OceanOptics

Technical Note

RS-232 Serial Protocol

For

Ocean Spectrometers

TABLE OF CONTENTS

TABLE OF FIGURES

OVERVIEW OF RS-232

1.1 DEFAULT DEVICE CONFIGURATION

Figure 1 shows a high-level overview of how the serial ports are connected for RS-232 and the direction of data transmission for them.

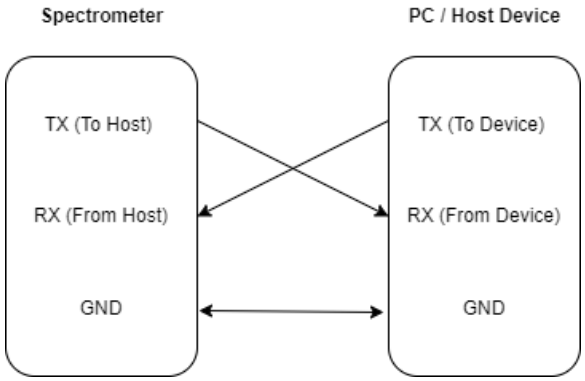

Figure 1 RS-232 Ports

The table below provides some specifications regarding the RS-232 implementation on Ocean spectrometers.

The voltage limits for TX (To Host) describe the voltage levels the spectrometer outputs when transmitting data. The voltage limits for the RX (From Host) describes the voltage levels the spectrometer can receive as inputs.

The RS-232 data configuration is fixed to (8N1). The default baud rate is also important to keep in mind when first interfacing with our spectrometers.

Your PC/Host Device will need to use the default baud rate **when first powering the spectrometer**.

Voltage Limits		
	Minimum	Maximum
Tx (To Host)	-6V	+6V
Rx (From Host)	-30V	+30V
RS-232 Data Configuration (8N1)		
8 bits of data	No Parity bit	1 stop bit
Default Baud Rate		
115200		

1.2 CONNECTING TO A PC

Below is an example list of items for connecting your spectrometer to a PC/Host device. Figure 2 shows how all the listed parts are connected. Both the list and diagram reflect what was used during testing and development of RS-232 for our spectrometers.

1. Spectrometer
2. A serial to USB adapter
3. Ocean's RS-232 accessory cable (CBL-ISDF-DB9)
4. PC/Host Device
5. External +5VDC power supply that is **center positive.**

Note: If you wish to use RS-232 you must have an external power source. You cannot use the USB-C cable to power our spectrometers when using RS-232. Our devices only allow one communication protocol at a time. If a USB cable is connected to the device, that will become the communication venue for the host.

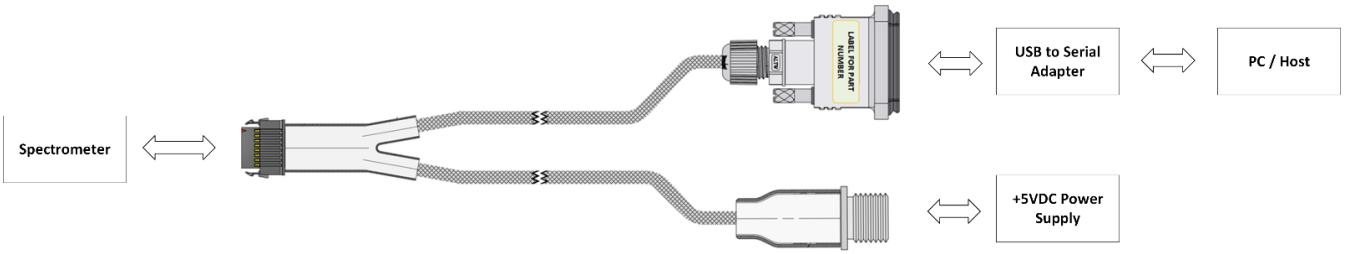

Figure 2 Connecting Spectrometer to PC Host

CABLE DESIGN CONSIDERATIONS

The RS-232 accessory cable sold by Ocean may not be long enough for your application. If that is the case, then you can use some mating connectors to effectively extend the cable. The following connectors or cables **are examples** of what is compatible with Ocean's accessory cable. There are similar alternatives on the market for these connectors or cables.

For power, you can mate Ocean's RS-232 cable with the part listed below. This part takes full advantage of the locking mechanism on Ocean's RS-232 accessory cable. **Note:** This part does not come with cabling/wiring, it is just the connector.

Figure 3 Mating Connector for Power

Manufacturer	Manufacturer Part #	Description
Tensility International Corp	50-00011	Power Barrel Connector Plug 2.50mm ID (0.098"), 5.50mm OD (0.217") Free Hanging (In-Line)

The DB9 female connector on Ocean's RS-232 cable can be mated with a male DB9 cable.

Figure 4 Male DB9 Cable

Manufacturer	Manufacturer Part #	Description
CNC Tech	710-10011-00200	D-Sub Cable Assembly DB09 Beige, Individual (Round) 6.56' (2.00m) Plug, Male Pins to Plug, Male Pins Shielded

If additional info related to the Samtec connector on our spectrometers is desired, refer to the device's user manual.

2.1 OCEAN OPTICS RS-232 CABLE ASSEMBLY

Below is a drawing for the Ocean Optics RS-232 cable accessory with details on the pinouts and how they connect to the previously mentioned connectors/ports. This visual should help in the case a custom cable design is desired.
Keep in mind again that Tx and Rx are from the perspective of the spectrometer itself.

Another thing to note is that power jack is center positive, so keep that in mind when choosing a +5VDC power supply.

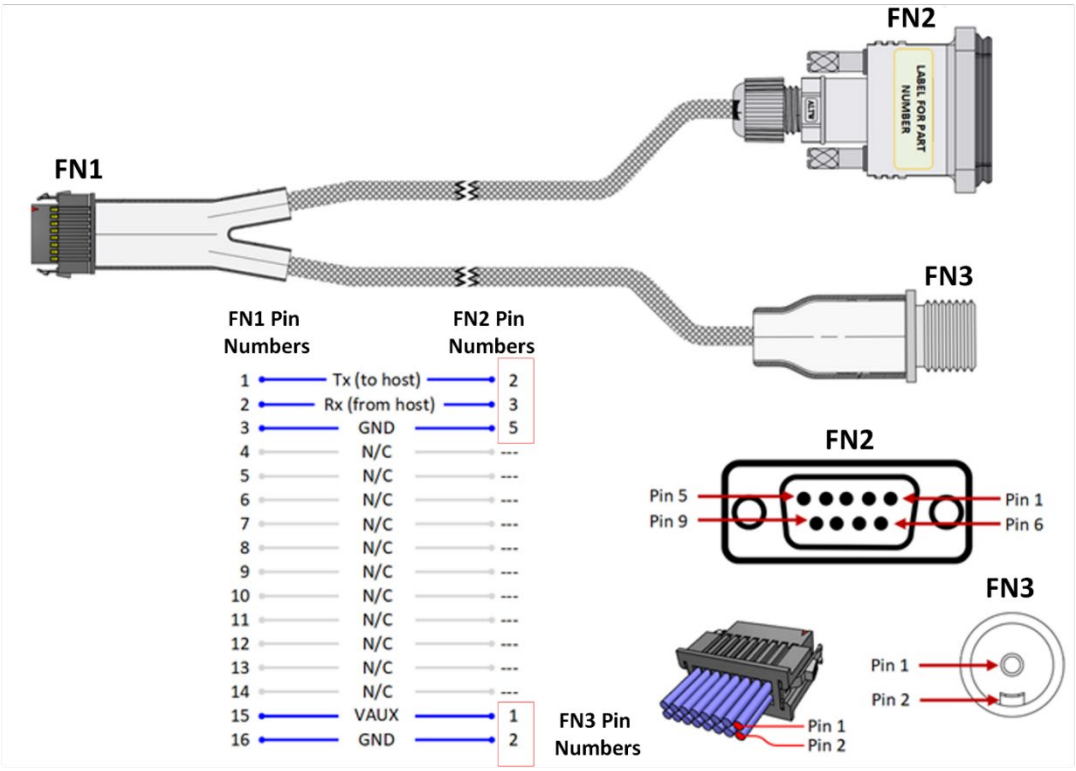

Figure 5 RS-232 Accessory Cable

2.2 SAMPLE PYTHON CODE

Contact your Sales Representative for sample code that exercises the RS-232 commands. Keep in mind when using this example program, you will need additional hardware to connect your development PC to the device RS-232 port.

Below is an example list of items used to effectively run the Python script on a PC.

- Spectrometer
- USB to Serial Adapter (Example: Digitus DA-70156)
- PC/Host Device
- Ocean RS-232 accessory cable.
- Python Development Environment (i.e. VSCode, Pycharm, Spyder, etc.)

COMMAND LINE SYNTAX

This section describes the general format and syntax of supported commands.

Keep in mind words enclosed in **<**angle brackets**>** are references to syntactical elements. Words enclosed in **[**square brackets**]** represent optional items which may be left out from the command line. The brackets are not used when the words appear in the command line.

3.1 CONTROL CHARACTERS

Table 1 shows the Controls characters used by the command protocol to manage the spectrometer. These control characters are not printable, and they are used to signal the end of the command strings.

Acronym	Hexadecimal Value	Definition
<CR>	0x0D	Carriage Return
<LF>	0x0A	Line Feed

Table 1: Control Characters

3.2 COMMAND ENCODING

All command set and read operations are text-based using printable ASCII characters except for the Acquire Spectra command response which is encoded as **binary data**.

When the Host software sends any command to the spectrometer, the spectrometer "echoes" the command back out the RS-232 port.

3.3 SET COMMAND SYNTAX

*<command>***=***<value><CR>*

The process for sending a Set command is as follows.

- Start with the command name.
- An equal sign follows the command name.
- Value parameter follows equal sign. Multiple values are separated by commas.
- End command with Carriage Return.

Once the request is received, the spectrometer responds with one of the response strings described in Table 2.

Response	Description
OK	Request processed successfully
ERROR	Error condition encountered

Table 2: Command response

Figure 6 shows the command sequence to set a parameter.

I=10000<CR>
OK<CR><LF>

Figure 6: Set command sequence

Commands requiring multiple values are passed as a comma-separated list.

<command>=<parameter 1>,< parameter 2>,< parameter n><CR>

3.4 READ COMMAND SYNTAX

*<command>**?**[option]<CR>*

The process for Read commands is as follows.

- Start with the command name.
- A question mark follows the command name.
- Depending on the command an option argument may be given.
- End command with Carriage Return.

Once the request is processed by the spectrometer, the response is one of the following formats:

- An alphanumeric string
- A numeric string
- Binary format

Figure 7 shows the command sequence for a Read operation. Notice control characters are used by the Host and Device.

N?<CR>
ST00253<CR><LF>

Figure 7: Read command example

Commands requiring multiple parameters are reported as a comma-separated list.

<parameter 1>,< parameter 2>,< parameter n><CR><LF>

3.5 COMMAND SUMMARY

Table 3 provides a summary of supported commands. All upper and lower limits for the command values can be found in the spectrometer User Manual.

End users should check section 3.6 Command Support Exclusions as not all commands are available on each spectrometer model.

NOTE: All command Names are upper case.

Name	Access	Response	Description
A	Read / Set	String	Scans to Average
B	Read / Set	String	Single Strobe
C	Read / Set	String	Continuous Strobe
I	Read / Set	String	Integration Time
J	Read / Set	String	Lamp Enable Feature
K	Read / Set	String	Baud Rate
L	Read / Set	String	LED Status
M	Read	String	Spectrometer Model
N	Read	String	Serial Number
P	Read / Set	String	Partial Pixel Mode
S	Read	Binary	Acquire Spectra
T	Read / Set	String	Trigger mode
V	Read	String	Firmware version
X	Read	String	Calibration coefficients

Table 3: Command Name List

3.6 COMMAND SUPPORT EXCLUSIONS

Table 3 provides an overview for all known RS232 commands. The following sections provide a detailed view of each spectrometer model and the corresponding unsupported commands.

3.6.1 ST spectrometer

Table 4 lists all unsupported commands by firmware versions for this spectrometer family.

Item	ST
Micro version	**1.2.5**
FPGA version	**1.3.1**
Unsupported commands	**A**
	B
	C
	L

Table 4

3.6.2 SR2 and HR2 spectrometer

Table 5 lists all unsupported commands by firmware versions for this spectrometer family.

Item	SR2		HR2	
Micro	**1.2.5**	**2.0.7**	**1.2.5**	**2.0.7**
FPGA	**1.3.8**	**1.3.7**	**1.3.8**	**1.3.8**
Unsupported commands	**A**	**A**	**A**	**A**
	B	**B**	**B**	**B**
	C	**C**	**C**	**C**

Table 5

3.6.3 SR6 and HR6 spectrometer

Table 6 lists all unsupported commands by firmware versions for this spectrometer family.

Item	SR6		HR6	
Micro	**1.2.5**	**2.0.7**	**1.2.5**	**2.0.7**
FPGA	**1.3.3**	**1.3.3**	**1.3.3**	**1.3.3**
Unsupported commands	**A**	**A**	**A**	**A**
	B	**B**	**B**	**B**
	C	**C**	**C**	**C**

Table 6

3.6.4 SR4 and HR4 spectrometer

Table 7 and Table 8 list all unsupported commands by firmware versions for this spectrometer family.

Item	SR4	HR4
Micro	**1.2.5**	**1.2.5**
FPGA	**1.3.3**	**1.3.3**
Unsupported commands	**A**	**A**
	B	**B**
	C	**C**

Table 7

Item	SR4	HR4
System version	**3.0.1**	**3.0.1**
Unsupported commands	**None**	**None**

Table 8

3.6.5 NR spectrometer

Table 9 list all unsupported commands by firmware versions for this spectrometer family.

Item	NR
Micro	**1.2.5**
FPGA	**1.3.0**
Unsupported commands	**A**
	B
	C

Table 9

3.7 COMMAND DETAILS

The following sections provide details on each command operation.

3.7.1 Scans to Average

Sets the number of discrete spectra to be summed together. Setting the scans to average parameter above 1 scan will set the pixel size to 32 bits. See Spectra Data Format section for details.

NOTE: Host software must divide 32 bits pixel value by the number of scans to normalize each pixel value.

A=5<CR>
OK<CR><LF>

To query the value, use the read command pattern below.

A?<CR>
5<CR><LF>

3.7.2 Single Strobe

Configures the Single Strobe signal and associated parameters shown in Table 10. All parameters are passed as a comma-separated list.

Parameter	Purpose	Notes
1	Strobe status	Enable = 1 Disable = 0
2	Strobe Delay	Microseconds
3	Strobe width	Microseconds

Table 10 Single Strobe

Below is an example to enable the Single Strobe functionality with a delay of 150us and strobe width of 1000us.

B=1,150,1000<CR>
OK<CR><LF>

It is possible to disable the Single Strobe functionality as below.

B=0<CR>
OK<CR><LF>

To query the strobe configuration, use the read command pattern below.

B?<CR>
1,150,1000<CR><LF>

3.7.3 Continuous Strobe

Configures the Continuous Strobe signal and associated parameters shown in Table 11. All parameters are passed as a comma-separated list.

Parameter	Purpose	Notes
1	Strobe status	Enable = 1 Disable = 0
2	Strobe Period	Microseconds

Table 11 Continuous Strobe

Below is an example to enable the Continuous Strobe functionality with a strobe period of 1500us.

C=1,1500<CR>
OK<CR><LF>

It is possible to disable the Single Strobe functionality as below.

C=0<CR>
OK<CR><LF>

To query the value, use the read command pattern below.

C?<CR>
1,1500<CR><LF>

3.7.4 Integration Time

Refer to spectrometer user manual for minimum and maximum integration times. Integration time value is specified in microseconds.

I=325910<CR>
OK<CR><LF>

To query the value, use the read command pattern below.

I?<CR>
325910<CR><LF>

3.7.5 Lamp Enable

The Lamp Enable command allowed values are listed below.

Lamp Enable	Description
Details	High=1 Low=0

Table 12 Lamp Enable

A set operation is shown below.

J=1<CR>
OK<CR><LF>

Current state of the parameter can be verified with a read operation as shown below.

J?<CR>
1<CR><LF>

3.7.6 Change Baud Rate

The following table shows the supported baud rates:

Baud rates
2400, 9600, 14400,19200, 38400, 115200

When changing baud rates, the following sequence must be followed:

1. Host sends desired baud rate value, communicating at the current baud rate.
2. Device acknowledges request with **OK**, otherwise it responds with **ERROR**.
3. Host waits longer than 50 ms and switches to new baud rate.
4. Host resends the command baud rate value, communicating at the new baud rate.
5. Device confirms change, communicating at the new baud rate.

NOTE: If a deviation occurs at any step, the previous baud rate is utilized

K=9600<CR>
OK<CR><LF>
K=9600<CR>
OK<CR><LF>

3.7.7 LED Indicator

The LED indicator allowed values are listed below.

LED indicator values
Enabled = 1 Disable = 0

A set operation is shown below.

L=1<CR>
OK<CR><LF>

L?<CR>
1<CR><LF>

3.7.8 Spectrometer Model

Returns the spectrometer model.

M?<CR>
OceanST<CR><LF>

3.7.9 Serial Number

Returns the device serial number assigned during manufacturing.

N?<CR>
SR221234<CR><LF>

3.7.10 Partial Pixel Mode

Allows user to select which pixels are returned by an Acquire Spectra command. Note that all pixels are still acquired for every scan, but this command will allow the user to specify which ones are returned to the user.

Two command arguments must be provided, the lower pixel first and the upper pixel. Pixel values must be within a valid range. Refer to spectrometer user manual for the number of available pixels.

The current pixel range can be read back as shown below.

P?<CR>
25,200<CR><LF>

3.7.11 Acquire Spectra

Acquires single spectra with the current set of operation parameters.

S?<CR>

Returned data is in **binary format**. Refer to section **Spectra Data Format** for more details.

3.7.12 Trigger Mode

Set which trigger mode the spectrometer will function with.

Trigger Mode	Value
Software Trigger	0
External Edge Trigger	1
External Level Trigger	2

A set operation is shown below.

T=1<CR>
OK<CR><LF>

The current state of the parameter can be verified with a read operation as shown below.

T?<CR>
1<CR><LF>

3.7.13 Firmware Version

Returns the spectrometer microcontroller firmware version.

V?<CR>
1.2.0<CR><LF>

3.7.14 Calibration Coefficients

Returns values set by Ocean during manufacturing. Each individual calibration parameter is mapped to an index as shown on Table 13.

NOTE: All coefficients are IEEE single-precision floating-point values occupying 32 bits. Each coefficient query response is encoded as a string, maximum count is 16 characters.

Index	Calibration Entry
0	Wavelength Polynomial Order
1	Wavelength Coefficient 0
2	Wavelength Coefficient 1
3	Wavelength Coefficient 2
4	Wavelength Coefficient 3
10	Non-linearity Polynomial Order
11	Non-linearity Coefficient 0
12	Non-linearity Coefficient 1
13	Non-linearity Coefficient 2
14	Non-linearity Coefficient 3
15	Non-linearity Coefficient 4
16	Non-linearity Coefficient 5
17	Non-linearity Coefficient 6
18	Non-linearity Coefficient 7

Table 13

The example below shows the command to query the Wavelength Coefficient 1.

X?2<CR>
1.2857E-08<CR><LF>

SPECTRA DATA FORMAT

When using the Acquire Spectra command every spectra message response starts with a 32-byte header called “metadata header” followed by the pixel data.

4.1 METADATA HEADER

The metadata header provides detailed information about the spectra data. The header format is the same across all Ocean spectrometers. The “metadata header” is of a fixed length, so it can always be assumed that 32 bytes can be read and processed to determine additional reads.

Metadata header fields that have multi-byte values are Least Significant Byte first.

Field	Offset (Bytes)	Length (Bytes)	Notes
Metadata version	0	1	**1** = Version 1
Trigger Mode	1	1	See Trigger mode command
Reserved	2	2	
Spectra Size	4	2	Spectra data length in bytes
Scan Count	6	4	32 bits counter
Tick Count	10	8	
Integration Time	18	4	Time in microseconds
Pixel Format	22	1	**1** = 16 bits **2** = 32 bits
Reserved	23	9	
Total Bytes		32	

Table 14: Metadata header

4.2 SPECTRA DATA FORMAT

Pixel format can be determined from the **Pixel Format** metadata field. End user software can use this metadata field to determine the amount of bytes leftover to retrieve the spectra data.

Pixel	Offset (Bytes)	Description
0	0	Pixel 0 least significant byte
0	1	Pixel 0 most significant byte
1	2	Pixel 1 least significant byte
1	3	Pixel 1 most significant byte

Table 15: Pixel format (16 bits)

If the “Scans to Average” setting is set to a value above 1 scan, the pixel format will be set to 32 bits format. Host software must divide pixel counts value by the number of “Scans to Average” to normalize each pixel value.

Pixel	Offset (Bytes)	Description
0	0	Pixel 0 least significant byte
0	1	
0	2	
0	3	Pixel 0 most significant byte

Table 16: Pixel format (32 bits)

DRIVER DEVELOPMENT

Developing low-level software is always a challenging task, especially when it comes to the details. This section provides data samples of the serial data exchange over the RS-232 interface.

5.1 QUERY COMMAND

Table 17 shows the **X** command data exchange from the host's perspective.

Host	Description	Bytes (hex)
TX	**X?2** <CR>	58 3F 32 0D
RX	Command Echo	58 3F 32 0D
RX	3.447893e-01 <CR> <LF>	33 2E 34 34 37 38 39 33 65 2D 30 31 0D 0A

Table 17

5.2 GET SPECTRA COMMAND

Table 18 shows the S command data exchange from the host's perspective. The response includes the metadata header and a few pixels.

Host	Description	Bytes (hex)
TX	**S?** <CR>	53 3F 0D
RX	Command Echo	53 3F 0D
RX	Spectra data	01 00 02 00 D8 0B 03 00 00 00 C8 5F 00 00 00 00 00 00 00 35 0C 00 01 00 00 00 00 00 00 00 04 00 14 02 F8 01 06 02 09 02 1B 02 …

Table 18

Table 19 shows the metadata header decoding plus some pixels. Metadata header fields that have multi-byte values are Least Significant Byte first.

Field	Serial data (hex)	Value (decimal)
Metadata	01	1
Trigger mode	00	0
Reserved	02 00	-
Spectra size	D8 0B	3,032
Scan count	03 00 00 00	3
Tick count	C8 5F 00 00 00 00 00 00	24,520
Integration	00 35 0C 00	800,000
Pixel Format	01	1
Reserved	00 00 00 00 00 00 00 04 00	-
Pixel 1	14 02	532
Pixel 2	F8 01	504
Pixel 3	06 02	518

Table 19