830 Douglas Ave.
Dunedin, FL 34698
(727)733-2447
Fax:(727)733-3962
www.OceanOptics.com

A HALMA GROUP COMPANY

USB4000 Data Sheet

Description

The Ocean Optics USB4000 Spectrometer is designed from the USB2000 Spectrometer to include an advanced detector and powerful high-speed electronics to provide both an unusually high spectral response and high optical resolution in a single package. The result is a compact, flexible system, with no moving parts, that's easily integrated as an OEM component.

The USB4000 features a 16-bit A/D with autonulling (an enhanced electrical dark signal correction), 4 total triggering options, a dark-level correction during temperature changes, and a 22-pin connector with 8 user-programmable GPIOs. The modular USB4000 is responsive from 200-1100 nm and can be configured with various Ocean Optics optical bench accessories, light sources and sampling optics, to create application-specific systems for thousands of absorbance, reflection and emission applications.

The USB4000 interfaces to a computer via USB 2.0 or RS-232 communications. Data unique to each spectrometer is programmed into a memory chip on the USB4000; SpectraSuite Spectroscopy Cross-platform Operating Software reads these values for easy setup and hot swapping among computers, whether they run on Linux, Mac or Windows operating systems. The USB4000 operates from the +5V power, provided through the USB, or from a separate power supply and an RS-232 interface.

The detector used in the USB4000 spectrometer is a high-sensitivity 3648-element CCD array from Toshiba, product number TCD1304AP. (For complete details on this detector, visit Toshiba's web site at **www.toshiba.com**. Ocean Optics applies a coating to all TCD1304AP detectors, so the optical sensitivity could vary from that specified in the Toshiba datasheet).

The USB4000 operates off of a single +5VDC supply and either a USB or RS-232 interface. It has a 22-pin external interface to easily integrate with Ocean Optics' other modular components for an entire system.

Features

- TCD1304AP Detector
  - High-sensitivity detector
  - Readout rate: 1MHz
  - Shutter mode
- Responsive from 200 to 1100 nm, specific range and resolution depends on your grating and entrance slit choices
- An optical resolution of ~1.5 nm (FWHM)
- A wide variety of optics available
  - 14 gratings
  - 6 slit widths
  - 3 detector coatings
  - 6 optical filters
- Integration times from 10 µs* to 10 seconds
- 16-bit, 3MHz A/D Converter
- Embedded microcontroller allows programmatic control of all operating parameters and standalone operation
  - USB 2.0 480Mbps (high speed) and 12Mbps (full speed)
  - RS232 115K baud
  - Multiple communication standards for digital accessories (SPI, I^2C)
- Onboard Pulse Generator
  - 2 programmable strobe signals for triggering other devices
  - Software control of nearly all pulse parameters
  - Onboard GPIO – 8 user-programmable digital I/O
- EEPROM storage for
  - Wavelength Calibration Coefficients
  - Linearity Correction Coefficients
  - Absolute Irradiance Calibration (optional)
- Low power consumption of only 250 mA @ 5 VDC
- 4 triggering modes
- 24-pin connector for interfacing to external products
- Programmable for standalone operation
- CE Certification

*10µs to 3.79ms integration times require the use of Shutter Mode.

Specifications

Specifications	Criteria
Absolute Maximum Ratings:	
V_{CC}	+ 5.5 VDC
Voltage on any pin	Vcc
Physical Specifications:	
Physical Dimensions	89.1 mm x 63.3 mm x 34.4 mm
Weight	190 g
Power:	
Power requirement (master)	230 mA at +5 VDC
Supply voltage	4.5 – 5.5 V
Power-up time	~5s depending on code size
Spectrometer:	
Design	Asymmetric crossed Czerny-Turner
Focal length (input)	42mm
Focal length (output)	68mm (75, 83, and 90mm focal lengths are also available)
Input Fiber Connector	SMA 905
Gratings	14 different gratings
Entrance Slit	5, 10, 25, 50, 100, or 200 µm slits. (Slits are optional. In the absence of a slit, the fiber acts as the entrance slit.)
Detector	Toshiba TCD1304AP linear CCD array
Filters	2^{nd} and 3^{rd} order rejection, long pass (optional)
Spectroscopic:	
Integration Time	10µs – 10 seconds
Dynamic Range	3.4×10^6 (system); 1300:1 for a single acquisition
Signal-to-Noise	300:1 (at full signal)
Dark Noise	50 counts RMS
Resolution (FWHM)	~1.5 nm
Stray Light	<0.05% at 600 nm; <0.10% at 435 nm
Spectrometer Channels	One
Environmental Conditions:	
Temperature	-30° to +70° C Storage & -10° to +50° C Operation
Humidity	0% - 90% noncondensing
Interfaces:	
USB	USB 2.0, 480 Mbps
RS-232	2-wire RS-232

Mechanical Diagram

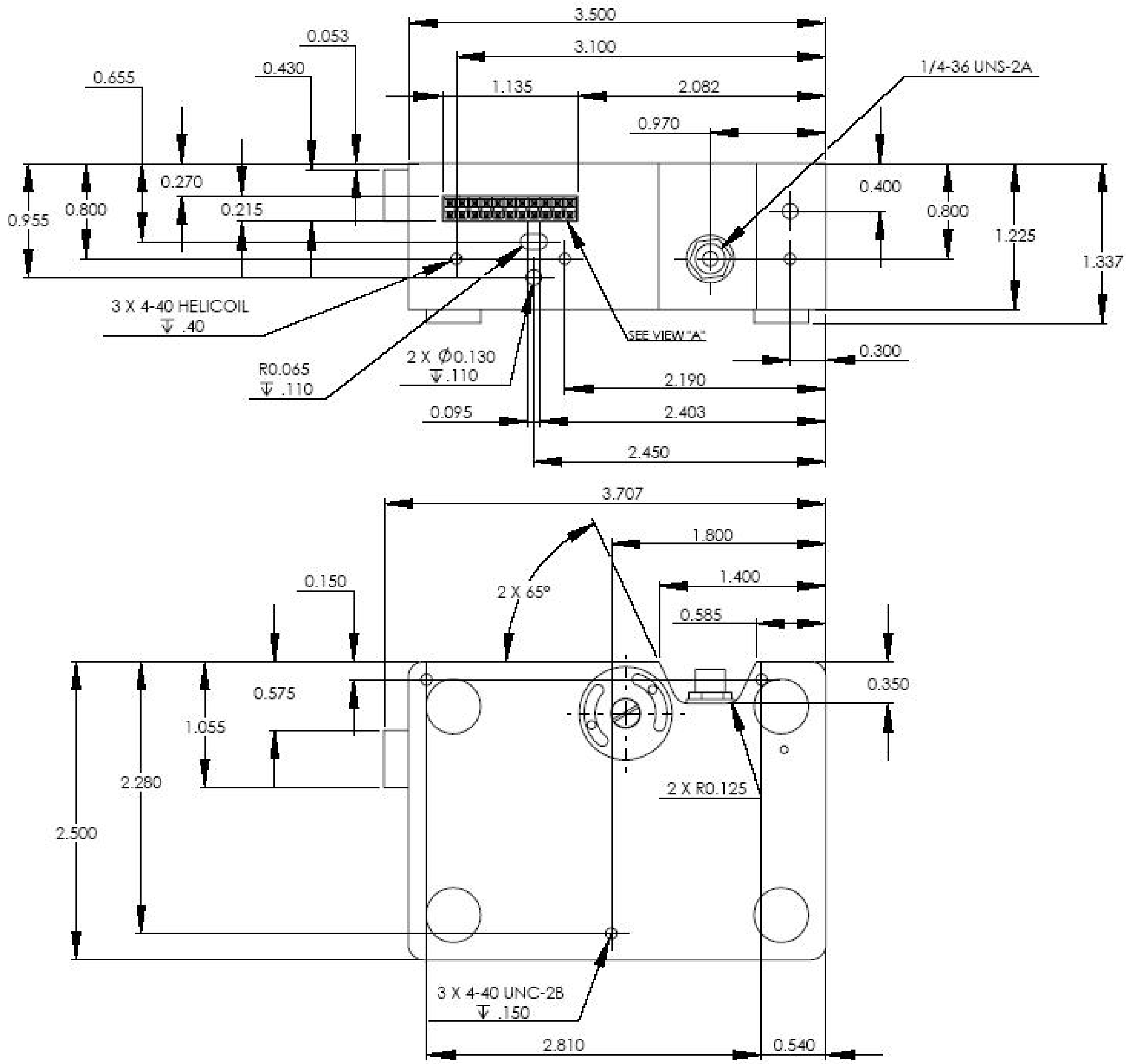
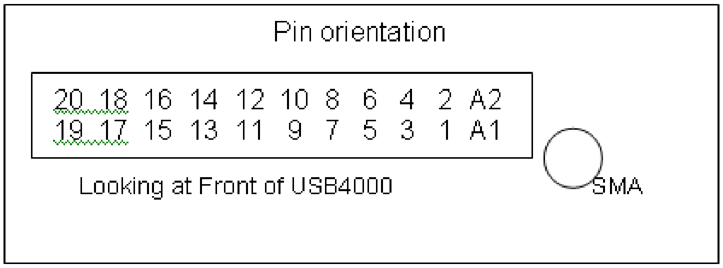


Figure 1: USB4000 Outer Dimensions

Electrical Pinout

Listed below is the pin description for the USB4000 Accessory Connector located on the front vertical wall of the unit. The connector is a Samtec part # IPT1-111-01-S-D-RA connector. The vertical mate to this is part #IPS1-111-01-S-D-VS and the right angle PCB mount is part #IPS1-111-01-S-D-RA.



Pin #	Function	Input/Output	Description
1	V_{CC}, V_{USB}, or $5V_{IN}$	Input or Output	Input power pin for USB4000 – When operating via USB, this pin can power other peripherals – Ensure that peripherals comply with USB specifications
2	RS232 Tx	Output	RS232 transmit signal – Communicates with a computer over DB9 Pin 2
3	RS232 Rx	Input	RS232 receive signal – Communicates with a computer over DB9 Pin 3
4	Lamp Enable	Output	TTL signal driven Active HIGH when the Lamp Enable command is sent to the spectrometer
5	Continuous Strobe	Output	TTL output signal used to pulse a strobe – Divided down from the master clock signal
6	Ground	Input/Output	Ground
7	External Trigger In	Input	TTL input trigger signal – See External Triggering Options document for info
8	Single Strobe	Output	TTL output pulse used as a strobe signal – Has a programmable delay relative to the beginning of the spectrometer integration period
9	I^2C SCL	Input/Output	The I^2C clock signal for communications to other I^2C peripherals.
10	I^2C SDA	Input/Output	The I^2C Data signal for communications to other I^2C peripherals.
11	MOSI	Output	SPI Master Out Slave In (MOSI) signal for communication to other SPI peripherals
12	MISO	Input	SPI Master In Slave Out (MISO) signal for communication to other SPI peripherals
13	GPIO-1(1P)*	Input/Output	General purpose software-programmable, digital input/output (channel number)

Pin #	Function	Input/Output	Description
14	GPIO-0(2P)*	Input/Output	General purpose software-programmable, digital input/output (channel number)
15	GPIO-3(1N)*	Input/Output	General purpose software-programmable, digital input/output (channel number)
16	GPIO-2(2N)*	Input/Output	General purpose software-programmable, digital input/output (channel number)
17	GPIO-5(3P)*	Input/Output	General purpose software-programmable, digital input/output (channel number)
18	GPIO-4(4P)*	Input/Output	General purpose software-programmable, digital input/output (channel number)
19	GPIO-7(3N)*	Input/Output	General purpose software-programmable, digital input/output (channel number)
20	GPIO-6(4N)*	Input/Output	General purpose software-programmable, digital input/output (channel number)
A1	SPI_CLK	Output	SPI clock signal for communication to other SPI peripherals
A2	SPICS OUT	Output	The SPI Chip/Device Select signal for communications to other SPI peripherals
NOTE: GPIO *n*P and *n*N are for future LVDS capability			

CCD Overview

CCD Detector

The detector used for the USB4000 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 USB4000 invert and amplify this electrical signal.

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 ~12µs. The user needs to account for this time delay when the pixels are optically inactive, especially in the external triggering modes.

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 3000:1 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 USB4000 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 via USB:

Pixel	Description
1–5	Not usable
6–18	Optical black pixels
19–21	Transition pixels
22–3669	Optical active pixels
3670–3681	Not usable

In USB interface mode, Ocean Optics software displays 3648 pixels starting at pixel 1 above. In RS232 interface mode, the USB4000 transmits out the first 3670 pixels.

Timing Signals

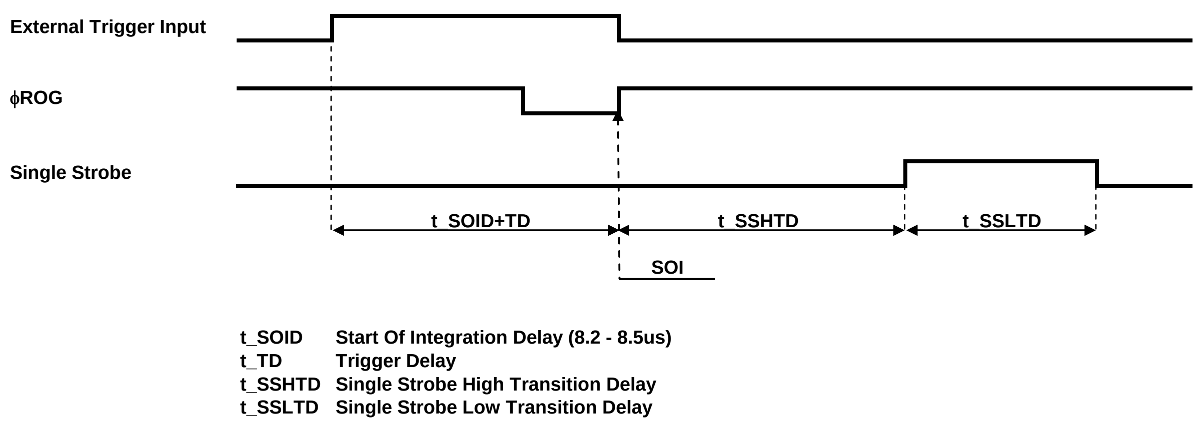
Strobe Signals

Single Strobe

The Single Strobe signal is a programmable TTL pulse that occurs at a user-determined time during each integration period. This pulse has a user-defined High Transition Delay and Low Transition Delay. The pulse width of the Single Strobe is the difference between these delays. It is only active if the Lamp Enable command is active.

Synchronization of external devices to the spectrometer's integration period is accomplished with this pulse. The Strobe Delay is specified by the Single Strobe High Transition Delay (SSHTD) and the Pulse Width is specified by the Single Strobe Low Transition Delay (SSLTD) minus the Single Strobe High Transition Delay (PW = SSLTD – SSHTD). Both values are programmable in 500ns increments for the range of 0 to 65,535 (32.7675ms).

The timing of the Single Strobe is based on the Start of Integration (SOI). SOI occurs on the rising edge of ϕROG which is used to reset the Sony ILX511 detector. In all trigger modes using an External Trigger, there is a fixed relationship between the trigger and the SOI. In the Normal mode and Software Trigger mode, the SOI still marks the beginning of the Single Strobe, but due to the nondeterministic timing of the software and computer operating system, this timing will change over time and is not periodic. That is, at a constant integration time, the Single Strobe will not be periodic, but it will indicate the start of the integration. The timing diagram for the Single Strobe in External Hardware Trigger mode is shown below:



Single Strobe (External Hardware Trigger/External Synchronous Trigger Mode)

The Trigger Delay (TD) is another user programmable delay which specifies the time in 500ns increments that the SOI will be delayed beyond the normal Start of Integration Delay (SOID).

An example calculation of the Single Strobe timing follows:

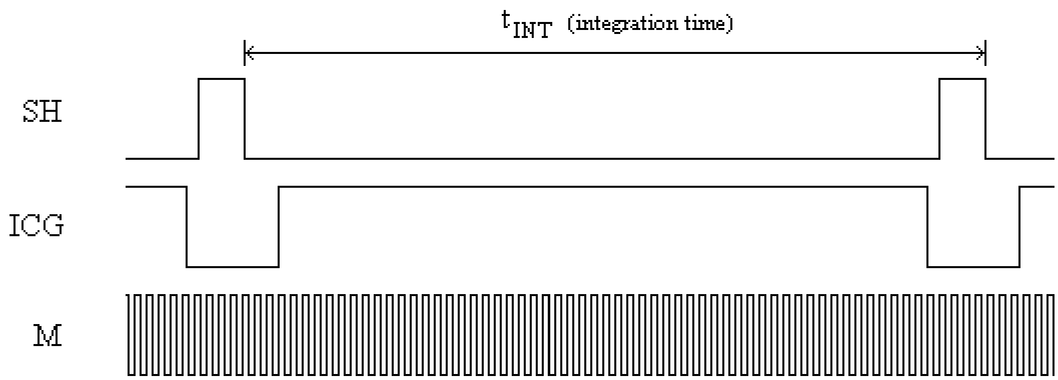
If the TD = 1ms, SSHTD = 50ms, and SSLTD = 70ms then, the rising edge of the Single Strobe will occur approximately 51.82ms (1ms + 50ms + 8.2us) after the External Trigger Input goes high and the Pulse Width will be 20ms (70ms – 50ms).

Continuous Strobe

The Continuous Strobe signal is a programmable frequency pulse-train with a 50% duty cycle. It is programmed by specifying the desired period whose range is 2µs to 60s. This signal is continuous once enabled, but is not synchronized to the Start of Integration or External Trigger Input. The Continuous Strobe is only active if the Lamp Enable command is active.

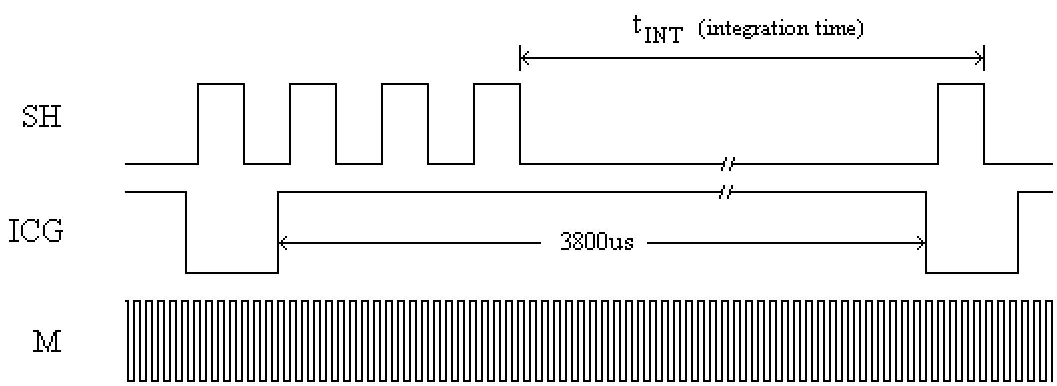
CCD Timing

The USB4000 uses the Toshiba TCD1304AP CCD detector. When synchronizing the USB4000 to an external trigger source and/or an external light source, especially if the application has strict timing requirements, it is helpful to have an understanding of how the timing of the signals to the CCD work. In addition to power and ground, the TCD1304 requires three signals for operation. These signals are the Shutter (SH), the Integration Clear Gate (ICG) and the Master Clock (M). In normal operation (i.e., when not using an external signal to trigger the spectrometer), all of these signals are provided automatically based on the user-defined integration time set in the software. In external triggering modes, these signals are also provided automatically, but are derived from the external trigger signal provided by the user. Below is a timing diagram of these signals during normal operation (no external trigger):



Non-Shutter Mode

SH defines the integration time, ICG is used to reset the detector between integration periods, and M is used to clock out the spectral data from the detector. In the diagram above, the detector is operating in "non-shutter" mode. Non-shutter mode is used whenever the integration time (tINT) is greater than or equal to 3800us. 3800us is the time it takes for the spectrometer to read out all of the data from the CCD. Integration times of less than 3800us can be achieved, though the time between integration periods must remain at least 3800us in order for the spectrometer to have enough time to read out all of the data from the CCD. This is achieved by using the "shutter-mode" feature of the TCD1304AP as shown below:



Shutter Mode

Again, these signals are provided automatically by the USB4000, but when trying to synchronize the spectrometer to an external device, understanding these signals is helpful.

Operational Modes

The USB4000 supports three free-run modes and two triggering modes of operation. They are described in the following sections. All Trigger Modes require the use of the TCD1304's Shutter Mode to minimize trigger latency. For more information on triggering modes, refer to the *External*

Triggering Options document for Firmware Versions 3.0 and above located on our website at **http://www.oceanoptics.com/technical/External-Triggering2.pdf**. The following paragraphs describe these modes. For firmware version below 3.0, see **http://www.oceanoptics.com/technical/External-Triggering.pdf.**

Also reference the Engineering Note *HR4000 and USB4000 Shutter Mode Performance in Hardware Trigger Mode* which is applicable to any operation using the TCD1304 Shutter Mode.

A detailed description of each triggering mode follows.

Normal Mode

In the Normal (Free-run) mode, the spectrometer will acquire back-to-back spectra based on the integration period specified. After the Integration Cycle completes, the data is read out of the detector and written into an internal FIFO where it is available for reading. In parallel to this read/write operation, another integration is occurring. If the data from the FIFO is completely read before the parallel integration completes, a back-to-back operation will occur. If the data is not read (FIFO Empty) in this time period, the FPGA will generate an Idle Cycle which is equivalent to one integration period and the data from the detector is discarded. After the Idle Cycle has completed, the FIFO Empty status is checked. If the FIFO is empty and a new spectrum is requested by the software, a new acquisition will begin. If either condition is false, additional Idle Cycles will be generated until both conditions are true.

Shutter Mode

Shutter Mode is always invoked when the specified integration period is less than 3800 microseconds. Back-to-back operations are not permitted in this mode because the parallel use of the shutter operation would corrupt the detector data as it is being read. Once the data is retrieved and written into the FIFO, a complete Idle Cycle is executed. As in the Normal Mode, the FIFO Empty and the spectrum request are evaluated to determine what occurs next. Due to the steps required for each acquisition, the minimum time per acquisition will be approximately three times the minimum detector cycle time (3 x 3800 us).

Normal (Shutter) Mode

Normal (Shutter) Mode is a hybrid operation which combines the Normal Mode Integration Cycle with the Detector Reset Cycle that is used in all Trigger Modes. This combination allows the spectrometer to exhibit the same behavior in a Free-Run mode as it does in the Trigger Modes. In actuality, the Normal (Shutter) Mode is the equivalent of the External Hardware Level Trigger Mode with the trigger input stuck high (logic ‘1’).

External Hardware Level Trigger Mode

In the External Hardware Level Trigger mode, a rising edge detected by the FPGA from the External Trigger input starts the Integration Cycle specified through the software interface. After the Integration Cycle completes, the spectrum is retrieved and written to the FIFO in the FPGA. As long as the trigger level remains active in a logic one state, continuous acquisitions will occur with the following exception. Each subsequent acquisition must wait until a minimum CCD Reset Cycle completes. This Reset Cycle insures that the CCD performance uniform on a scan-to-scan basis. The time duration for this reset cycle is relative to the Integration Cycle time and will change if the integration period is changed. So the timing sequence is Trigger, Trigger Delay, Integration Cycle, Read/Write Cycle, Reset Cycle, Idle Cycle(s), and Integration Cycle (if trigger is still high). The Idle Cycle will on last 2 µs if the trigger remains high and the FIFO is empty and a spectrum request is active, otherwise the Idle Cycle will continue until all 3 conditions are satisfied.

External Hardware Edge Trigger Mode

In the External Hardware Edge Trigger mode, a rising edge detected by the FPGA from the External Trigger input starts the Integration Cycle specified through the software interface. After the Integration Cycle completes, the spectrum is retrieved and written to the FIFO in the FPGA followed by a CCD Reset Cycle. Only one acquisition will be performed for each External Trigger pulse, no matter what the pulse's duration is. The Reset Cycle insures that the CCD performance uniform on a scan-to-scan basis. The time duration for this reset cycle is relative to the Integration Cycle time and will change if the integration period is changed. So the timing sequence is Trigger, Trigger Delay, Integration Cycle, Read/Write Cycle, Reset Cycle, and Idle Cycle(s). The Idle Cycle will until the next trigger occurs.

Strobe Signals

Single Strobe

The Single Strobe signal is a programmable TTL pulse that occurs at a user determined time during each integration period. This pulse has a user defined High Transition Delay and Low Transition Delay. The pulse width of the Single Strobe is the difference between these delays. It is only active if the Lamp Enable command is active.

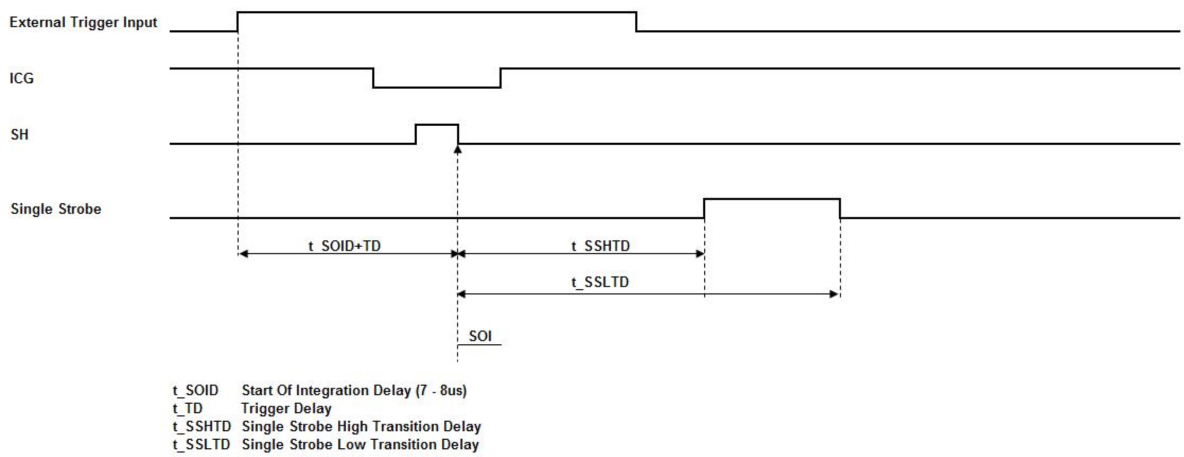
Synchronization of external devices to the spectrometer's integration period is accomplished with this pulse. The Single Strobe High Transition Delay (SSHTD) is a 16 bit value that defines the time period from the SOI until the Single Strobe pulse transitions high. The Single Strobe Low Transition Delay (SSLTD) is a 16 bit value that defines the time period from the SOI until the Single Strobe pulse transitions low. Thus the Pulse Width of the Single Strobe can be calculated using the following equation:

PW = SSLTD – SSHTD

Both delay values are programmable in 500ns increments for the range of 0 to 65,535 (32.7675ms).

The timing of the Single Strobe is based on the Start of Integration (SOI). SOI occurs on the rising edge of ϕROG which is used to reset the Sony ILX511 detector. In all Trigger Modes using an External Trigger, there is a fixed relationship between the trigger and the SOI. In the Normal Mode and Software Trigger Mode, the SOI still marks the beginning of the Single Strobe, but due to the nondeterministic timing of the software and computer Operating System, this timing will change over time and is not periodic. This is to say that at a constant integration time, the Single Strobe will not be periodic, but it will indicate the start of the integration.

The timing diagram for the Single Strobe is shown below:



Single Strobe Timing Diagram

The Trigger Delay (TD) is another user programmable delay which specifies the time in 500ns increments that the SOI will be delayed beyond the normal Start of Integration Delay (SOID).

An example calculation of the Single Strobe timing follows. If the TD = 1ms, SSHTD = 50ms, and SSLTD = 70ms then, the rising edge of the Single Strobe will occur approximately 51.82ms (1ms + 50ms + 8.2us) after the External Trigger Input goes high and the Pulse Width will be 20ms (70ms – 50ms).

Continuous Strobe

The Continuous Strobe signal is a programmable frequency pulse-train with a 50% duty cycle. It is programmed by specifying the desired period whose range is 2us to 60s. This signal is continuous once enabled, but is not synchronized to the Start of Integration or External Trigger Input. The Continuous Strobe is only active if the Lamp Enable command is active.

Digital Inputs & Outputs

General Purpose Inputs/Outputs (GPIO)

The USB4000 has 8 2.5V user-programmable digital Input/Output pins, which can be accessed at the 22-pin accessory connector. Through software, the state of these I/O pins can be defined and used for multi-purpose applications such as communications buses, sending digital values to an LCD/LED display, or even implementing complex feedback systems.

GPIO Recommended Operating Levels:

VIH(min) = 1.7V

VIL(max) = 0.7V

IOL = 24mA

IOH = -24mA

GPIO Absolute Maximum Ratings are as follows:

VIN(min) = -0.5V

VIN(max) = 3.0V

Communication and Interface

USB 2.0

480-Mbit Universal Serial Bus allows for ultra fast data transfer. This is the main communication standard for PC users. The USB BUS also provides power as well as communications over a single cord, thereby allowing the USB4000 to operate anywhere you can take a laptop computer without any bulky external power supplies.

RS-232

Also known as serial port communication, RS232 is a standard in PC and industrial device communications. Using transmit and receive signals this option allows the USB4000 to be a standalone device, which can output data to other logic devices/controllers such as a PLC or microcontroller. The USB4000 requires an external 5-Volt power source when operating in RS-232 mode.

USB4000 USB Port Interface Communications and Control Information

Overview

The USB4000 is a microcontroller-based Miniature Fiber Optic Spectrometer that can communicate via the Universal Serial Bus or RS-232. This section contains the necessary command information for controlling the USB200 via the USB interface. This information is only pertinent to users who do not want to use Ocean Optics' 32-bit driver to interface to the USB4000. Only experienced USB programmers should attempt to interface to the USB4000 via these methods.

Hardware Description

The USB4000 uses a Cypress CY7C68013 microcontroller that has a high speed 8051 combined with an USB2.0 ASIC. Program code and data coefficients are stored in external E^2PROM that are loaded at boot-up via the I^2C bus. The microcontroller has 8K of internal SRAM and 64K of external SRAM. Maximum throughput for spectral data is achieved when data flows directly from the external FIFO's directly across the USB bus. In this mode the 8051 does not have access to the data and thus no manipulation of the data is possible.

USB Information

Ocean Optics Vendor ID number is 2457 and the product ID is 0x1022.

Instruction Set

Command Syntax

The list of the commands is shown in the following table followed by a detailed description of each command. The length of the data depends on the command. All commands are sent to the USB4000 through End Point 1 Out (EP1). All spectra data is acquired through End Point 2 and 6 In and all other queries are retrieved through End Point 1 In (EP1). The endpoints enabled and their order is:

Pipe #	Description	Type	Hi Speed Size (Bytes)	Full Speed Size (Bytes)	Endpoint Address
0	End Point 1 Out	Bulk	64	64	0x01
1	End Point 2 In	Bulk	512	64	0x82
2	End Point 6 In	Bulk	512	64	0x86
3	End Point 1 In	Bulk	64	64	0x81

USB Command Summary

EP2 Command Byte Value	Description	Version
0x01	Initialize USB4000	0.90.0
0x02	Set Integration Time	0.90.0
0x03	Set Strobe Enable Status	0.90.0
0x05	Query Information	0.90.0
0x06	Write Information	0.90.0
0x09	Request Spectra	0.90.0
0x0A	Set Trigger Mode	0.90.0
0x0B	Query number of Plug-in Accessories Present	0.90.0
0x0C	Query Plug-in Identifiers	0.90.0
0x0D	Detect Plug-ins	0.90.0
0x60	General I^2C Read	0.90.0
0x61	General I^2C Write	0.90.0
0x62	General SPI I/O	0.90.0
0x68	PSOC Read	0.90.0
0x69	PSOC Write	0.90.0

EP2 Command Byte Value	Description	Version
0x6A	Write Register Information	0.90.0
0x6B	Read Register Information	0.90.0
0x6C	Read PCB Temperature	0.90.0
0x6D	Read Irradiance Calibration Factors	0.90.0
0x6E	Write Irradiance Calibration Factors	0.90.0
0xFE	Query Information	0.90.0

USB Command Descriptions

A detailed description of all USB4000 commands follows. While all commands are sent to EP1 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 USB4000

Description: Initializes certain parameters on the USB4000 and sets internal variables based on the USB communication speed the device is operating at. This command should be called at the start of every session however if the user does not call it, it will be executed on the first Request Scan command. The default vales are set as follows:

Parameter	Default Value
Trigger Mode	0 – Normal Trigger

Byte Format

Byte 0
0x01

Set Integration Time

Description: Sets the USB4000 integration time in microseconds. The value is a 32-bit value whose acceptable range is 10 – 65,535,000 µs. If the value is outside this range the value is unchanged. For integration times less than 3800 µs, the integration counter has a resolution of 1µs. For integration times greater than this, the integration counter has a resolution of 10 µs.

Byte Format

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0x02	LSW-LSB	LSW-MSB	MSW-LSB	MSW-LSB

MSW & LSW: Most/Least Significant Word

MSB & LSB: Most/Least Significant Byte

Set Strobe Enable Status

Description: Sets the USB4000 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 = 1 ➔ Lamp Enable HIGH/On

Byte Format

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

Query Information

Description: Queries any of the 31 stored spectrometer configuration variables. The Query command is sent to End Point 1 Out and the data is retrieved through End Point 1 In. The 31 configuration variables are indexed as follows:

Data Byte - Description
0 – Serial Number
1 – 0th order Wavelength Calibration Coefficient
2 – 1st order Wavelength Calibration Coefficient
3 – 2nd order Wavelength Calibration Coefficient
4 – 3rd order Wavelength Calibration Coefficient
5 – Stray light constant
6 – 0th order non-linearity correction coefficient
7 – 1st order non-linearity correction coefficient
8 – 2nd order non-linearity correction coefficient
9 – 3rd order non-linearity correction coefficient
10 – 4th order non-linearity correction coefficient
11 – 5th order non-linearity correction coefficient
12 – 6th order non-linearity correction coefficient
13 – 7th 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 – USB4000 configuration: AWL V
A – Array coating Mfg, W – Array wavelength (VIS, UV, OFLV), L – L2 lens installed, V – CPLD Version
17 – Autonulling information
18 – Power-up baud rate value
19-30 – User-configured

Byte Format

Byte 0	Byte 1
0x05	Data byte

Return Format (EP1)

The data is returned in ASCII format and read in by the host through End Point 1.

Byte 0	Byte 1	Byte 2	Byte 3	...
0x05	Configuration Index	ASCII byte 0	ASCII byte 1	…

Write Information

Description: Writes any of the 31 stored spectrometer configuration variables to EEPROM. The 31 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 17
0x06	Configuration Index	ASCII byte 0	ASCII byte 1	…	ASCII byte 15

Request Spectra

Description: Initiates a spectra acquisition. The USB4000 will acquire a complete spectra (3840 pixel values). The data is returned in bulk transfer mode through EP2 and EP6 depending on the USB Communication Speed. The table below provides the pixel orderint overview for the 2 different speeds. The pixel values are decoded as described below.

Byte Format

Byte 0
0x09

Return Format

The format for the returned spectral data is dependant upon the USB communication speed. The format for both High Speed (480 Mbps) and Full Speed (12Mbps) is shown below. All pixel values are 16 bit values which are organized in LSB | MSB order. There is an additional packet containing one value that is used as a flag to insure proper synchronization between the PC and USB4000.

USB High Speed (480Mbps) Packet Format

In this mode, the first 2K worth of data is read from EP6In and the rest is read from EP2In. The packet format is described below.

Packet #	End Point	# Bytes	Pixels
0	EP6In	512	0-255
1	EP6In	512	256-511
2	EP6In	512	512-767
3	EP6In	512	768-1023
4	EP2In	512	1024-1279
5	EP2In	512	1280-1535
…	EP2In	512	
14	EP2In	512	3584–3840
15	EP2In	1	Sync Packet

The format for the first packet is as follows (all other packets except the synch packet has a similar format except the pixel numbers are incremented by 256 pixels for each packet).

Packet 0

Byte 0	Byte 1	Byte 2	Byte 3
Pixel 0 LSB	Pixel 0 MSB	Pixel 1 LSB	Pixel 2 MSB

…

		Byte 510	Byte 511
		Pixel 255 LSB	Pixel 255 MSB

Packet 15 – Synchronization Packet (1 byte)

Byte 0
0x69

USB Full Speed (12Mbps) Packet Format

In this mode all data is read from EP2In. The pixel and packet format is shown below.

Packet #	End Point	# Bytes	Pixels
0	EP2In	64	0-31
1	EP2In	64	32-63
2	EP2In	64	64-95
…	EP2In	64	
119	EP2In	64	3808–3839
120	EP2In	1	Sync Packet

Packet 0

Byte 0	Byte 1	Byte 2	Byte 3
Pixel 0 LSB	Pixel 0 MSB	Pixel 1 LSB	Pixel 2 MSB

…

		Byte 62	Byte 63
		Pixel 31 LSB	Pixel 31 MSB

Packet 120 – Synchronization Packet (1 byte)

Byte 0
0x69

Set Trigger Mode

Description: Sets the USB4000 Trigger mode to one of four states. If an unacceptable value is passed, then the trigger state is unchanged.

Data Value = 0 ➔ Normal (Free running) Mode Data Value = 1 ➔ Software Trigger Mode Data Value = 2 ➔ External Synchronization 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

Description: Query's 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

Description: 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 I^2C addresses 2 through 7 respectively. The I^2C address are reserved for various categories of devices and the value for each category is shown below. I^2Caddresses 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^2C address 2	Value @ I^2C address 3	…	Value @ I^2C address 7	0x00

Detect Plug-ins

Description: Reads all of the plug-in accessories that are plugged into the I^2C bus. No data values are returned.

Byte Format

Byte 0
0x0D

General I²C Read

Description: Performs a general purpose read on the I^2C pins for interfacing to attached peripherals. The time to complete the command is determined by the amount of data transferred and the response time of the peripheral. The I^2C bus runs at 400KHz. The maximum number of bytes that can be read is 61.

Command Byte Format

Byte 0	Byte 1	Byte 2
0x60	I^2C Address	Bytes to Read

Return Byte Format

Byte 0	Byte 1	Byte 2	Byte 3	…	Byte N+3
I^2C Results	I^2C Address	Bytes to Read	Data Byte 0	…	Data byte N

I²C Result Value	Description
0	I^2C bus Idle
1	I^2C bus Sending Data
2	I^2C bus Receiving Data
3	I^2C bus Receiving first byte of string
5	I^2C bus in waiting for STOP condition
6	I^2C experienced Bit Error
7	I^2C experience a Not Acknowledge (NAK) Condition
8	I^2C experienced successful transfer
9	I^2C bus timed out

General I²C Write

Description: Performs a general purpose write on the I^2C pins for interfacing to attached peripherals. The time to complete the command is determined by the amount of data transferred and the response time of the peripheral. The I^2C bus runs at 400KHz. The results codes are described above.

Command Byte Format

Byte 0	Byte 1	Byte 2	Byte 3	...	Byte N+3
0x61	I^2C Address	Bytes to Write	Data Byte 0	…	Data byte N

Return Byte Format

Byte 0
I^2C Results

General SPI Input/Output

Description: Performs a general-purpose write and read on the SPI bus for interfacing to attached peripherals. The time to complete the command is determined by the amount of data transferred and the response time of the peripheral. The SPI bus runs at ~25KHz Clock. The maximum number of bytes that can be written or read is 61. During this transfer the SPI Chip Select signal is driven to an active LOW TTL level. Data is transmitted out the MOSI (Master Out Slave In) line on the rising edge of the clock signal. Data is also latched in the from the MISO line on the falling edge of the clock signal.

Command Byte Format

Byte 0	Byte 1	Byte 2	Byte 3	...	Byte N+2
0x62	# of Bytes (N)	Write Byte 0	Write Byte 1	…	Write Byte N

Return Byte Format

Byte 0	Byte 1	Byte 2	Byte 3	...	Byte N+1
# of Bytes (N)	Read Byte 0	Read Byte 1	Read Byte 2	…	Read Byte N

Write Register Information

Description: Most all of the controllable parameters for the USB4000 are accessible through this command (e.g., GPIO, strobe parameters, etc). A complete list of these parameters with the associate register information is shown in the table below. Commands are written to End Point 1 Out typically with 4 bytes (some commands may require more data bytes). All data values are 16 bit values transferred in MSB | LSB order. This command requires 100µs to complete; the calling program needs to delay for this length of time before issuing another command. In some instances, other commands will also write to these registers (i.e., integration time), in these cases the user has the options of setting the parameters through 2 different methods.

Byte Format

Byte 0	Byte 1	Byte 2	Byte 3
0x6A	Register Value	Data Byte LSB	Data Byte MSB

Register Address	Description	Default Value	Min Valu e	Max Value	Time Base	Version
0x00*	Master Clock Counter Divisor	24	1	0xFFFF	48MHz	1.00.0
0x04	FPGA Firmware Version (Read Only)					1.00.0
0x08	Continuous Strobe Timer Interval Divisor	48000	0	0xFFFF	Continuous Strobe Base Clock (see Register 0x0C)	1.00.0
0x0C	Continuous Strobe Base Clock Divisor	4800	0	0xFFFF	48MHz	1.00.0
0x0C	Continuous Strobe LSB Register	4800	0	0xFFFF	48MHz	3.00.0
0x10*	Integration Period Base Clock Divisor	480	0	0xFFFF	48MHz	1.00.0
0x10*	Integration Period LSB Register	480	0	0xFFFF	1KHz	3.00.0
0x14	Set base_clk or base_clkx2 0: base_clk 1: base_clkx2	0	0	1	N/A	1.00.0
0x18*	Integration Clock Timer Divisor	600	0	0xFFFF	Integration Period Base Clock (see Register 0x10)	1.00.0
0x18*	Integration Period MSB Register					3.00.0
0x20	Reserved					
0x28	Hardware Trigger Delay – Number of Master Clock cycles to delay when in External Hardware Trigger mode before the start of the integration period	0	0	0xFFFF	2MHz	1.00.0
0x28	Hardware Trigger Delay – Delay the start of integration from the rising edge of the trigger in 500ns increments	0	0	0xFFFF		3.00.0

Register Address	Description	Default Value	Min Value	Max Value	Time Base	Version
0x2C[&*]	Trigger Mode 0 = Free Running 1 = Software 2 = External Synchronization 3 = External Hardware Trigger	0	0	2	N/A	1.00.0
0x2C[&*]	Trigger Mode 0 = Free Running 1 = Software 2 = External Hardware Level Trigger 3 = Normal (Shutter) 4 = External Hardware Edge Trigger	0	0	2	N/A	3.00.0
0x30	Reserved					1.00.0
0x38	Single Strobe High Clock Transition Delay Count	1	0	0xFFFF	2MHz	1.00.0
0x3C	Single Strobe Low Clock Transition Delay Count	5	0	0xFFFF	2MHz	1.00.0
0x40	Lamp Enable	0	0	1	N/A	1.00.0
0x48	GPIO Mux Register 0: pin is GPIO pin 1: pin is alternate function	0	0	0x03FF	N/A	1.00.0
0x50	GPIO Output Enable 1: pin is output 0: pin is input	0	0	0x03FF	N/A	1.00.0
0x54	GPIO Data Register For Ouput: Write value of signal For Input: Read current GPIO state	0	0	0x03FF	N/A	1.00.0
0x58	Reserved					1.00.0
0x5C	Reserved					1.00.0

Register Address	Description	Default Value	Min Value	Max Value	Time Base	Version
0x74	Offset Value	0	0	0xFFFF	N/A	1.00.0
0x78	Offset Control Bit 0 = Enable Auto-Nulling Bit 1 = Enable Auto-Nulling Saturation	0	0	0xFFFF	N/A	1.00.0
0x7C	FPGA Programmed (Read Only)	0x5501	N/A	N/A	N/A	1.00.0
0x80	Maximum Saturation Level	0x55F0	0	0xFFFF	N/A	1.00.0
0xD4	ADC Convert Delay	0x002E	0	0xFFFF	N/A	3.00.0
0XD8	ADC Convert Width	0x0010	0	0xFFFF	N/A	3.00.0

Notes: * - User should not change these values because spectrometer performance can be affected. This information is included just for completeness
& - These values are controlled by other command interfaces to the USB4000 (i.e , Set integration time command).

Read Register Information

Description: Read the values from any of the registers above. This command is sent to End Point 1 Out and the data is retrieved through End Point 1 In.

Byte Format

Byte 0	Byte 1
0x6B	Register Value

Return Format (EP1In)

Byte 0	Byte 1	Byte 2
Register Value	Value MSB	Value LSB

Read PCB Temperature

Description: Read the Printed Circuit Board Temperature. The USB4000 contains a DS1721 temperature sensor chip which is mounted to the under side of the PCB. This command is sent to End Point 1 Out and the data is retrieved through End Point 1 In. The value returned is a signed 16-bit A/D conversion value, which is equated to temperature by:

Temperature (^{o}C) = .003906 * ADC Value

Byte Format

Byte 0
0x6C

Return Format (EP1In)

Byte 0	Byte 1	Byte 2
Read Result	ADC Value LSB	ADC Value MSB

If the operation was successful, the Read Result byte value will be 0x08. All other values indicate the operation was unsuccessful.

Read Irradiance Factors

Description: Reads 60 bytes of data, which is utilized for Irradiance Calibration information from the desired EEPROM memory address.

Byte Format

Byte 0	Byte 1	Byte 2
0x6D	EEPROM Address LSB	EEPROM Address MSB

Return Byte Format

Byte 0	Byte 1	...	Byte 59
Byte 0	Byte 1	…	Byte 59

Write Irradiance Factors

Description: Write 60 bytes of data, which is used for Irradiance Calibration information to the desired EEPROM memory address.

Byte Format

Byte 0	Byte 1	Byte 2	Byte 3	...	Byte 62
0x6E	EEPROM Address LSB	EEPROM Address MSB	Byte 0	…	Byte 59

Query Status

Description: 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
2-5	Integration Time - WORD	Integration time in µs – LSW | MSW. Within each word order is LSB | MSB
6	Lamp Enable	0 – Signal LOW 1 – Signal HIGH
7	Trigger Mode Value	
8	Spectral Acquisition Status	
9	Packets In Spectra	Returns the number of Packets in a Request Spectra Command.
10	Power Down Flag	0 – Circuit is powered down 1 – Circuit is powered up
11	Packet Count	Number of packets that have been loaded into End Point Memory
12	Reserved	
13	Reserved	
14	USB Communications Speed	0 – Full Speed (12Mbs) 0x80 – High Speed (480 Mbps)
15	Reserved	

Appendix A: USB4000 Serial Port Interface Communications and Control Information

Overview

The USB4000 is a microcontroller-based Miniature Fiber Optic, which can communicate via the Universal Serial Bus or RS-232. This document contains the necessary command information for controlling the USB4000 via the RS-232 interface.

Hardware Description

The USB4000 utilizes a Cypress FX2 microcontroller, which has a high speed 8051, combined with an USB ASIC. Program code and data coefficients are stored in external E^2PROM, which are loaded at boot-up via the I^2C bus.

Spectral Memory Storage

The USB4000 can store a single spectrum in the spectral data section. While spectra is being accumulated, it is being co-added to the existing spectra in memory. With this approach it is capable to accumulate any number of spectra (previous limit was 4).

Instruction Set

Command Syntax

The list of the command is shown in the following table along with the microcode version number they 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 USB4000 “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.

Upgrading from USB2000

The following is a summary of the changes that may be required if you are upgrading from a USB2000 to USB4000

- Baud rates
    - The startup baud rate is programmable through the EEPROM Calibration Entry #18
    - The unit operates at 115.2K Baud, but does not run at 57.6K Baud
- Operating Parameters
    - The I (upper case) command will set the integration time in milliseconds
    - To take advantage of the microsecond integration time capability, use the i (lower case) command
    - Most new operating parameters are set through the FPGA (W command)
- Spectral Data
    - If only one spectra is "Accumulated", then data is returned in 16 bit format
    - If additional spectra is "Accumulated", then data is returned in 32 bit format
    - The limitation of "Accumulating" 15 spectra is eliminated

Command Summary

Letter	Description	Version
A	Adds scans	1.00.0
B	Set Pixel Boxcar	1.00.0
C		
D		
E		
F		
G	Set Data Compression	1.00.0
H		
I	Sets integration time (ms increments)	1.00.0
J	Sets Lamp Enable Signal	1.00.0
K	Changes baud rate	1.00.0
L	Clear Memory	
M		
N		
O		
P	Partial Pixel Mode	1.00.0
Q		

Letter	Description	Version
R		
S	Starts spectral acquisition with previously set parameters	1.00.0
T	Sets trigger mode	1.00.0
U		
V		
W	Set FPGA Register Information	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
i	Set integration value (32-bit value and us increments)	1.00.0
k	Sets Checksum mode	1.00.0
v	Provides microcode version #	1.00.0
x	Sets calibration coefficients	1.00.0
?	Queries parameter values	1.00.0
+	Reads the plugged-in accessories	1.00.0

Command Descriptions

A detailed description of all USB4000 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 the USB4000 has the ability to return 32 bit values, overflow of the raw 16-bit ADC value is not a concern.

Command Syntax:	A{DATA WORD}
Response:	ACK or NAK
Range:	1-5000
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.

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

Set Data Compression

Description: Specifies whether the data transmitted from the USB4000 should be compressed to speed data transfer rates. For more information on USB4000 Data Compression, see Technical Note 1.

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

Integration Time (16 Bit)

Description: Sets the USB4000's integration time, in milliseconds, to the value specified. This command accepts just a 16-bit value and is expressed in ms for backward compatibility with the USB2000. Use the "i" command for full 32-bit functionality.

Command Syntax:	I{16 bit DATA WORD}
Response:	ACK or NAK
Range:	1 – 65,000,000
Default value:	6ms

Integration Time (32 Bit)

Description: Sets the USB4000's integration time, in microseconds, to the value specified.

Command Syntax:	i{32-bit DATA DWORD}
Response:	ACK or NAK
Range:	10 – 65,000,000
Default value:	6,000

Lamp Enable

Description: Sets the USB4000'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 USB4000's baud rate.

Command Syntax:	K{DATA WORD}
Value:	0=2400 1=4800 2=9600 3=19200 4=38400 5=Not Supported 6=115,200 7=230,400
Response:	See below
Default value:	2

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

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.

Notes

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

The power-up Baud rate can be set by setting the EEPROM Memory slot to the desired value (i.e., 6 for a value of 115,200 Baud)

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.

Command Syntax:	P{DATA WORD}	
Value:	Description 0 = all 3870 pixels 1 = every nth pixel with no averaging 2 = N/A 3 = pixel x through y every n pixels 4 = up to 10 randomly selected pixels between 0 and 2047 (denoted p1, p2, … p10)	Example P 0 (spaces for clarity only) P 1<Enter> N<Enter> P 2 N/A P3<Enter> x<Enter> y<Enter> n<Enter> P 4<Enter> n<Enter> p1<Enter> p2<Enter> p3<Enter> … p10<Enter>
Response:	ACK or NAK	
Default value:	0	

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.

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. If the Scans to AcculULate is 1, then the data is returned as WORDs. However, if it is greater than 1, then the data is returned as DWORDs to avoid overflow.

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 Data size flag (0=Data is WORDs, 1=Data is DWORDs)

WORD Number of Scans Accumulated

WORD Integration time in milliseconds

WORD FPGA Established Baseline value (MSW)

WORD FPGA Established Baseline value (MSW)

WORD pixel mode

WORDs if pixel mode not 0, indicates parameters passed to the Pixel Mode command (P)

(D)WORDs spectral data depending on Data size flag

WORD 0xFFFD – end of spectrum

Trigger Mode

Description: Sets the USB4000's external trigger mode to the value specified.

Command Syntax:	T{DATA WORD}
Value:	0 = Normal – Continuously scanning 1 = Software trigger 2 = External Hardware Level Trigger 3 = Normal (Shutter) 4 = External Hardware Edge Trigger
Response:	ACK or NAK
Default value:	0

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 USB4000 "echoes" the command and data values back out the RS-232 port.

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

Note

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 USB4000 is in the ASCII data mode.

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.

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

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.

Checksum Mode

Description: Specifies whether the USB4000 will generate and transmit a 16-bit checksum of the spectral data. This checksum can be used to test the validity of the spectral dat, and its use is recommended when reliable data scans are required. See Technical Note 2 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

Description: 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

Set FPGA Register Value

Description: Sets the appropriate register within the FPGA. The list of register setting is in the USB command set information. This command requires two data values, one to specify the register and the next to specify the value.

Command Syntax:	W{DATA WORD 1}{DATA WORD 2}
Value:	Data Word 1 – FPGA Register address Data Word 2 – FPGA Register Value
Response:	ACK or 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 USB4000 "echoes" the command and data values back out the RS-232 port.

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

Note

This command requires that the string "aA" be sent without any CR or LF. This is an attempt to ensure that this mode is not entered inadvertently. A legible response to the version number query (v command) indicates the USB4000 is in the ASCII data mode.

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.

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

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.

Checksum Mode

Description: Specifies whether the USB4000 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 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

Description: 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

Description: Writes one of the 32 possible calibration constant to EEPROM. The calibration constant is specified by the first DATA WORD which follows the x. 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.

Command Syntax:	x{DATA WORD}{ASCII STRING}
Value:	DATA WORD Index description 0 – Serial Number 1 – 0th order Wavelength Calibration Coefficient 2 – 1st order Wavelength Calibration Coefficient 3 – 2nd order Wavelength Calibration Coefficient 4 – 3rd order Wavelength Calibration Coefficient 5 – Stray light constant 6 – 0th order non-linearity correction coefficient 7 – 1st order non-linearity correction coefficient 8 – 2nd order non-linearity correction coefficient 9 – 3rd order non-linearity correction coefficient 10 – 4th order non-linearity correction coefficient 11 – 5th order non-linearity correction coefficient 12 – 6th order non-linearity correction coefficient 13 – 7th 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 – USB4000 configuration: AWL V A – Array coating Mfg, W – Array wavelength (VIS, UV, OFLV), L – L2 lens installed, V – CPLD Version 17 – Auto-nulling configuration information 18 – Startup Baud rate entry 19-30 – Reserved
Response:	ACK or NAK
Default value:	N/A

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

Query Variable

Description: 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) and ?W, which requires an additional data word be passed to indicate which calibration constant is to be queried.

Command Syntax:	?{ASCII character}
Response:	ACK followed by {DATA WORD}
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. In ASCII mode, the USB4000 transmits the prompt "> ", which is shown.

The desired operating conditions are: acquire spectra with a 200ms integration time, set number of scan to add to 5, transmit back every 4th pixel and operate at 115,200 Baud.

aA	(Set ASCII Data Mode)
K6<CR>	(Start baud rate change to 115,200)
	Wait for ACK, change to 115200, wait for 20ms
K6<CR>	(Verify command, communicate at 115200)
A2<CR>	(Add 5 spectra)
I200<CR>	(Set integration time to 200ms)
P1<CR>	(Set to acquire every 4th pixel)
4<CR>	
S	(Acquire spectra)
…	Repeat as necessary

Application Tips

- During the software development phase of a project, the operating parameters of the USB4000 may become out-of-synch with the controlling program. It is good practice to cycle power on the USB4000 when errors occur.
- If you question the state of the USB4000, you can transmit a space (or another non-command) using a terminal emulator. If you receive a NAK, the USB4000 is awaiting a command; otherwise, it is still completing the previous command.
- For Windows 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 **OK** in Port Settings and in Properties dialog boxes.

Technical Note 1: USB4000 Data Compression

Transmission of spectral data over the serial port is a relatively slow process. Even at 115,200 baud, the transmission of a complete 3840 point spectrum takes around 600 msec. The USB4000 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 USB4000 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 USB4000. Compression rates of 35-48% can easily be achieved with this algorithm.

The following shows a section of a spectral line source spectrum and the results of the data compression algorithm.

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
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

Pixel Value	Value Difference	Transmitted Bytes
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

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.

Technical Note 2: USB4000 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, 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

Data Value	Value Difference	Transmitted Bytes	Value added to Checksum
87	-2	0xFE	0x00FE
89	2	0x02	0x0002
86	-3	0xFD	0x00FD
88	2	0x02	0x0002
98	10	0x0A	0x000A
121	23	0x17	0x0017
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.