

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

## SM32Pro SDK Spectrometer Operating -Software Development Kit- USER MANUAL

For USB 2.0 Multi-channel User Only



# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

## Table of Contents

Warranty and Liability .....	3
Location of the SDK source code for USB 2.0 .....	4
System Requirements .....	5
Getting Started .....	6
Using the SP Libraries .....	8
General Overview .....	9
SDK Functions .....	11
SDK Function Descriptions .....	12-20
spTestAllChannels .....	12
spSetupAllChannels .....	12
spSetupGivenChannel .....	13
spInitAllChannels .....	13
spInitGivenChannel .....	14
spSetIntEx .....	14
spSetDbIntEx .....	15
spSetTrgEx .....	16
spReadDataEX .....	16
spCloseAllChannels .....	17
spCloseGivenChannel .....	17
spReadChannelID .....	18
spWriteChannelID .....	18
spGetAssignedChannelID .....	19
spPolyFit .....	19
spPolyCalc .....	20

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

## ***Warranty And Liability***

This SM product is warranted against defects in material and workmanship for a period of one year from the date of shipment. During the warranty period, **Spectral Products (SP)** will, without charge, repair or replace, at its discretion, the defective product or component parts.

For warranty service or repair, this product must be returned to a service facility designated by **SP**. For products returned under warranty, the Buyer shall prepay shipping charges (including shipping charges, duties, and taxes for products returned to **SP** from another country), and **SP** will pay for shipping charges to return the product to the Buyer.

This warranty does not apply in the event of misuse or abuse of the product or as a result of unauthorized alterations, modifications or repairs, if the serial number is altered, defaced or removed, the improper or inadequate maintenance by the Buyer, Buyer-supplied software or interfacing, operation outside of the environmental specifications for the product, or improper site preparation or maintenance. No other warranty is expressed or implied. **SP** shall not be liable for any consequential damages, including without limitation, damages resulting from loss of use, as permitted by law.

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834· FAX (860) 928-2676  
<http://www.spectralproducts.com>

## ***The Location of the SDK source code for USB 2.0***

If the software was installed with the installation CD, the SDK source codes are located in the directory that the software was installed. If the user didn't change the installation directory and the Windows OS had been installed on "C:\", the location of the SDK source codes is "C:\Program Files\Spectral Products\SM32Pro\SDK Examples\VC++ SDKs\SDK2XX-USB2.0" in general. The SDK examples can be downloaded from **SP**'s website (Click [here](#)).

In case of other interface users, please refer to the manual located in each SDK example directory. The SDK directory name indicates the interface type.

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834· FAX (860) 928-2676  
<http://www.spectralproducts.com>

## ***System Requirements***

### **Requirements for the Software**

- Any IBM compatible computer
- A hard drive with at least 50 MB free space
- A CD-ROM
- A VGA or compatible display
- 32 MB RAM or higher
- A mouse or other pointing device
- Microsoft Windows® 98, 98SE, ME, NT, 2000, and XP

### **Check System Package Contents**

Check that your SM system package contains all of the required components.

**Common system packages contain the following:**

- Spectrometer
- USB 2.0 Cable
- Any accessories ordered

**Note:** Package contents may vary from unit to unit and order to order. If you have any questions about the contents of your package, please contact the support team referred to in the back of this manual.

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

## ***Getting Started***

Welcome to the **Spectral Products'** SM32Pro Software Development Kit.

This kit documents a library of functions for accessing the USB 2.0/1.1 Data Acquisition Board used by the CCD detector units. Two fundamental CCD imaging concepts are the Collection of Data and Generating Meaningful Values from that data.

### **Collection of Data**

In our CCD spectrometer, the light is dispersed across a charge-coupled device (CCD) 2048 (Sony) or 3648 (Toshiba) pixel detector array. Data is collected by each pixel and converted to a relative value by the analog-to-digital (A/D) converter ranging from 0 to 65535 and representing the intensity of the light at each pixel. We are able to control the amount of time that the pixels collect light and thus read signals of varying strengths. Using this library of functions you will be able to adjust the signal capture time (Integration Time) and collect the converted signals from each pixel. The criteria to adjust the integration is to try different lengths of time with a reference signal representing the maximum possible signal level during the measurement until the peak value across the range of pixels is close to, but not at 65535. We recommend trying to get the peak at 65000 around. This way you can be sure that you are getting an optimized measurement condition with no saturation of CCD elements.

### **Generating Meaningful Values**

The information you collect from the system is only the relative signal size at each detector pixel until it has been calibrated to absolute (certified reference) values. That is in the raw data, the pixel location represents spatial distribution of the CCD elements and the corresponding intensity is represented by a digital value. Since we can adjust the strength of the light source and the integration time of the CCD, this A/D value only provides a relative representation of the real world parameters.

We address both of these problems. We connect pixels to wavelengths by use of known emission lines (e. g. **SP's** calibration light sources or narrow band filters) across the range of pixels together with a table relating pixels to wavelengths in nanometers. Curve fitting functions can then be applied to

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

generate a polynomial function for the conversion of all other pixels to wavelengths.

We address the second problem in one of two ways. Both ways involve “normalization.” Normalization involves measurements of signal strength based on its ratio with respect to reference signal intensity. If we are only interested in the spectral distribution of the sample signal, then we can normalize a sample signal scan to one reference value by, for example, intensities of all the array elements divided by the peak intensity. If we want to measure percent transmission, the light can be measured first with only air in the light path as 100% reference and then the sample can be inserted into the light path and a sample scan followed. Consequently, the divisions of sample scan readings by the 100% reference yields the relative transmission values, or percent transmission when multiplied by 100%. For reflectance measurement similar practice can be applied and a high reflector can be used as a reference air in many cases.

It is strongly recommended that a measurement be taken prior to any sample scan, in which no external light but only the detector noise will be sensed and subsequently subtracted from all the following measurements. This will subtract thermally generated “background” DC offset level and statistically establish a baseline of zero for all subsequent measurements.

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

## ***Using the SP Libraries***

For Windows C/C++, you must include the correct “include” file and “lib” file for the functions you are calling:

For Visual Basic, please include the correct “BAS” file. The function declarations listing in this file demonstrate the correct data types. To pass a “pointer” or array to a DLL from Visual basic, simply pass the first element of the array (IE SampleData[0]).

Please refer to the samples or give us a call if you need any further technical assistance.

### **Procedure for using the .dll with VC++**

1. The SPdbUSBm.dll and SPdbUSBm.lib need to be copied to the work or release folder. Those are located in the “[Software installed directory]\SDK Examples\VC++ SDKs\SDK2XX-USB2.0\DLL”.
2. Add SPdbUSBm.lib to the project/settings/link/object/library module.



# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

## General Overview

### Some basic fundamentals of utilizing the SP SDK functions

First spTestAllChannels() has to be called to get the total number of channels that are currently connected to the computer. It returns the total number of channels connected. Then spSetupAllChannels() or spSetupGivenChannel() should be called to set up each USB port connection. The splnitAllChannels() or splnitGivenChannel() needs to be called to initialize the setup components and values. After consecutively calling the three functions above, you can call any function in the DLL.

Integration time is the time period the CCD pixels are exposed to light before the resulting charges are read out. A longer integration time can allow you to detect a lower light level signal. The longer your integration time is the more background signal will accumulate.

### Using Curve Fitting to Calibrate SM32Pro

Curve fitting is used in SM32Pro/SDK to correlate the physical locations of pixels on the CCD with the known wavelength of the radiation falling on them.

This is done by identifying the pixel locations where the maximal of the known wavelength is at. These peak intensity wavelengths and pixels are used by spPolyCalc() and spPolyFit() to generate a correlating polynomial function which best represents all the data points. We have found that using a third order polynomial function produced the most desirable results for most cases. In cases where high dispersion elements are used, lower order polynomial functions may have to be utilized due to the limited known wavelengths available from the calibration lamps.

### Calibration Files

Each unit's calibration set is included in the SM32Pro software settings, "SM32Pro.ini". This text file contains calibration data of the form "DataX=Wavelength;Pixel". The file also contains the regression coefficients "A<sub>0</sub> value", "A<sub>1</sub> value", ... , "B<sub>3</sub> value" that satisfy the equations.

$$I_i = A_0 + A_1 P_i + A_2 P_i^2 + A_3 P_i^3$$
$$P_i = B_0 + B_1 I_i + B_2 I_i^2 + B_3 I_i^3$$

The A values are the coefficients for conversions from a pixel number to a wavelength in "nm" by use of the above first third order polynomial function. The B values allow the conversions from a desired wavelength in "nm" to a pixel number.

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

## ***SDK Functions***

### **Data Acquisition**

#### **spTestAllChannels**

- Used to check up the total number of channels connected. The USB connection order type needs to be set in this function.

#### **spSetupAllChannels**

#### **spSetupGivenChannel**

- Used to set up and establish the USB connection in all channels or each channel.

#### **spInitAllChannels**

#### **spInitGivenChannel**

- Used to initialize the components and values. The total pixel number of the array detector, the detector type (Sony CCD, Toshiba CCD, or Hamamatsu PDA) and the initial integration time are set in this function.

#### **spSetTrgEx**

- Used to set triggering type. The external triggering option can be set in this function.

#### **spSetIntEx**

- Used to set the CCD integration time.

#### **spReadDataEx**

- Used to collect spectral data from the CCD.

#### **spCloseAllChannels**

#### **spCloseGivenChannel**

- Used to close the USB connection. This function should be called at the exiting of program.

### **Channel Configuration**

#### **spReadChannelID**

- Used to read the channel ID assigned and saved in the EEPROM on the USB board.

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

## **spWriteChannelID**

- Used to write a new channel ID on the EEPROM on the USB board.

## **spGetAssignedChannelID**

- Used to get the information of all channel IDs assigned.

## **Calibration**

### **spPolyFit**

- Used to generate a calibration function from pixels to wavelength. This function calculates the coefficients for a polynomial curve fitting function given an array of independent variables and a corresponding array of dependent variables.

### **spPolyCalc**

- Used for determining the wavelength for a given pixel location. This function calculates a polynomial function given the independent variable and a coefficient array.

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

## ***SDK Function Descriptions***

### **spTestAllChannels**

short spTestAllChannels

```
(  
    short sOrderType = SP_ORDER_BY_CHANNELID  
        // The order of USB connection:  
        // SP_ORDER_BY_CHANNELID or  
        // SP_ORDER_BY_USBPORTNUM  
)
```

This function is used to test and check the connection of USB boards.

**sOrderType** is the type of the USB port connection order. If the user wants to define each channel by the order of the USB port connection, set the **sOrderType** as "SP\_ORDER\_BY\_USBPORTNUM" which is "0". If the user wants to define each channel by the assigned channel ID, set the value as "SP\_ORDER\_BY\_CHANNELID" which is "1".

The default is "1" i.e. SP\_ORDER\_BY\_CHANNELID. When the "SP\_ORDER\_BY\_CHANNELID" was selected but the USB boards connected don't have proper channel IDs assigned, this function will reassign the channel ID as the same as the USB port number of each USB board.

### **RETURN**

If the function works properly, it will return the total number of channels connected. If not, it will return a negative number.

### **spSetupAllChannels**

short spSetupAllChannels

```
(  
)
```

This function is used to set up and check the connections of all USB boards at once.

### **RETURN**

If there is no error, it will return a positive value (SP\_NO\_ERROR). If not, a negative one.

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834· FAX (860) 928-2676  
<http://www.spectralproducts.com>

## **spSetupGivenChannel**

short spSetupGivenChannel

```
(  
    short sChannel = 0    // The channel ID or USB port number  
)
```

This function is used to set up and check the connection of the given USB board.

**sChannel** could be the assigned channel ID or the USB port number according to the setting of the spTestAllChannels() function.

## **RETURN**

If there is no error, it will return a positive value (SP\_NO\_ERROR). If not, a negative one.

## **spInitAllChannels**

short spInitAllChannels

```
(  
    short sCCDType = SP_CCD_SONY // the type of CCD installed in the  
                                // spectrometer  
)
```

This function is used to initialize all USB boards connected at once.

**sCCDType** indicates what kind of detector is installed in the spectrometer. If the detector is Sony ILX511, then it should be "SP\_CCD\_SONY (0)", if Toshiba TCD1304AP, then "SP\_CCD\_TOSHIBA (1)" and if Hamamatsu, then "SP\_CCD\_PDA (2)".

When all spectrometers have the same type of CCD, use this function to initialize.

## **RETURN**

If there is no error, it will return a positive value (SP\_NO\_ERROR). If not, a negative one.

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834· FAX (860) 928-2676  
<http://www.spectralproducts.com>

## **spInitGivenChannel**

short spInitAllChannels

```
(  
    short sCCDType = SP_CCD_SONY, // the type of CCD installed in the  
                                   // spectrometer  
    short sChannel = 0    // The channel ID or USB port number  
)
```

This function is used to initialize the given USB board.

**sCCDType** indicates what kind of detector is installed in the spectrometer. If the detector is Sony ILX511, then it should be "SP\_CCD\_SONY" which is 0, if Toshiba TCD1304AP, then "SP\_CCD\_TOSHIBA" which is 1 and if Hamamatsu PDA, then "SP\_CCD\_PDA" which is 2.

**sChannel** could be the assigned channel ID or the USB port number according to the setting of the spTestAllChannels() function.

Use this function to initialize each USB board separately when each spectrometer has a different CCD.

## **RETURN**

If there is no error, it will return a positive value (SP\_NO\_ERROR). If not, a negative one.

## **spSetIntEx**

short spSetIntEx

```
(  
    long const lIntTime, //new integration time  
    short sChannel = 0    // The channel ID or USB port number  
)
```

This function is used to set and change the integration time.

**lIntTime** is amount of time in milliseconds for Sony CCD or Hamamatsu PDA detector or in 10 microseconds for Toshiba CCD to set as a new integration time. This number should range between 1 and 65535 for Sony CCD or

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

Hamamatsu PDA detector and between 1 and 6553500 for Toshiba CCD. In the case of Toshiba CCD, the minimum integration time is 10 microseconds and the value setting is based on 10 microseconds. For example, **IntTime** of "100" is 1 millisecond. To avoid the misleading, it is recommended for Toshiba CCD spectrometer users to use the **spSetDbIntEx()** function for setting the integration time.

**sChannel** could be the assigned channel ID or the USB port number according to the setting of the **spTestAllChannels()** function.

## RETURN

If there is no error, it will return a positive value (SP\_NO\_ERROR). If not, a negative one.

## **spSetDbIntEx**

short spSetDbIntEx

```
(  
    double const dIntTime,    //new integration time  
    short sChannel = 0        // The channel ID or USB port number  
)
```

This function is used to set and change the integration time.

**dIntTime** is amount of time in milliseconds to set as a new integration time. This number should range between 0.01 and 65535.0 for Toshiba CCD and between 1.0 and 65535.0 for Sony CCD or Hamamatsu PDA detector. In the case of Toshiba CCD, the value will be rounded off to two decimal places. In the case of other detectors, all numbers below decimal point will be off so it is recommended using the **spSetIntEx()** function rather than this one.

**sChannel** could be the assigned channel ID or the USB port number according to the setting of the **spTestAllChannels()** function.

## RETURN

If there is no error, it will return a positive value (SP\_NO\_ERROR). If not, a negative one.

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834· FAX (860) 928-2676  
<http://www.spectralproducts.com>

## spSetTrgEx

short spSetTrgEx

```
(  
    short const sTrgMode //Sets triggering mode  
    short sChannel = 0    // The channel ID or USB port number  
)
```

This function is used to set and change the triggering mode.

**sTrgMode** is used to set the trigger mode. If this value was set as “SP\_TRIGGER\_OFF” which is 10, the USB board runs freely. If an external triggering is needed, this value should be set as “SP\_TRIGGER\_EXTERNAL” which is 12. When the external trigger mode needs to be release to make the USB board run freely, set this value as “SP\_TRIGGER\_INTERNAL” which is 11. The default value is “SP\_TRIGGER\_OFF”. If the trigger mode doesn’t need to be set, then do not call this function.

**sChannel** could be the assigned channel ID or the USB port number according to the setting of the spTestAllChannels() function.

## RETURN

If there is no error, it will return a positive value (SP\_NO\_ERROR). If not, a negative one.

## spReadDataEx:

short spReadDataEx

```
(  
    long * pArray;    // The array in which spectral data is stored  
    short sChannel = 0    // The channel ID or USB port number  
)
```

This function is used to read the CCD signal data through USB board.

**pArray** points to a read memory address. Its memory size should be larger than (or at least the same as) the detector pixel number defined at **spInitAllChannels** or **spInitGivenChannel**. The total CCD pixel number to be read is one of “SP\_CCD\_PIXEL\_PDA” which is 1024, “SP\_CCD\_PIXEL\_SONY” which 2080, or “SP\_CCD\_PIXEL\_TOSHIBA” which 3680. Some detector contains the dummy pixels or optical blank pixels. The real data pixel number is



# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

“SP\_CCD\_PIXEL\_PDA\_REAL” which is 1024, “SP\_CCD\_PIXEL\_SONY\_REAL” which 2048, or “SP\_CCD\_PIXEL\_TOSHIBA\_REAL” which 3648. In case of Sony and Toshiba CCDs, the first 32 pixels are optical blank.

**sChannel** could be the assigned channel ID or the USB port number according to the setting of the spTestAllChannels() function.

## RETURN

If there is no error, it will return a positive value (SP\_NO\_ERROR). If not, a negative one.

## **spCloseAllChannels**

```
short spCloseAllChannels  
(  
)  
)
```

This function is called to close all USB board connections. It should be called when exiting the application.

## RETURN

If there is no error, it will return a positive value (SP\_NO\_ERROR). If not, a negative one.

## **spCloseGivenChannel**

```
short spCloseGivenChannel  
(  
    short sChannel = 0    // The channel ID or USB port number  
)
```

This function is called to close the given USB board connection.

**sChannel** could be the assigned channel ID or the USB port number according to the setting of the spTestAllChannels() function.

## RETURN

If there is no error, it will return a positive value (SP\_NO\_ERROR). If not, a negative one.

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

## **spReadChannelID**

short spReadChannelID

```
(  
    short *spNum,      // The pointer that the Channel ID will be saved in  
    short sChannel = 0 // The channel ID or USB port number  
)
```

This function is called to read the channel ID of the given USB board.

**spNum** is the pointer that the assigned channel ID of the USB board will be saved in.

**sChannel** could be the assigned channel ID or the USB port number according to the setting of the spTestAllChannels() function.

## **RETURN**

If there is no error, it will return a positive value (SP\_NO\_ERROR). If not, a negative one.

## **spWriteChannelID**

short spWriteChannelID

```
(  
    short sNum,          // The Channel ID to save  
    short sChannel = 0   // The channel ID or USB port number  
)
```

This function is called to write the channel ID on the given USB board.

**sNum** is the new channel ID to be saved on the USB board.

**sChannel** could be the assigned channel ID or the USB port number according to the setting of the spTestAllChannels() function.

## **RETURN**

If there is no error, it will return a positive value (SP\_NO\_ERROR). If not, a negative one.

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834· FAX (860) 928-2676  
<http://www.spectralproducts.com>

## **spGetAssignedChannelID**

void spGetAssignedChannelID

```
(  
    short *spChannelID,          // The pointer of the Channel ID assigned  
)
```

This function is called to get the assigned channel ID information.

It is only available when the “SP\_ORDER\_BY\_CHANNELID” was set on the **spTestAllChannels()** function.

**spChannelID** is the pointer of the assigned channel ID information. The memory of this pointer has to be allocated before using this function. The total memory size must be the same as the total USB connection number which is returned by the **spTestAllChannels()** function.

## **RETURN**

None.

## **spPolyFit**

short spPolyFit

```
(  
    double *x,      // Array of independent variables  
    double *y,      // Array of dependent variables  
    short numPts,   // Number of points in independent and dependent arrays  
    double *coefs,  // Pointer to array to hold calculated coefficients [index: 0 - order]  
    short order     // Order of polynomial  
)
```

This curve fitting function is used to find a polynomial function to calculate the wavelength of a given pixel. This function is used for calibration purposes. Either a calibration light source or a series of narrow band filters are scanned and the pixel locations of all known peaks are identified along with the known wavelength at that peak. These peak locations and wavelengths are stored in the arrays x and y, respectively. The arrays indices should range from 0 to [Number\_of\_Points - 1]. They are passed to the function along with a requested order for the polynomial fitting function and an array large enough to hold the coefficients (). This array is then used with **spPolyCalc** to calculate wavelength from pixels.

# Spectral Products

Spectrometers · Spectrophotometers · Color Instruments · Spectrographs · Monochromators

---

111 Highland Drive · Putnam, CT · 06260 · USA  
PHONE (860) 928-5834 · FAX (860) 928-2676  
<http://www.spectralproducts.com>

**x** is an array containing the independent variables. It should range from 0 to (numPts-1).

**y** is an array containing the dependent variables. It should range from 0 to (numPts-1).

**numPts** is the number of points in the variable arrays.

**coefs** is a pointer to the array that will contain the polynomial coefficients. It should range from 0 to (order-1).

**order** is the desired order of the polynomial. We have determined third order to be the optimum for wavelength calibration for most cases.

## RETURN

This function will return 1 if the function is successful. Otherwise it will return negative.

## spPolyCalc

void spPolyCalc

```
(  
    double *coefs,  
    short order,  
    double x,  
    double *y  
)
```

This function calculates for the following formula:

$y = a_0 + a_1 * x^1 + a_2 * x^2 + \dots + a_N * x^N$ , where \* specified multiplication.

**coefs** is a pointer to an array containing the polynomial coefficients. These can be calculated using **spPolyFit**.

**order** specified the order of the polynomial equation and must be less than or equal to the number of elements in coefs.

**x** is the independent variable, in this case, the pixel number.

**y** is the value to be calculated.

## RETURN

None