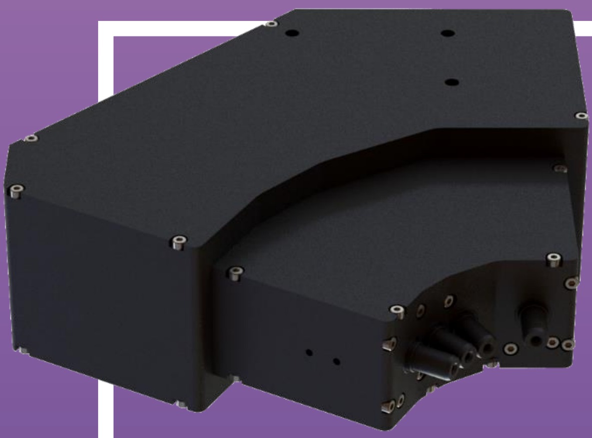

HYPERIONQ COLORIMETER

Operating Manual

ADMESY

colorimeters | spectroradiometers | lightmeters

CONTENTS

1 INTRODUCTION

1.1 HyperionQ

The HyperionQ colorimeter is a multi-angle colorimeter based on the Hyperion colorimeter series featuring 0°, 30°, 45°, 60° simultaneous measurement.

1.2 HyperionQ highlights

- Multiple angle simultaneous measurement of luminance and colour.
- Flicker measurement
- Mechanical shutter
- Low noise
- Auto-range function
- Excellent linearity
- Dark current compensated
- USBTMC compliant, SCPI command set, high speed device
- USB, RS232 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

1.3 Standards

The HyperionQ 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 HyperionQ 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 HyperionQ is external powered, please use included power supply.

2.2 RS232 interface

RS232 is provided to connect any host that does not provide USB or Ethernet or for which no USBTMC drivers exist. Using RS232, the functions that generate a lot of data (:sample:Y) 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'

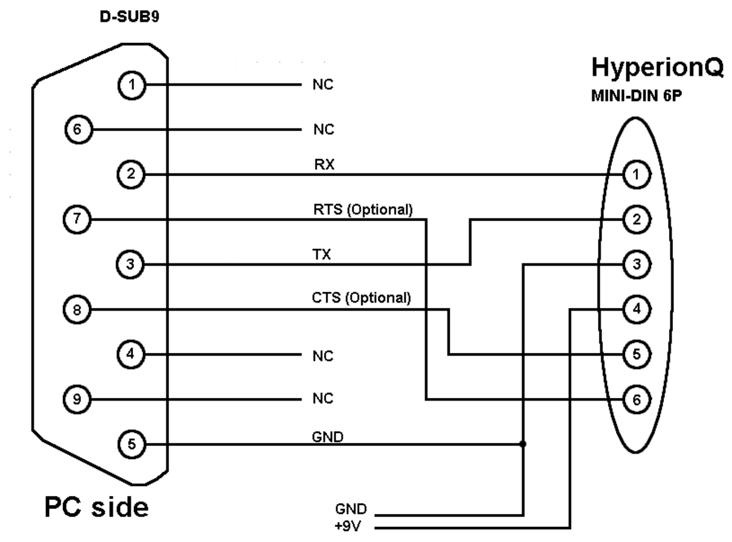

Fig 1 HyperionQ RS232 connection.

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

2.3 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 (:MEASURE or :SAMPLE) and send the result to the host via the selected interface (last used interface when sending the command).

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).

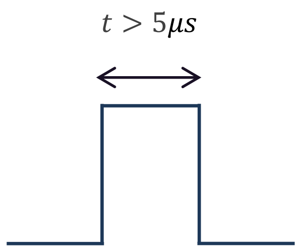

Fig 3 Trigger-in timing.

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	0 V	5 V
Trigger In	0 V	5 V

2.4 Power connections (FIXME)

The HyperionQ 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 9V DC voltage, reinforced separated from Mains, with a limited energy of < 150VA and < 8A.

Connection	Min. Voltage	Typ. voltage	Max. voltage
DC powered	8.50 V	9.00 V	15.00 V

3 COMMUNICATIONS PROTOCOL

3.1 USB

The HyperionQ 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 HyperionQ directly usable in popular programming languages like NI's Labview and Labwindows but also C++, Visual basic, C#, Java etc. The HyperionQ has two interfaces build in, which require a different device driver to be used.

- HyperionQ bootloader - USB RAW device driver, Vendor ID: 0x23CF, Product ID 0x0105
- HyperionQ - USBTMC device driver, Vendor ID: 0x23CF, Product ID 0x108F

When the HyperionQ is connected to the host, it will start the HyperionQ 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 HyperionQ is USBTMC compliant and can be used with libraries that contain a USBTMC compliant driver like NI-VISA. The HyperionQ is a USB 2.0 high speed device.

3.2 RS232

All commands are equal for all interfaces. Note that for high speed transfers it is best to use USB.

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 overview 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 in our SDK 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 HyperionQ.

5 COMMAND SET OVERVIEW

5.1 Command set categories

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

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

The HyperionQ 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
IDO	—	device name	identification query
:*RST	—	—	reset command
:*FWD?	—	firmware date	reads firmware date
:SYSTem:VERSion?	—	firmware version	returns firmware version
:BOOT:REBOOT	—	—	Reboot the instrument

5.3 Configuration commands

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

Command	Write Parameter	Read Result	Description
:ANGLE:SELECT	Angle	—	Selects the specified angle
:ANGLE:DESELECT	Angle	—	Deselects the specified angle
:ANGLE:ISSELECTED	Angle	0 or 1	Checks if the specified angle is selected
:ANGLE:SELECTED?	—	State per angle	Checks selection state of all angles
:SENSe:SHUTter	Shutter state	—	close (1) or open (0) shutter
:SENSe:SHUTter?	—	Shutter state	Reads the shutter state
:SENSe:AUTORANGE	Autorange	—	autorange on (1) or off (0) for allchannels
:SENSe:INT	Integration time	—	Wrtites the integration time for all selected angles
:SENSe:INT?	Angle	—	Reads the integration time of the specified angle
:SENSe:AVERage	average	—	sets averaging for all selected angles
:SENSe:AVERage?	Angle	average	Reads the averaging of the specified angle
:SENSe:CHAVER?	Angle	Samples per channel	reads nr of samples per angle
:SENSe:GAIN	Gain (1-3)	—	Sets the gain for all selected angles
:SENSe:GAIN?	Angle	Gain	Reads the gain of the specified angle
:SENSe:SBW	matrix	—	set the calibration matrix for all selected angles
:SENSe:AUTOMODE	mode (fast, normal, slow, hisense, manual)	—	Writes the mode for all selected angles
:SENSe:FREQ	Frequency	—	Writes the DUT frequency
:SENSe:FREQ?	—	Frequency	Reads the DUT frequency
:SENSe:ARFREQ	Autorange frequency	—	Sets the frequency to be used for auto ranging
:SENSe:ARFREQ?	—	Autorange frequency	Reads the frequency that is used for auto ranging
:SENSe:SATURATION?	Angle	SaturationX, SaturationY, SaturationZ [0-1 float]	Reads the saturation level of the last performed measurement

5.4 Measurement commands

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

Command	Write Parameter	Read Result	Description
:MEASure:TEMPerature	—	temperature	measure temperature for all selected angles
:MEASure:XYZ	—	X, Y, Z, clip, noise	measures XYZ for all selected angles
:MEASure:YXY	—	Y, CIEx, CIEy, clip, noise	measures Yxy for all selected angles
:MEASure:YUV	—	Y, u, v, clip, noise	measures Yuv for all selected angles
:MEASure:CCT	—	CCT, clip, noise	measures CCT for all selected angles
:MEASure:DWL	—	DWL, Pe, Y, CIEx, CIEy, clip, noise	measure dominant wavelength for all selected angles
:MEASure:Y	—	Y, clip, noise	measures luminance for all selected angles
:MEASure:FREQuency	Samples	frequency, amplitude, clip	Measures the detected frequency of the DUT plus the amplitude of this frequency
:MEASure:FLICker	method, samples, delay	Flicker value	Measures MIN/MAX, RMS, JEITA, VESA, percentage or index flicker
:MEASure:MVFLicker	samples, delay	Mmflicker, MMflickerdB, RMSflicker, RMSflicerdB, JEITA%, JEITAdB, VESA%, VESAdB, clip	Measures multiple flicker methods using one function.
:SAMPle:XYZ	samples, delay	dt, clip, XYZ data	XYZ data per sample
:SAMPle:Y	samples, delay	dt, clip, Y data	Y data per sample

Command	Write Parameter	Read Result	Description
:MEASure:FLINdex	samples, delay, freq, StoreSignal	flicker, clip	measures flicker index
:READ:FLINDATA	samples	signal data (similar to :sample:Y)	reads raw signal data stored by the ":MEASure:FLINdex" command
:MEASure:FFLINdex	samples, delay, freq, viewingArea, StoreSignal	flicker, clip	measures fusion flicker index
:READ:FFLINDATA	samples	signal data (similar to :sample:Y)	reads raw signal data stored by the ":MEASure:FFLINdex" command
:MEASure:XYZFLINdex	samples, delay, freq, StoreSignal	flicker, clip, X, Y, Z	Measures flicker index and XYZ
:READ:XYZFLINDATA	samples	signal data (similar to :sample:XYZ)	reads raw signal data stored by the ":MEASure:XYZFLINdex" command
:MEASure:XYZFFLINdex	samples, delay, freq, viewingArea, StoreSignal	flicker, clip, X, Y, Z	measures fusion flicker index and XYZ
:READ:XYZFFLINDATA	samples	signal data (similar to :sample:XYZ)	reads raw signal data stored by the ":MEASure:XYZFFLINdex" command

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 HyperionQ after writing new values to the EEPROM.

Command	Write Parameter	Read Result	Description
:EEPROM:STARTUP:WRITE	—	—	stores startup values to eeprom
:EEPROM:STARTUP:READ	—	—	read all startup eeprom values
:EEPROM:CONFigure:BAUDRATE	baudrate setting	—	write RS232 baudrate setting
:EEPROM:CONFigure:BAUDRATE?	—	baudrate setting	read RS232 baudrate setting
: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]
:EEPROM:CONFigure:LPFREQ	Low pass frequency	—	sets the low pass frequency
:EEPROM:CONFigure:LPFREQ?	—	Low pass frequency	reads the low pass frequency
:EEPROM:CONFigure:AVG	average	—	sets the initial averaging for all channels
:EEPROM:CONFigure:AVG?	—	average	reads the initial averaging
:EEPROM:CONFigure:WHITE	reference white ("D65")	—	sets the reference point for dominant wavelength calculation.
:EEPROM:CONFigure:WHITE?	—	reference white	reads the reference white point
:EEPROM:WRITE:ID	Device ID	—	Writes an ID number to the device
:EEPROM:READ:ID	—	Device ID	Reads the device ID number from the device
:EEPROM:CONFigure:AUTO:FREQ	Frequency	—	writes the DUT frequency
:EEPROM:CONFigure:AUTO:FREQ?	—	Frequency	reads the DUT frequency
:EEPROM:CONFigure:AUTO:FRAMES	Frames	—	write the number of frames to measure
:EEPROM:CONFigure:AUTO:FRAMES?	—	Frames	reads the number of frames
:EEPROM:CONFigure:AUTOrange	State, angle	—	read autorange state
:EEPROM:CONFigure:AUTORANGE?	angle	State	write autorange state
:EEPROM:CONFigure:AUTO:ADJMIN	adjmin, angle	—	write the adjmin value per angle
:EEPROM:CONFigure:AUTO:ADJMIN?	angle	Adjmin	reads the adjmin value per angle
:EEPROM:CONFigure:AUTO:MAXINT	maxint, angle	—	writes maximum integration time per angle
:EEPROM:CONFigure:AUTO:MAXINT?	angle	maxint	reads the maximum integration time per angle
:EEPROM:CONFigure:AUTO:MODE	mode, angle	—	write the autorange mode per angle
:EEPROM:CONFigure:AUTO:MODE?	angle	mode	reads the autorange mode per angle
:EEPROM:CONFigure:INT	inttime, angle	—	write the initial integration time per angle
:EEPROM:CONFigure:INT?	angle	inttime	reads the initial integration time
:EEPROM:CONFigure:GAIN	gain, angle	—	writes the initial gain per angle
:EEPROM:CONFigure:GAIN?	angle	Gain	reads the gain per angle
:EEPROM:CONFigure:SBW	matrixID, angle	—	Write the initial calibration matrix per angle
:EEPROM:CONFigure:SBW?	angle	MatrixID	reads the initial calibration matrix per angle

Command	Write Parameter	Read Result	Description
:EEPROM:READ:SN	—	Serial number	Reads the instrument serial number

5.6 User EEPROM calibration commands

Below table shows the commands which can be used to store user calibrations (matrices).

It is advised to reboot the HyperionQ 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:READ:USERCAL	—	—	Transfers user calibration data from EEPROM to RAM
:EEPROM:WRITE:USERCAL	—	—	Transfers user calibration data from RAM to EEPROM
:EEPROM:SBW:WRITE	matrixID, row, col, value, angle	—	Write matrix values
:EEPROM:SBW:READ	matrixID, row, col, angle	value	reads matrix values
:EEPROM:READ:SBWNAME	angle, matrixID	name	Reads the matrix name

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. HyperionQ	LF	ASCII

6.1.2 IDO

Reads the identification query of the device

Action	Data	Termination	Format
write	IDO	LF	ASCII
read	Admesy B.V. HyperionQ	LF	ASCII

6.1.3 :*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		

6.1.4 :*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

6.1.5 :SYSTem:VERSion?

Reads the firmware version

Action	Data	Termination	Format
write	:SYSTem:VERSion?	LF	ASCII
read	0.44	LF	ASCII

6.1.6 :BOOT:REBOOT

Reads the firmware version

Action	Data	Termination	Format
write	:BOOT:REBOOT	LF	ASCII
read	Not applicable		

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.

6.2.1 :ANGLE:SELECT

With this command an angle is marked to be active

Action	Data	Termination	Format
write	:ANGLE:SELECT 1	LF	ASCII
Read	Not applicable	LF	ASCII

Angle can be 1,2,3 or 4. When 0 or 255 is specified, all angles will be selected.

6.2.2 :ANGLE:DESELECT

With this command an angle is marked to be inactive

Action	Data	Termination	Format
write	:ANGLE:DESELECT 1	LF	ASCII
Read	Not applicable	LF	ASCII

Angle can be 1,2,3 or 4. When 0 or 255 is specified, all angles will be deselected.

6.2.3 :ANGLE:ISSELECTED?

With this command it can be checked if an angle is active.

Action	Data	Termination	Format
write	:ANGLE:ISSELECTED 1	LF	ASCII
Read	1	LF	ASCII

Angle can be 1,2,3 or 4. When 0 or 255 is specified, all angles will be selected.

6.2.4 :ANGLE: SELECTED?

This command returns which angles are active (1) and inactive (0)

Action	Data	Termination	Format
write	:ANGLE:SELECTED?	LF	ASCII
Read	1,0,1,0	LF	ASCII

6.2.5 :ANGLE:ISSELECTED?

With this command it can be checked if an angle is active.

Action	Data	Termination	Format
write	:ANGLE:ISSELECTED 1	LF	ASCII
Read	1	LF	ASCII

Angle can be 1,2,3 or 4. When 0 or 255 is specified, all angles will be selected.

6.2.6 :SENSe:SHUTter

With this command it can be checked if an angle is active.

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

6.2.7 :SENSe:SHUTter?

With this command it can be checked if the shutter is open(0) or closed(1)

Action	Data	Termination	Format
write	:SENSe:SHUTter?	LF	ASCII
Read	0	LF	ASCII

6.2.8 :SENSe:AUTORANGE

With this command, autorange can be set on (1) or off (0) for all angles.

Action	Data	Termination	Format
Write	:SENSe:AUTORANGE 1	LF	ASCII
Read	Not applicable	LF	ASCII

6.2.9 :SENSe:INT

This command sets the integration time for all selected angles.

Action	Data	Termination	Format
Write	:SENSe:INT 16666	LF	ASCII
Read	Not applicable	LF	ASCII

6.2.10 :SENSe:INT?

This command gets the current inttime for the specified angle.

Action	Data	Termination	Format
Write	:SENSe:INT? 1	LF	ASCII
Read	16666	LF	ASCII

6.2.11 :SENSe:AVERage

This command sets the average for all selected angles.

Action	Data	Termination	Format
Write	:SENSe:AVERage 1	LF	ASCII
Read	Not applicable	LF	ASCII

6.2.12 :SENSe:AVERage?

This command gets the current averaging for the specified angle.

Action	Data	Termination	Format
write	:SENSe:AVERage? 1	LF	ASCII
Read	1	LF	ASCII

6.2.13 :SENSe:CHAVER?

This command gets the number of samples per angles.

For example, when in high luminance mode a frame is 16.666ms and the integration time is 1666ms, the number of samples will become 10 to match to the frame time.

Action	Data	Termination	Format
write	:SENSe:CHAVER? 1	LF	ASCII
Read	10	LF	ASCII

6.2.14 :SENSe:GAIN

This command sets the gain for all selected angles.

Action	Data	Termination	Format
Write	:SENSe:GAIN 1	LF	ASCII
Read	Not applicable	LF	ASCII

Gain 3 is the least sensitive (for high luminance). Gain 1 is the most sensitive, to be used for low luminance measurements.
When in auto-range mode, it can save tact time to set the gain to a correct value. For example when it is known that the next pattern to be measured is a dark pattern it saves a little time to manually set gain=1 and then perform ":meas:XYZ". The same applies for a high luminance pattern, setting the gain to 3.

6.2.15 :SENSe:GAIN?

This command gets the gain of the specified angle

Action	Data	Termination	Format
Write	:SENSe:GAIN? 1	LF	ASCII
Read	3	LF	ASCII

6.2.16 :SENSe:SBW

This command sets the calibration matrix for all selected angles.

Action	Data	Termination	Format
write	:SENSe:SBW user1	LF	ASCII
Read	Not applicable	LF	ASCII

6.2.17 :SENSe:AUTOMODE

This command sets the autorange mode for all selected angles.

Action	Data	Termination	Format
write	:SENSe:AUTOMODE 1	LF	ASCII
Read	Not applicable	LF	ASCII

Automode can be fast(0),normal(1),slow(2),hisense(3),manual(4)

6.2.18 :SENSe:FREQ

This command sets the DUT frequency.

Action	Data	Termination	Format
write	:SENSe:FREQ 59.567	LF	ASCII
Read	Not applicable	LF	ASCII

6.2.19 :SENSe:FREQ?

This command reads the DUT frequency setting

Action	Data	Termination	Format
Write	:SENSe:FREQ?	LF	ASCII
Read	59.567	LF	ASCII

6.2.20 :SENSe:ARFREQ

This command sets the frequency to be used during autorange mode. In case FREQ is set to a low frequency it is possible to speed up autoranging by setting this to a higher value. This frequency however should match the real display frame frequency (60Hz for example).

Action	Data	Termination	Format
Write	:SENSe:ARFREQ 60	LF	ASCII
Read	Not applicable	LF	ASCII

6.2.21 :SENSe:ARFREQ?

This command reads the autorange frequency setting

Action	Data	Termination	Format
Write	:SENSe:ARFREQ?	LF	ASCII
Read	60	LF	ASCII

6.2.22 :SENSe:SATURATION?

This command reads the saturation of the last performed measurement of the specified angle.

Action	Data	Termination	Format
write	:SENSe:SATURATION? 1	LF	ASCII
Read	0.678,0.898,0.776	LF	ASCII

6.3 Measurement commands

6.3.1 :MEASure:TEMPerature

Returns the temperature per angle

Action	Data	Termination	Format
write	:MEASure:TEMPerature	LF	ASCII
read	38.123,39.010,38.789,38.12	LF	ASCII

Note, the output depends on how many angles are selected.

6.3.2 :MEASure:XYZ

Measures XYZ of all selected angles. The example shows 2 angles.

Action	Data	Termination	Format
write	:MEASure:XYZ	LF	ASCII
read	100.1234,100.432,100.78932,0,0, 50.12,50.3323,49.234,0,0	LF	ASCII

Note, the output depends on how many angles are selected.

6.3.3 :MEASure:YXY

Measures Yxy of all selected angles. The example shows 2 angles.

Action	Data	Termination	Format
write	:MEASure:YXY	LF	ASCII
Read	100.01,0.312,0.329,0,0, 50.01,0.33332,0.33315,0,0	LF	ASCII

6.3.4 :MEASure:YUV

Measures Yuv of all selected angles.

Action	Data	Termination	Format
write	:MEASure:YUV	LF	ASCII
Read	100.01,0.257,0.378,0,0, 50.01,0.198,0.358,0,0	LF	ASCII

6.3.5 :MEASure:CCT

Measures CCT of all selected angles. The example shows 2 angles.

Action	Data	Termination	Format
write	:MEASure:CCT	LF	ASCII
read	2800.265,0,0, 3150.123,0,0	LF	ASCII

6.3.6 :MEASure:DWL

Measures dominant wavelenght of all selected angles. The example shows 2 angles.

Action	Data	Termination	Format
write	:MEASure:DWL	LF	ASCII
read	550.3,0.88,100.2561,0.456,0.4533,0,0, 420.3,0.95,50.156,0.1512,0.045,0,0	LF	ASCII

6.3.7 :MEASure:Y

Measures the luminance of all selected angles. The example shows 2 angles.

Action	Data	Termination	Format
write	:MEASure:Y	LF	ASCII
read	100.2561,0,0, 50.156,0,0	LF	ASCII

6.3.8 :MEASure:FREQuency

Measures the detected frequency of the DUT plus the amplitude of this frequency

Action	Data	Termination	Format
write	:MEASure:FREQuency	LF	ASCII
read	59.578,288.258,0	LF	ASCII

6.3.9 :MEASure:FLICKer

Measures MIN/MAX, RMS, JEITA, VESA, percentage or index flicker. The example measures JEITA flicker.

Action	Data	Termination	Format
write	:MEASure:FLICKer 2,2000,0	LF	ASCII
read	-39.45	LF	ASCII

The value of JIETA is in dB.

6.3.10 :MEASure:MVFLICKer

Measures MIN/MAX, RMS, JEITA, VESA, percentage or index flicker. The example measures JEITA flicker.

Action	Data	Termination	Format
write	:MEASure:MVFLICKer 2000,0	LF	ASCII
read	5.234,4.322,-39.45,- 36.45,3.234,0.555	LF	ASCII

6.3.11 :SAMPLE:Y

Measures in sample mode using a fixed integration time of 500us.

Action	Data	Termination	Format
write	:SAMPLE:Y 2000,0	LF	ASCII
read	0.0005,0.0000,100.234,99.234 etc		Binary (4-bytes float array)

6.3.12 :SAMPLE:XYZ

Measures in sample mode using either autorange or a fixed integration time.

Action	Data	Termination	Format
write	:SAMPLE:XYZ 200,0	LF	ASCII
Read	0.001,0.000,100.34,99.234,100.54 etc….		Binary (4-bytes float array)

6.3.13 :MEASure:FLINdex

Measures the flicker index value based on the number of samples and delay. Samples and delay are the same parameters as used for the :SAMPLE:Y function.

Action	Data	Termination	Format
write	:MEASure:FLINdex 2000,0,30,0	LF	ASCII
Read	13.1234,0	LF	ASCII

6.3.14 :READ:FLINDATA

When the last parameter of :MEASure:FLINdex is 1, the raw signal will be stored. This signal can be read back from the device, just like the :SAMPLE:Y command does.

Action	Data	Termination	Format
write	:READ:FLINDATA 2000	LF	ASCII
Read	100.88,100.99,99.78…..		Binary (4-bytes float array)

The command does not return dt, like the :sample:Y command. The timing of this function is however the same.

6.3.15 :MEASure:FFLINdex

Measures the fusion flicker index value based on the number of samples and delay.

Action	Data	Termination	Format
write	:MEASure:FFLINdex 2000,0,30,10,1	LF	ASCII
Read	13.1234,0	LF	ASCII

6.3.16 :READ:FFLINDATA

When the last parameter of :MEASure:FFLINdex is 1, the raw signal will be stored. This signal can be read back from the device, just like the :SAMPLE:Y command does.

Action	Data	Termination	Format
write	:READ:FLINDATA 2000	LF	ASCII
Read	100.88,100.99,99.78…..		Binary (4-bytes float array)

The command does not return dt, like the :sample:Y command. The timing of this function is however the same.

6.3.17 :MEASure:XYZFLINdex

Measures the flicker index value based on the number of samples and delay.

Action	Data	Termination	Format
write	:MEASure:XYZFLINdex 2000,0,30,0	LF	ASCII
Read	13.1234,0,50.123,55.123,60.4322	LF	ASCII

6.3.18 :READ:XYZFLINDATA

When the last parameter of :MEASure:XYZFLINdex is 1, the raw signal will be stored. This signal can be read back from the device, just like the :SAMPLE:XYZ command does.

Action	Data	Termination	Format
write	:READ:XYZFLINDATA 2000	LF	ASCII
Read	100.88,100.99,99.78…..		Binary (4-bytes float array)

The command does not return dt, like the :sample:XYZ command. The timing of this function is however the same.

6.3.19 :MEASure:XYZFFLINdex

Measures the fusion flicker index value based on the number of samples and delay.

Action	Data	Termination	Format
write	:MEASure:FFLINdex 2000,0,30,10,1	LF	ASCII
Read	13.1234,0	LF	ASCII

6.3.20 :READ:XYZFFLINDATA

When the last parameter of :MEASure:XYZFFLINdex is 1, the raw signal will be stored. This signal can be read back from the device, just like the :SAMPLE:XYZ command does.

Action	Data	Termination	Format
write	:READ:XYZFFLINDATA 2000	LF	ASCII
Read	100.88,100.99,99.78…..		Binary (4-bytes float array)

The command does not return dt, like the :sample:Y command. The timing of this function is however the same.

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 copies the values from RAM to EEPROM. All the other commands write to RAM only.

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 or reboot (:BOOT:REBOOT) the device.

The angle numbering in the configure commands is 0 to 3, where angle 0 is the perpendicular angle.

6.4.1 :EEPROM:STARTUP:WRITE

This command stores all the values from RAM to 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 the values from EEPROM back to RAM.

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:TRIG

Sets the trigger mode to on (1) or off (0)

Action	Data	Termination	Format
Write	:EEPROM:CONFigure:TRIG 0	LF	ASCII
Read	Not applicable	LF	ASCII

6.4.6 :EEPROM:CONFigure:TRIG?

Reads the trigger mode.

Action	Data	Termination	Format
Write	:EEPROM:CONFigure:TRIG?	LF	ASCII
Read	0	LF	ASCII

6.4.7 :EEPROM:CONFigure:TRIGDELAY

Sets the trigger delay in microseconds. HyperionQ measurement will start after this delay.

Action	Data	Termination	Format
Write	:EEPROM:CONFigure:TRIGDELAY 166666	LF	ASCII
Read	Not applicable	LF	ASCII

6.4.8 :EEPROM:CONFigure:TRIGDELAY?

Reads the trigger delay value

Action	Data	Termination	Format
write	:EEPROM:CONFigure:TRIGDELAY?	LF	ASCII
read	166666	LF	ASCII

6.4.9 :EEPROM:CONFigure:LPFREQ

Sets the low pass frequency. This is used for MIN/MAX and RMS flicker method only.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:LPFREQ 40	LF	ASCII
read	Not applicable	LF	ASCII

6.4.10 :EEPROM:CONFigure:LPFREQ?

Reads the low pass frequency.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:LPFREQ?	LF	ASCII
read	40	LF	ASCII

6.4.11 :EEPROM:CONFigure:AVG

Sets the initial averaging for all angles.

Action	Data	Termination	Format
Write	:EEPROM:CONFigure:AVG 1	LF	ASCII
Read	Not applicable	LF	ASCII

6.4.12 :EEPROM:CONFigure:AVG?

Reads the initial averaging value for all channels.

Action	Data	Termination	Format
Write	:EEPROM:CONFigure:AVG?	LF	ASCII
Read	1	LF	ASCII

6.4.13 :EEPROM:CONFigure:WHITE

Sets the white reference to be used for dominant wavelength calculation.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:WHITE D65	LF	ASCII
read	Not applicable	LF	ASCII

6.4.14 :EEPROM:CONFigure:WHITE?

Reads the white reference point that is used for dominant wavelength calculation.

Action	Data	Termination	Format
Write	:EEPROM:CONFigure:WHITE?	LF	ASCII
Read	D65	LF	ASCII

6.4.15 :EEPROM:CONFigure:AUTO:FREQ

Sets the DUT frequency to be used by all angles.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:AUTO:FREQ 59.789	LF	ASCII
read	Not applicable	LF	ASCII

6.4.16 :EEPROM:CONFigure:AUTO:FREQ?

Reads the DUT frequency.

Action	Data	Termination	Format
Write	:EEPROM:CONFigure:AUTO:FREQ?	LF	ASCII
Read	59.789	LF	ASCII

6.4.17 :EEPROM:CONFigure:FRAMES

Sets the number of frames to be measured. The default is 1. Only in case a display is not stable, this value can be set higher.

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

6.4.18 :EEPROM:CONFigure:FRAMES?

Reads the number of frames to be measured.

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

6.4.19 :EEPROM:CONFigure:AUTOrange

Sets the autorange per angle on (1) or off (0)

Action	Data	Termination	Format
write	:EEPROM:CONFigure:AUTOrange 1,2	LF	ASCII
read	Not applicable	LF	ASCII

The 2nd parameter is the angle.

6.4.20 :EEPROM:CONFigure:AUTOrange?

Reads whether an angle is using auto-range.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:AUTOrange? 2	LF	ASCII
read	1	LF	ASCII

6.4.21 :EEPROM:CONFigure:AUTOrange?

Reads whether an angle is using auto-range.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:AUTOrange? 2	LF	ASCII
read	1	LF	ASCII

6.4.22 :EEPROM:CONFigure:AUTO:ADJMIN

Sets the ADJMIN value per angle.

Action	Data	Termination	Format
Write	:EEPROM:CONFigure:AUTO:ADJMIN 15,1	LF	ASCII
Read	Not applicable	LF	ASCII

6.4.23 :EEPROM:CONFigure:AUTO:ADJMIN?

Reads the ADJMIN value per angle.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:AUTO:ADJMIN? 1	LF	ASCII
read	15	LF	ASCII

6.4.24 :EEPROM:CONFigure:AUTO:MAXINT

Sets the maximum integration time per angle.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:AUTO:MAXINT 200000,1	LF	ASCII
read	Not applicable	LF	ASCII

6.4.25 :EEPROM:CONFigure:AUTO:MAXINT?

Sets the maximum integration time per angle.

Action	Data	Termination	Format
Write	:EEPROM:CONFigure:AUTO:MAXINT? 1	LF	ASCII
Read	200000	LF	ASCII

6.4.26 :EEPROM:CONFigure:AUTO:MODE

Sets the autorange mode per angle.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:AUTO:MODE 1,2	LF	ASCII
read	Not applicable	LF	ASCII

Mode	Meaning	Adjmin	Maxint
0	Fast	20	50000
1	Normal	20	250000
2	Slow	20	600000
3	Hi-sense	20	1000000
4	Manual	manual	manual

6.4.27 :EEPROM:CONFigure:AUTO:MODE?

Reads the autorange mode per angle.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:AUTO:MODE? 2	LF	ASCII
read	1	LF	ASCII

6.4.28 :EEPROM:CONFigure:INT

Sets the initial integration time per angle when not using auto-range.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:INT 16666,2	LF	ASCII
read	Not applicable	LF	ASCII

6.4.29 :EEPROM:CONFigure:INT?

Reads the initial integration time per angle.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:INT? 2	LF	ASCII
read	16666	LF	ASCII

6.4.30 :EEPROM:CONFigure:GAIN

Sets the initial gain per angle when not using auto-range.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:GAIN 3,2	LF	ASCII
read	Not applicable	LF	ASCII

6.4.31 :EEPROM:CONFigure:GAIN?

Reads the initial gain per angle when not using auto-range.

Action	Data	Termination	Format
Write	:EEPROM:CONFigure:GAIN 2	LF	ASCII
Read	3	LF	ASCII

6.4.32 :EEPROM:CONFigure:SBW

Sets the initial calibration matrix per angle.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:SBW 0,1	LF	ASCII
read	Not applicable	LF	ASCII

Note: user1=0, user2=1 etc…

6.4.33 :EEPROM:CONFigure:SBW?

Sets the initial calibration matrix per angle.

Action	Data	Termination	Format
Write	:EEPROM:CONFigure:SBW? 1	LF	ASCII
Read	0	LF	ASCII

6.4.34 :EEPROM:READ:SN

Reads the instrument serial number.

Action	Data	Termination	Format
write	:EEPROM:CONFigure:SBW 0,1	LF	ASCII
read	Not applicable	LF	ASCII

6.4.35 :EEPROM:WRITE:ID

Writes a device ID to the instrument. This ID can be used by applications to identify the device without using the serial number.

Action	Data	Termination	Format
write	:EEPROM:WRITE:ID 1	LF	ASCII
read	Not applicable	LF	ASCII

6.4.36 :EEPROM:READ:ID

Reads the device ID from the instrument.

Action	Data	Termination	Format
write	:EEPROM:READ:ID	LF	ASCII
read	1	LF	ASCII

6.5 User EEPROM calibration commands

With the user EEPROM calibration commands the user can program calibration matrices. Each angle can contain up to 30 matrices.

6.5.1 :EEPROM:READ:USERCAL

This command all calibration data from EEPROM to RAM.

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

6.5.2 :EEPROM:WRITE:USERCAL

This command stores the written values of the 3x3 matrices 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			

6.5.3 :EEPROM:SBW: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: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 five parameters.

After each write wait 5ms.

Command	Write Parameter
:EEPROM:SBW:WRITE	matrixID, row, col, value, angle

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

To write the matrix name, use index 11:

Action	Data	Wait	Termination	Format
write	:EEPROM:SBW:WRITE 0,11,0,myMatrix,0	5ms	LF	ASCII

To write the matrix calibration date, use index 12:

Action	Data	Wait	Termination	Format
write	:EEPROM:SBW:WRITE 0,12,0,20200706,0	5ms	LF	ASCII

Note : For names and dates, only use pure ASCII code without spaces, commas or any other special characters except underscores if needed.

6.5.4 :EEPROM:SBW:READ

This command reads the values of a 3x3 matrix from the device

This command has four parameters:

Command	Write Parameter
:EEPROM:SBW:READ	matrixID, row, col, angle

Returned value is the correction factor.

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

Action	Data	Termination	Format
read	0.021	LF	ASCII
write	:EEPROM:SBW:READ 0,1,2,0	LF	ASCII
read	-0.016	LF	ASCII
write	:EEPROM:SBW:READ 0,2,2,0	LF	ASCII
read	1.102	LF	ASCII

To read the matrix name, use index 11:

Action	Data	Termination	Format
write	:EEPROM:SBW:READ 0,11,0,0	LF	ASCII
read	myMatrix	LF	ASCII

To read the matrix calibration date, use index 12:

Action	Data	Wait	Termination	Format
write	:EEPROM:SBW:READ 0,12,0,0	5ms	LF	ASCII
read	20200706	LF	ASCII	

6.5.5 :EEPROM:READ:SBWNAME

This command reads the matrix name. It has the same functionality as :EEPROM:SBW:READ with index 11.

Action	Data	Wait	Termination	Format
write	:EEPROM:WRITE:USERCAL	20ms	LF	ASCII
read	myMatrix			

7 3X3 MATRIX – HOW TO CALCULATE

Below the formula is given 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

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

8 CODING EXAMPLES

8.1 Configuration and measurement examples

First example selects and deselects angles to perform measurements of the selected angles.

Command	Result	Remark
:ANGLE:SELECT 255	---	Select all angles
:SENSe:AUTORANGE 1	---	set autorange on
:MEAS:XYZ	XYZ data	Measure XYZ of all angles.
:ANGLE:DESELECT 4		Make angle 4 inactive.
:ANGLE:DESELECT 2		Make angle 2 inactive.
:ANGLE:SELECTED?	1,0,1,0	Show which angles are active.
:MEAS:XYZ	XYZ data	Measure XYZ of angle 1 and angle 3.

8.2 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 AUTO-RANGE FUNCTION

9.1 Introduction

The HyperionQ uses a gain and integration time setting for the sensor to perform the measurement. To manually set these values can be quite complex because the optimal settings depend on the luminance of the measured DUT.

In this case, the HyperionQ 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 30. 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.

9.2 How autoranging works

The auto ranging works, by first setting a default integration time (1/FREQ). 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 HyperionQ work.

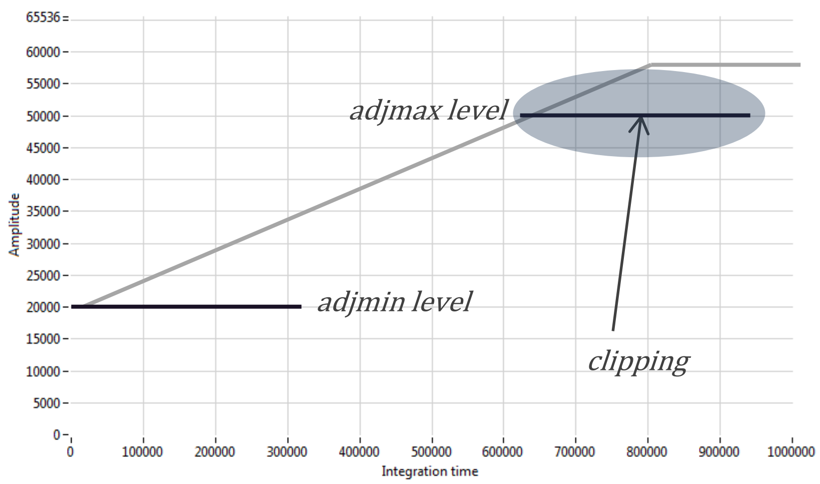

Fig 5 Autorange levels

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.

9.3 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 1-30, which means 1-30% 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.

9.4 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 but it doesn’t hurt when setting a timeout value longer. The timeout setting does not affect the tact time.

9.5 HyperionQ auto-range good practice settings

- Frequency: Frequency of your light source
- Display (DUT) 60Hz
- Adjmin: 10%
- Maximum integration time: 300000 us (0.3sec)
- 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.

ADMESY

colorimeters | spectroradiometers | lightmeters

Sleestraat 3
6014 CA Ittervoort
The Netherlands

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

admesy.com

Version 1.0.3 October 2020