

USB2000 Data Sheet

Description

The Ocean Optics USB2000 Spectrometer includes the linear CCD-array optical bench, plus all the circuits necessary for spectrometer operation. The result is a compact, flexible system, with no moving parts, that's easily integrated as an OEM component.



The USB2000 spectrometer is a unique combination of technologies providing users with both an unusually high spectral response and good optical resolution in a single package. The electronics have been designed for considerable flexibility in connecting to various USB2000 series modules as well as external interfaces. The USB2000 interfaces to PCs, PLCs and other embedded controllers through either USB or RS-232 communications. The information included in this guide provides detailed instructions on the connection and operation of the USB2000.

The detector used in the USB2000 spectrometer is a high-sensitivity 2048-element CCD array from Sony, product number ILX511. (For complete details on this detector, visit Sony's web site at www.sony.com. Ocean Optics applies a coating to all ILX511 detectors, so the optical sensitivity could vary from that specified in the Sony datasheet)

The USB2000 operates off of a single +5VDC supply and either a USB or RS-232 interface. The USB2000 is a microcontroller-controlled spectrometer, thus all operating parameters are implemented through software interfacing to the unit.

The USB2000 has a 10 pin external interface to easily integrate with Ocean Optics other modular components to form an entire system.

Features

- ❑ High sensitivity of up to 90 photons/counts
- ❑ An optical resolution of $\sim 0.3\text{nm}$ (FWHM)
- ❑ A wide variety of optics available
 - 14 gratings
 - 6 slit widths
 - 3 detector coatings
 - 6 optical filters
- ❑ Integration times from 3 to >30000 ms
- ❑ Embedded microcontroller allows programmatic control of all operating parameters
- ❑ EEPROM storage for
 - Wavelength Calibration Coefficients
 - Linearity Correction Coefficients
 - Other Configuration Parameters
- ❑ Low power of only 450 mW
- ❑ 12 bit, 1MHz A/D Converter
- ❑ 3 triggering modes
- ❑ 2 strobe signals for triggering other devices
- ❑ Programmable for Standalone Operation
- ❑ Plug-n-Play Interface for PC applications
- ❑ CE Certification

Specifications

Specifications	Criteria
Absolute Maximum Ratings: V_{CC} Voltage on any pin	+ 5.5 VDC $V_{CC} + 0.2$ VDC
Physical Specifications: Physical Dimensions Weight	90 mm x 65 mm x 35 mm 190 g
Power: Power requirement (master) Supply voltage Power-up time	95 mA at +5 VDC 4.5 – 5.5 V ~5s depending on code size
Spectrometer: Design Focal length (input) Focal length (output) Input Fiber Connector Gratings Entrance Slit Detector Filters	Asymmetric crossed Czerny-Turner 42mm 68mm (75, 83, and 90mm focal lengths are also available) SMA 905 14 different gratings 5, 10, 25, 50, 100, or 200 μ m slits. (Slits are optional. In the absence of a slit, the fiber acts as the entrance slit.) Sony ILX511 CCD 2 nd and 3 rd order rejection, long pass (optional)
Spectroscopic: Integration Time Dynamic Range Signal-to-Noise Readout Noise (single dark spectrum) Resolution (FWHM) Stray Light Spectrometer Channels	3 – >30,000 msec 2×10^8 250:1 single acquisition 3.5 counts RMS, 20 counts peak-to-peak 0.03 – 10.0 nm varies by configuration (see www.Oceanoptics.com for configuration options) <0.05% at 600 nm; <0.10% at 435 nm One
Environmental Conditions: Temperature Humidity	-30° to +70° C Storage & -10° to +50° C Operation 0% - 90% noncondensing
Interfaces: USB RS-232	USB 1.1, 12 Mbps 2-wire RS-232

Mechanical Diagrams

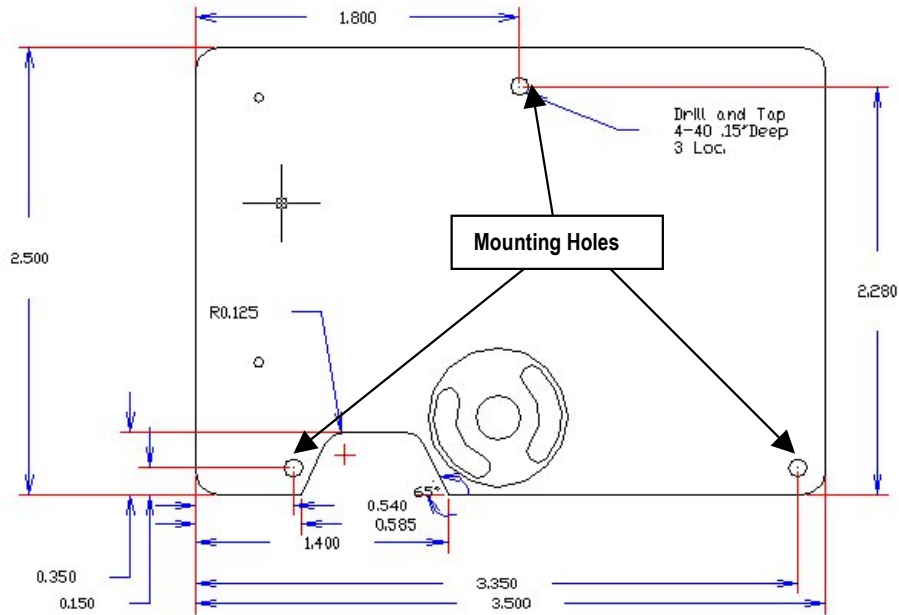


Figure 1: USB2000 Outer Dimensions (Bottom View)

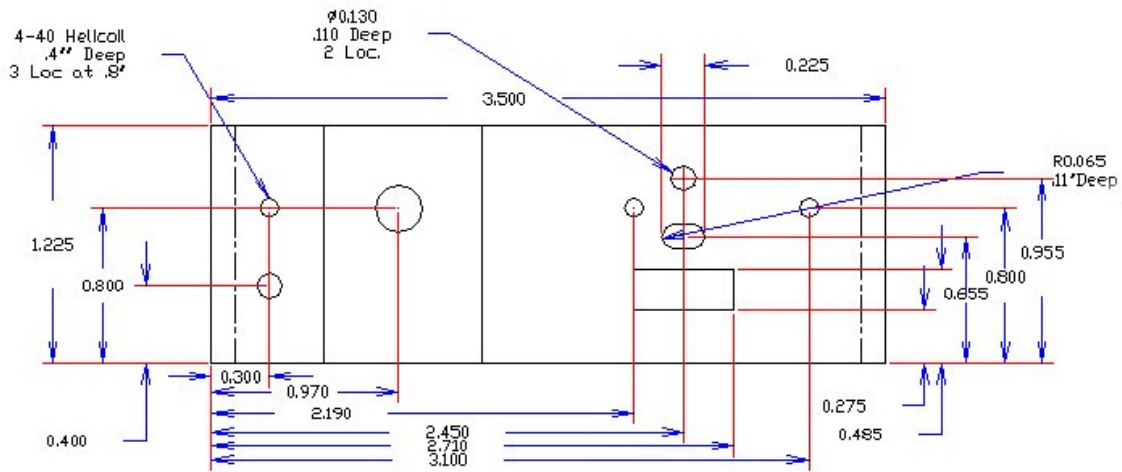
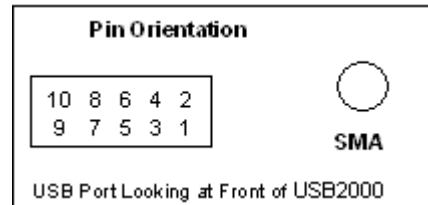


Figure 2: USB2000 Outer Dimensions (Inverted Front View)

Electrical Pinout

Listed below is the pin description for the USB2000 Accessory Connector (J2) located on the front vertical wall of the unit. The connector is model IPT1-105-01-S-D-RA from Samtec and a right-angle mating connector is IPS1-105-01-S-D-RA. Connector information and samples are available from www.Samtec.com.

Pin#	Description
1	V_{CC} , V_{USB} or $5V_{in}$
2	RS232 Tx
3	RS232 Rx
4	Lamp Enable
5	Continuous Strobe
6	Ground
7	External Trigger
8	Single Strobe
9	I ² C SCL
10	I ² C SDA



Function	Input/Output	Description
V_{CC} , V_{USB} or $5V_{in}$	Input or Output	This is the input power pin to the USB2000. Additionally when operating via a Universal Serial Bus (USB) this is the USB power connection (+5V) which can be used to power other peripherals (Care must be taken to insure that the peripheral complies with USB Specifications).
RS232 Tx	Output	RS232 Transmit signal – for communication with PC connect to DB9 pin 2
RS232 Rx	Input	RS232 Receive signal – for communication with PC connect to DB9 pin 3.
Lamp Enable	Output	A TTL signal that is driven Active HIGH when the Lamp Enable command is sent to the USB2000.
Continuous Strobe	Output	TTL output signal used to pulse a strobe that is divided down from the Master Clock signal.
Ground	Input/Output	Ground
Single Strobe	Output	TTL output signal used to pulse a strobe that is high at the start of each integration period.
External Trigger In	Input	The TTL input trigger signal. In External Hardware Trigger mode this is a rising edge trigger input. In Software Trigger Mode this is a Active HIGH Level signal.
I ² C SCL	Input/Output	The I ² C Clock signal for communications to other I ² C peripherals.
I ² C SDA	Input/Output	The I ² C Data signal for communications to other I ² C peripherals

CCD Overview

CCD Detector

The detector used for the USB2000 is a charge transfer device (CCD) that has a fixed well depth (capacitor) associated with each photodetector (pixel).

Charge transfer, reset and readout initiation begin with the integration time clock going HIGH. At this point, the remaining charge in the detector wells is transferred to a shift register for serial transfer. This process is how the array is read.

The reset function recharges the photodetector wells to their full potential and allows for nearly continuous integration of the light energy during the integration time, while the data is read out through serial shift registers. At the end of an integration period, the process is repeated.

When a well is fully depleted by leakage through the back-biased photodetector, the detector is considered saturated and provides the maximum output level. The CCD is a depletion device and thus the output signal is inversely proportional to the input photons. The electronics in the USB2000 invert and amplify this electrical signal.

CCD Well Depth

We strive for a large signal-to-noise (S:N) in optical measurements so that small signal variations can be observed and a large dynamic range is available. The S:N in photon noise-limited systems is defined and measured as the square root of the number of photons it takes to fill a well to saturation. In the USB2000, the well depth of the CCD pixels is about 160,000 photons, providing a S:N of 400:1 (S:N can also be measured as the saturation voltage divided by near-saturation RMS noise). There is also a fixed readout noise component to all samples. The result is a system with a S:N of ~275:1.

There are two ways to achieve a large S:N (e.g., 6000:1) in CCD detectors where photon noise is predominant.

1. Use a large-well device that integrates to saturation over a long period of time until the photon noise is averaged out by the root of n multiples of a defined short Δt .
2. Use a small-well device that integrates to saturation at one short Δt and then signal average mathematically n times.

Theoretically, both approaches achieve the same results, though there are large differences in actual operation. Traditional spectroscopic instruments use large-well devices and 16-bit ADCs to achieve the defined S:N. The USB2000 uses a small-well device and utilizes signal averaging to achieve the same S:N. A brief comparison of large and small-well devices is shown in the table below.

Well Depth Comparison

Large-well CCDs	Small-well CCDs
Low photon noise	Medium photon noise that can be averaged out
Low optical sensitivity	High optical sensitivity
High power consumption	Low power consumption
>10 MHz operating speeds	Moderate operating speeds (~2 MHz)

Signal Averaging

Signal averaging is an important tool in the measurement of spectral structures. It increases the S:N and the amplitude resolution of a set of samples. The types of signal averaging available in our software are time-based and spatial-based.

When using the time-base type of signal averaging, the S:N increases by the square root of the number of samples. Signal averaging by summing is used when spectra are fairly stable over the sample period. Thus, a S:N of 2500:1 is readily achieved by averaging 100 spectra.

Spatial averaging or pixel boxcar averaging can be used to improve S:N when observed spectral structures are broad. The traditional boxcar algorithm averages n pixel values on each side of a given pixel.

Time-based and spatial-based algorithms are not correlated, so therefore the improvement in S:N is the product of the two processes.

In review, large-well devices are far less sensitive than small-well devices and thus, require a longer integration time for the same output. Large-well devices achieve a good S:N because they integrate out photon noise. Small-well devices must use mathematical signal averaging to achieve the same results as large-well devices, but small-well devices can achieve the results in the *same period of time*. This kind of signal averaging was not possible in the past because analog-to-digital converters and computers were too slow.

Large-well devices consume large amounts of power, resulting in the need to build thermoelectric coolers to control temperature and reduce electronic noise. Then, even more power is required for the temperature stabilization hardware. But small-well devices only need to use signal averaging to achieve the same results as large-well devices, and have the advantages of remaining cool and less noisy.

Internal Operation

Pixel Definition

A series of pixels in the beginning of the scan have been covered with an opaque material to compensate for thermal induced drift of the baseline signal. As the USB2000 warms up, the baseline signal will shift slowly downward a few counts depending on the external environment. The baseline signal is set between 90 and 140 counts at the time of manufacture. If the baseline signal is manually adjusted, it should be left high enough to allow for system drift. The following is a description of all of the pixels:

Pixel	Description
0–1	Not usable
2–24	Optical black pixels
24–25	Transition pixels
26–2074	Optical active pixels

It is important to note that the USB2000 only digitizes the first 2048 pixels.

CCD Detector Reset Operation

At the start of each integration period, the detector transfers the signal from each pixel to the readout registers and resets the pixels. The total amount of time required to perform this operation is $\sim 20 \mu\text{s}$. The user needs to account for this time delay when the pixels are optically inactive, especially in the external triggering modes.

Timing Signals

Strobe Signals

Single Strobe

The Single Strobe signal is a TTL HIGH pulse that occurs at the start of each integration period. This pulse is a $2 \mu\text{s}$ wide pulse that occurs at the start of each integration period. The pulse is only active if the Lamp Enable command is active.

Continuous Strobe

The Continuous Strobe signal is a 976Hz pulse-train (50% duty cycle). The pulse is only active if the Lamp Enable command is active

Synchronizing Strobe Events

If the application requires more than one pulse per integration period, the user needs to insure the continuous strobe and integration period are synchronized. The integration time must be set so that an equal number of strobe events occurs during any given integration period. This synchronization only occurs when the integration period is a multiple of a power of 2.

Triggering Modes

The USB2000 supports three triggering modes, which are set with the Trigger Mode command. Detailed information of each triggering mode follows.

Normal

In this mode, the USB2000 uses the user-supplied integration clock and continuously scans the CCD array.

External Software Trigger

In this mode, the USB2000 uses the user-defined integration clock; however, the A/D converter is required to wait until the Trigger Input Signal goes HIGH before it acquires the data at the start of the next integration period. This is an asynchronous trigger mode by allows the user to define an integration period.

External Hardware Trigger

In this mode, the USB2000 uses an internally generated clock to generate the integration period. On the rising edge of this signal, the internal logic resets the CCD array, integrates for 50ms. This is a synchronous trigger mode but the integration time is fixed. OEMs can contact Ocean Optics for other periods.

USB2000 USB Port Interface Communications and Control Information

Overview

The USB2000 is a microcontroller-based Miniature Fiber Optic Spectrometer that can communicate via the Universal Serial Bus or RS-232. This document contains the necessary command information for controlling the USB2000 via the RS-232 interface. Refer to the OOIPWinIP software package for driver level USB control information.

Hardware Description

The USB2000 uses a Cypress AN2131 microcontroller, which has a high speed 8051, combined with an USB ASIC. Program code and data coefficients are stored in external E²PROM that are loaded at boot-up via the I²C bus.

Depending on the user's application, there is an important trade-off that needs to be made between maximum baud rate and integration time. Onboard the USB2000, there are 2 timers (an 8-bit and 16-bit) that control the integration clock and the baud rate generator. This allocation is performed with the `y` command. If the bit timer is allocated to the integration clock then the maximum integration time is 255ms. If it is allocated to the baud rate, then the max baud rate is 38,400Baud. This command should be sent at the start of the session since both timers are reset by its operation. Depending upon the application, the user will allocate the 16 bit timer to the function that provides the most flexibility.

Spectral Memory Storage

The microcontroller has 8K of internal SRAM and 128K of external SRAM. Due to memory mapping and paging constraints, only 96K of the external SRAM is useable. This 96K is further broken down into 64K for Spectral data memory and the remainder for internal program variables.

The USB2000 can store up to 16 spectra in the spectral data section. The full spectra (2048 points) is stored. Spectra are organized in a stack formation (i.e. LIFO). Its important to realize that the spectral math (averaging and boxcar smoothing) is performed when the data is transmitted out and not when the spectra are acquired. This allows up to 16 scans to be acquired with one command (set A=16 and send the S command) and then read out on scan at a time (set A=1 and use Z1 command).

Instruction Set

Command Syntax

The list of the commands is shown in the following table along with the microcode version number the commands were introduced with. All commands consist of an ASCII character passed over the serial port, followed by some data. The length of the data depends on the command. The format for the data is either ASCII or binary (default). The ASCII mode is set with the “a” command and the binary mode with the “b” command. To insure accurate communications, all commands respond with an ACK (ASCII 6) for an acceptable command or a NAK (ASCII 21) for an unacceptable command (i.e. data value specified out of range).

In the ASCII data value mode, the USB2000 “echoes” the command back out the RS-232 port. In binary mode all data, except where noted, passes as 16-bit unsigned integers (WORDS) with the MSB followed by the LSB. By issuing the “v command” (Version number query), the data mode can be determined by viewing the response (ASCII or binary).

In a typical data acquisition session, the user sends commands to implement the desired spectral acquisition parameters (integration time, etc.). Then the user sends commands to acquire spectra (S command) with the previously set parameters. If necessary, the baud rate can be changed at the beginning of this sequence to speed up the data transmission process.

Pipe #	Description	Type	Full Speed Size (Bytes)	Endpoint Address
0	End Point 2 Out	Bulk	64	0x02
1	End Point 2 In	Bulk	64	0x82
2	End Point 7 Out (unused)	Bulk	64	0x07
3	End Point 7 In	Bulk	64	0x87

USB Command Summary

EP2 Command Byte Value	Description	Version
0x01	Initialize USB2000	1.00.0
0x02	Set Integration Time	1.00.0
0x03	Set Strobe Enable Status	1.10.0
0x04	Reserved	1.10.0
0x05	Query Information	1.10.0
0x06	Write Information	1.10.0
0x07	Write Serial Number	1.10.0
0x08	Get Serial Number	1.10.0
0x09	Request Spectra	1.10.0

EP2 Command Byte Value	Description	Version
0x0A	Set Trigger Mode	1.10.0
0x0B	Query Number of Plug-in Accessories Present	2.00.0
0x0C	Query Plug-in Identifiers	2.00.0
0x0D	Detect Plug-ins	2.00.0
0xFE	Query Status	2.41.0
	USB-LS450 Specific Commands	
0x20	Read Temperature	2.00.0
0x21	Set LED Mode	2.00.0
0x23	Query Calibration Constant	2.00.0
0x24	Send Calibration Constant	2.00.0
0x25	Set Analog Output	2.00.0
0x26	Load All Calibration Values from EEPROM	2.00.0
0x27	Write All Calibration Coefficients to EEPROM	2.00.0
	USB-ISS- UV/VIS Specific Commands	
0x40	Set Digital Potentiometer	2.41.0
0x41	Set Power Up Potentiometer Values	2.41.0
0x42	Read Potentiometer Values	2.41.0

Command Descriptions

A detailed description of all USB2000 commands follows. While all commands are sent to EP2 over the USB port, the byte sequence is command dependent. The general format is the first byte is the command value and the additional bytes are command specific values.

Byte 0	Byte 1	Byte 2	...	Byte n-1
Command Byte	Command Specific	Command Specific	...	Command Specific

Initialize USB2000

Initializes all values and Digital I/O ports on the USB2000. This command should be called at the start of every session. In addition to initializing the device, this command will also acquire a spectra and load the end points with the spectral data. The application needs to get this spectral data to maintain effective communications (refer to the spectral acquisition command for details). The default values are set as follows:

Parameter	Value
Integration Time	100ms
A/D Frequency	1 MHz
Trigger Mode	0 – Normal Trigger
Integration timer	16-bit timer -> Integration range is 3 to 65535 ms
Lamp Enable	0 - Off

Byte Format

Byte 0
0x01

Set Integration Time

Sets the USB2000+ integration time in milliseconds. The acceptable range is 3 – 65,535. If the value is less than 3 ms then the integration time is unchanged.

Byte Format

Byte 0	Byte 1	Byte 2
0x02	Integration Time LSB	Integration Time MSB

Set Strobe Enable Status

Sets the USB2000 Lamp Enable line (J2 pin 4) as follows. The Single Strobe and Continuous Strobe signals are enabled/disabled by this Lamp Enable Signal.

Data Byte = 0 → Lamp Enable Low/Off
Data Byte = !0 → Lamp Enable HIGH/On

Byte Format

Byte 0	Byte 1	Byte 2
0x03	Data Byte LSB	Data Byte MSB

Query Information

Queries any of the 19 stored spectrometer configuration variables. The Query command is sent to EP2 and the data is retrieved through End Point 7. The 19 configuration variables are indexed as follows:

Data Byte - Description
0 – Serial Number
1 – 0 th order Wavelength Calibration Coefficient
2 – 1 st order Wavelength Calibration Coefficient
3 – 2 nd order Wavelength Calibration Coefficient
4 – 3 rd order Wavelength Calibration Coefficient
5 – Stray light constant
6 – 0 th order non-linearity correction coefficient
7 – 1 st order non-linearity correction coefficient
8 – 2 nd order non-linearity correction coefficient
9 – 3 rd order non-linearity correction coefficient
10 – 4 th order non-linearity correction coefficient
11 – 5 th order non-linearity correction coefficient
12 – 6 th order non-linearity correction coefficient
13 – 7 th order non-linearity correction coefficient
14 – Polynomial order of non-linearity calibration
15 – Optical bench configuration: gg fff sss gg – Grating #, fff – filter wavelength, sss – slit size
16 – USB2000 configuration: AWL V A – Array coating Mfg, W – Array wavelength (VIS, UV, OFLV), L – L2 lens installed, V – CPLD Version
17 – Reserved
18 – Reserved
19 – Reserved

Byte Format

Byte 0	Byte 1
0x05	Data byte

Return Format (EP7)

The data is returned in ASCII format and read in by the host through End Point 7.

Byte 0	Byte 1	Byte 2	Byte 3	...	Byte 16
0x05	Configuration Index	ASCII byte 0	ASCII byte 1	...	ASCII byte 14

Write Information

Writes any of the 19 stored spectrometer configuration variables to EEPROM. The 19 configuration variables are indexed as described in the Query Information. The information to be written is transferred as ASCII information.

Byte Format

Byte 0	Byte 1	Byte 2	Byte 3	...	Byte 16
0x06	Configuration Index	ASCII byte 0	ASCII byte 1	...	ASCII byte 15

Write Serial Number

Writes the serial number to EEPROM. The information to be written is transferred as ASCII information.

Byte Format

Byte 0	Byte 1	Byte 2	...	Byte 16
0x07	ASCII byte 0	ASCII byte 1	...	ASCII byte 15

Query Serial Number

Queries the unit's serial number. The Query command is sent to EP2 and the data is retrieved through End Point 7. The information to be read is transferred as ASCII information.

Byte Format

Byte 0
0x08

Return Format

The data is returned in ASCII format and read in by the host through End Point 7.

Byte 0	Byte 1	Byte 2	...	Byte 16
0x08	ASCII byte 0	ASCII byte 1	...	ASCII byte 15

Request Spectra

Initiates spectra acquisition. The USB2000+ will acquire a complete spectrum (2048 pixel values). The data is returned in bulk transfer mode through EP2 in 64 packets, each containing 64 bytes. The table below provides the pixel order overview for the 2 different speeds. The pixel values are decoded as described below.

Byte Format

Byte 0
0x09

Return Format

The data is returned in bulk transfer mode through EP2 in 64 packets each containing 64 bytes. There is an additional packet containing one value that is used as a flag to insure proper synchronization between the PC and USB2000. The pixel values are decoded as described below.

Packet 0 - LSBs for first 64 pixels

Byte 0	Byte 1	...	Byte 63
Pixel 0 LSB	Pixel 1 LSB	...	Pixel 63 LSB

Packet 1 – MSBs for first 64 pixels

Byte 0	Byte 1	...	Byte 63
Pixel 0 MSB	Pixel 1 MSB	...	Pixel 63 MSB

Packet 2 – LSBs for second group of 64 pixels

Byte 0	Byte 1	...	Byte 63
Pixel 64 LSB	Pixel 65 LSB	...	Pixel 127 LSB

Packet 3 – MSBs for second group of 64 pixels

Byte 0	Byte 1	...	Byte 63
Pixel 64 MSB	Pixel 65 MSB	...	Pixel 127 MSB

...

Packet 62 – LSBs for last group of 64 pixels

Byte 0	Byte 1	...	Byte 63
Pixel 1984 LSB	Pixel 1985 LSB	...	Pixel 2047 LSB

Packet 3 – MSBs for last group of 64 pixels

Byte 0	Byte 1	...	Byte 63
Pixel 1984 MSB	Pixel 1985 MSB	...	Pixel 2047 MSB

Packet 64 – Synchronization Packet

Byte 0
0x69

Set Trigger Mode

Sets the USB2000 Trigger mode to one of four states. If an unacceptable value is passed then the trigger state is unchanged (refer to the [External Triggering Options Instructions](#) for a description of the trigger modes).

Data Value = 0 → Normal (Free running) Mode
Data Value = 1 → Software Trigger Mode
Data Value = 3 → External Hardware Trigger Mode

Byte Format

Byte 0	Byte 1	Byte 2
0x0A	Data Value LSB	Data Value MSB

Query Number of Plug-in Accessories

Queries the number of Plug-in accessories preset. This is determined at power up and whenever the Plug-in Detect command is issued.

Byte Format

Byte 0
0x0B

Return Format

The data is returned in Binary Format and read in by the host through End Point 7.

Byte 0
Value (BYTE)

Query Plug-in Identifiers

Queries the Plug-in accessories identifiers. This command returns 7 bytes with the last byte always being zero at this point. Each of the first 6 bytes correspond to Ocean Optics compatible devices which responded appropriately for I²C addresses 2 through 7 respectively. The I²C addresses are reserved for various categories of devices and the value for each category is shown below. I²C addresses 0-1 are reserved for loading program code from EEPROMS

Byte Format

Byte 0
0x0C

Return Format

The data is returned in Binary format and read in by the host through End Point 7.

Byte 0	Byte 1	...	Byte 5	Byte 6
Value @ I ² C address 2	Value @ I ² C address 3	...	Value @ I ² C address 7	0x00

Value	I ² C Addresses					
	2	3	4	5	6	7
	Light Sources & Sample Chambers	Expansion (A/D, D/A)	TBD	TBD	Power Sources	Customer Specific
1	ISS-VIS/NIR					
2	LS-450					
3	ISS/UV/VIS					
...						
255						

Detect Plug-ins

Reads all of the plug-in accessories that are plugged into the I²C bus. No data values are returned.

Byte Format

Byte 0
0x0D

Query Status

Returns a packet of information, which contains the current operating information. The structure of the status packet is given below:

Byte Format

Byte 0
0xFE

Return Format

The data is returned in Binary format and read in by the host through End Point 1 In. The structure for the return information is as follows:

Byte	Description	Comments
0-1	Number of Pixels - WORD	LSB MSB order
2-3	Integration Time - WORD	Integration time in ms – MSB LSB.
4	Lamp Enable	0 – Signal LOW !0 – Signal HIGH
5	Trigger Mode Value	
6	Request Spectra	0 – No Spectra Requested !0 – Spectral request in progress
7	Timer Swap	0 – 16-bit timer for Integration time 1 – 8-bit timer for Integration time
8	Spectral Data Ready	0 – Data Not yet Available !0 – Data is present for transfer
9 - 15	Reserved	Returns 0

RS232 Command Summary

RS232 Command Summary

Letter	Description	Version
A	Adds scans	1.00.0
B	Set Pixel Boxcar	1.00.0
C		
D		
E		

USB2000 Data Sheet

Letter	Description	Version
F	***Non functional but follows SAD500 command format***	
G	Set Data Compression	1.00.0
H	***Non functional but follows SAD500 command format***	
I	Sets integration time	1.00.0
J	Sets Lamp Enable Line	1.00.0
K	Changes baud rate	1.00.0
L	Clear Memory	
M	Set Data Storage Mode	1.00.0
N		
O		
P	Partial Pixel Mode	1.00.0
Q		
R		
S	Starts spectral acquisition with previously set parameters	1.00.0
T	Sets trigger mode	1.00.0
U		
V		
W	Query scans in memory	1.00.0
X		
Y		
Z	Read out Scan from memory	1.00.0
A	Set ASCII mode for data values	1.00.0
b	Set binary mode for data values	1.00.0
k	Sets Checksum mode	1.00.0
o	Oxygen (USB-LS-450) related commands	1.00.0
t		
u	Set Oxygen Calibration Coefficients	1.00.0
v	Provides microcode version #	1.00.0
x	Sets calibration coefficients	1.00.0
y	Sets 16-bit timer operation	1.00.0
?	Queries parameter values	1.00.0
+	Reads the plugged-in accessories	1.00.0
-	USB2000 Identifier	1.00.0

“Nonfunctional but follows SAD500 command format” means that the USB2000 requires a data value following the command variable and responds with just one NAK (instead of 3). This allows customer device drivers that were written for the SAD500 to be functional with minimal change.

RS232 Command Descriptions

A detailed description of all USB2000 commands follows. The {} indicates a data value which is interpreted as either ASCII or binary (default). The default value indicates the value of the parameter upon power-up.

Add Scans

Description: Sets the number of discrete spectra to be summed together. Since this routine can add up to 15 spectra, each with a maximum intensity of 4096, the maximum returned intensity is 65535.

Command Syntax:	A{DATA WORD}
Response:	ACK or NAK
Range:	1-15
Default value:	1

Pixel Boxcar Width

Description: Sets the number of pixels to be averaged together. A value of n specifies the averaging of n pixels to the right and n pixels to the left. This routine uses 32-bit integers so that intermediate overflow will not occur; however, the result is truncated to a 16-bit integer prior to transmission of the data. This math is performed just prior to each pixel value being transmitted out. Values greater than ~ 3 will exceed the idle time between values and slow down the overall transfer process.

Note

Prior to version 1.05.0, this command would accept a value but not perform the math.

Command Syntax:	B{DATA WORD}
Response:	ACK or NAK
Range:	0-15
Default value:	0

Set Spectral Pointer Value

Description: When scans are being stored in internal spectral memory (RAM), it's possible to adjust the pointer to readout the desired scan. Memory is organized in a LIFO fashion (Last In First Out) or like a stack. The beginning and end of the stack are 0 and 15 respectively.

Caution

The programmer must insure that this value is not set to a value that contains invalid data. For example, if there are only four scans in memory, its possible to set the value to something greater than this. When this is done, the user will receive garbage data from the Z command.

USB2000 Data Sheet

Command Syntax:	E{DATA WORD}
Response:	ACK or NAK
Range:	0 – 15
Default value:	0

Set Data Compression

Description: Specifies whether the data transmitted from the USB2000 should be compressed to speed data transfer rates. For more information on USB2000 Data Compression, see [Technical Note 1: USB2000 Data Compression](#).

Command Syntax:	G{DATA WORD}
Response:	ACK or NAK
Range:	0 – Compression off !0 – Compression on
Default value:	0

Integration Time

Description: Sets the USB2000's integration time, in milliseconds, to the value specified. If the Integration clock is using the 8-bit timer (y command) and a value greater than 255 is desired, the integration time is set to the LSB of the data word (i.e. value MOD 255).

Command Syntax:	I{DATA WORD}
Response:	ACK or NAK
Range:	5 - 65535
Default value:	100

Lamp Enable

Description: Sets the USB2000's Lamp Enable line to the value specified

Command Syntax:	J{DATA WORD}
Value:	0 = Light source/strobe off—Lamp Enable low 1 = Light source/strobe on—Lamp Enable high
Response:	ACK or NAK
Default value:	0

Baud Rate

Description: Sets the USB2000's baud rate.

Command Syntax:	K{DATA WORD}
Value:	0=2400 1=4800 2=9600 3=19200 4=38400 5=57600 6=115,200
Response:	See below
Default value:	2

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

1. Controlling program sends K with desired baud rate, communicating at the old baud rate.
2. A/D responds with ACK at old baud rate, otherwise it responds with NAK and the process is aborted.
3. Controlling program waits longer than 50 milliseconds.
4. Controlling program sends K with desired baud rate, communicating at the new baud rate.
5. A/D responds with ACK at new baud rate, otherwise it responds with NAK and old baud rate is used.

Note

If a deviation occurs at any step, the previous baud rate is used.

If the baud rate generator is using the 8-bit timer (y command), then the maximum baud rate is 38400 and not 57,600.

Clear Memory

Description: Clears spectral data memory based upon the value specified. Clearing memory is immediate since only pointer values are reinitialized.

Caution

All stored spectra are lost when the clear memory command is executed.

Command Syntax:	L{DATA WORD}
Value:	0 = Clear Spectral memory 1 = Clear Spectral memory
Response:	ACK or NAK
Default value:	N/A

Data Storage Mode

Description: Sets the data storage mode for future spectral acquisitions.

Command Syntax:	M{DATA WORD}
Value:	0 = Scans transmitted through the serial port 1 = Scans stored in spectral memory and not transmitted
Response:	ACK or NAK
Default value:	0

Pixel Mode

Description: Specifies which pixels are transmitted. While all pixels are acquired on every scan, this parameter determines which pixels will be transmitted out the serial port.

Note

Since most applications only require a subset of the spectrum, this mode can greatly reduce the amount of time required to transmit a spectrum while still providing all of the desired data. This mode is helpful when interfacing to PLCs or other processing equipment.

Command Syntax:	P{DATA WORD}	
Value:	<p>Description</p> <p>0 = all 2048 pixels</p> <p>1 = every nth pixel with no averaging</p> <p>2 = N/A</p> <p>3 = pixel x through y every n pixels</p> <p>4 = up to 10 randomly selected pixels between 0 and 2047 (denoted p1, p2, ... p10)</p>	<p>Example</p> <p>P 0 (spaces for clarity only)</p> <p>P 1<Enter></p> <p>N<Enter></p> <p>P 2 N/A</p> <p>P3<Enter></p> <p>x<Enter></p> <p>y<Enter></p> <p>n<Enter></p> <p>P 4<Enter></p> <p>n<Enter></p> <p>p1<Enter></p> <p>p2<Enter></p> <p>p3<Enter> ...</p> <p>p10<Enter></p>
Response:	ACK or NAK	
Default value:	0	

Spectral Acquisition

Description: Acquires spectra with the current set of operating parameters. When executed, this command determines the amount of memory required. If sufficient memory does not exist, an ETX (ASCII 3) is immediately returned and no spectra are acquired. An STX (ASCII 2) is sent once the data is acquired and stored. If the Data Storage Mode value is 0, then the data is transmitted immediately.

Command Syntax:	S
Response:	If successful, STX followed by data If unsuccessful, ETX

The format of returned spectra includes a header to indicate scan number, channel number, pixel mode, etc. The format is as follows:

- WORD 0xFFFF – start of spectrum
- WORD channel number ALWAYS 0
- WORD scan number ALWAYS 0
- WORD scans in memory ALWAYS 0
- WORD integration time in milliseconds
- WORD integration time counter ALWAYS 0
- WORD pixel mode
- WORDS if pixel mode not 0, indicates parameters passed to the Pixel Mode command (P)
- WORDS spectral data
- WORD 0xFFFD – end of spectrum

Trigger Mode

Description: Sets the USB2000's external trigger mode to the value specified.

Command Syntax:	T{DATA WORD}
Value:	0 = Normal – Continuously scanning 1 = Software trigger 3 = External Hardware Trigger
Response:	ACK or NAK
Default value:	0

Number of Scans in Memory

Description: Returns the number of scans in spectral data memory.

Command Syntax:	W{DATA WORD}
Value:	1 = Return number of scans in fast memory
Response:	ACK or NAK
Default value	N/A

Read Out Scan from Memory

Description: Reads out one scan from the type of memory specified. The data is returned with the header information as described in the Spectral Acquisition command (S).

Command Syntax:	Z{DATA WORD}
Value:	1 = Read scan from spectral memory
Response:	If successful, ACK followed by data (See Spectral Acquisition [S] for header information). If unsuccessful, NAK.
Default value:	N/A

ASCII Data Mode

Description: Sets the mode in which data values are interpreted to be ASCII. Only unsigned integer values (0 – 65535) are allowed in this mode and the data values are terminated with a carriage return (ASCII 13) or linefeed (ASCII 10). In this mode the SAD500 “echoes” the command and data values back out the RS-232 port.

Notes

The command requires that the string “aA” be sent without any CR or LF. This is an attempt to insure that this mode is not entered inadvertently.

A legible response to the Version number query (v command) indicates the SAD500 is in the ASCII data mode.

Command Syntax:	aA
Response:	ACK or NAK
Default value	N/A

Binary Data Mode

Description: Sets the mode in which data values are interpreted to be binary. Only 16 bit unsigned integer values (0 – 65535) are allowed in this mode with the MSB followed by the LSB.

Note

The command requires that the string “bB” be sent without any CR or LF. This is an attempt to insure that this mode is not entered inadvertently.

Command Syntax:	bB
Response:	ACK or NAK
Default value	Default at power up – not changed by Q command

Checksum Mode

Specifies whether the USB2000 will generate and transmit a 16-bit checksum of the spectral data. This checksum can be used to test the validity of the spectral data, and its use is recommended when reliable data scans are required. See [Technical Note 2: USB2000 Checksum Calculation](#) for more information on checksum calculation.

Command Syntax:	k{DATA WORD}
Value:	0 = Do not transmit checksum value !0 = transmit checksum value at end of scan
Response:	ACK or NAK
Default value:	0

Version Number Query

Returns the version number of the code running on the microcontroller. A returned value of 1000 is interpreted as 1.00.0.

Command Syntax:	v
Response:	ACK followed by {DATA WORD}
Default value	N/A

Calibration Constants

Writes one of the 16 possible calibration constant to EEPROM. The calibration constant is specified by the first DATA WORD that follows the x. The calibration constant is stored as an ASCII string with a max length of 15 characters.

Command Syntax:	x{DATA WORD}{ASCII STRING}
Value:	<p>DATA WORD Index description</p> <p>0 – Serial Number</p> <p>1 – 0th order Wavelength Calibration Coefficient</p> <p>2 – 1st order Wavelength Calibration Coefficient</p> <p>3 – 2nd order Wavelength Calibration Coefficient</p> <p>4 – 3rd order Wavelength Calibration Coefficient</p> <p>5 – Stray light constant</p> <p>6 – 0th order non-linearity correction coefficient</p> <p>7 – 1st order non-linearity correction coefficient</p> <p>8 – 2nd order non-linearity correction coefficient</p> <p>9 – 3rd order non-linearity correction coefficient</p> <p>10 – 4th order non-linearity correction coefficient</p> <p>11 – 5th order non-linearity correction coefficient</p> <p>12 – 6th order non-linearity correction coefficient</p> <p>13 – 7th order non-linearity correction coefficient</p> <p>14 – Polynomial order of non-linearity calibration</p> <p>15 – Optical bench configuration: gg fff sss gg – Grating #, fff – filter wavelength, sss – slit size</p> <p>16 – USB2000 configuration: AWL V A – Array coating Mfg, W – Array wavelength (VIS, UV, OFLV), L – L2 lens installed, V – CPLD Version</p> <p>17 – Reserved</p> <p>18 – Reserved</p> <p>19 – Reserved</p>
Response:	ACK or NAK
Default value:	N/A

To query the constants, use the ?x{DATA WORD} format to specify the desired constant

16-bit Timer Operation

As described above, there is one 16-bit timer and one 8-bit timer that are capable of being multiplexed for either the integration time counter or the baud rate generator. This command sets that operation. This command re-initializes the baud rate to 9600, the Trigger mode to 0, and Lamp Enable to 0. Typically it is issued at the beginning of a session. When setting the integration timer to a 8 bit value, the LSB of the current integration time is used.

Command Syntax:	y{DATA WORD}
Value:	<p>0 – 16 bit timer for Integration clock, 8 bit timer for baud rate generator</p> <p>!0 - 8 bit timer for Integration clock, 16 bit timer for baud rate generator</p>
Response:	ACK or NAK
Default value	1

Query Variable

Returns the current value of the parameter specified. The syntax of this command requires two ASCII characters. The second ASCII character corresponds to the command character which sets the parameter of interest (acceptable values are B, A, I, K, T, J, y). A special case of this command is ?x (lower case) which requires an additional data word be passed to indicate which calibration constant is to be queried.

Note

There is a special query “?+” which returns a code describing the accessories plugged into the USB2000. See the [Read Plugged-in Accessories](#) command for more information.

Command Syntax:	?{ASCII character}
Response:	ACK followed by {DATA WORD}
Default value:	N/A

Read Plugged-in Accessories

This command causes the microcontroller to poll addresses 2 – 7 on the I²C bus and determines what Ocean Optics Compatible accessories are plugged in. This command is executed upon power up. The results are queried by sending the “?+” command and are decoded as described below.

Command Syntax:	+
Response:	ACK followed by {DATA WORD}
Default value:	N/A

The Query Plugged-in Accessories command (?+) returns a ACK followed by 6 data values (ASCII or binary depending upon the data format). These six values correspond to Ocean Optics compatible devices that responded appropriately for I²C addresses 2 through 7 respectively. The I²C addresses are reserved for various categories of devices and the value for each category is shown below. I²C addresses 0-1 are reserved for loading program code from EEPROMS.

Value	I ² C Addresses					
	2	3	4	5	6	7
	Light Sources & Sample Chambers	Expansion (A/D, D/A, temperature)	TBD	TBD	Power Sources	Customer Specific
1	1cm Vis/NIR		Battery Pack (Low voltage)		Battery Pack	
2	Vis/NIR					
3	LS450					
...						
255						

USB2000 Identifier

Description: Device drivers can differentiate between a USB2000 and SAD500 with the use of this command. The USB2000 will generate an ACK in response to this command while the SAD500 will respond with a NAK.

Command Syntax:	_
Response:	ACK
Default value	N/A

Examples

Below are examples on how to use some of the commands. Commands are in **BOLD** and descriptions are in parentheses. For clarity, the commands are shown in the ASCII mode (a command) instead of the default binary mode.

The desired operating conditions are: acquire spectra from spectrometer channel 0 (master) with a 200ms integration time, set number of scan to add to 5 and operate at 57,600 Baud.

- aA** (Set ASCII Data Mode)
- y1<CR>** (Allocate timers to allow faster baud rate)
- K5<CR>** (Start baud rate change to 57,600)
Wait for ACK, change to 57600, wait for 50ms
- K5<CR>** (Verify command, communicate at 57600)
- A5<CR>** (Add 5 spectra)
- I200<CR>** (Set integration time to 200ms)
- S** (Acquire spectra)
- ... Repeat as necessary

Application Tips

During the software development phase of a project, the operating parameters of the USB2000 may become out-of-synch with the controlling program. It is good practice to cycle power on the USB2000 when errors occur.

If you question the state of the USB2000, you can transmit a space (or another non-command) using a terminal emulator. If you receive a NAK, the USB2000 is awaiting a command; otherwise, it is still completing the previous command.

For Windows 95/98 users, use HyperTerminal as a terminal emulator after selecting the following:

1. Select **File | Properties**.
2. Under **Connect using**, select **Direct to Com x**.
3. Click **Configure** and match the following Port Settings:
 - Bits per second (Baud rate): Set to desired rate
 - Data bits: 8
 - Parity: None
 - Stop bits: 1
 - Flow control: None
4. Click the **Advanced** button and slide the **Receive Buffer** and **Transmit Buffer** arrows to the Low 1 value. Click **OK**.
5. Click **OK** in **Port Settings** and in **Properties** dialog boxes.

Appendix A: USB-LS-450 Specific Commands

USB Command Summary

Read Temperature

Reads the next A/D conversion from the A/D converter and returns the raw A/D value. The A/D converter requires ~70ms to perform a conversion thus this command should only be sent once every 70ms. The values that is returned should be used in the following equation to obtain temperature:

$$\text{Temperature (}^\circ\text{C)} = \text{Temperature Calibration Intercept} + (\text{Reading} * \text{Temperature Calibration Slope})$$

Where the Temperature Calibration values are obtained from the Query Calibration Constant Command.

Byte Format

Byte 0
0x20

Return Format

The data is returned in ASCII format and read in by the host through End Point 7.

Byte 0	Byte 1	Byte 2	Byte 3
A/D Byte 0	A/D Byte 1	A/D Byte 2	A/D Byte 3

The format for the 32 bit A/D reading is

Bits	31-29	28	27-0
Description	Always 0	Over-range 0 = 1 1 = Out of Range	28 Bit A/D Value. Lower 4 bits are noise

Set LED Mode

Sets the LED to either CW or pulsed operation. The Pulse rate of the LED is determined by the Continuous Strobe signal from the USB2000. Typically this rate is 976Hz. The LED is also Enabled/Disabled by the Lamp/Strobe Enable command/signal.

Byte Format

Byte 0	Byte 1
0x21	0 = CW Mode !0 = Pulsed Mode

Query USB-LS450 Calibration Constant

Queries any of the 22 stored spectrometer configuration variables. The Query command is sent to EP2 and the data is retrieved through End Point 7. The 22 configuration variables are indexed as follows:

DATA WORD	Index	Description
DATA WORD Index description		
0		Serial Number
1		1 st order Temperature Calibration Coefficient
2		0 th order Temperature Calibration Coefficient
3		Oxygen Probe Cal Coefficient: 0 th order I_0
4		Oxygen Probe Cal Coefficient: 1 st order I_0
5		Oxygen Probe Cal Coefficient: 2 nd order I_0
6		Oxygen Probe Cal Coefficient: 3 rd order I_0
7		Oxygen Probe Cal Coefficient: 0 th order K_1
8		Oxygen Probe Cal Coefficient: 1 st order K_1
9		Oxygen Probe Cal Coefficient: 2 nd order K_1
10		Oxygen Probe Cal Coefficient: 3 rd order K_1
11		Oxygen Probe Cal Coefficient: 0 th order K_2
12		Oxygen Probe Cal Coefficient: 1 st order K_2
13		Oxygen Probe Cal Coefficient: 2 nd order K_2
14		Oxygen Probe Cal Coefficient: 3 rd order K_2
15		Oxygen Probe Cal Coefficient: 0 th order K_2
16		Oxygen Probe Cal Coefficient: 0 th order K_2
17		Oxygen Probe Cal Coefficient Reserved
18		Oxygen Probe Cal Coefficient Reserved
19		Oxygen Probe Cal Coefficient Reserved
20		Oxygen Probe Cal Coefficient Reserved
21		Oxygen Probe Cal Coefficient Reserved

Byte Format

Byte 0	Byte 1
0x23	Data byte

Return Format

The data is returned in ASCII format and read in by the host through End Point 7.

Byte 0	Byte 1	Byte 2	Byte 3	...	Byte 17
0x23	Configuration Index	ASCII Byte 0	ASCII Byte 1	...	ASCII Byte 15

Set Analog Output

USB-LS450s can be ordered with an optional 4-20mA analog output. This command sets the analog output. The analog output value is a 16-bit value that is linearly scaled between 4 and 20mA.

Byte Format

Byte 0	Byte 1	Byte 1
0x25	Analog Output Value LSB	Analog Output Value MSB

Read USB-LS450 Constants From EEPROM

This command instructs the microcontroller to read the calibration coefficients from the USBLS450 and store them in internal volatile memory.

Byte Format

Byte 0
0x26

Write USB-LS450 Constants To EEPROM

This command instructs the microcontroller to write the calibration coefficients to the EEPROM on the USB-LS450. This command requires 3.5 seconds to complete, during this time no other commands should be sent to the USB2000.

Byte Format

Byte 0
0x27

RS232 Command Summary

There are programmable features in the USB-LS450 that are accessed through the USB2000. The commands for these features are documented below.

Command Summary

Letter	Description	Version
A	Set Analog Output (optional hardware feature)	2.00.1
j	Set LED Operational Mode	2.00.1
t	Read Temperature	2.00.1
u	Set USB-LS-450 Calibration Coefficient	2.00.1

?	Query variable	2.00.1
Note: all commands are preceded by an 'o'		

Set Analog Output

Description: Sets the analog output to a value between 0 and 20 mA. The value transmitted is a 16 bit unsigned integer that is scaled to the 0-20mA output.

Command Syntax:	oA{DATA WORD}
Response:	ACK or NAK
Range:	
Default value:	0

Set LED Operational Mode

Description: Sets the operational mode of the LED to either CW or pulsed. The pulse rate is set by the USB2000 and is typically 976Hz.

Note

LED is also enabled/disabled by the Strobe On/Off command (J command).

Command Syntax:	oj{DATA WORD}
Response:	ACK or NAK
Range:	0 – LED is on CW !0 – LED is Pulsed
Default value:	0

Set USB-LS-450 Calibration Constants

Description: Writes one of the 21 possible calibration constant to the USB-LS-450's EEPROM. The calibration constant is specified by the first DATA WORD that follows the u. The calibration constant is stored as an ASCII string with a max length of 15 characters. The string is not check to see if it makes sense. To query the constants, use the o?u{DATA WORD} format to specify the desired constant.

Command Syntax:	ou{DATA WORD}{ASCII STRING}
-----------------	-----------------------------

Value:	<p>DATA WORD Index description</p> <p>0 – Serial Number</p> <p>1 – 1st order Temperature Calibration Coefficient</p> <p>2 – 0th order Temperature Calibration Coefficient</p> <p>3 – Oxygen Probe Cal Coefficient: I₀ No Temperature Correction</p> <p>4 – Oxygen Probe Cal Coefficient: 0th order I₀ with Temp Correction</p> <p>5 – Oxygen Probe Cal Coefficient: 1st order I₀ with Temp Correction</p> <p>6 – Oxygen Probe Cal Coefficient: 2nd order I₀ with Temp Correction</p> <p>7 – Oxygen Probe Cal Coefficient: K No Temperature Correction</p> <p>8 – Oxygen Probe Cal Coefficient: 0th order K with Temp Correction</p> <p>9 – Oxygen Probe Cal Coefficient: 1st order K with Temp Correction</p> <p>10 – Oxygen Probe Cal Coefficient: 2nd order K with Temp Correction</p> <p>11 – Oxygen Probe Cal Coefficient: K₁ No Temperature Correction</p> <p>12 – Oxygen Probe Cal Coefficient: 0th order K₁ with Temp Correction</p> <p>13 – Oxygen Probe Cal Coefficient: 1st order K₁ with Temp Correction</p> <p>14 – Oxygen Probe Cal Coefficient: 2nd order K₁ with Temp Correction</p> <p>15 – Oxygen Probe Cal Coefficient: K₂ No Temperature Correction</p> <p>16 – Oxygen Probe Cal Coefficient: 0th order K₂ with Temp Correction</p> <p>17 – Oxygen Probe Cal Coefficient: 1st order K₂ with Temp Correction</p> <p>18 – Oxygen Probe Cal Coefficient: 2nd order K₂ with Temp Correction</p> <p>19 – Oxygen Probe Cal Coefficient Reserved</p> <p>20 – Oxygen Probe Cal Coefficient Reserved</p>
Response:	ACK or NAK
Default value:	N/A

Read Temperature

Description: Command reads the A/D converter in the USB-LS450 and calculates the temperature in Degrees C. The calculation is performed by reading the temperature calibration factors (ou command) at startup.

Command Syntax:	ot
Value:	N/A
Response:	ACK {Temperature in Degrees C-Always an ASCII string}
Default value	N/A

Query Variable

Description: Returns the current value of the USB-LS450 parameter specified. The syntax of this command requires three ASCII characters. The third ASCII character corresponds to the command character which sets the parameter of interest (acceptable values are A, j & u).

Command Syntax:	o?{ASCII character}
Response:	ACK followed by {DATA WORD}
Default value:	N/A

Appendix B: USB-ISS-UV/VIS Specific Commands

There are programmable features in the USB-ISS-UV/VIS that are accessed through the USB2000. The commands for these features are documented below.

USB Command Summary

The USB-ISS-UV/VIS contains 4 digitally controlled 6-bit potentiometers. The functionality of the potentiometers is as follows:

Pot #	Description
0	Controls the intensity of the Visible lamp 0x0=low (not off), 0x3f=high
1	Controls the intensity of the D2 lamp 0x0=low (not off), 0x3f=high
2	Enable/Disable the Visible Power - This is the only way to turn the visible lamp completely off 0x0=On, 0x3f= off (always one or the other)
3	Enable/Disable the Both Lamps - This is the only way to turn the both lamps completely off. It also Enables/Disables the Fan. The Potentiometer values are retained in this state. 0x0=On, 0x3f= off (always one or the other)

Write Potentiometer Value

Sets the specified Potentiometer to the desired value.

Byte Format

Byte 0	Byte 1	Byte 2
0x40	Potentiometer Number	Potentiometer Value

Set Power Up Potentiometer Values

Transfers the current potentiometer values to nonvolatile memory for recall upon power up.

Byte Format

Byte 0	Byte 1	Byte 2
0x41	0x01	0x80

Read Potentiometer Values

Returns the values of all 4 digitally controlled Potentiometers.

Byte Format

Byte 0
0x42

Return Format

Byte 0	Byte 1	Byte 2	Byte 3
Pot 0 Value	Pot 2 Value	Pot 3 Value	Pot 3 Value

RS232 Command Summary

Command Summary

Letter	Description	Version
p	Set Potentiometer value	2.40.0
C	Store values to non-volatile memory	2.40.0
?	Query variable	2.40.0
Note: all commands are preceded by an 'm'		

Set Potentiometer Value

Description: Sets one of four potentiometers to the desired value. The function of each 6-bit potentiometer is given in the table below.

Potentiometer #	Function	Values
0	Sets the intensity of the Visible lamp	0x00 = lowest 0x3f = highest
1	Sets the intensity of the D2 lamp	0x00 = lowest 0x3f = highest
2	Enable/Disable the Visible Lamp. This setting should be either 0x00 or 0x3f since it implements a digital function	0x00 = Visible Lamp On 0x3f = Visible Lamp off
3	Enable/Disable the power supply of both lamps This setting should be either 0x00	0x00 = Module Powered 0x3f = Module Shutdown

	or 0x3f since it implements a digital function	
--	--	--

Command Syntax:	mp{DATA WORD}{DATA WORD}
Value:	This command requires 2 DATA WORDS. The first DATA WORD is the pot number and the second DATA WORD is the setting.
Range:	Pot Number is 0 – 3 and the setting is 0x00 – 0x3f
Response:	ACK or NAK
Default value:	N/A

Store Potentiometer Values to Non-volatile Memory

Description: This command writes the current Potentiometer values to non-volatile memory. These are the values that are used at power-up of the device.

Command Syntax:	mC
Value:	N/A
Response:	ACK or NAK
Default value	N/A

Query Variable

Description: Returns the current value of the USB-ISS-UV/VIS parameter specified. The syntax of this command requires three ASCII characters and a DATA VALUE.

Command Syntax:	o?{ASCII character p} {DATA WORD}
Response:	ACK followed by {DATA WORD}
Default value:	N/A

Technical Note 1: USB2000 Data Compression

Transmission of spectral data over the serial port is a relatively slow process. Even at 57,600 baud, the transmission of a complete 2048-point spectrum takes around 600 msec. The USB2000 implements a data compression routine that minimizes the amount of data that needs to be transferred over the RS-232 connection. Using the “G” command (Compressed Mode) and passing it a parameter of 1 enables the data compression. Every scan transmitted by the USB2000 will then be compressed. The compression algorithm is as follows:

1. The first pixel (a 16-bit unsigned integer) is always transmitted uncompressed.
2. The next byte is compared to 0x80.
 - If the byte is equal to 0x80, the next two bytes are taken as the pixel value (16-bit **unsigned** integer).
 - If the byte is not equal to 0x80, the value of this byte is taken as the difference in intensity from the previous pixel. This difference is interpreted as an 8-bit **signed** integer.
3. Repeat step 2 until all pixels have been read.

Using this data compression algorithm greatly increases the data transfer speed of the SAD500. The table below shows the data transfer speed, in milliseconds, for various light sources and baud rates. Keep in mind that these rates are for demonstration purposes only, and the speed of your computer may impact the data transfer rates.

	Comp	57.6 kb	% faster	38.4 kb	% faster	19.2 kb	% faster	9.6 kb	% faster
dark	on	426	45.2%	624	46.7%	1148	47.5%	2247	48.8%
	off	778		1170		2188		4391	
LS-1	on	429	44.9%	624	46.6%	1141	49.6%	2192	50.1%
	off	779		1169		2266		4390	
HG-1	on	465	40.2%	679	41.9%	1238	43.5%	2424	44.8%
	off	777		1169		2193		4391	

The following table shows a section of a spectral line source spectrum and the results of the data compression algorithm. In this example, spectral data for 40 pixels is transmitted using only 60 bytes. If the same data set were transmitted using uncompressed data, it would require 80 bytes.

Pixel Value	Value Difference	Transmitted Bytes
185	0	0x80 0x00 0xB9
2151	1966	0x80 0x08 0x67
836	-1315	0x80 0x03 0x44
453	-383	0x80 0x01 0xC5

Pixel Value	Value Difference	Transmitted Bytes
210	-243	0x80 0x00 0xD2
118	-92	0xA4
90	-28	0xE4
89	-1	0xFF
87	-2	0xFE
89	2	0x02
86	-3	0xFD
88	2	0x02
98	10	0x0A
121	23	0x17
383	262	0x80 0x01 0x7F
1162	779	0x80 0x04 0x8A
634	-528	0x80 0x02 0x7A
356	-278	0x80 0x01 0x64
211	-145	0x80 0x00 0xD3
132	-79	0xB1
88	-44	0xD4
83	-5	0xFB
86	3	0x03
82	-4	0xFC
91	9	0x09
92	1	0x01
81	-11	0xF5
80	-1	0xFF
84	4	0x04
84	0	0x00
85	1	0x01
83	-2	0xFE
80	-3	0xFD
80	0	0x00
88	8	0x08
94	6	0x06
90	-4	0xFC
103	13	0x0D
111	8	0x08
138	27	0x1B

Technical Note 2: USB2000 Checksum Calculation

For all uncompressed pixel modes, the checksum is simply the unsigned 16-bit sum (ignoring overflows) of all transmitted spectral points. For example, if the following 10 pixels are transferred, the calculation of the checksum would be as follows:

Pixel Number	Data (decimal)	Data (hex)
0	15	0x000F
1	23	0x0017
2	46	0x002E
3	98	0x0062
4	231	0x00E7
5	509	0x01FD
6	1023	0x03FF
7	2432	0x0980
8	3245	0x0CAD
9	1984	0x07C0

Checksum value: 0x2586

When using a data compression mode, the checksum becomes a bit more complicated. A compressed pixel is treated as a 16-bit **unsigned** integer, with the most significant byte set to 0. Using the same data set used in [Technical Note 1: USB2000 Data Compression](#), the following shows a section of a spectral line source spectrum and the results of the data compression algorithm.

Data Value	Value Difference	Transmitted Bytes	Value Added to Checksum
185	0	0x80 0x00 0xB9	0x0139
2151	1966	0x80 0x08 0x67	0x08E7
836	-1315	0x80 0x03 0x44	0x03C4
453	-383	0x80 0x01 0xC5	0x0245
210	-243	0x80 0x00 0xD2	0x0152
118	-92	0xA4	0x00A4
90	-28	0xE4	0x00E4
89	-1	0xFF	0x00FF
87	-2	0xFE	0x00FE
89	2	0x02	0x0002
86	-3	0xFD	0x00FD
88	2	0x02	0x0002
98	10	0x0A	0x000A
121	23	0x17	0x0017

Data Value	Value Difference	Transmitted Bytes	Value Added to Checksum
383	262	0x80 0x01 0x7F	0x01FF
1162	779	0x80 0x04 0x8A	0x050A
634	-528	0x80 0x02 0x7A	0x02FA
356	-278	0x80 0x01 0x64	0x01E4
211	-145	0x80 0x00 0xD3	0x0153
132	-79	0xB1	0x00B1
88	-44	0xD4	0x00D4
83	-5	0xFB	0x00FB
86	3	0x03	0x0003
82	-4	0xFC	0x00FC
91	9	0x09	0x0009
92	1	0x01	0x0001
81	-11	0xF5	0x00F5
80	-1	0xFF	0x00FF
84	4	0x04	0x0004
84	0	0x00	0x0000
85	1	0x01	0x0001
83	-2	0xFE	0x00FE
80	-3	0xFD	0x00FD
80	0	0x00	0x0000
88	8	0x08	0x0008
94	6	0x06	0x0006
90	-4	0xFC	0x00FC
103	13	0x0D	0x000D
111	8	0x08	0x0008
138	27	0x1B	0x001B

The checksum value is simply the sum of all entries in the last column, and evaluates to 0x2C13.