RHEA 02 SPECTRORADIOMETER

Operating Manual

ADMESY

colorimeters | spectroradiometers | lightmeters

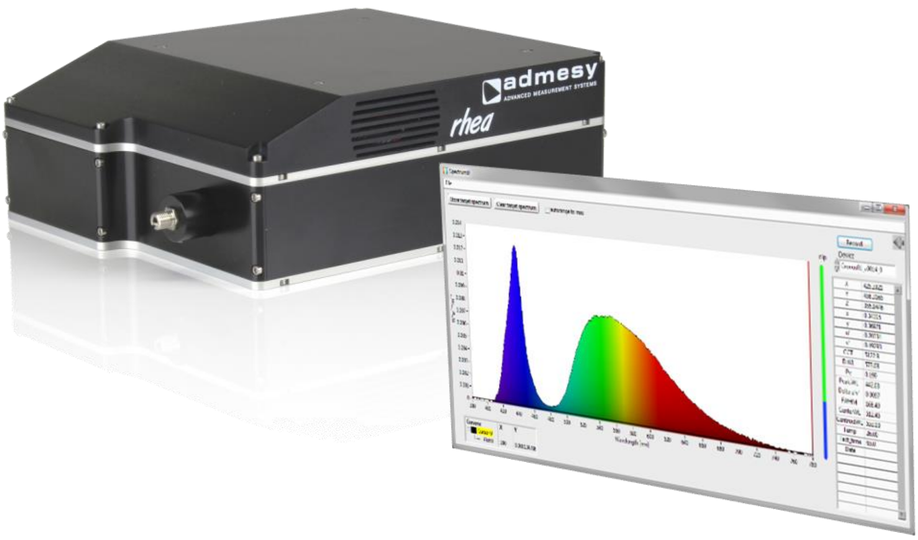
CONTENTS

1 INTRODUCTION

1.1 Rhea02

The Rhea02 series spectrometer offers a unique combination of ease of use and accurate measurement capabilities combined in a robust package. The Rhea utilizes a high-end cooled CCD detector for low noise and high dynamic range. The mechanical shutter makes it possible to measure accurate at extreme low light levels. An ideal device for measurements where ease of use, stability, performance and price are of the essence. All in all, the perfect solution for hassle free integration in your product or process. The Rhea spectrometer can virtually cover any wavelength range in the 200-1100nm range. Standard versions (like UV, VIS and NIR) are available. Additionally, we also support a broad range of gratings for specific applications.

The Rhea02 series is available in a variety of optical fibre connected lenses or cosine correctors. The M8 fixed position fibre connector has been developed to ensure that the optical fibre has a fixed and uniform optical connection for both calibration and customer application. Also, due to this fixed position measurement results are more stable. Additionally, Admesy also offers SMA connection.



1.2 Rhea02 highlights

- Various spectral ranges including UV, VIS and NIR within the 200-1100nm range
- Cooled high-end CCD detector, cooled to -10 degrees Celsius
- Mechanical shutter
- Shutter function
- Low noise
- Auto-range function
- Wavelength calibrated
- Very low stray light
- Excellent linearity, internally compensated within 1%
- Dark current compensated, virtually zero over entire integration range
- USBTMC compliant, SCPI command set, high speed device
- USB, RS232, Ethernet connections and trigger in & out for ideal system integration
- Internal calculations for most common parameters, saving processing power in production environments
- Robust housing, optimized for mounting and protection in harsh production environments
- M8 or SMA fibre connection

1.3 Standards

The Rhea02 is compliant to the USBTMC standard and can be used in combination with external provided USBTMC compliant drivers. Currently it has been tested on Windows, Linux and Apple OSX using NI VISA (www.ni.com/visa) and using the open source drivers on Linux (i686, x86_64 and ARM).

2 INTERFACES

2.1 USB-interface

The USB B connector is used to connect the Hera to a PC/Laptop. USB uses the USBTMC class protocol and can therefore be used directly with third party provided VISA compliant libraries like NI-VISA. The Rhea02 is external powered, please use included power supply.

2.2 Ethernet interface

Ethernet is 100Mbit and is 10Mbit/GigE compatible. Ethernet can be used in the same way as USB. All commands have the same format. Ethernet is preferred over USB in situations where the distance between device and a PC are more than 5 meters.

2.3 RS232 interface

RS232 is provided to connect any host that doesn't provide USB or Ethernet or for which no USBTMC drivers exist. Using RS232, the functions that generate a lot of data (spectrum) are still available, but the use of it is highly discouraged due to the low speed of RS232.

Baud rate	Data bits	Parity	Stop bits	Flow control	Termination character
115200[1]	8	None	1	None	LF='\n'

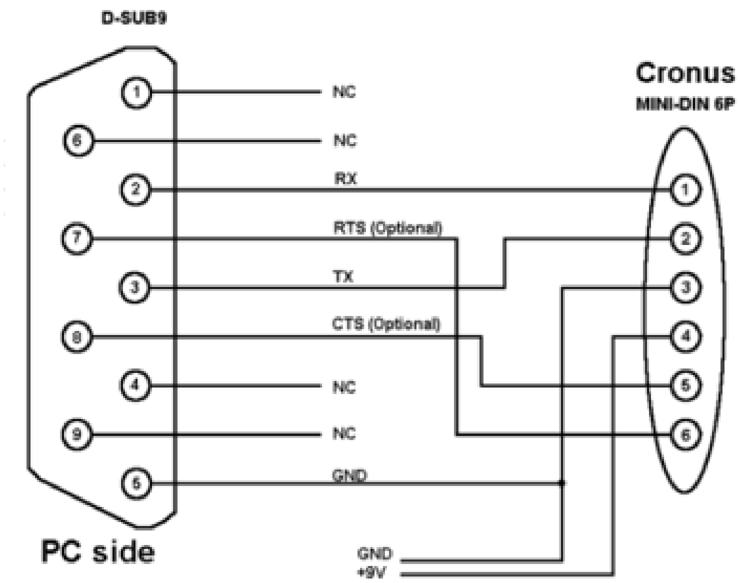


Fig 1 Rhea02 RS232 connection.

Fig 2 Rear side mini-DIN-6 pin connector.

2.4 Trigger in/out

Two trigger connections are available. One trigger output and one trigger input. The connectors are SMA connectors. When triggering is enabled, the trigger output line will be set to a high level once the measurement has finished and the measurement result is available. It will stay at a high level until the next command is carried out but has a minimum high level of 5µs. A trigger will carry out the last send command and send the result to the host via the selected interface. The colorimeter main application allows external triggering in the data-logging tab. Supplied code examples show how to use this feature in an application. The trigger output line is used to indicate that the measurement is ready. Trigger signals should comply to the following timing (Fig 3).

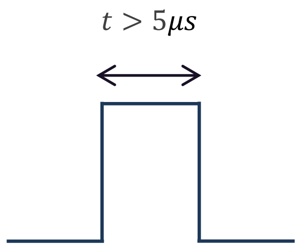


Fig 3 Trigger-in timing.

Trigger pulses arriving faster than the device can measure will be ignored, but it may slowdown overall performance. Trigger pulses should not arrive faster than the measurement takes to complete. The best way is to use the trigger output to make sure measurement was finished.

The trigger out port provides a continuous high signal when the device is in use. Its signal turns low when the device is not executing any commands.

Connection	Low level	High level
Trigger out	0V	5 V
Trigger In	0V	5V

2.5 Power connections

The Rhea02 must be externally powered, by default we recommend using the supplied power supply however using your own supply is also possible.

The unit shall be powered by a 15V DC voltage, reinforced separated from Mains, with a limited energy of < 150VA and < 8A.

Connection	Min. Voltage	Typ. voltage	Max. voltage
DC powered	14.50 V	15.00 V	15.50 V

3 COMMUNICATIONS PROTOCOL

3.1 USB

The Rhea02 can be connected to any USB host and is a USBTMC compliant device which is a standard USB class. Device drivers for this class are available for most popular operating systems (also embedded). This makes the Hera directly usable in popular programming languages like NI's Labview and Labwindows but also C++, Visual basic, C#, Java etc. The Hera has two interfaces build in, which require a different device driver to be used.

- Rhea bootloader - USB RAW device driver, Vendor ID: 0x23CF, Product ID 0x0105
- Rhea02 - USBTMC device driver, Vendor ID: 0x23CF, Product ID 0x1040

When the Rhea02 is connected to the host, it will start the Rhea02 firmware. As soon as the firmware is idle to receive commands, the Power LED goes to the ON state. The Admesy bootloader is a RAW USB device and in order to use this device in Windows, a driver must be installed which is supplied by Admesy. Besides upgrading to new firmware, it is also allowed to downgrade firmware in case this is required. Note that older firmware also may require the use of older software libraries and/or executable versions of software. The Rhea02 is USBTMC compliant and can be used with libraries that contain a USBTMC compliant driver like NI-VISA. The Rhea02 is a USB 2.0 Full speed device.

3.2 RS232

All commands are equal for all interfaces. Note that for high speed transfers it is best to use USB. When RS232 will be used, the device mode should be set to RS232. This is done via software and stored in the device EEPROM memory.

3.3 Ethernet

All commands are equal for all interfaces. When the Ethernet connection will be used, the device mode should be set to ETH (Ethernet). This is done via the Iliad application software: Device→Start-up settings→Rhea02 and stored in the device EEPROM memory. In the start-up menu the IP address, gateway and network mask can also be changed.

4 DEVICE DRIVERS

4.1 USB

The following table shows an overview of USB support on various operating systems.

OS	NI-VISA	Libusb	Native kernel	Agilent USBTMC
Windows XP 2	✓	✓	Not available	Not tested[1]
Windows VISTA	✓	✓	Not available	Not tested[1]
Windows 7	✓	✓	Not available	Not tested[1]
Windows 8(.1)	✓	Not tested[1]	Not available	Not tested[1]
Windows 10	✓	✓	Not available	Not tested[1]
Windows CE	✓	Not tested[1]	Not available	Not tested[1]
Apple OSX PPC	✓	Not tested[1]	Not available	Unknown
Apple OSX Intel	✓	Not tested[1]	Not available	Unknown
Linux i386 (32bit)	✓	✓	✓	✓
Linux i386 (64bit)	✓	✓	✓	✓
Linux ARM	Not available	✓	✓	✓
Linux other	Not available	✓	✓	✓

1 Not tested: Available, but not tested by Admesy, 2 Native Kernel: Driver included with OS.
2 Windows XP SP3 is supported: Windows official support has ended as of April 8 2014

Admesy supports all tested platforms but does not provide standard applications on all platforms. The matrix is provided to show the possible platforms for software development. Admesy does however provide software examples for most of the tested platforms. Most of these examples can be found on our support web page.

4.2 RS232

When no USB driver is available or the host system does not provide USB, RS232 can be used as it does not require additional drivers for the Hera.

4.3 Ethernet

No special drivers are needed for Ethernet operation. The Hera can be directly addressed through a TCP/IP socket on port 10000. In case firewalls are used, the TCP/IP port needs to be opened.

5 COMMAND SET OVERVIEW

5.1 Command set categories

The functions of the Rhea02 can be described through the following categories.

- System commands
- Configuration commands
- Measurement commands
- User EEPROM configuration commands
- User EEPROM calibration commands

The Rhea02 uses SCPI like commands for control and measurement. These are ASCII based commands and follow specific rules regarding syntax.

5.2 System commands

The following commands can be used set to read back information. These commands are general system commands.

Command	Write Parameter	Read Result	Description
:*IDN?	—	device name	identification query
:*RST	—	—	reset command
:*FWD?	—	firmware date	reads firmware date
:SYSTem:VERSion?	—	firmware version	returns firmware version

5.3 Configuration commands

Configuration commands are used to set parameters of the Rhea02 that are used by the measurement functions. The settings are used by measurement functions.

Command	Write Parameter	Read Result	Description
:SENSe:SHUTter	state	—	close (1) or open (0) shutter
:SENSe:SP:AVERage	average	—	sets average
:SENSe:SP:AVERage?	—	average	reads average
:SENSe:AVERage	average	—	sets average
:SENSe:AVERage?	—	average	reads average
:SENSe:SP:AUTORANGE	state	—	autorange on (1) or off (0)
:SENSe:SP:AUTORANGE?	—	state	get autorange state
:SENSe:AUTORANGE	state	—	autorange on (1) or off (0)
:SENSe:AUTORANGE?	—	state	get autorange state
:SENSe:SP:SBW	matrix	—	set matrix
:SENSe:SP:SBW?	—	matrix	get matrix
:SENSe:SBW	matrix	—	set matrix
:SENSe:SBW?	—	matrix	get matrix
:SENSe:SP:INT	integration time	—	set integration time [us]
:SENSe:SP:INT?	—	integration time	get integration time [us]
:SENSe:INT	integration time	—	set integration time [us]
:SENSe:INT?	—	integration time	get integration time [us]
:SENSe:ARPARMS	frame freq, Adjmin, Max Int time, Average	—	set auto-range parameters
:SENSe:ARPARMS?	—	frame freq, Adjmin, Max Int time, Average	read auto-range parameters
:SENSe:CALPARMS	interpol, startwl, stopwl, res, abs cal, wl cal		configure spectral calibration parameters
:SENSe:CALPARMS?		interpol, startwl, stopwl, res, abs cal, wl cal	read spectral calibration parameters
:SENSe:TRIG	mode	—	set trigger mode on (1) or off (0)
:SENSe:TRIG?	—	mode	read trigger mode
:SENSe:TRIGDELAY	delay	—	set trigger delay [us]
:SENSe:TRIGDELAY?	—	delay	read trigger delay [us]

5.4 Measurement commands

The following table shows the measurement commands available of the Hera.

Command	Write Parameter	Read Result	Description
:GET:SPECSize	—	size	get spectrum size in bytes
:MEASure:SPECtrum	dark mode	clip level, intensity spectrum	meausre spectrum function
:MEASure:RAWSPECtrum	dark mode,hotpix comp, linearity	clip level, intensity spectrum	measure raw spectrum function
:GET:WAVElengths	—	wavelength array	get values of wavelength axis
MEASure:TEMP	—	T sensor, T cold, T hot, T power	measure temperature function
MEASure:XYZ	—	XYZ, clip, noise	measure XYZ function
MEASure:YXY	—	Yxy, clip, noise	measure Yxy function

5.5 User EEPROM configuration commands

Below table shows the configuration setting commands which can be used to store values in the user EEPROM space. It is advised to reboot the Rhea02 after writing new values to the EEPROM.

Command	Write Parameter	Read Result	Description
:EEPROM:STARTUP:WRITE	—	—	stores values to eeprom
:EEPROM:STARTUP:READ	—	—	read all eeprom values
:EEPROM:CONFigure:BAUDRATE	baudrate setting	—	write RS232 baudrate setting
:EEPROM:CONFigure:BAUDRATE?	—	baudrate setting	read RS232 baudrate setting
:EEPROM:CONFigure:AUTOrange	state		read autorange state
:EEPROM:CONFigure:AUTOrange?			write autorange state
:EEPROM:CONFigure:ARPARMS	frame freq, Adjmin, Max Int time	—	write autorange parameters
:EEPROM:CONFigure:ARPARMS?	—	frame freq, Adjmin, Max Int time	read autorange parameters
:EEPROM:CONFigure:SPAVG	average	—	write average value to
:EEPROM:CONFigure:SPAVG?	—	average	read average value
:EEPROM:CONFigure:SPINT	average	—	write integration time value
:EEPROM:CONFigure:SPINT?	—	average	read integration time value
:EEPROM:CONFigure:CALPARMS	interpol, startwl, stopwl, res, abs cal, wl cal	—	write spectral calibration parameters
:EEPROM:CONFigure:CALPARMS?	—	interpol, startwl, stopwl, res, abs cal, wl cal	read spectral calibration parameters
:EEPROM:CONFigure:SPSBW	matrix	—	write which matrix to use
:EEPROM:CONFigure:SPSBW?	—	matrix	read which matrix is used
:EEPROM:CONFigure:IP	IP address	—	write IP address
:EEPROM:CONFigure:IP?	—	IP address	read IP address
:EEPROM:CONFigure:GW	gateway	—	write gateway address
:EEPROM:CONFigure:GW?	—	gateway	read gateway address
:EEPROM:CONFigure:MASK	mask	—	write network mask
:EEPROM:CONFigure:MASK?	—	mask	read network mask
:EEPROM:CONFigure:TRIG	mode	—	write trigger mode on (1) or off (0)
:EEPROM:CONFigure:TRIG?	—	mode	read trigger mode
:EEPROM:CONFigure:TRIGDELAY	delay	—	write trigger delay [us]
:EEPROM:CONFigure:TRIGDELAY?	—	delay	read trigger delay [us]

5.6 User EEPROM calibration commands

Below table shows the commands which can be used to store user calibrations. There are three possible user calibrations:

- Absolute calibration, also called intensity calibration. If the user would like to use its own lightsource or reference for absolute calibration this is possible by this calibration
- wavelength calibration, this enables the user to do its own wavelength calibration. If this calibration is done we also advice to do an absolute calibration
- Matrix calibration, the 3x3 user matrix can be programmed. This function is only valid for the XYZ domain and will only be applied on the :MEASure:XYZ and :MEASure:YXY command.

It is advised to reboot the Rhea02 after writing new values to the EEPROM.

Physically storing the values into the eeprom is divided into 2 parts, first part is for storing the wavelength and absolute calibration values. The second part is for storing the matrix values.

Command	Write Parameter	Read Result	Description
:EEPROM:ABS:FIX	—	—	store calibration values to eeprom
:EEPROM:ABS:WRITE	param,index,value	—	write absolute calibration values
:EEPROM:ABS:READ	param,index	value	read absolute calibration values
:EEPROM:WLLUTWRITE	param,index,value	—	write wavelength lut values
:EEPROM:WLLUTREAD	param,index	value	read wavelength lut values
:EEPROM:WRITE:USERCAL	—	—	store 3x3 matrix (SPSBW) in eeprom
:EEPROM:SPSBW:WRITE	row,col,value	—	write 9 values of matrix
:EEPROM:SPSBW:READ	—	row,col,value	read 9 values of matrix

6 DETAILED COMMAND LIST

6.1 System commands

6.1.1 :*IDN?

Reads the identification query of the device

Action	Data	Termination	Format
write	:*IDN?	LF	ASCII
read	Admesy B.V. Rhea02	LF	ASCII

6.1.2 :*RST

Resets the device and settings are set back to original start up value

Action	Data	Termination	Format
write	:*RST	LF	ASCII
read	Not applicable		

6.1.3 :*FWD?

Reads the date of the firmware

Action	Data	Termination	Format
write	:*FWD?	LF	ASCII
read	Mon Mar 23 14:32:19 2020	LF	ASCII

6.1.4 :SYSTem:VERSion?

Reads the firmware version

Action	Data	Termination	Format
write	:SYSTem:VERSion?	LF	ASCII
read	1.04	LF	ASCII

6.2 Configuration commands

Configuration commands are in general commands which have impact on measurement commands, many of these commands determine the quality of the measurement or at least have an influence on it. In the list many commands can be found which seem double. Since Admesy offers also other devices there are devices which contain more than one sensor. In order to keep everything compatible we have also applied the same structure in the software so customers who use these devices can still use the same structure. In the case of the Rhea02 this means the same function can react exactly the same on two different (but nearly the same) input commands.

6.2.1 :SENSe:SHUTter

With this command the mechanical shutter inside the device can be closed or opened. This can come in use for very low light levels. 1 means shutter is closed (no light can enter), 0 means shutter is open.

Action	Data	Termination	Format
write	:SENSe:SHUTter 1	LF	ASCII
read	Not applicable	LF	ASCII

6.2.2 :SENSe:SP:AVERage

Sets the average of the spectrometer. Average can be up to 255.

Action	Data	Termination	Format
write	:SENSe:SP:AVERage 2	LF	ASCII
read	Not applicable		

The average command can also be set in the :SENSe:ARPARMS command but it also exists as a single global variable. This is to enhance the flexibility for the customer

6.2.3 :SENSe:SP:AVERage?

Reads the average of the spectrometer.

Action	Data	Termination	Format
write	:SENSe:SP:AVERage?	LF	ASCII
read	2	LF	ASCII

6.2.4 :SENSe:AVERage

Sets the average of the spectrometer. Average can be up to 255

Action	Data	Termination	Format
write	:SENSe:AVERage 2	LF	ASCII
read	Not applicable	—	—

The average command can also be set in the :SENSe:ARPARMS command but it also exists as a single global variable. This is to enhance the flexibility for the customer

6.2.5 :SENSe:AVERage?

Reads the average of the spectrometer.

Action	Data	Termination	Format
write	:SENSe:AVERage?	LF	ASCII
read	2	LF	ASCII

6.2.6 :SENSe:SP:AUTORANGE

Set the auto-range state to on (1) or off (0). The functionality of the auto-range algorithm is described in the auto-range chapter.

Action	Data	Termination	Format
write	:SENSe:SP:AUTORANGE 1	LF	ASCII
read	Not applicable	—	—

6.2.7 :SENSe:SP:AUTORANGE?

Read the auto-range state, 0 means off and 1 means on

Action	Data	Termination	Format
write	:SENSe:SP:AUTORANGE?	LF	ASCII
read	1	LF	ASCII

6.2.8 :SENSe:AUTORANGE

Set the auto-range state to on (1) or off (0). The functionality of the auto-range algorithm is described in the auto-range chapter.

Action	Data	Termination	Format
write	:SENSe:AUTORANGE 1	LF	ASCII
read	Not applicable	—	—

6.2.9 :SENSe:AUTORANGE?

Read the auto-range state, 0 means off and 1 means on

Action	Data	Termination	Format
write	:SENSe:AUTORANGE?	LF	ASCII
read	1	LF	ASCII

6.2.10 :SENSe:SP:SBW

Set matrix to “off” or to “user”. This function can only be used when using the :MEASure:XYZ or :MEASure:Yxy command. In general this is only used when measuring color, when using the XYZ command the spectral data is converted to the XYZ domain. To achieve a higher accuracy (especially from inter-instrument agreement point of view) a matrix can be applied to the XYZ result.

When setting the value to “ off” no matrix is applied. When “user” is set the value programmed inside the EEPROM is set.

Action	Data	Termination	Format
write	:SENSe:SP:SBW user	LF	ASCII
read	Not applicable	—	—

6.2.11 :SENSe:SP:SBW?

Read matrix value, can be either “off” or “user”.

Action	Data	Termination	Format
write	:SENSe:SP:SBW?	LF	ASCII
read	off	LF	ASCII

6.2.12 :SENSe:SBW

Set matrix to “off” or to “user”. This function can only be used when using the MEASure:XYZ or MEASureYxy command. In general this is only used when measuring color, when using the XYZ command the spectral data is converted to the XYZ domain. To achieve a higher accuracy (especially from inter-instrument agreement point of view) a matrix can be applied to the XYZ result.

When setting the value to “ off” no matrix is applied. When “user” is set the value programmed inside the EEPROM is set.

Action	Data	Termination	Format
write	:SENSe:SBW user	LF	ASCII
read	Not applicable	—	—

6.2.13 :SENSe:SBW?

Read matrix value, can be either “off” or “user”.

Action	Data	Termination	Format
write	:SENSe:SBW?	LF	ASCII
read	off	LF	ASCII

6.2.14 :SENSe:SP:INT

Set integration time of the spectrometer. Integration time set is in [us] and can be set between 4700us and up to 60 minutes (this extreme long integration time is not guaranteed).

Action	Data	Termination	Format
write	:SENSe:SP:INT 20000	LF	ASCII
read	Not applicable	—	—

6.2.15 :SENSe:SP:INT?

Read back the integration time used by the device during its last measurement

Action	Data	Termination	Format
write	:SENSe:SP:INT?	LF	ASCII
read	20000	LF	ASCII

6.2.16 :SENSe:INT

Set integration time of the spectrometer. Integration time set is in [us] and can be set between 4700us and up to 60 minutes (this extreme long integration time is not guaranteed) and is not supported in auto-range mode.

Action	Data	Termination	Format
write	:SENSe:INT 20000	LF	ASCII
read	Not applicable	—	—

6.2.17 :SENSe:INT?

Read back the integration time used by the device during it's last measurement.

Action	Data	Termination	Format
write	:SENSe:INT?	LF	ASCII
read	20000	LF	ASCII

6.2.18 :SENSe:ARPARMS

Set the auto range parameters and average.

Set parameter:	**Range:**	**Description:**
Frame Frequency [Hz]	0 – 250.0	frequency of source
Adjmin [%]	1 – 40	scale of ADC to regulate to in percentage
Max int time [µs]	1/freq – 60s.	maximum integration time to use in µs
Average	1 - 255	number of measurements to average

Action	Data	Termination	Format
Write	:SENSe:ARPARMS 60,20,1000000,1	LF	ASCII
Read	Not applicable	—	—

6.2.19 :SENSe:ARPARMS?

Returns the auto range parameters and average

Return parameter:	**Range:**	**Description:**
Frame Frequency [Hz]	0 – 250.0	frequency of source
Adjmin [%]	1 – 40	scale of ADC level to regulate to in percentage
Max int time [µs]	1/freq – 60s.	maximum integration time to use in µs
Average	1 - 255	number of measurements to average

Action	Data	Termination	Format
write	:SENSe:ARPARMS?	LF	ASCII
read	60,20,1000000,1	LF	ASCII

6.2.20 :SENSe:CALPARMS

Parameters to be set are interpolation, start wavelength, stop wavelength resolution, abs calibration mode and wavelength calibration mode. Resolution determines the step size between two consecutive wavelengths. For wavelength and abs calibration mode there is an option for the customer to use its own wavelength calibration and absolution calibration.

Set parameter:	**Range:**	**Description:**
Interpolation	0	Output is in native wavelength resolution, sensor full range
	1	Output is from start to stop, step size is resolution setting
	2	Output is from start to stop, step size is native resolution
Start wl	200-1100	Start wavelength to be returned
Stop wl	201-1100	Stop wavelength to be returned
Resolution	0-10	Wavelength step size, minimal value is 0.01nm
ABS Cal mode	0	Factory
	1	User
	2	Off
WL CAL mode	0	Factory
	1	User

Stop wavelength should always be bigger than the start wavelength

Action	Data	Termination	Format
write	:SENSe:CALPARMS 1,400,800,0.5,1,0	LF	ASCII
read	Not applicable	—	—

6.2.21 :SENSe:CALPARMS?

Returns interpolation, start wavelength, stop wavelength resolution, abs calibration mode and wavelength calibration mode.

Return Parameter:	**Range:**	**Description:**
Interpolation	0	Output is in native wavelength resolution, sensor full range
	1	Output is from start to stop, step size is resolution setting
	2	Output is from start to stop, step size is native resolution
Start wl	200-1100	Start wavelength to be returned
Stop wl	201-1100	Stop wavelength to be returned
Resolution	0-10	Wavelength step size, minimal value is 0.01nm
ABS Cal mode	0	Factory
	1	User
	2	Off
WL CAL mode	0	Factory
	1	User

Action	Data	Termination	Format
write	:SENSe:CALPARMS?	LF	ASCII
read	1,400,800,0.5,1,0	LF	ASCII

6.2.22 :SENSe:TRIG

Set the device into trigger mode, if set to "1" the device is set into trigger mode. If set to "0" the trigger mode is switched off.

Action	Data	Termination	Format
write	:SENSe:TRIG 1	LF	ASCII
read	Not applicable	—	—

6.2.23 :SENSe:TRIG?

Returns the device trigger mode, if set to "1" the device is set into trigger mode. If set to "0" the trigger mode is switched off.

Action	Data	Termination	Format
write	:SENSe:TRIG?	LF	ASCII
read	1	LF	ASCII

6.2.24 :SENSe:TRIGDELAY

Set a delay for trigger mode, measurement will start after the trigger is received and this delay has passed. Value is in [us].

Action	Data	Termination	Format
write	:SENSe:TRIGDELAY 100	LF	ASCII
read	Not applicable	—	—

6.2.25 :SENSe:TRIGDELAY?

Returns the trigger delay value, the value is in [us].

Action	Data	Termination	Format
write	:SENSe:TRIGDELAY?	LF	ASCII
read	100	LF	ASCII

6.3 Measurement commands

6.3.1 :GET:SPECSize

Returns the size of the wavelength array which is the same size as the intensity array used in measure spectrum function and get wavelength function

Size returned is nr. of Bytes, this size can be used for reading back the wavelength values (x-axis of the spectrum) or intensity values (y-axis of the spectrum). For these two axis each 4 bytes equal a single float which represents a wavelength or intensity value.

If wavelength range would be set to 400nm to 800nm this would mean there would be 401 step/wavelengths, when the resolution setting is set to 1nm. The value returned in this case would be 401x4 = 1604.

Action	Data	Termination	Format
write	:GET:SPECSize	LF	ASCII
read	1604	LF	ASCII

6.3.2 :MEASure:SPECtrum

Measures the spectrum and returns the spectrum values, nr. of values returned depends on the settings set in the configuration commands, output is intensity value. Input parameter which can be set is the dark mode. When the dark mode is set to 1 (shutter mode) every measurement is followed by a measurement with the shutter closed. The dark spectrum is internally subtracted from the measured (light) spectrum. For the software compensation mode (0) an internal software compensation is applied for dark. We strongly advise to use this one in most circumstances, except for ultra-low light levels, since excessive use of the shutter should be avoided. Additionally, the dark software compensation works excellent for almost all conditions and saves half the measurement time.

Set parameter:	**Range:**	**Description:**
dark mode	0	Software compensation
	1	Shutter mode

Returned values are the clip level and the spectrum which are returned as a single float. The first 4 bytes are the clip level (first single float), when this is 0 this means there is virtually no light and 1 means the device is clipping. Everything in between indicates a certain amount of light. 0.5 for example indicates that the highest pixel intensity value has an output of 50% on the saturation scale (ADC output scale). For the spectrum 4 bytes are 1 intensity value. Size of the intensity spectrum array can be determined by GET:SPECSize (already in bytes). Nr of bytes for the spectrum is the return value of the command GET:SPECSize. Order of the Bytes is big endian (single float big endian).

Return Parameter:	**Range:**	**Description:**
Clip level	0-1	First 4 bytes are clip level, value between 0 and 1
Intensity level	float	length is GET:SPECSize

When reading back data be careful since the return values are sent back as a byte stream instead of ASCII code and are not followed by a termination character.

Nr of read cycles which should be done are 4 + the return value of GET:SPECSize

Action	Data	Termination	Format
write	:MEASure:SPECtrum 0	LF	ASCII
read	0x00,0xAA…….0x16	—	float

6.3.3 :MEASure:RAWSPECtrum

Measures the raw spectrum and returns the raw spectrum values, nr. of values returned are 1024(full rang sensor), output is in ADC counts.

Input parameters which can be set are the dark mode, hotpixel compensation and linearity. When the dark mode is set to 1 (shutter mode) every measurement is followed by a measurement with the shutter closed. The dark spectrum is internally subtracted from the measured (light) spectrum. For the software compensation mode (0) an internal software compensation is applied for dark. We strongly advise to use this one in most circumstances, except for ultra-low light levels, since excessive use of the shutter should be avoided. Additionally, the dark software compensation works excellent for almost all conditions and saves half the measurement time. When the dark mode is set to 0 no dark compensation is done, additionally also the linearity is automatically switched off.

Hotpixel compensation can also be switched on or off. In the case of a CCD it can occur that a pixel is damaged or has a too high dark current to perform properly. In order to fix this the average of the 2 neighbours is used to interpolate the value.

Linearity can also be switched on or off, when linearity is on the non-linear behaviour of the CCD is compensated.

Parameter:	**Range:**	**Description:**
Raw dark mode	0	software compensation
	1	Shutter mode
	2	No dark compensation
Hotpixel comp	0	hot pixel compensation off
	1	Hot pixel compensation on
Linearity	0	linearity off
	1	linearity on

Returned values are the clip level and the raw spectrum (in ADC counts) which are returned as single float. The first 4 bytes are the clip level (first single float), when this is 0 this means there is virtually no light and 1 means the device is clipping. Everything in between indicates a certain amount of light. 0.5 for example indicates that the highest pixel intensity value has an output of 50% on the saturation scale (ADC output scale). For the spectrum 4 bytes are 1 raw intensity value. Size of the raw intensity spectrum array is 1024 pixels (nr. of pixels of the sensor). Nr of bytes for the spectrum are 4 x 1024. Order of the Bytes is big endian (single float big endian).

Return Parameter:	Range:	Description:
Clip level	0-1	First 4 bytes are clip level, value between 0 and 1
ADC counts	float	length is 4x1024 Bytes

When reading back data be careful since the return values are sent back as a byte stream instead of ASCII code and are not followed by a termination character.

Nr of read cycles which should be done are 4 + 1024 x 4

Action	Data	Termination	Format
write	:MEASure:RAWSPECtrum 0,1,1	LF	ASCII
read	0x00,0xA9.......0x1B	—	float

6.3.4 :GET:WAVElengths

Returns the wavelength values, nr. of values returned depends on the settings set in the configuration commands, output is wavelength thus nm. When constructing a spectrum graph this is the x-axis. Usually this is only read back once. Since the values are always the same if you don't change the settings, which have influence on the wavelength, in the configuration commands.

Returned values are the wavelength values, they are returned as a single float. Size of the wavelength array can be determined by GET:SPECSize (already in bytes). Nr of bytes for the wavelength array is the return value of the command GET:SPECSize. Order of the Bytes is big endian (single float big endian, 4 bytes are one value).

Return Parameter:	**Range:**	**Description:**
wavelength	float	length is GET:SPECSize

Action	Data	Termination	Format
write	:GET:WAVElengths	LF	ASCII
read	0x02,0xBC…….0x90	—	float

6.3.5 :MEASure:TEMP

The Rhea02 contains 4 temperature sensor, one sensor is placed on the the sensor pcb, two sensors are available for the TEC cooler, one for the cold side and one for the hot side. And there is also a sensor on the power pcb. All values are returned in consecutive order.

Action	Data	Termination	Format
write	MEASure:TEMP	LF	ASCII
read	26.21,-10.232,31.454,27.6	LF	ASCII

6.3.6 :MEASure:XYZ

This command returns the XYZ result of the spectrometer and the clip and noise, Parameters read back are X,Y,Z,clip,noise. The parameters are separated by a comma ",". In total 5 parameters are returned. If clip and noise (both boolean) are 0 the measurement is fine, if clip is 1 the measurement is clipped (thus wrong). And if noise is 1 the luminance level is not high enough. This function should only be used when measuring color, so when the spectrometer is configured at least from 380-780nm.

Action	Data	Termination	Format
write	:MEASure:XYZ	LF	ASCII
Read	100.01,50.33,200.356,0,0	LF	ASCII

6.3.7 :MEASure:YXY

This command returns the Yxy result of the spectrometer and the clip and noise, Parameters read back are Y,x,y,clip,noise. The parameters are separated by a comma “,”. In total 5 parameters are returned. If clip and noise (both boolean) are 0 the measurement is fine, if clip is 1 the measurement is clipped (thus wrong). And if noise is 1 the luminance level is not high enough. This function should only be used when measuring color, so when the spectrometer is configured at least from 380-780nm.

Action	Data	Termination	Format
write	:MEASure:YXY	LF	ASCII
read	50.33,0.3331,0.4501,0,0	LF	ASCII

6.4 User EEPROM configuration commands

With the user eeprom configuration commands the user can program the device start-up values, this determines the configuration on how the devices powers up without writing any command. If a user has multiple devices and configures the eeprom values with exactly the same values the program to control the device can be dramatically simplified. Programming the user eeprom values can be done by the software supplied by Admesy or by writing your own program with below described commands.

When writing the EEPROM configuration commands one should always finish the sequence with the :EEPROM:STARTUP:WRITE command. This command physically tells the eeprom to store the value. All the other commands write to the eeprom and don't do a store.

A small example is given in the sample section

It is recommended after writing and storing the values into the eeprom that you re-power the device.

6.4.1 :EEPROM:STARTUP:WRITE

This command stores all the written eeprom configuration values into the eeprom. It is important to wait 20ms after sending this command. Storing physically to the eeprom requires some time.

Action	Data	Wait	Termination	Format
write	:EEPROM:STARTUP:WRITE	20ms	LF	ASCII
read	Not applicable	—	—	—

6.4.2 :EEPROM:STARTUP:READ

This command reads back all the stored eeprom configuration values. All configuration values are set to the eeprom values.

Before reading eeprom values (by EEPROM:CONFigure:…? Commands) it is recommend to send this command first.

Action	Data	Termination	Format
Write	:EEPROM:STARTUP:READ	LF	ASCII
Read	Not applicable	—	—

6.4.3 :EEPROM:CONFigure:BAUDrate

The baudrate of RS232 can be set with this command. Baudrate setting is the following

Value	Baudrate
0	9600
1	19200
2	38400
3	57600
4	115200

Action	Data	Termination	Format
write	:EEPROM:CONFigure:BAUDRATE 0	LF	ASCII
read	Not applicable	—	—

6.4.4 :EEPROM:CONFigure:BAUDrate?

The baudrate of RS232 can be read back. Baudrate values read back correspond to the following Baudrate:

Value	Baudrate
0	9600
1	19200
2	38400
3	57600
4	115200

Action	Data	Termination	Format
write	:EEPROM:CONFigure:BAUDRATE?	LF	ASCII
read	0	LF	ASCII

6.4.5 :EEPROM:CONFigure:AUTOrange

Set the eeprom auto-range value

Action	Data	Termination	Format
write	:EEPROM:CONFigure:AUTOrange 1	LF	ASCII
read	Not applicable	—	—

6.4.6 :EEPROM:CONFigure:AUTOrange?

Read the eeprom auto-range value

Action	Data	Termination	Format
write	:EEPROM:CONFigure:AUTOrange?	LF	ASCII
read	1	LF	ASCII

6.4.7 :EEPROM:CONFigure:ARPARMS

Set the auto range parameters.

Set parameter:	**Range:**	**Description:**
Frame Frequency [Hz]	0 – 250.0	frequency of source
Adjmin [%]	1 – 40	scale of ADC to regulate to in percentage
Max int time [µs]	1/freq – 60s.	maximum integration time to use in µs

Compared to the sense command the configure command does not have the average setting. The average is stored separately.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:ARPARMS 60,20,1000000	LF	ASCII
read	Not applicable	—	—

6.4.8 :EEPROM:CONFigure:ARPARMS?

Read the auto range parameters.

Return parameter:	**Range:**	**Description:**
Frame Frequency [Hz]	0 – 250.0	frequency of source
Adjmin [%]	1 – 40	scale of ADC level to regulate to in percentage
Max int time [µs]	1/freq – 60s.	maximum integration time to use in µs

Action	Data	Termination	Format
write	:EEPROM:CONFigure:ARPARMS?	LF	ASCII
read	60,20,1000000	LF	ASCII

6.4.9 :EEPROM:CONFigure:SPAVG

Set the average.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:SPAVG 2	LF	ASCII
read	Not applicable	—	—

6.4.10 :EEPROM:CONFigure:SPAVG?

Read the average.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:SPAVG?	LF	ASCII
read	2	LF	ASCII

6.4.11 :EEPROM:CONFigure:SPINT

Sets the default integration time of the spectrometer. Be aware that this value will be overruled when the auto-range is on.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:SPINT 10000	LF	ASCII
read	Not applicable	—	—

6.4.12 :EEPROM:CONFigure:SPINT?

Read the spectrometer integration time.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:SPINT?	LF	ASCII
read	10000	LF	ASCII

6.4.13 :EEPROM:CONFigure:CALPARMS

Write the calibration parameters. For a detailed explanation please refer to :SENSe:CALPARMS

Action	Data	Termination	Format
write	:EEPROM:CONFigure:CALPARMS 1,400,800,0.5,1,0	LF	ASCII
read	Not applicable	—	—

6.4.14 :EEPROM:CONFigure:CALPARMS?

Read the calibration parameters. For a detailed explanation please refer to :SENSe:CALPARMS?

Action	Data	Termination	Format
write	:EEPROM:CONFigure:CALPARMS?	LF	ASCII
read	1,400,800,0.5,1,0	LF	ASCII

6.4.15 :EEPROM:CONFigure:SPSBW

Set spectrometer user matrix

Action	Data	Termination	Format
write	:EEPROM:CONFigure:SPSBW user	LF	ASCII
read	Not applicable	—	—

6.4.16 :EEPROM:CONFigure:SPSBW?

Read spectrometer user matrix

Action	Data	Termination	Format
write	:EEPROM:CONFigure:SPSBW?	LF	ASCII
read	off	LF	ASCII

6.4.17 :EEPROM:CONFigure:IP

Set the IP address, applicable when using ethernet communication

Action	Data	Termination	Format
write	:EEPROM:CONFigure:IP 192.168.1.50	LF	ASCII
read	Not applicable	—	—

6.4.18 :EEPROM:CONFigure:IP?

Read the IP address, applicable when using ethernet communication

Action	Data	Termination	Format
write	:EEPROM:CONFigure:IP?	LF	ASCII
read	192.168.1.50	LF	ASCII

6.4.19 :EEPROM:CONFigure:GW

Set the gateway address, applicable when using ethernet communication

Action	Data	Termination	Format
write	:EEPROM:CONFigure:GW 192.168.1.1	LF	ASCII
read	Not applicable	—	—

6.4.20 :EEPROM:CONFigure:GW?

Read the gateway address, applicable when using ethernet communication

Action	Data	Termination	Format
write	:EEPROM:CONFigure:GW?	LF	ASCII
Read	192.168.1.1	LF	ASCII

6.4.21 :EEPROM:CONFigure:MASK

Set the network mask, applicable when using ethernet communication

Action	Data	Termination	Format
write	:EEPROM:CONFigure:MASK 255.255.255.0	LF	ASCII
read	Not applicable	—	—

6.4.22 :EEPROM:CONFigure:MASK?

Read the network mask, applicable when using ethernet communication

Action	Data	Termination	Format
write	:EEPROM:CONFigure:MASK	LF	ASCII
read	255.255.255.0	LF	ASCII

6.4.23 :EEPROM:CONFigure:TRIG

Write the device into trigger mode, if set to “1” the device eeprom value is set to trigger mode. If set to “0” the trigger mode is switched off

Action	Data	Termination	Format
write	:EEPROM:CONFigure:TRIG 1	LF	ASCII
read	Not applicable	—	—

6.4.24 :EEPROM:CONFigure:TRIG?

Read the device trigger mode, if set to “1” the device is set into trigger mode. If set to “0” the device trigger mode is switched off

Action	Data	Termination	Format
write	:EEPROM:CONFigure:TRIG?	LF	ASCII
read	1	LF	ASCII

6.4.25 :EEPROM:CONFigure:TRIGDELAY

Write a delay for trigger mode, measurement will start after the trigger is received and this delay has passed. Value is in [us].

Action	Data	Termination	Format
write	:EEPROM:CONFigure:TRIGDELAY 100	LF	ASCII
read	Not applicable	—	—

6.4.26 :EEPROM:CONFigure:TRIGDELAY?

Returns the trigger delay value, the value is in [us].

Action	Data	Termination	Format
write	:EEPROM:CONFigure:TRIGDELAY?	LF	ASCII
read	100	LF	ASCII

6.5 User EEPROM calibration commands

With the user eeprom calibration commands the user can program the following:

- Absolute calibration, also called intensity calibration. If the user would like to use its own light source or reference for absolute calibration this is possible by this calibration
- wavelength calibration, this enables the user to do its own wavelength calibration. If this calibration is done, we also advice to do an absolute calibration
- Matrix calibration, the 3x3 user matrix can be programmed. This function is only valid for the XYZ domain and will only be applied on the :MEASure:XYZ and :MEASure:YXY

After the description of the calibration commands an example will be given for each calibration in the examples section.

It is recommended after writing and storing the values into the eeprom that you re-power the device.

6.5.1 :EEPROM:ABS:FIX

This command stores the written absolute calibration values and wavelength LUT values into the eeprom. It is important to wait 20ms after sending this command. Storing physically to the eeprom requires some time.

Action	Data	Wait	Termination	Format
write	:EEPROM:ABS:FIX	20ms	LF	ASCII
read	Not applicable	—	—	—

6.5.2 :EEPROM:ABS:WRITE

This command stores the absolute user calibration values into the eeprom.

The structure is the following:

Param	Index	Value
0	0	length
1	0	int time
2	index	wavelength
3	index	calibration value

When writing the calibration values the first parameter which needs to be written is the length of the nr of correction factors you are going to write, this should be the same length for the nr. of wavelength values. It is very important that you write correctly the length correction values and wavelengths. Length is determined as parameter 0.

Parameter 1 is the integration time which you used for calibration. We added this parameter for traceability purpose. This is value is not used in calibration/compensation itself, it is purely meant for traceability.

Parameter 2 is the wavelength. Here the index also becomes important, when writing the absolute calibration the calibration/correction factor is set per wavelength. First write all the wavelength values which you want to compensate. For each next wavelength increment the index value.

Parameter 3 is the correction factor also called the calibration value which belongs to the wavelength you have written.

Then nr of wavelength values you write should be exactly the same as the nr of calibration values you write, both should have the same length and this length should be the value of parameter 0 (length).

When calibrating there is no need to match the exact wavelength steps of the spectrometer, inside the spectrometer the values are interpolated to the native resolution of the spectrometer
Very important is to wait after each write 5ms.

In the example the 4 parameter sections have been highlighted by different colors

The length can never be larger than 2048

Action	Data	Wait	Termination	Format
write	:EEPROM:ABS:WRITE 0,0,301	5ms	LF	ASCII
write	:EEPROM:ABS:WRITE 1,0,10000	5ms	LF	ASCII
write	:EEPROM:ABS:WRITE 2,0,400	5ms	LF	ASCII
write	:EEPROM:ABS:WRITE 2,1,401	5ms	LF	ASCII
	……			
	……			
write	:EEPROM:ABS:WRITE 2,299,699	5ms	LF	ASCII
write	:EEPROM:ABS:WRITE 1,300,700	5ms	LF	ASCII
write	:EEPROM:ABS:WRITE 3,0,0.171255	5ms	LF	ASCII
write	:EEPROM:ABS:WRITE 3,1,0.169139	5ms	LF	ASCII
	……			
	……			
write	:EEPROM:ABS:WRITE 3,299,0.077330	5ms	LF	ASCII
write	:EEPROM:ABS:WRITE 3,300,0.0776182	5ms	LF	ASCII
read	Not applicable			

6.5.3 :EEPROM:ABS:READ

This command enables you to read back the absolute user calibration values. The structure is the same as the write command and is the following:

Param	Index	Value
0	0	length
1	0	int time
2	index	wavelength
3	index	calibration value

Action	Data	Termination	Format
write	:EEPROM:ABS:READ 0,0	LF	ASCII
read	301	LF	ASCII
write	:EEPROM:ABS:READ 1,0	LF	ASCII
read	10000	LF	ASCII
write	:EEPROM:ABS:READ 2,0	LF	ASCII
read	400	LF	ASCII
write	:EEPROM:ABS:READ 2,1	LF	ASCII
read	401		
	……		
	……		
write	:EEPROM:ABS:READ 2,299	LF	ASCII
read	699	LF	ASCII
write	:EEPROM:ABS:READ 1,300	LF	ASCII
read	700	LF	ASCII
write	:EEPROM:ABS:READ 3,0	LF	ASCII
read	0.171255	LF	ASCII
write	:EEPROM:ABS:READ 3,1	LF	ASCII
read	0.169139	LF	ASCII
	……		
	……		
write	:EEPROM:ABS:READ 3,299	LF	ASCII
read	0.07733	LF	ASCII
write	:EEPROM:ABS:READ 3,300	LF	ASCII
read	0.0776182	LF	ASCII

6.5.4 :EEPROM:WLLUTWRITE

Be aware that this command does not have a : before the WRITE. This command stores the wave-length user calibration values into the eeprom. The structure is the following:

Param	Index	Value
0	0	length
1	index	pixel value
2	index	wl value

The basic theory behind this calibration is that you are going to write a LUT (Look Up Table) where on the one hand you have the pixel values and on the other hand you have corresponding wavelengths value which belong to that certain pixel. In other words you are telling the device that pixel “x” in reality is wavelength "y". You are mapping strategic pixel values to a wavelength. All values in between are being interpolated.

When writing the calibration values the first parameter which needs to be written is the length of the nr of pixel values you are going to write, this should be the same length for the nr. of wavelength values. It is very important that you write correctly the length. Length is determined as parameter 0.

Parameter 1 is the pixel value. Here the index also becomes important, because you want to do the wavelength calibration with multiple pixels. First write all the wavelength values which you want to use for wavelength calibration. For each next pixel increment the index value.

Parameter 2 are the wavelength values which belong the pixels you have just written. The index also needs to be incremented.

Very important is to wait after each write 5ms.

Maximum length of wavelength calibration values which you can program are 20.

Action	Data	Wait	Termination	Format
write	:EEPROM:WLLUTWRITE 0,0,9	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 1,0,148.54	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 1,1,252.313	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 1,2,320.032	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 1,3,382.812	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 1,4,451.937	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 1,5,548.667	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 1,6,639.53	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 1,7,712.472	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 1,8,756.655	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 2,0,404.656	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 2,1,467.815	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 2,2,508.582	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 2,3,546.047	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 2,4,587.092	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 2,5,643.847	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 2,6,696.543	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 2,7,738.398	5ms	LF	ASCII
write	:EEPROM:WLLUTWRITE 2,8,763.511	5ms	LF	ASCII
read	Not applicable			

6.5.5 :EEPROM:WLLUTREAD

Be aware that this command does not have a : before the READ. This command enables you to read back the user wavelength calibration values. The structure is the same as the write command and is the following:

Param	Index	Value
0	0	length
1	index	pixel value
2	index	wl value

Action	Data	Termination	Format
write	:EEPROM:WLLUTREAD 0,0	LF	ASCII
read	9	LF	ASCII
write	:EEPROM:WLLUTREAD 1,0	LF	ASCII
read	148.54	LF	ASCII
write	:EEPROM:WLLUTREAD 1,1	LF	ASCII
read	252.313	LF	ASCII
write	:EEPROM:WLLUTREAD 1,2	LF	ASCII
read	320.032	LF	ASCII
write	:EEPROM:WLLUTREAD 1,3	LF	ASCII
read	382.812	LF	ASCII
write	:EEPROM:WLLUTREAD 1,4	LF	ASCII
read	451.937	LF	ASCII
write	:EEPROM:WLLUTREAD 1,5	LF	ASCII
read	548.667	LF	ASCII
write	:EEPROM:WLLUTREAD 1,6	LF	ASCII
read	639.53	LF	ASCII
write	:EEPROM:WLLUTREAD 1,7	LF	ASCII
read	712.472	LF	ASCII
write	:EEPROM:WLLUTREAD 1,8	LF	ASCII
read	756.655	LF	ASCII
write	:EEPROM:WLLUTREAD 2,0	LF	ASCII
read	404.656	LF	ASCII
write	:EEPROM:WLLUTREAD 2,1	LF	ASCII
read	467.815	LF	ASCII
write	:EEPROM:WLLUTREAD 2,2	LF	ASCII
read	508.582	LF	ASCII
write	:EEPROM:WLLUTREAD 2,3	LF	ASCII
read	546.047	LF	ASCII
write	:EEPROM:WLLUTREAD 2,4	LF	ASCII
read	587.092	LF	ASCII
write	:EEPROM:WLLUTREAD 2,5	LF	ASCII
read	643.847	LF	ASCII
write	:EEPROM:WLLUTREAD 2,6	LF	ASCII
read	696.543	LF	ASCII
write	:EEPROM:WLLUTREAD 2,7	LF	ASCII
read	738.398	LF	ASCII
write	:EEPROM:WLLUTREAD 2,8	LF	ASCII
read	763.511	LF	ASCII

6.5.6 :EEPROM:WRITE:USERCAL

This command stores the written values of the 3x3 matrix into the eeprom.

Wait 20ms after writing this command. This is the necessary time for the eeprom to physically store everything.

Action	Data	Wait	Termination	Format
write	:EEPROM:WRITE:USERCAL	20ms	LF	ASCII
read	Not applicable			

6.5.7 :EEPROM:SPSBW:WRITE

This command write the values of the 3x3 matrix to the device

XYZ values returning from the devices are multiplied with this matrix when SBW is set to “user”. This can be done with the :SENSe:SP:SBW command.

Calculation inside the device will be the following

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \end{bmatrix} = M \times \begin{bmatrix} X_{dut} \\ Y_{dut} \\ Z_{dut} \end{bmatrix}$$

M is the matrix, the formula can also be written in the following equation

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \end{bmatrix} = \begin{bmatrix} CF_{0,0} & CF_{0,1} & CF_{0,2} \\ CF_{0,1} & CF_{1,1} & CF_{1,2} \\ CF_{0,2} & CF_{2,1} & CF_{2,2} \end{bmatrix} \times \begin{bmatrix} X_{dut} \\ Y_{dut} \\ Z_{dut} \end{bmatrix}$$

Writing the correction factors must be done in 9 steps where in each step you write three parameters. First write the row value, after that the column value and after that the Correction Factor.

After each write wait 5ms.

Action	Data	Wait	Termination	Format
write	:EEPROM:SPSBW:WRITE 0,0,1.01	5ms	LF	ASCII
write	:EEPROM:SPSBW:WRITE 1,0,0.002	5ms	LF	ASCII
write	:EEPROM:SPSBW:WRITE 2,0,0.041	5ms	LF	ASCII
write	:EEPROM:SPSBW:WRITE 0,1,-0.035	5ms	LF	ASCII
write	:EEPROM:SPSBW:WRITE 1,1,0.98	5ms	LF	ASCII
write	:EEPROM:SPSBW:WRITE 2,1,0.11	5ms	LF	ASCII
write	:EEPROM:SPSBW:WRITE 0,2,0.021	5ms	LF	ASCII
write	:EEPROM:SPSBW:WRITE 1,2,-0.016	5ms	LF	ASCII
write	:EEPROM:SPSBW:WRITE 2,2,1.102	5ms	LF	ASCII
Read	Not applicable			

6.5.8 :EEPROM:SPSBW:READ

This command write the values of the 3x3 matrix to the device

This command has two parameters, first one is the row and second is the column.

Returned value is the correction factor.

Action	Data	Termination	Format
write	:EEPROM:SPSBW:READ 0,0	LF	ASCII
read	1.01	LF	ASCII
write	:EEPROM:SPSBW:READ 1,0	LF	ASCII
read	0.002	LF	ASCII
write	:EEPROM:SPSBW:READ 2,0	LF	ASCII
read	0.041	LF	ASCII
write	:EEPROM:SPSBW:READ 0,1	LF	ASCII
read	-0.035	LF	ASCII
write	:EEPROM:SPSBW:READ 1,1	LF	ASCII
read	0.98	LF	ASCII
write	:EEPROM:SPSBW:READ 2,1	LF	ASCII
read	0.11	LF	ASCII
write	:EEPROM:SPSBW:READ 0,2	LF	ASCII
read	0.021	LF	ASCII
write	:EEPROM:SPSBW:READ 1,2	LF	ASCII
read	-0.016	LF	ASCII
write	:EEPROM:SPSBW:READ 2,2	LF	ASCII
read	1.102	LF	ASCII

7 ABSOLUTE CALIBRATION – HOW TO CALCULATE

Coming soon

8 3X3 MATRIX – HOW TO CALCULATE

Below the formula is give on how to determine a 3x3 matrix.

- Matrix M is determined on RGB colors
- Xrr means X value of reference for the red color
- Ydg means Y value of DUT for the green color
- T is a transposed matrix
- -1 is an inverse matrix

$$M = \begin{bmatrix} CF_{0,0} & CF_{0,1} & CF_{0,2} \\ CF_{1,0} & CF_{1,1} & CF_{1,2} \\ CF_{2,0} & CF_{2,1} & CF_{2,2} \end{bmatrix} = \left(\begin{bmatrix} X_{rr} & X_{rg} & X_{rb} \\ Y_{rr} & Y_{rg} & Y_{rb} \\ Z_{rr} & Z_{rg} & Z_{rb} \end{bmatrix} \times \begin{bmatrix} X_{dr} & X_{dg} & X_{db} \\ Y_{dr} & Y_{dg} & Y_{db} \\ Z_{dr} & Z_{dg} & Z_{db} \end{bmatrix}^T \right) \times \left(\begin{bmatrix} X_{dr} & X_{dg} & X_{db} \\ Y_{dr} & Y_{dg} & Y_{db} \\ Z_{dr} & Z_{dg} & Z_{db} \end{bmatrix} \times \begin{bmatrix} X_{dr} & X_{dg} & X_{db} \\ Y_{dr} & Y_{dg} & Y_{db} \\ Z_{dr} & Z_{dg} & Z_{db} \end{bmatrix}^T \right)^{-1}$$

9 CODING EXAMPLES

9.1 Configuration and measurement examples

First example is a simple, whereby we switch auto-range "on". We will also configure the auto-range parameters and we want a calibrated spectrum with a wavelength range of 400-700nm with a wave-length step of 1nm.

Example sequence of commands to initialize the instrument and measure spectrums.

Command	Result	Remark
:SENSe:AUTORANGE 1		set autorange on
:SENSe:ARPARMS 60,20,1000000,1		set auto-range parameters.
:SENSe:CALPARMS 1,400,700,1,0,0		set calibration parameters
:GET:SPECSize	301	size of wavelength array which is the same size of the spectrum intensity array
:GET:WAVElengths	0x02,0xBC.......0x90	wavelength array returned is 400,401,....699,700
:MEASure:SPECtrum 0	0x00,0xAA.......0x16	measure spectrum 1 number of intensity values returned are 401
:MEASure:SPECtrum 0	0x00,0xAB.......0x23	measure spectrum 2 number of intensity values returned are 401

9.2 User eeprom configuration example

A simple example is given for writing the baudrate and auto-range setting to the eeprom.

Command	Wait	Remark
:EEPROM:CONFigure:AUTOrange 1		write auto-range start up setting. Auto-range setting is switched on
:EEPROM:CONFigure:BAUDRATE 0		Baudrate setting is set to 9600
:EEPROM:STARTUP:WRITE	20ms	auto-range and baudrate setting are stored inside the eeprom.

9.3 User eeprom absolute calibration example

Coming soon

9.4 User eeprom wavelength calibration example

Coming soon

10 AUTO-RANGE FUNCTION

10.1 Introduction

The Rhea02 includes an auto-range function for the spectrometer sensor. This function is useful in case the measured object shows an unknown luminance value. In this case, the Rhea02 will try to find the optimum setting which is a trade-off between speed and the stability of the instrument. The auto-range function can also be fine-tuned to reach better stability levels by setting a few parameters. Auto-ranging can be controlled by 3 parameters.

- Frequency : supposed to be frame frequency of the source (display) that is measured
- Adjmin : The minimum level to adjust to, this is the scale of the ADC expressed in %
- MAX int time: after this time has passed the device will not increase the integration time beyond this value even if the Adjmin level was not reached. This is to avoid a measurement time which takes forever. This can occur when virtually no light is present.

Adjmin can vary between 1 and 40. Adjmin effects the stability of the result. The higher the value the better the S/N ratio. If higher stability is needed, this setting must be increased (but measurements will become slower). If a faster measurement is needed and the stability level may be lowered a little, the value of adjmin can be decreased. When the Rhea02 measures in auto range mode, it can occur that the found integration time is very low. For example when measuring high light levels it may be just 7ms. When this happens, the Rhea will automatically increase the averaging so that the total measurement time equals >= (1/frequency) · averaging.

10.2 How autoranging works

The auto ranging works, by first setting a default integration time. If this already meets the criteria for a good measurement, the measurement will be done using that integration time. It should be clear that this is the fastest because no adjustment will be done. In case the measured result is either to low or too high (clip) the Rhea will adjust to a new level by using steps of at least 1/frequency. The following graph shows how levels inside the Rheawork.

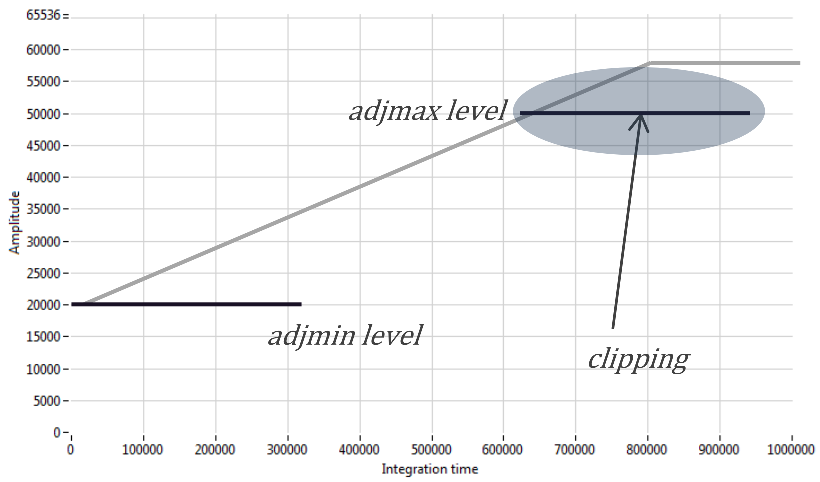


Fig 5 Autorange levels

When auto-ranging is set, the Rhea will accept any level between “Adjmin” and “Adjmax” as a good signal. The Adjmin level can be set by the user through a software command, allowing some fine tuning of the stability and the speed of the instrument.

10.3 Auto-range measurement parameters

Auto-range controls the following parameters:

- Frequency: This setting must be set to the frame frequency of the measured sample. The range is 1 to 255 Hz.
- Adjmin: The adjmin parameter is very important. The higher this level is set, the more stable the measurement becomes, but it also slows down the measurement if set too high. That means, there is a trade-off between speed and stability. It mainly affects the lower grey scale levels. If these levels appear not stable, adjmin needs to be increased. Adjmin is set from 1-40, which means 1-40% of the Adjmax level (Adjmax is fixed by Admesy).
- Maximum integration time: Although not really part of the auto-ranging algorithm, this parameter is used when the auto-range algorithm would result in an integration time exceeding this value. This is to avoid measurement times which would last forever (in the case no light is present)
- Averaging: Average can be set by the user, average will not have influence on the time needed for finding the appropriate level, however when the appropriate level is found the "real" measurement is executed with the nr. of averages set.

10.4 Auto-range measurement time

The measurement time when using autorange usually is about 2 times the used integration time. Please consider this for your software time-out value. A good time-out value is in general 3x the max integration time.

10.5 Rhea Auto-range good practice settings

- Frequency: Frequency of your light source
- Display 60Hz
- Lighting 100Hz
- Adjmin: 10%
- Maximum integration time: 5000000 us (5sec)
- Averaging: 1

The above settings are based on our experience and every application can be different. It is the responsibility of the end-user to validate the correct setting, no rights can be derived from above recommendation. It must be considered as a push in the right direction.

11 RHEA02 MEASUREMENT RESULTS

The Rhea02 is a spectrometer that on request can provide direct calibrated output. In general the Rhea02 will supply you with the calibrated or non-calibrated spectral data. Processing of this data is up to the user, however Admesy also supplies some dll's whereby spectral data can be inputted and some appropriate calculations can be applied.

Available dll and examples can be found on the website.

12 TRIGGER MODE

A trigger actives only one command, for example ":MEASure:XYZ". This command needs to be set first by the host. After a trigger is received, the command will execute, and the result will be presented on the selected interface as shown above. The device responds to a rising edge of the trigger signal.

colorimeters | spectroradiometers | lightmeters

Sleestraat 3
6014 CA Ittervoort
The Netherlands

+31 (0)475 600 232
info@admesy.com

admesy.com

Version 1.0.4 June 2020