Technical Manual

SDCM3

Firmware Description

Revision 1.0

INSION GmbH
Weipertstraße 8 - 10
74076 Heilbronn
Germany

Phone +49 71 31 97 36 06 - 0
Fax: +49 71 31 97 36 06 - 99
E-Mail: info@insion.de
Web: www.insion.de

Table of contents

1. General Remarks

1.1. Introduction

The direct communication with SDCM3 will be realized using special commands via USB interface. This interface is designed as a virtual COM port, so it can be handled similarly to a serial port with the settings 8n1/ no protocol. The allowed transfer rates are 38 400, 115 200, 230 400, 921 600 and 3 000 000 baud.

The command syntax of SDCM3 is SCPI compatible. Therefore, the commands consist of a main category and a more specific part, both connected by a colon. Some very special commands contain only one part. This document lists the firmware commands summarized in logical categories.

1.2. Command Categories

The following list shows the available command categories. They begin with "*" and include the following keywords:

Command	**Description**
*PARAmeter	Get and set general parameters
*MEASure	Configure, start and receive response of a measurement
*FETCH	Get data from the previous measurement
*CONTRol	Control of peripherical components
*MMEMory	Handling of data storage
*SC30	SC30 calculation
*STATus	Information about error and enquiry requests

These keywords are followed by an additional word, separated by colon, and by arguments. It is only necessary to use the indicated capital letters, other letters can be omitted.

1.3. Commands with Arguments

Many commands can be extended by arguments. The meanings of the arguments are as follows:

Argument	**Meaning**
tint	Integration time in ms, range from 0.01 … 65 000 ms
av	Average counts for measurement (1 … 10 000)
arg	Other argument, described in text

If parameter commands with get and set options are used with "?", the stored values will be displayed. If an argument is used, this argument will be set. This can be done permanently by using the *PARA:SAVE command or temporarily, if this command will not be given.

Several commands can be written successively in one line separated by semicolons. A space sign between command and argument is necessary. In case a command which accepts arguments is used without arguments, then the pre-configured arguments will be used. If a command was successfully executed it will be acknowledged (ACK, 06 hex), otherwise negative acknowledgement (NAK, 15 hex) will be returned. The reason of an error can be read by the command:
*STATUS:ERR<CR>

1.4. Abbreviations

The description of the abbreviations above and in the following paragraphs can be found in the next table:

Sign	**Description**	**Hex code**
<CR>	Carriage return	0x0D
<SP>	Space	0x20
<ACK>	Acknowledgement	0x06
<NAK>	Negative acknowledgement	0x15
<ETX>	End of text	0x03
<TAB>	Tabulator	0x09
<BEL>	Bell	0x07

2. Request and Setting of Parameters

These commands are used to read/write the basic settings of the instrument. The values can be checked using the appropriate command, followed by a question mark and a <CR>. They can be changed by specifying arguments.

Remark:	Keep in mind that changed parameters can cause erroneous measurement results.

2.1. General Parameter Commands

The following commands can be used to identify an instrument after checking all virtual COM ports.

Command	**Comment**	**Valid range**	**Response**
*IDN?<CR>	Get device ID and spectrometer number	up to 63 characters	JETI_SDCM3 1500012<CR>
*VERSion? <CR>	Get firmware version	up to 63 characters	SDCM3_INSION VERSION 1.0.0 150415<CR>
*RST<CR>	Device software reset		Performing software reset …<CR>
*PARAmeter:SPNUMber<CR>	Get/ set spectrometer serial number	up to 15 characters (0 ... 9, a ... z)	1500012<CR>
*PARAmeter:SERNumber<CR>	Get/ set electronic serial number	up to 15 characters (0 … 9, a … z)	9999<CR>
*BOOT<CR>	Jump to boot loader		

2.2. Settings Concerning the Detector Array and the AD Converter

Different sensor types can be set as parameter according to the hardware version (see hardware description). The following sensor types are supported:

Sensor Number	**Valid Pixel Count**	**Sensor Type**
8	256	S837x
21	1024	S9226
100	Up to 2048	S11639
121	256	G9203

Command	Comment	Valid Value Range	Response
*PARAmeter:SENSor<SP> Sensor Number<SP> Pixel Count<CR>	Get/ set imaging sensor type and pixel count	See table above	8<SP>256<SP>(S11639)<CR>
*PARAmeter:PIXEL?<CR>	Get effective pixel count (depending on pixel binning – see *PARA:PIXBIN)		256<CR>
*PARAmeter:PIXBINning <CR>	Get/ set the pixel to bins	1 ... 16	3<CR>
*PARAmeter:DIRECTion arg<CR>	Get/ set sensor scan direction	0 - lowest pixel first 1 - highest pixel first	0<CR>
*PARAmeter:FASTscan arg<CR>	Get/ set time to next fastscan cycle in ms (integer value, 0 – fastscan switched off)	0 ... 350	50<SP>ms<CR>
*PARAmeter:PRESCan<CR>	Get/set pre-scan count (integer value, 0 – pre-scan switched off)	0 … 8	0<CR>
*PARAmeter:PDAGain<CR>	Set line array gain	0 - low 1 - high	0<SP>(low)<CR>
*PARAmeter:ADCResolution arg<CR>	Get/ set ADC resolution (in bit)	8 ... 16	16<CR>
*PARAmeter:ADCVoltage arg <CR>	Get/ set ADC full scale value (in V)	2 or 4	2<SP>V<CR>
*PARAmeter:OFFSet arg<CR>	Get/ set offset value (in mV, integer value)	-300 ... +300	50<SP>mV<CR>
*PARAmeter:GAIN arg<CR>	Get/ set gain value	1.0 ... 5.0	1.7<CR>
*PARAmeter:OVSAMPling arg <CR>	Get/ set oversampling of ADC	1 ... 32	7<CR>

2.3. Settings Concerning the System and the Power Consumption

Command	Comment	Valid Value Range	Response

*PARAmeter:BAUD arg <CR>	Get/ set baudrate	38400, 115200, 230400, 921600, 3000000	3000000<CR>
*SLEEP<CR>	Enter power save mode		<ACK>

To put the device into power save mode one can call the command *SLEEP<CR>. The device will send an <ACK> byte and go to sleep mode. Due to technical restrictions one has to wake up the device by sending a single byte (e.g. <CR>) before any other command will be recognized. The device will leave sleep mode, returns an <NAK> and set the error code to 300 (ERR_WAS_SLEEPING) to signalize, that it is awake.

2.4. Settings for Peripheral Components

It is possible to connect several peripheral components to the SDCM3 board. The following commands are used to control these components.

2.4.1. Lamps/ Shutter

Pin 7 of the SDCM3 board can be used as LV-TTL lamp or shutter control output. This signal is programmable in active polarity (0/1). If the parameter lamp enable is set to enabled, it will be set active before the scan of the line array for the time determined by scan delay parameter and during the light scan or inactive during the dark scan. Additionally the signal can be controlled by control lamp command. If lamp enable is set to disabled, the signal will be left inactive and switching with control command will return an error.

Command	Comment	Valid Value Range	Response
*PARAmeter:LAMPEnable arg<CR>	Get/ set enable state of external lamp/ shutter control output	0 - disabled 1 - enabled	1<SP>(enabled) <CR>
*PARAmeter:LAMPPolarity arg<CR>	Get/ set polarity of external lamp/shutter control	0 - low 1 - high	1<SP>(high)<CR>
*PARAmeter:SDELay arg<CR>	Get/ set scan delay time in ms (integer value)	0 ... 60000	100<SP>ms<CR>
*CONTRol:LAMP<CR>	Get/ set the lamp status	0 - lamp off 1 - lamp on	1<SP>(lamp is on)<CR>

2.4.2. Auxiliary Out

Pin 8 can be used as digital output 0 (output zero) and can be controlled with control out0 (out zero) command.

Command	Comment	Valid Value Range	Response
*CONTRol:OUT0 arg<CR>	Get/ set the digital out0	0 - low 1 - high	

2.5. Pixel-Wavelength Relation

The correlation between the pixel numbers of the detector array and the corresponding wavelength is done with a fit polynomial up to the 4th order.

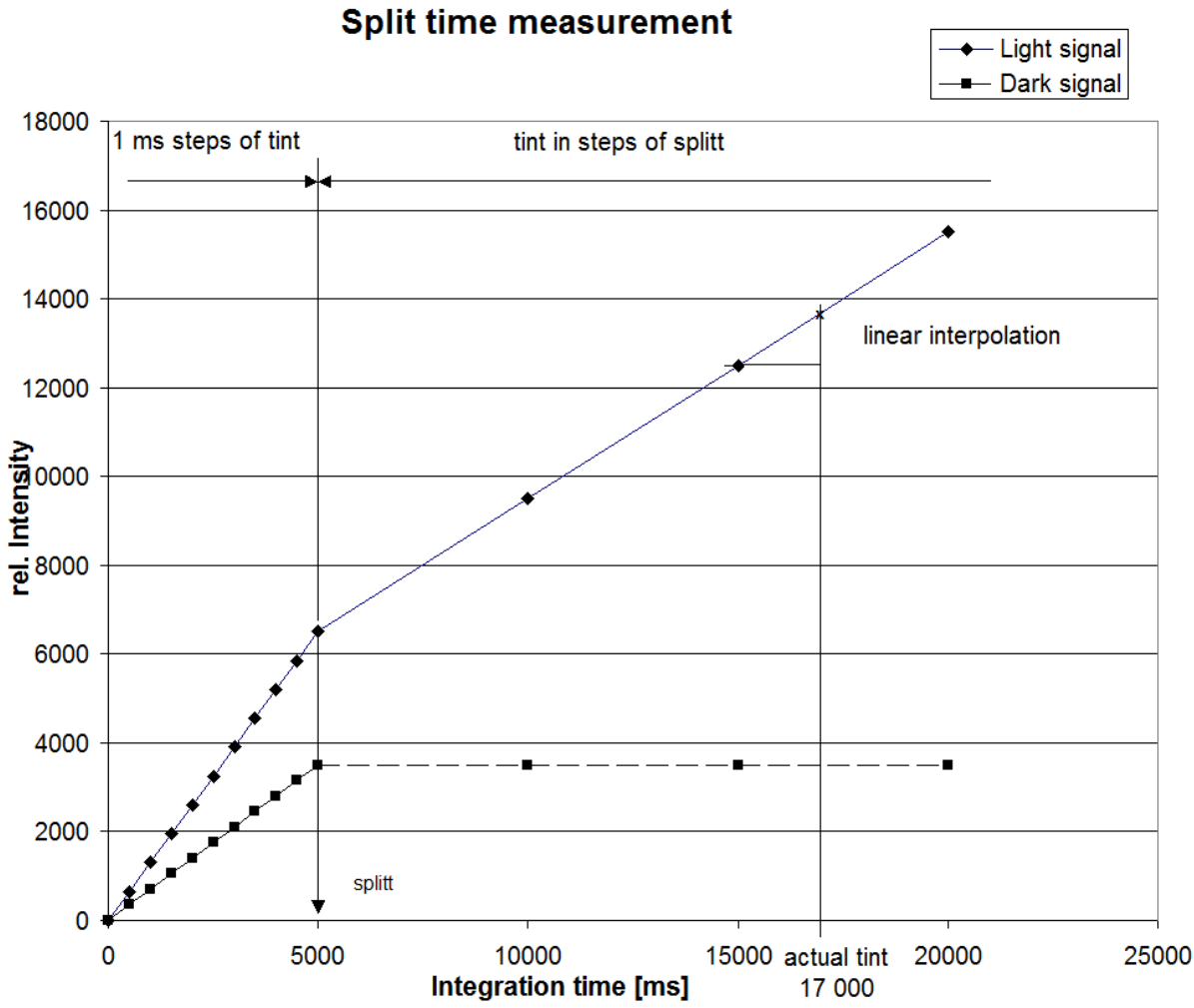
Command	Comment	Valid Value Range	Response
*PARAmeter:FIT0 arg<CR>	Get/ set polynomial coefficient fit0 for wavelength-pixel	any valid float number	3.800000e+02<CR>
*PARAmeter:FIT1 arg<CR>	Get/ set polynomial coefficient fit1 for wavelength-pixel	any valid float number	4.075535e-01<CR>
*PARAmeter:FIT2 arg<CR>	Get/ set polynomial coefficient fit2 for wavelength-pixel	any valid float number	5.642718e-05<CR>
*PARAmeter:FIT3 arg<CR>	Get/ set polynomial coefficient fit3 for wavelength-pixel	any valid float number	-1.261602e-08<CR>
*PARAmeter:FIT4 arg<CR>	Get/ set polynomial coefficient fit4 for wavelength-pixel	any valid float number	-2.181461e-14<CR>

2.6. Time Settings

Command	Comment	Valid Value Range	Response
*PARAmeter:TINT tint<CR>	Get/ set default integration time (in ms)	any float from 0.01 up to 65000.0	10.000<SP>ms<CR>
*PARAmeter:SPLITTime arg<CR>	Get/set splitting interval for the integration time (integer value)	400 … 6000 (0 – no split)	1000<SP>ms<CR>

Explanation of Split Time Concept

Some low light applications demand a long integration time of the detector. However it is not the case for every detector to be compatible with SDCM3. Therefore the split of the integration time was introduced. Above the preset split time (splitt) the measurement scan is divided into integer parts of the split time. For example, 17 s will be split into three measurements of 5 s in case of splitt = 5000 ms (see diagram below). The dark signal is calculated by averaging the individual split scans. Therefore the dark signal does not rise any more if *tint* > *splitt*. The light signal is calculated by a summation of the different scans. Furthermore there is done a linear interpolation to get the data for integration times between multiples of the split time. The following diagram shows the scheme of the split time measurement.



2.7. Measurement Pre-configuration

Command	Comment	Valid Value Range	Response
*PARAmeter:FORMat format<CR>	Get/ set predefined output format	0, 1, 3-7 (see Format)	2<CR>
*PARAmeter:FUNCtion function<CR>	Get/ set predefined measurement function	1 - light measurement 2 - dark measurement 3 - reference measurement	1<CR>

2.8. Additional Settings

2.8.1. Correction and Smoothing

Command	Comment	Valid Value Range	Response
*PARAmeter:PIXBINning arg<CR>	Get/ set pixel binning	1 … 10	3<CR>

2.8.2. Request of Parameters and Handling of User Data

It is possible to read and store the parameter block with one command. Furthermore it is possible to use a user data block.

***PARAmeter:ALLPARA?<CR>** Get a list of all parameters. An example for the response is provided below.

*PARAmeter:BAUDrate 3000000<CR>
*PARAmeter:TINT 10.0 ms
*PARAmeter:FORMat 1
*PARAmeter:FUNCtion 2
*PARAmeter:FIT0 1.395770e+02
*PARAmeter:FIT1 4.075535e-01
*PARAmeter:FIT2 5.642718e-05
*PARAmeter:FIT3 -1.261602e-08
*PARAmeter:FIT4 -2.181461e-14
*PARAmeter:SERNumber 1500135
*PARAmeter:SPNUMber 99999
*PARAmeter:SDELay 20 ms
*PARAmeter:SPLITTime 1000 ms
*PARAmeter:SENSor 100 2048 (S11639)
*PARAmeter:PDAGain 0 (low)
*PARAmeter:OVSAMPling 16
*PARAmeter:PIXBINning 1
*PARAmeter:OFFSet -180 mV
*PARAmeter:GAIN 2.1
*PARAmeter:ADCResolution 16
*PARAmeter:ADCVoltage 4 V
*PARAmeter:TEMPCorr 0.00 K
*PARAmeter:SPIENable 1 (enabled)
*PARAmeter:SPIMODe 0
*PARAmeter:SPIMASTer 1 (Master)
*PARAmeter:SPIBRate 10000000 Hz
*PARAmeter:FASTscan 0 ms
*PARAmeter:LAMPEnable 1 (enabled)
*PARAmeter:LAMPPolarity 1 (high)
*PARAmeter:TRIGger 0
*PARAmeter:TRSLope 0
*PARAmeter:LDCURRent 40 mA
*PARAmeter:LDLIMIT 80 mA
*PARAmeter:SCALTI 0
*PARAmeter:DIRECTion 0

All parameter settings are stored within a special flash memory area of the SDCM3 board, so all settings will be kept even after powering off the device. It is possible to backup all parameter settings from this area to another special protected flash memory area. If one has changed and saved illegal parameter settings accidently one can restore it from the protected area. To backup or restore a password must be given as an argument to the command (e.g. *para:backup password<CR>).

Command	Comment	Valid Value	Response
*PARAmeter:BACKUP<CR>	backup parameter, password protected		<ACK> or <NAK>
*PARAmeter:RESTORE<CR>	restore parameter, password protected		<ACK> or <NAK>

2.8.3. Trigger Commands

Command	Comment	Valid Value Range	Response
*PARAmeter:TRIGger<CR>	Get/ set trigger mode for external trigger	0 - disabled 1 - enabled (measure mode) 2 - enabled (enquiry mode)	1<SP>(measure mode)<CR>
*PARAmeter:TRSLope<CR>	Get/ set trigger slope for external	0 - rising edge 1 - falling edge	0<SP>(rising edge)<CR>

There are two possibilities to trigger measurement scans of SDCM3 by external signals:

- Normal trigger by a TTL signal
- Release an enquiry by a TTL signal

Voltage requirements for the TTL signal:

- Low 0 … 0.4 V
- High 2.4 … 5.5 V

The trigger slope (falling or rising edge) can be set by *PARA:TRSL command.

Different Trigger Modes

Normal Trigger (*PARA:TRIG 1)

The measurement scan will be executed directly after the trigger signal. The delay time can be adjusted by the scan delay (*PARA:SDEL). Use pin 12 (GND) and pin 2 (TRIG) of the I/O connector to connect the trigger source to the SDCM3 electronics. The type of measurement has to be defined before using the following commands:

*PARA:FUNC

*PARA:TINT

After the measurement scan is finished the firmware will send an enquiry byte <ENQ> to signalize that a triggered measurement was performed.

Use command *STAT:ENQU? to determine which measurement was triggered:

Status code 20: dark measurement

Status code 21: light measurement

Status code 22: reference measurement

One can fetch the measured spectra with the corresponding *FETCH command.

Enquiry Trigger (*PARA:TRIG 2)

It is also possible to trigger only an enquiry (0x5H), which can be used by the software for other actions. An enquiry can be detected e.g. by using a COM callback function. The status of the

enquiry can be obtained by the following command:

*STAT:ENQU?

The enquiry code for trigger event is 1.

2.9. Permanent Saving of Parameters

After any change of one or more parameters this change has to be saved to make it permanent using the following command:

Command	Comment	Valid Value Range	Response
*PARAmeter:SAVE<CR>	Save parameters to internal flash storage		<ACK>

3. Measurements

All commands of the *MEAS category provide a measurement scan and the following output of data. Alternatively it is possible to proceed a measurement without output and to output the data later try a fetch command.

3.1. Raw Data Measurements

Normally each light measurement will be combined with an individual dark measurement. This measure will suppress base line drift effects.

Command	Comment	Arguments	Response
*MEASure?<CR>	Get a help list of the measurement commands		
*MEASure:DARKspectra tint av format<CR>	Perform a dark scan	integration time, average, format (if left one or more argument standard values set with *param will be used, average will be 1)	(see Format for further information)
*MEASure:LIGHT tint av format<CR>	Perform a light scan	integration time, average, format (if left one or more argument standard values set with *param will be used, average will be 1)	(see Format for further information)
*MEASure:REFERence tint av format<CR>	Perform a reference scan	integration time, average, format (if left one or more argument standard values set with *param will be used, average will be 1)	(see Format for further information)
A reference measurement performs a light measurement and subtracts a previously measured dark spectra. Therefore it is only possible if a dark measurement with the same integration time was performed before.			
*MEASure:TEMPE<CR>	Perform a temperature measuremen	output of temperature	25.48<SP>°C<CR>

- after a measurement was started (e.g. with *meas:dark 100<CR>) the device will return <ACK>
- when the measurement was completed the device will return <BEL>
- after this the device will return the measurement values in the selected format as described in paragraph Abbreviations

Format = 0: **no output**

Format = 1: **L/H binary output without length**

All data transmitted as 16 bit word, low-byte first (Little Endian)

Byte	Value	Definition
0	xx yy	first Pixel
2	xx yy	second Pixel
⋮	⋮	⋮
⋮	⋮	⋮
2·n-1	xx yy	last Pixel

n: number of pixel per line

Format = 3: **L/H binary output with length**

All data transmitted as 16 bit word, low-byte first (Little Endian)

Byte	Value	Definition
0	xx yy	length
2	xx yy	first Pixel
4	xx yy	second Pixel
⋮	⋮	⋮
⋮	⋮	⋮
2 + 2·n-1	xx yy	last Pixel

n: number of pixel per line

Format = 4: **ASCII output, separated by <CR>, closed by <ETX>**

819<CR>
858<CR>
807<CR>
...<CR>
<ETX>

Format = 5: **H/L binary output without length**

All data transmitted as 16 bit word, high-byte first (Big Endian)

Byte	Value	Definition
0	xx yy	first Pixel
2	xx yy	second Pixel
⋮	⋮	⋮
⋮	⋮	⋮
2·n-1	xx yy	last Pixel

n: number of pixel per line

Format = 6: **H/L binary output with length**

All data transmitted as 16 bit word, high-byte first (Big Endian)

Byte	Value	Definition
0	xx yy	length
2	xx yy	first Pixel
4	xx yy	second Pixel
⋮	⋮	⋮
⋮	⋮	⋮
2 + 2·n-1	xx yy	last Pixel

n: number of pixel per line

Format = 7: **ASCII output with wavelength, separated by <CR>, closed by <ETX>**

250.1<TAB>4153<CR>
250.5<TAB>4118<CR>
250.9<TAB>4126<CR>
251.4<TAB>3690<CR>
.....<TAB>.....<CR>
<ETX>

4. Fetch of Measured Data

A *FETCH command can only be used if the appropriate measurement was proceeded before.

Command	Comment	Response
*FETCH?<CR>	Get a help list of the fetch commands	
*FETCH:DARK format<CR>	Read out of meas. dark spectrum (if no format argument the standard value set with *param:form will be used)	(see Format for further information)
*FETCH:LIGHT format<CR>	Read out of meas. light spectrum (if no format argument the standard value set with *param:form will be used)	(see Format for further information)
*FETCH:REFERence format<CR>	Read out of meas. reference spectrum (if no format argument the standard value set with *param:form will be used)	(see Format for further information)
*FETCH:LEVEL<CR>	Read out of maximum counts and percentage of exposure	47186<SP>72<CR>

5. Mass-Memory

Command	Comment
*MMEMory:CATalog?	Returns a list of all file names in user flash
*MMEMory:DATA	Writes user data to internal flash (up to 15 files with max. 4KByte per file)
*MMEMory:DATA?	Returns the associated file data in block format
*MMEMory:DELete	Delete a file
*MMEMory:COPY	Copy a file
*MMEMory:MOVE	Move/rename a file

*MMEMory:CATalog?<CR>

This query outputs a list of the files. The return data will be in the following form:

<filename1><CR>

<filename2><CR>

...<CR>

<ETX>

*MMEMory:DATA<SP>filename<CR>#ABC

This command writes <data> into filename, where <data> is in 488.2 block format. The SDCM3 expects to see user data as block data (binary files). The IEEE standard 488.2-1992 section 7.7.6 defines block data.

The following example shows how to structure a SCPI command for downloading user data where #ABC represents the block data.

filename	The filename can be any name up to 63 character. White-spaces are not allowed to be part of the filename!
#	This character indicates the beginning of the data block. A Number of decimal digits present in B (can be 1 - 4)
B	Decimal number specifying the number of data bytes to follow in C (up to 4096 bytes are allowed)
C	Actual binary user data

After sending *MMEM:DATA<SP>filename<CR>#AB there is 10 seconds left to send C, so one can prepare a data transfer via a terminal program and send any file (up to 4KByte size) to the device

*MMEMory:DATA?<SP>filename<CR>

This query returns the associated <data> in 488.2 block format.

*MMEMory:DELete<SP>filename<CR>

This command removes a file from the device.

*MMEMory:COPY<SP>sourcefile<SP>destinationfile<CR>

This command makes a duplicate of the requested <sourcefile> and copies its content to <destinationfile>. Source and destination must be different.

6. SC30 Calculation

6.1. SC30 Parameter Commands

Command	Comment	Valid Value Range	Response
*SC30:CORR arg<CR>	Get/ set SC30 correction factor	0.1 … 5.0	1.0<CR>
*SC30:POWer arg<CR>	Get / set SC30 power function	0 – off 1 – on	1 (on)<CR>
*SC30:SAMPling arg<CR>	Get/ set SC30 sampling width	2 or 4	4<CR>
*SC30:ITERation arg<CR>	Get/ set SC30 iteration count	Up to 20	10<CR>
*SC30:COLUMNwidth arg<CR>	Get/ set SC30 column width	Up to 20	5<CR>
*SC30:ROWwidth arg<CR>	Get / set SC30 row width	Up to 20	0<CR>
*SC30:STARTPIXel?<CR>	Get SC30 start pixel		0<CR>
*SC30:STARTWAVelen arg<CR>	Get / set SC30 start wavelength		253.52<CR>
*SC30:ENDWAVelen arg<CR>	Get / set SC30 end wavelength		872.12<CR>
*SC30:PIXel?<CR>	Get SC30 pixel count		220<CR>

6.2. SC30 Correction Data Commands

***SC30:DATA<CR>#ABC**

This command writes <data> into internal storage, where <data> is in 488.2 block format. The SDCM3 expects to see SC30 data as ASCII data table, where <TAB> separates columns and <CR> separates rows. The IEEE standard 488.2-1992 section 7.7.6 defines block data.

The following example shows how to structure a SCPI command for downloading SC30 data where #ABC represents the block data.

#	This character indicates the beginning of the data block. A Number of decimal digits present in B (can be 1 - 5)
B	Decimal number specifying the number of data bytes to follow in C (up to 16384 bytes are allowed)
C	Actual SC30 correction data set

After sending *SC30:DATA<CR> there is 10 seconds left to send the SC30 data set, so one can prepare a data transfer via a terminal program and send any calibration file (up to 16KByte size)

to the device.

NOTE: **If a new SC30 calibration data set was send to the device, all SC30 parameters are set to their default values:**

Power function:	1 (on)
Iterations:	10
Column width:	5
Row width:	0
Correction factor:	1.0
Start pixel:	0
Sampling width:	4
Pixel count:	sampling width * number of rows
Start Wavelength:	corresponding wavelength of first pixel
End wavelength:	corresponding wavelength of last pixel

***SC30:DATA? <CR>**

This query returns the internal stored SC30 data set in 488.2 block format.

6.3. SC30 Calculation Command

***SC30:CALC<SP>format<CR>**

This command performs an SC30 calculation with the last measured light or reference spectra and returns the corrected spectra. The original spectra will be unmodified. If the last measurement was a light measurement, the light values will be used for calculation and the reference values if the last measurement was a reference measurement. Possible values for format are 4, 7, 20 and 21. If no format argument is given, then format 7 will be used.

Format = 4: **ASCII output, separated by <CR>, closed by <ETX>**

819<CR>
858<CR>
807<CR>
...<CR>
<ETX>

Format = 7: **ASCII output with wavelength, separated by <CR>, closed by <ETX>**

250.1<TAB>4153<CR>
250.5<TAB>4118<CR>
250.9<TAB>4126<CR>
251.4<TAB>3690<CR>
.....<TAB>.....<CR>
<ETX>

Format = 20: **L/H binary output without length**

All data transmitted as 32 bit float, low-byte first (Little Endian)

Byte	Value	Definition
0	ww xx yy zz	first Pixel
4	ww xx yy zz	second Pixel
⋮	⋮	⋮
⋮	⋮	⋮
4·n-1	ww xx yy zz	last Pixel

n: number of pixel per line

Format = 21: **L/H binary output with length**

All data transmitted as 32 bit float, low-byte first (Little Endian)

Byte	Value	Definition
0	xx yy	length
2	ww xx yy zz	first Pixel
6	ww xx yy zz	second Pixel
⋮	⋮	⋮
⋮	⋮	⋮
2 + 4·n-1	ww xx yy zz	last Pixel

n: number of pixel per line

7. Help-, Status- and Error-Messages

Command	Comment	Response
*HELP?	Help text for all commands	List of available
*PARAmeter?	Help text for parameter commands	List of available parameter
*PARAmeter:HELP?	Help text for parameter commands	

The status commands are used to get the information of the error and configuration conditions.

Command	Comment	Response
*STATus:ERRor?	Get the error code	0<CR>
*STATus:TXTERRor?	Get error code and description of the error	0<SP>no error<CR>
*STATus:ENQUiry?	Get enquiry status code	20<CR>

7.1. Meaning of Error Codes

Error Code	Error String	Description
0	No error	No error occurred
4	Unknown command	The firmware command could not be recognized or doesn't exist
7	Wrong password	The given password was wrong
10	Invalid argument 1	The first argument of a command was invalid
11	Invalid argument 2	The second argument of a command was invalid
12	Invalid argument 3	The third argument of a command was invalid
13	Invalid argument 4	The fourth argument of a command was invalid
15	Missing argument	An argument was needed but missing
16	No dark measurement	To fetch dark values or perform reference measurement a dark measurement is obligatory
17	No light measurement	To fetch light values a light measurement is obligatory
18	No reference measurement	To fetch reference values a reference measurement is obligatory

19	Unable to perform split measurement	To use split-time-concept a lamp or shutter must be enabled (see 2.6 Time
30	No backup available	No parameter backup is available
50	Lamp is disabled	To switch lamp/shutter it must be enabled (see 2.4.1 Lamps/Shutter)
226	No memory left	No more memory free for user data (see 5. Mass-Memory)
227	File doesn't exist	The selected user file doesn't exist (see 5. Mass-Memory)
228	Wrong file size	The selected file size is wrong (see 5. Mass-Memory and 6.2 SC30 Correction Data)
229	Source and destination are identical	It is not possible to copy ore move to an existing user file (see 5. Mass-
300	Device was in sleep mode	(see 2.3 Settings Concerning the System and the Power Consumption)
400	Invalid SC30 parameter	(see 6.1 SC30 Parameter Commands)
401	Error in SC30 data set	(see 6.2 SC30 Correction Data Commands)
500	No ram left	No internal RAM left – contact your supplier