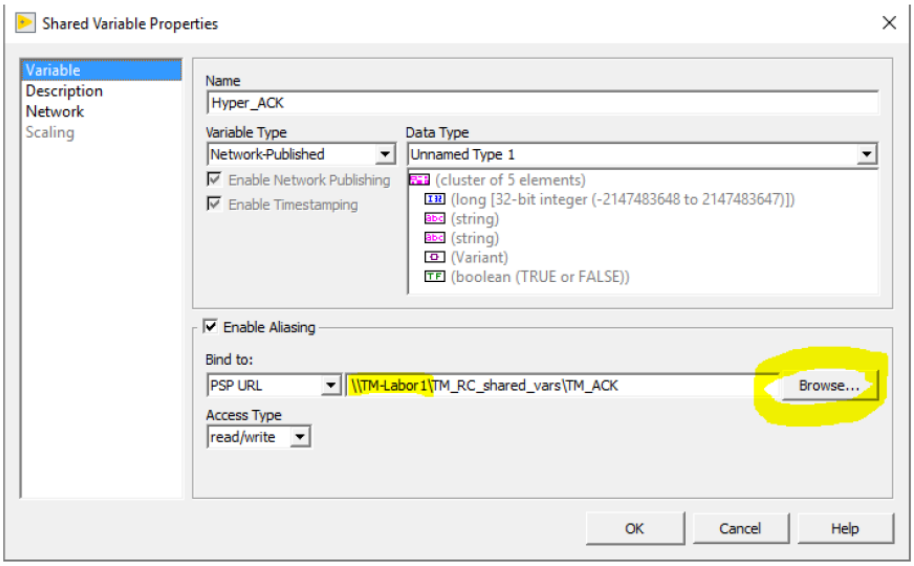
The shared variables are deployed in the background at start of Hyperchromator.exe. The SVE is installed together with the LV runtime engine and started as a background service. There are two variables used, TM_cntrl for control (i.e. to send a command) and TM_ACK for acknowledgement of the receipt or execution of the command. Both variables are published on the network, providing access for the external application or VI.

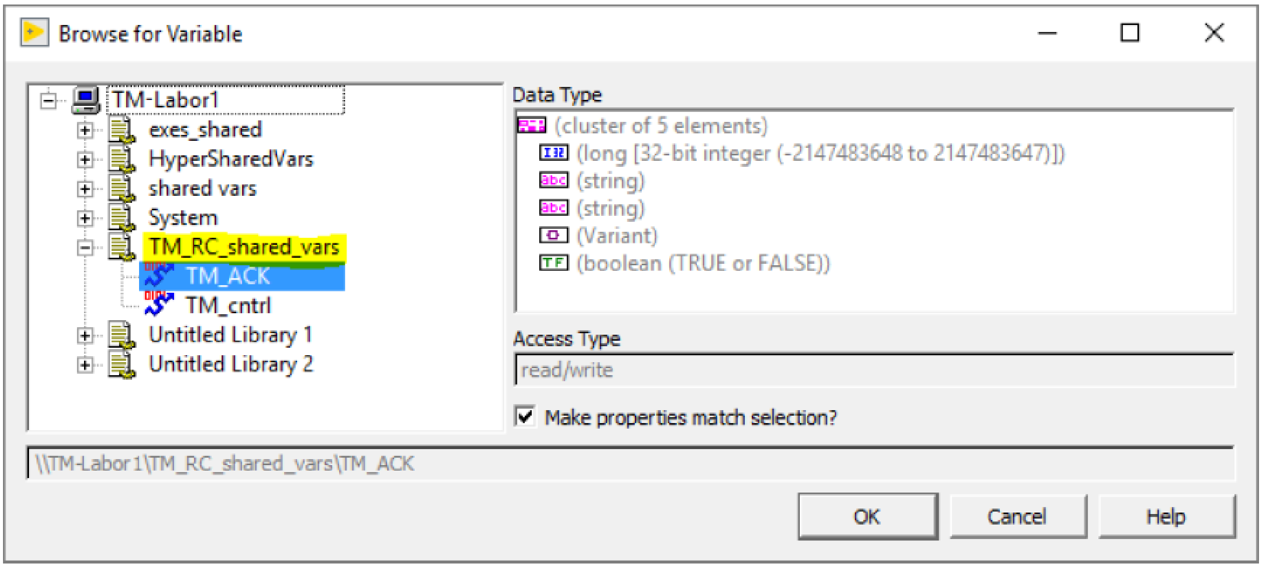
A sample project Hyperchromator_RC.lvproj is provided, which includes the function VIs and the shared variable: With the Hyperchromator.exe (GUI) running, use your LV development system to open …\Hyperchromator_RC\Hyperchromator_RC_VIs\Hyperchromator_RC.lvproj:



Note the shared variable library HyperSharedVars.lvlib with a local instance of the variables for control and acknowledgement, Hyper_ACK and Hyper_cntrl. You will need to link these local instances of the variables to the ones published by Hyperchromator.exe.

Double-click on Hyper_ACK

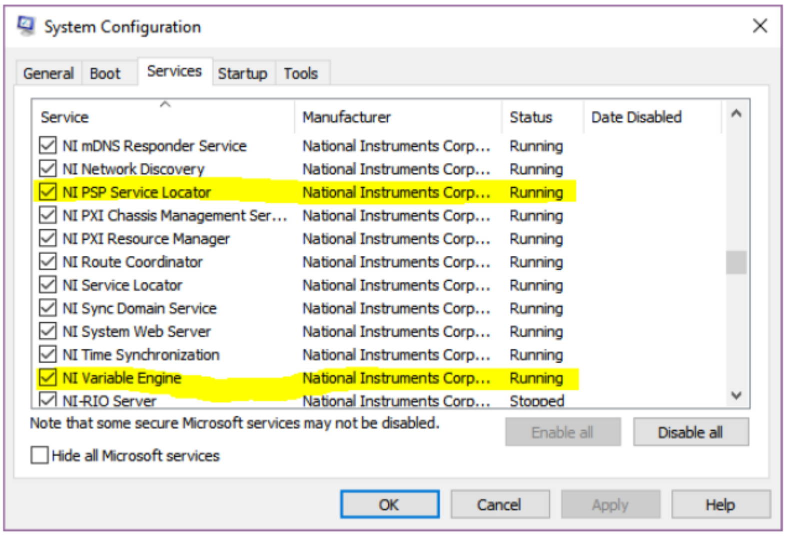




As you can see, at this point it is still linked to our local computer here ("TM-Labor1"), so use the browse button to locate …\TM_RC_shared_vars\TM_ACK in your local environment.

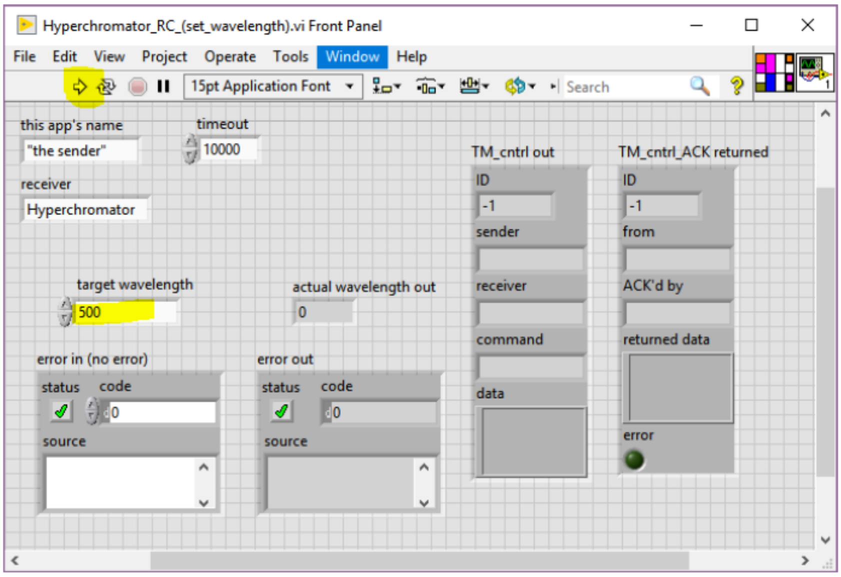
(Note: it is also possible to do this programmatically. Consult LabVIEW for documentation.)

In case you can't see and select any variables, check if the NI PSP Service Locator and the NI Variable Engine are both running: in START / search windows, type msconfig to access System Configuration. On the tab "Services", sort the table by name (click on the column label "Service") and locate these services.



Should one of it be assigned as "stopped", please restart the computer and check again. These services may also be restarted via the task manager, but there sort for the service description to locate these services. You may try to restart them via right-click.

Once the connection to the shared variable server is established, you can run one of the supplied command VI e.g. Hyperchromator_RC_(set_wavelength). The operation is straight-forward. For example, to set the wavelength, enter the target wavelength and click on the run button:



If you wish to use one of these VIs as a subVI in your own project, you will need to create shared variable items in your project tree (...New\Variable) and link them as described above. The easiest way to do this is to drag & drop (with Cntrl button to ADD, not to MOVE) the HyperSharedVars.lvlib from this sample project.

(Note that connection to the shared variable engine is established only once if you place this VI as a subVI into a loop.)

9 Appendix C: Low-level Remote Control (Protocol_2)

-this manual describes low-level RS232 control for Protocol_2 used for new systems or firmware upgrades following May 2020-

Remote control via the LabVIEW Shared Variable Interface to the running Hyperchromator program allows to send high-level command like “set_wave”, where the desired wavelength may be directly specified. The wavelength calibration is evaluated, the associated grating is being selected (in dual-grating systems) and the motor is set accordingly, as well as switching to the correct long-path order sorting filter.

On the other hand, it is also possible to use low-level remote control without the need for having the Hyperchromator program running in the background. By doing so, the user may command the grating to a motor position directly and setting filters or the shutter as wanted.

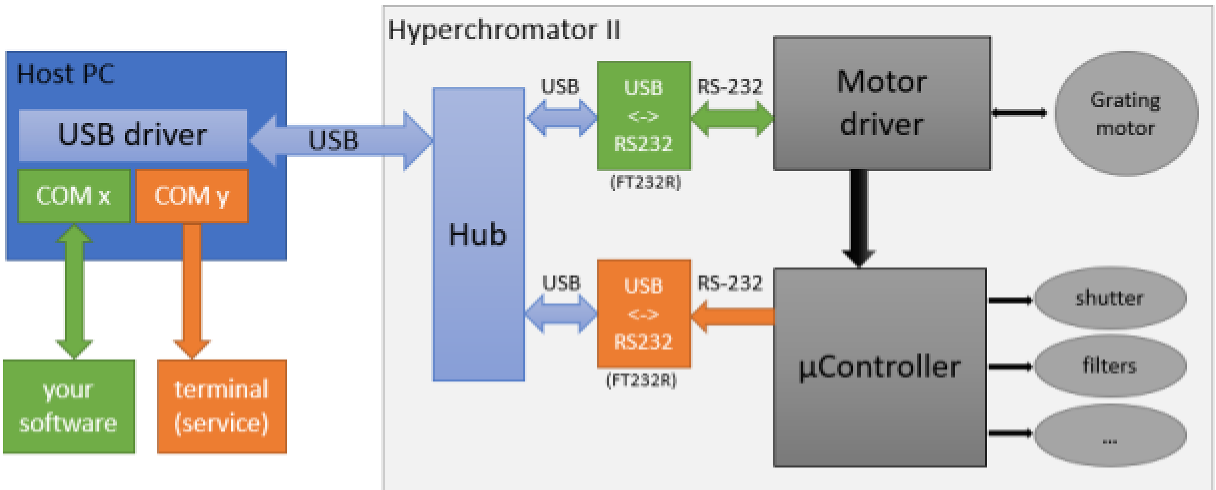
Note: when using low-level remote control, it’s the user’s responsibility to ensure proper choice of motor position and order sorting filter as well as monitoring if the target position has been reached before proceeding to the next step.

Important: setting out-of-range values to some internal parameters, e.g. for acceleration or speed, or other low-level commands may cause technical malfunction of the system or even permanent error. Please use only the low-level commands described in this manual.

9.1 Principle of low-level remote control

After switching on the Hyperchromator hardware, the internal motor controller starts motor control loops, performs motor indexing, and then goes to idle state, waiting for commands over its COM port (RS-232) interface. The COM ports are so-called “virtual COM ports” (VCP) because they are emulated via an USB connection. Current hardware of the Hyperchromator uses the FTDI FT232R chip to perform the USB to RS-232 transmission.

There’s a USB hub on-board, so there actually are 2 COM ports: one accesses the motor driver (and you will be communicating thru this port). The other accesses the embedded microcontroller for subsystems like the shutter and the filter wheel. You would normally not use this port, and it may even be switched off in some cases. However, some response of the µController may be monitored, if required.



Only the COMx port is used to control the device, with the motor driver occasionally forwarding some set-and-forget commands like “open shutter” to the µController. Thus, the motor driver does not receive any response from the µController. In current hardware it is not implemented to receive a “shutter open position reached” signal. Rather one must set a reasonable delay, typ. 300ms, before issuing further actions.

It is possible to send commands via any type of serial (COM) port interface, e.g., a terminal program or by using the respective commands in your programming environment. Sometimes the enumeration of the COM ports is not obvious. Usually, these are sequential numbers, e.g., COM4 and COM5, but it may not be clear which one belongs to the motor driver and the µC.

To identify the proper port for the motor driver, you may send a reset command and observe the response. If it returns the “OK” byte 0x4F it’s the correct port. The binary sequence for RESET is given in the command listing below.

In a Windows environment, the steps would be:

1. In the device manager, under “COM and LPT”, check the COM port.
2. Open a terminal or VISA session for this COM port, using the following settings:
   115200 baud, 8 data bits, 2 stop bits, no parity
3. If the port can be connected (i.e., not being used already) send the “Reset” cmd (see explanations and listing below)
4. Evaluate the return string -if any- for the OK byte 0x4F
5. Repeat 2-4 until you find the port returning OK

In your programming environment, use the respective command to receive a list of available COM ports and cycle thru it sending RESET.

The secondary COM port may be monitored for responses from the µController. This port is mainly for servicing, e.g., to upload modified firmware. The setup is 115200 baud, 8 data bits, 1 stop bit, no parity. See the “Protocol_2” commands table for available commands to produce a response on this channel.

9.2 TML command language

The internal motor controller is a high-performance motion controller from Swiss company Technosoft SA. It uses the proprietary programming language TML. Via the RS-232 interface, direct TML commands may always be send, overriding the internal program loops. A detailed description of the TML language may be found here (requiring registration):

https://www.technosoftmotion.com/en/downloads

The TML commands are not send directly as ASCII strings but encoded into binary command strings. We will use the TML syntax to describe the functions first and describe the conversion into the binary strings later.

9.2.1 Turning the grating: Selecting a wavelength

For our purposes, we will use only a small fraction of the TML language and registers: the most important function will be to command the grating motor to a new position for a desired wavelength. In single- or dual-grating systems, 1 or more calibration tables (wavelength_calib, or wavelength_calib_A&B or named files, e.g. wavelength_calib_FirstGrating…) may be found in configuration directory. In these editable text files, the first column is the wavelength (nm), the second is FWHM (nm), the third column is the motor position in internal motor units (“counts”).

The grating files actually used are selected in the file config.txt.

To drive the grating to a desired wavelength, the motor position column must be interpolated (a linear interpolation shall usually be sufficient).

Example: wavelength_calib.txt reads like this:

400 5 100000

500 5 200000

600 5 300000

Target wavelength = 440 nm -> target motor position = 140000 (counts)

Note that switching between dual gratings is simply achieved by rotating the grating motor by ca. 180°, as the gratings are facing back-to-back. Accordingly, the two calibration tables for each grating have a wide spacing between their motor position regimes, and the user must wait sufficient time or monitor completion of the movement.

The motion controller, after driving to some target position in the previous move, is now in idle mode and waits for commands. The relevant parameter here is CPOS, the “commanded position” (= target position of the next move). The controller is in “absolute position mode”, thus CPOS is given as absolute position. Changing CPOS, followed by the update command UPD will cause the controller to resume motion with the previously set parameters for speed and acceleration, until the new target position is reached:

Example:

CPOS = 140000; //target position for 440 nm in the case of the calibration table above

UPD; // go to it!

9.2.2 TML binary message structure

The TML strings need to be converted to binary code prior to sending it. The generic structure of the code is:

Byte 1: Message length (number of bytes minus 2)

Byte 2: 0F (AxisID=255 /Group ID – high byte)

Byte 3: F0 (AxisID=255/Group ID – low byte)

Byte 4: Operation code – high byte

Byte 5: Operation code – low byte

Byte 6: Data (1) – high byte

Byte 7: Data (1) – low byte

Byte 8: Data (2) – high byte

…

Byte13: Data (4) – low byte

Last byte: Checksum CS (low byte value of the sum obtained by adding up previous byte values)

For sending 32 bit LONG numeric data like motor positions hwhb, hwlb, lwhb, lwlb means high_word_high_byte, high_word_low_byte, low_word_high_byte, low_word_low_byte

Example: decimal 123400 = 0x0001E208 -> resorts to lwhb lwlb hwhb hwlb as E2 08 00 01

Binary data is given as hexadecimal numbers (0x).

If the command syntax is OK, the motor driver will return 0x4F (‘O’ for OK) as the (first) returned byte.

9.2.3 Error handling

If any error occurs during the message reception, for example the checksum computed by the driver doesn’t match with the one sent by your program, it will not send the 0x4F byte. If you don’t receive any acknowledge byte within a few ms after the end of the checksum byte transmission, this means that at some point during the last message transmission, information was lost and the synchronization between the host PC and the motor driver has been lost.

To restore the synchronization your program should do the following:

1. Send a SYNC byte having value 0x0D
2. Wait a programmed timeout (>= 2ms) period for the answer
3. If the drive sends back the same SYNC byte, synchronization was restored, and the host can send again the last message
4. Repeat steps 1 to 3 up to 15 times until the drive answers with a SYNC byte. If after 15 attempts the drive still doesn’t answer, then there is a serious communication problem, and the serial link must be checked.

The operation code may be either a command code of the TML language or the register address of a TML register or a combination. For example, RESET is a command with a fixed code, while CPOS is a register, which may change for different hardware configurations. Using CPOS=1234; or similar in TML means: write the following data to the CPOS, thus implicitly contains the command “write data to register”. The request ?CPOS combines the command “read current value of register” (“?”) and the register name (CPOS)

9.2.4 Register addresses

Make sure to check the register addresses for your specific device first! To find the valid register addresses related to internal registers or variables, open the motion controller configuration, which is of the type *.t.zip (e.g. Hyperchromator_P2.1.t.zip) within the configs/calibration directory of your Hyperchromator. Within this ZIP file, you can view the file variables.cfg (DO NOT MAKE ANY CHANGES TO THIS FILE!) and scroll through the list of registers for the register you want, e.g. the position register CPOS. It may look like this:

....

```
UINT      COUNTINGDIR     @0x0805
LONG      CPOS    @0x029E       %4
INT       CRTERR @0x0231
FIXED     CSPD     @0x02A0
```

...

For **reading** the register CPOS (in TML: ?CPOS;) it is accessed via the address 0x29E in this device. However, to **set** CPOS (or others), you cannot use the register address directly. Depending on the type of the register (INT, LONG, FIXED) the address needs to be reformatted (think of it as encoding some information token for the register type into some unused bits of the 2 address bytes). The reformatting of the address works like this: use the lower 9 bits of the register address and combine them by bitwise OR with a 2 byte starting code depending on the address range:

Type of register	Address range	New address code	Followed by
INT (16 bits)	0x0200 to 0x03FF	0x2000 | 9LSB(address)	2 bytes INT value
	0x0800 to 0x09FF	0x2200 | 9LSB(address)	
LONG (32 bits)	0x0200 to 0x03FF	0x2400 | 9LSB(address)	4 bytes LONG value
	0x0800 to 0x09FF	0x2600 | 9LSB(address)	
FIXED (32 bits)	0x0200 to 0x03FF	0x2400 | 9LSB(address)	4 bytes FIXED value
	0x0800 to 0x09FF	0x2600 | 9LSB(address)	

Example: we found that CPOS is of type long and resides at 0x029E (from variables.cfg file, where we found “LONG CPOS @0x029E”). The lower 9 bits of 0x029E is 0x009E. As 0x029E is in the address range between 0x0200 and 0x03FF, we need to bitwise OR 0x2400 and 0x009E, thus the reformatted address is 0x249E.

So, finally if

CPOS = 123400; //sets the commanded position to 123400.

Binary code: 08 0F F0 24 9E E2 08 00 01 B4

08: length of message – 2

0F F0: code for axisID=255, standard for the Hyperchromator

24 9E: code for writing a LONG value into the 0x029E register, as described above.

E2 08 00 01: LONG integer value of 123400 = 0x0001E208, ordered as lwhb, lwlb, hwhb, hwlb

B4: low byte of the checksum CS of bytes 1-9 = 0x02B4

9.2.5 List of commands

Note: the following binary command codes are based on the Hyperchromator's standard axisID =255 and the register addresses, which may be different for your device, as described in section "Register addresses" above.

Set Target Position: set the target position in internal units (counts) for the next grating motor move. The move will not be started until UPD is send, see next command.

TML: CPOS = position (LONG);
Register address (re-coded): 24 9E (see "Register addresses" !)
Binary: 08 0F F0 24 9E lwhb lwlb hwhb hwlb CS

Update Position: start motor move to last set CPOS

TML: UPD;
Command code: 0108
Binary: 04 0F F0 01 08 0C

Set Speed: sets the motor speed in internal units for the next grating motor move. The change will be effective with the next CPOS and UPD command. Speed is a fixed point number, with the first word describing the decimals (in multiples of 1/65536, for example 0.1 will be shown as 0.1*65536 rounded to 6554 = 0x199A) and the second word describing the integer part. Usually it should be OK to use only the integer part, thus setting the lower word to zero.

NOTE: make sure that speed is not above the safe maximum value, which is given by the key "speed" in config.txt.

TML: CSPD = speed (fixed point);
Register address (re-coded): 24 A0
Binary: 08 0F F0 24 A0 lwhb(0) lwlb(0) hwhb hwlb CS

Reset drive: reset the drive and index the motor after control errors. The behavior is similar to power-on of the hardware.

TML: RESET;
Command code: 04 02
Binary: 04 0F F0 04 02 09

Read grating pos: reads the actual position (APOS) of the grating motor, given by the motor encoder.The grating motor position is related to the wavelength via the wavelength calibration table, as explained above.

TML: ?APOS ;
Command code: B0 05
Axis requested: 0F F0 (code for axisID=255, standard for the Hyperchromator)
Register address: 02 28 (APOS reg_h reg_l)
Expeditor address: 0F F1 (master axis for communication in a multi-axis system)
Binary: 08 0F F0 B0 05 0F F1 reg_h reg_l CS (CS=low byte of the checksum of bytes 1-9)

The command returns the current APOS value in the following form
Binary: 4C 0C 0F F1 B4 05 0F F0 02 28 pos_lwhb pos_lwlb pos_hwhb pos_hwlb CS

Call a Function: calls a function stored in non-volatile memory with the motion controller. The function name is not transferred directly. Instead, it is associated to an internal memory address, which is being transferred.

TML: CALL function_label;

Command code: 74 01

Binary: 06 0F F0 74 01 reg_high reg_low CS

With Protocol_2, you will only need the SENDCODE function, as described above. To find its memory address, again search in the motion controller configuration file variables.cfg (as for the registers), which will be located close to the end:

…

LABEL SENDCODE @0x405A

…

Thus, 405A (hex) would be the required register address of the SENDCODE function:

CALL SENDCODE: 06 0F F0 74 01 40 5A 14

9.3 Other components: Protocol_2 syntax

The Hyperchromator contains an order-sorting filter wheel (i.e. special long-pass cut-off filters are used to get rid of higher order reflections of the gratings) and a shutter on the lamp input side. Optionally, also a high speed shutter (HSS) may be present on the output side.

To access these (and some internal) components, a simple protocol is used following a generic syntax. It consists of 3 variables and an "execute" command. First the variables are set, then it is executed:

dev: this INT (16bit) numeric variable indicates the devices (=component) which will be addressed

cmd: an INT (16bit) numeric code indicating the command to be performed by the device

prm: a LONG (32bit) numeric parameter, which may be needed for some commands

You must find their register addresses from the variables.cfg file, as for other registers (these ones are close to the end of the file).

sendcode: this firmware function executes the Protocol_2 command.

A complete command string could look like this is TML:

dev=2; cmd=0; prm=2; CALL sendcode;

The meaning is: select the filter wheel (device 2), command is GOTO_POS_n, go for filter no. 2, and execute it.

Let's say the variable registers are dev: 0x03B6, cmd: 0x03B7, prm: 0x03B9. Then, according to the address reformatting, writing to these registers has the form

06 0F F0 21 B6/B7 hb lb CS for dev and cmd (INT)

08 0F F0 21 B9 lwhb lwlb hwhb hwlb CS for prm (LONG).

Thus, for opening the shutter, dev=1; cmd=0; prm=0; CALL SENDCODE; the binary code to write looks like this:

06 0F F0 21 B6 00 01 DD

06 0F F0 21 B7 00 00 DD

08 0F F0 25 B9 00 00 00 00 E5

06 0F F0 74 01 40 5A 14

(Setting prm could be omitted in this case, as it is not used, but for generality it is better to do it).

9.3.1 List of implemented Protocol_2 commands

Note that the blue commands are not intended for general use.

dev	cmd	prm	comment
0: Generic	0: SYNC	- (1:VERBOSITY)	To sync communication after start or error. If verbosity is set, (prm=1), the system responds on a secondary RS232 channel
	1: STORE_EEPROM	-	Writes current (motor pos) values as defaults to EEPROM
	2: RETRIEVE_EEPROM	-	Retrieve default values
	3: RESPOND	0: FW 1: Type 2: S/N	Query firmware, device type or serial number (4 alphanumeric bytes each)
	4: FW_PROG	1: enter 0: leave	Enter the firmware programming mode, usually followed by cmd STORE_EEPROM, then leave.
1: Shutter	0: CLOSE	-	
	1: OPEN	-	
	2: OPEN_POS_SET	POS	Set the open position of the shutter (as given in config.txt)
	3: CLOSE_POS_SET	POS	Set the closed position…
2: Filter (order-sorting)	0: GOTO_POS_n	n (1..4)	Switch filter by number
	1: GOTO_POS_1		
	2: GOTO_POS_2		Direct cmd w/o need to give the number
	3: GOTO_POS_3		
	4: GOTO_POS_4		
	5: GOTO_POS_5		only for Gen.2+ Hyperchromator
	6: SET_POS_n	POS<<3 | (n & b111)	followed by STORE_EEPROM during calib
3: Relais	0: OFF		Switching the internal relais, e.g for lamp control.
	1: ON		

4: HSS HighSpeedShutter (optional)	0: CLOSE		
	1: OPEN		
	2: TIMED		Open for a preset time
	3: SET_TIME	OPEN_TIME	OPEN_TIME in ms
	4: SETUP	(b01 & MODE) | (POLARITY & b10) 0:direct/high 1: timed/high 2:direct/low 3:timed/low	Change the function of the trigger input MODE 0 = direct, 1 = timed POLARITY 0=ActiveHigh, 1= ActiveLow
	5: DISABLE	-	Disabled the HSS
	6: TEST	TESTPIN_LVL: POLARITY ActiveLow: 0:OPEN 1:CLOSED ActiveHigh: 0:CLOSED 1:OPEN	Connect test pin (8) to TrgIn (3)
	7: TIME_OFFSET	OFFSET	OFFSET in µs, typ. 2700

Mountain Photonics GmbH
Albert-Einstein-Str. 18
D-86899 Landsberg am Lech

P: +49 (0) 8191-985199-0
www.mphotonics.de
info@mphotonics.de