PDQ80S1, PDP90S1

Position Detector Systems

Operation Manual

Table of Contents

Table of Figures

Part 1. Introduction

The PDQ80S1 and PDP90S1 (from now on referred to as the PSD System) are a visible to IR beam alignment solution consisting of one detector and one sensor processing hub. Up to four sensors can be connected to a single hub.

The PDQ80S1 is designed for laboratory applications requiring precise path alignment of a beam in the 400 to 1050 nm wavelength range. It may be used in conjunction with a mirror or beamsplitter to verify proper beam path alignment. It may also be used with a motorized optical mount or translation stage to provide beam path control in a more sophisticated application.

The other PSD System is the PDP90S1. The PDP90A continuous position sensor replaces the PDQ80A quadrant sensor. This detector will detect a spot and give positional information of the centroid over the full active region of the sensor. It will be less precise for alignment measurements than the quadrant detectors, but will provide measurement information not available with the quadrant sensors.

The initial setup is fast and easy. Simply connect the sensor to the processing hub and the PSD is ready to use. It may be operated as a stand-alone instrument when powered by an AC USB power adaptor or it in conjunction with a PC by connecting it to a USB 2.0 high speed port and running the display application software supplied. This manual provides information on the setup procedure and features of the PSD. A troubleshooting section and detailed specifications of the various components are provided to further assist the user.

Part 2. Installation

2.1. Parts List

Below is a list of all components shipped with the PSD.

Description	PDQ80S1	PDP90S1
USB 2.0 Four Channel Detector Hub	1	1
PDQ80A – Quadrant Photodiode Sensor	1	
PDP90A –Position Photodiode Sensor		1
PSD User Manual	1	1
CD-ROM with PSD Software	1	1
USB 2.0 A-B Cable, 2 Meters	1	1

2.2. Connections

The sensor device connects to the PSD hub device via a 6-pin HRS connector to one of the four sensor ports on the top of the PSD hub device (see figure below). The four sensor ports are designated A through D, and each may be identified by its associated green LED which, when lit, indicates the sensor has detected beam alignment.

Figure 1: Four Channel Hub (PDQ80S1 Shown)

The hub can be powered by connecting the device to either a PC or a USB AC power converter. Once connected to the PC, the device can be controlled with the PSD Beam Alignment Display program. If the device is powered with the USB AC power converter, it will operate in Stand-Alone Mode, and the Sensor Status LED is used to verify beam alignment.

2.3. USB Port Requirements

This device requires a USB 2.0 port to interface with a PC; the older USB 1.1 port is not supported. To achieve the maximum benefit from your PSD, a dedicated USB 2.0 High-Speed port is highly recommended. If it is necessary to connect the PSD through a USB hub, a powered hub is required in order to supply the required power.

Part 3. Using the PSD Hub with the PDQ80A/PDP90A Sensor

The following is a functional description of the hub and the sensor.

3.1. Sensor Functional Description

PDQ80A

The PDQ80A detector is a segmented, position sensing, silicon, quadrant detector for precise path alignment of light in the 400 to 1050 nm range. Each quadrant detector of the position sensor produces a photocurrent proportional to the power of the incident beam. The four signals are compared to determine the position of the beam. For beams that have a non-uniform power density, the center determined by the sensor may not be the geometric center of the beam.

PDP90A

The PDP90A detector is a two dimensional, lateral effect, silicon, position-sensing detector for light in the 320 to 1100 nm range. These sensors can be used for beam alignment; however they are better suited for measuring relative positional changes or for distance measurements. The PDP90A sensor is constructed of a single large area photodetector with a resistive layer separating the anode connections. Incident light will generate a photocurrent which will be proportionally distributed to each of the four corner anode electrodes based on the location. The four signals are converted to difference X and Y signals relative to the center of the detector. A SUM signal is also provided to normalize the signals, removing positional errors due to dark current and temperature effects, typical of all photo detectors.

Beam Size Considerations

PDQ80A

The light spot applied to the PDQ80A must be smaller than the diameter of the quadrant photodiode array, which is 7.8 mm. The maximum recommended beam diameter is 3.9 mm. A decrease in output signal strength is observed as the light spot crosses the separation boundary of the quadrants, usually referred to as the gap. This effect is more pronounced as the diameter of the light spot decreases since a larger percentage of the light spot's power falls within the non-active gap. For this reason, the minimum light spot diameter is 1mm. It is recommended that a lens be used to adjust the size of the beam to fall between these two dimensions.

PDP90A

The large detection surface allows beam diameters of up to 9 mm, but for best results, the beam size should be less than 80% of the active area. Additionally, the beam should be no smaller than 0.2mm. Therefore, a range of 0.2 to 7.2 mm is recommended for best results. Unlike quadrant sensors which require overlap in all quadrants, the lateral sensor will provide positional information of any spot within the detector region, independent of beam shape, size and power distribution.

Use of the Sum and Difference Signals for Alignment

The sum output signal may be used to provide preliminary alignment of either device to the source light beam. First, the beam position is adjusted for maximum sum output signal. Second the beam is adjusted until the Y-Axis and X-Axis signals are at minimum. This procedure results in the beam being centered on the sensor.

Temperature Considerations

The operating temperature must be between 10 to 40 °C. For best resolution, the temperature should be kept at or below 25 °C. Thermal gradients across the detector will cause position errors and should be avoided.

3.2. PSD Hub Functional Description

The hub receives the three signals from each sensor connected to the hub. Up to four sensors and in any combination of PDPs or PDQs can be connected to one hub. The device hub processes the sensor data to determine the beam position relative to the center of the sensor and displays the alignment status for each sensor by toggling the green alignment status LED. When the LED is on, the beam is centered.

Hub Power and Initialization

The hub is powered via its USB port connector. It may be powered either by a PC USB 2.0 port or by a USB compatible AC power converter capable of supplying 5V DC with 500 mA. The green LED adjacent to the USB port will be lit when the device is powered.

Once powered up, the hub initializes in one of two modes depending on whether or not it is connected to an active USB port on a PC. If connected to a PC, the hub attempts to initialize in USB device mode and will appear in the PC’s device manager.

If the hub is not connected to a PC or fails to initialize in USB device mode, it will initialize in Stand-Alone Mode and indicates beam alignment status exclusively via the alignment status LED on the front panel. This permits beam alignment without the need for a computer. It is possible to disable the Stand-Alone Mode. In that case, if the device is not connected to a PC or it fails to initialize properly, it will be unusable.

Sensor Data Handling

The hub retrieves and processes sensor data in the same manner in either operation mode. The only difference is that in Stand-Alone Mode, no data is passed to the USB port.

The hub constantly monitors the three analog signals from each of the four sensor inputs. Once per scan interval each analog signal is sampled and digitally represented by a value ranging from -2047 to 2047. If the hub has initialized in USB device mode and sensor data has been requested by a host PC application, each of the sensor samples is sent to the USB port in 16-bit 2’s complement format using the following 12 word packet configuration:

Sensor	Parameter
A	X-Axis Difference
	Y-Axis Difference
	Total Beam Power
B	X-Axis Difference
	Y-Axis Difference
	Total Beam Power
C	X-Axis Difference
	Y-Axis Difference
	Total Beam Power
D	X-Axis Difference
	Y-Axis Difference
	Total Beam Power

The X-Axis Difference value is positive when the beam is positioned in the left half plane of the sensor area and negative when the beam is positioned in the right half plane of the sensor area. The Y-Axis Difference value is positive when the beam is positioned in the upper half plane of the sensor area and negative when the beam is positioned in the lower half plane of the sensor area.

The scan interval is user selectable as described on page 24. The default factory setting is 1 ms.

Processing and Detection of Beam Sensor Data

Once every four scan intervals, a beam alignment computation is performed for each sensor, and the status LED is updated. The beam alignment computation is calculated by normalizing the Total Beam Power parameter for each sensor to a value ranging from 1024 to 2047 inclusive. The resulting scale factor is used to normalize the X and Y difference parameters. For each sensor, the beam alignment status LED is toggled on if the normalized X and Y difference parameters are such that

$$-X_{Aligned} \leq \text{X-Axis Difference} \leq X_{Aligned}$$

and

$$-Y_{Aligned} \leq \text{Y-Axis Difference} \leq Y_{Aligned},$$

respectively, where $X_{Aligned}$ is the Horizontal Alignment Window setting and $Y_{Aligned}$ is the Vertical Alignment Window setting on the device. The default setting for both $X_{Aligned}$ and $Y_{Aligned}$ is 100 and may be set to any value between 1 and 254 using the Beam Alignment Display program panel.

Sensor Data Precision

PDQ80A

The precision of the beam alignment data for a PDQ80A sensor is dependent on the beam size, beam geometry, and beam power density characteristics. Given a beam size and geometry that fits within the 7.8 mm diameter of the PDQ80A sensor, a zero value for both the X and Y difference signals from the PDQ80A indicates the beam is centered about its centroid with respect to power density (i.e. the center of power) which may not be the geometric centroid of the beam. Apart from beam alignment, the actual beam position relative to true alignment can only be determined if the beam size, beam geometry, and beam power density gradient are well understood. Beam alignment is best achieved for a stable circular beam with a diameter between 1 and 3.9 mm and a uniform power density over the beam area.

PDP90A

The same is true for the PDP90A senor between 0.2 and 7.2 mm; however the PDP90A centroid will provide position information over its full operating region where the quadrant detector determines alignment with respect to the center. As such, the PDP90A provides a centroid position with respect to its location in the active region and is relatively independent of beam size. Resolution is typically lower since it is over a much larger detection range.

Beam Alignment Accuracy

The PSD is capable of converting sensor input analog signal data into a digital format with a maximum resolution of ±2047. However, due to the sensor signal output levels, the sensor resolution is limited to approximately ±1500. A stable beam with a diameter between 1 and 3.9 mm for the PDQ80A and 0.2 and 7.2 mm for the PDP90A can be aligned with respect to its center of power with known accuracy using the X-Axis Difference and Y-Axis Difference values as described on page 8 with the following procedure:

1. Position the beam such that the sensor X-Axis Difference and Y-Axis Difference values are as close to zero as possible. Note that large oscillations about zero in the X-Axis Difference and Y-Axis Difference values may indicate the source is not stable with respect to power density.
2. Move the beam to the left until the X-Axis Difference no longer increases and note the value. This is the negative X position limit for the beam and indicates that the entire beam is positioned in the left half plane of the sensor area. Note that moving the beam farther to the left causes the sensor Total Beam Power value to drop. This indicates a portion of the beam has moved outside the sensor area.
3. Repeat the above step moving the beam to the right of center to determine the positive X position limit. Note that the limits of the X-Axis Difference value may not be symmetric about zero. This indicates the beam geometry is not symmetric about its center of power. In this case, although the beam may still be aligned, the alignment precision cannot be determined.
4. Repeat steps 1 and 2 to determine the upper and lower limits of the Y-Axis Difference value by moving the beam in the vertical direction.

Given the beam width along the X-Axis and Y-Axis – Wx and Wy – are known and the X-Axis Difference and Y-Axis Difference values for the positive and negative beam position limits: $X_{positive}$, $X_{negative}$, $Y_{positive}$, and $Y_{negative}$ have been determined, the beam alignment precision can be calculated.

The beam alignment accuracy with respect to X-axis alignment is

$$\pm\left(\frac{Wx}{\left|X_{negative}\right|+\left|X_{positive}\right|}\right).$$

The beam alignment accuracy with respect to Y-axis alignment is

$$\pm\left(\frac{Wy}{\left|Y_{negative}\right|+\left|Y_{positive}\right|}\right).$$

Beam Position Accuracy

The PSD can be used to determine the beam position relative to alignment with known precision assuming a highly stable circular beam with a known radius R that falls between 1mm and 3.9mm and a uniform power density.

First, determine the X-Axis Difference and Y-Axis Difference values for the positive and negative beam position limits: $X_{positive}$, $X_{negative}$, $Y_{positive}$, and $Y_{negative}$ using the method above. The magnitudes of all four values should be equal or very close to equal.

Given an X-Axis Difference value X and a Y-Axis Difference value Y, the beam position with respect to

X-Axis alignment is

$$X\left(\frac{2R}{\left|X_{negative}\right|+\left|X_{positive}\right|}\right)\pm\left(\frac{2R}{\left|X_{negative}\right|+\left|X_{positive}\right|}\right),$$

and the beam position with respect to Y-Axis alignment is

$$Y\left(\frac{2R}{\left|Y_{negative}\right|+\left|Y_{positive}\right|}\right)\pm\left(\frac{2R}{\left|Y_{negative}\right|+\left|Y_{positive}\right|}\right),$$

3.3. Using the Hub as a Stand-Alone Instrument

As previously described, the PSD may be used as a stand-alone device independent of a PC when powered by an AC power supply with a USB port. In this configuration, the hub is only capable of indicating beam alignment status for each of the four sensor ports, A through D, by toggling the corresponding green status LED on the face of the device. This permits the user to monitor simple alignment status without needing a computer.

The ability to operate the PSD as a stand-alone instrument, herein referred to as Stand-Alone Mode, may be enabled or disabled. Although, the device will function properly during USB operation with Stand-Alone Mode enabled, the user may experience some USB initialization problems when using the device with slower computers. Therefore we recommend that the user disable the Stand-Alone Mode when the device is to be used solely in conjunction with a computer running the display software and operating the device via a USB 2.0 high-speed port.

The hub is shipped with the Stand-Alone Mode disabled. The device cannot operate as a stand-alone instrument using an AC to USB power supply until the user enables the Stand-Alone Mode using the Thorlabs Beam Alignment Display software supplied on the provided CDROM. The procedure for enabling and disabling Stand-Alone Mode operation on the PDQ80S1 device is detailed on page 26.

The ability to enable or disable Stand-Alone Mode operation on the PSD is only available on devices running firmware revision 1.03 or later. The user may determine which version of the firmware is running on the device by using the “Display Device Information” command option described on page 25.

Part 4. Software

The PSD is delivered with a software kit CDROM which includes the following components:

Component	Description
Beam Alignment Display	This is a CVI based instrument panel display program for providing a GUI for determining precise beam alignment.
USB Application Program Interface Library (API)	This is a Microsoft Visual C++ 6.0 static library providing a USB Application Program Interface for the PSD including C callable methods for all application-specific USB commands. The Beam Alignment Display application uses this library to communicate with the hub over its USB interface.
USB Device Driver Software	This is the software that allows the Beam alignment Display program and the API to communicate with the hub via its USB interface.
National Instruments CVI Runtime Engine Software version 7.1	The Beam Alignment Display program requires the National Instruments CVI Runtime Engine for its controls to function and display properly. If the host PC does not already have an installation of the CVI Runtime Engine, an installer package is provided for version 7.1 with this CDROM. Note that installation and use of this software is subject to terms and conditions set by National Instruments.
Software End User License Agreement	Please read the file *license.rtf*. All software distributed as part of this CDROM is subject to terms and conditions of the End User License Agreement.
This User Manual in PDF format	

4.1. Installing the Software

To install the software on the PC, insert the CDROM. The Installation Menu program should launch automatically. If not, browse to the CDROM directory: *\CD-Starter* and run the program *CD-Starter.exe*.

Application Software Installation

To install the Beam Alignment Display application and the USB Application Program Interface Library, mouse click on the "Install Software" menu button. Unless an alternative installation path is specified during the installation, the PDQ80S1 Beam Alignment Display application executable QDdisplay.exe is installed in the directory %ProgramFiles%\Thorlabs\PDQ80S1\ and the PDQ80S1 USB Application Program Interface Library files are installed in the directory ProgramFiles%\Thorlabs\PDQ80S1\API\[1]

USB Driver Installation

To ensure that the PSD functions properly, please follow the USB driver installation instructions carefully.

[1] Note: %ProgramFiles% refers to the Windows system variable ProgramFiles i.e. "C:\Program Files" in English Language installations of Windows 2000 or XP.

The hub must not be connected to your PC while the USB driver software is being installed. Installing the USB driver software while the hub is connected to a USB port can corrupt the driver installation process and cause the PSD to fail USB initialization.

To install the USB driver, perform the following sequence of steps.

1. Make sure the hub is not connected to a USB port.
2. Click on the "Install Drivers" menu button and follow the pop-up prompts. Note: you must accept the terms of the End User License Agreement for the driver software installation to complete successfully.
3. Be patient. The driver installation may take a few minutes. When it is complete, you will see a pop-up message that reads: "The USB driver for the PDQ80S1 has been successfully installed."
4. Attach the hub to a USB 2.0 High Speed port. If the device has not previously been installed and registered on this computer, the Found New Hardware Wizard will run. If the Found New Hardware Wizard fails to launch, proceed to step 7.
5. Click the radio button next to "No, not this time" and click "Next".
6. Click the radio button next to "Install software automatically" and click "Next". Allow the driver initialization to complete.
7. Following execution of the Found New Hardware Wizard or if the wizard fails to launch, open the Device Manager (open the *Control Panel*, click on the *System icon*, click the *Hardware tab*, and click on *Device Manager*). A device icon with the label "Jungo" should appear. Expand the icon and verify an additional device icon is present with the label "Thorlabs PDQ80S1 Quadrant Detector"

IMPORTANT
To function properly, the device must be reset and allowed to re-initialize following driver installation. To complete the driver installation sequence, un-plug and re-attach the hub to the USB port.

If problems arise, please refer to Part 5 on page 37 of the manual regarding troubleshooting the driver installation.

NI CVI Runtime Engine

The Thorlabs Beam Alignment Display program is a National Instruments CVI application requiring the CVI runtime engine. If this is not installed on the computer, mouse click on the "Install CVI Runtime" menu button. This will install version 7.1 of the National Instruments CVI Runtime Engine. For more information, visit CVI Runtime web page.

4.2. Beam Alignment Display Application

The following is a description of the functionality of the Beam Alignment Display application and some guidelines for how it may be used.

Running the Beam Alignment Display Application

To activate the Beam Alignment Display program go to the *Start Menu → Thorlabs → PDQ80S1 → PDQ80S1 Beam Alignment Detector*. The Beam Alignment Display panel should appear as shown below.

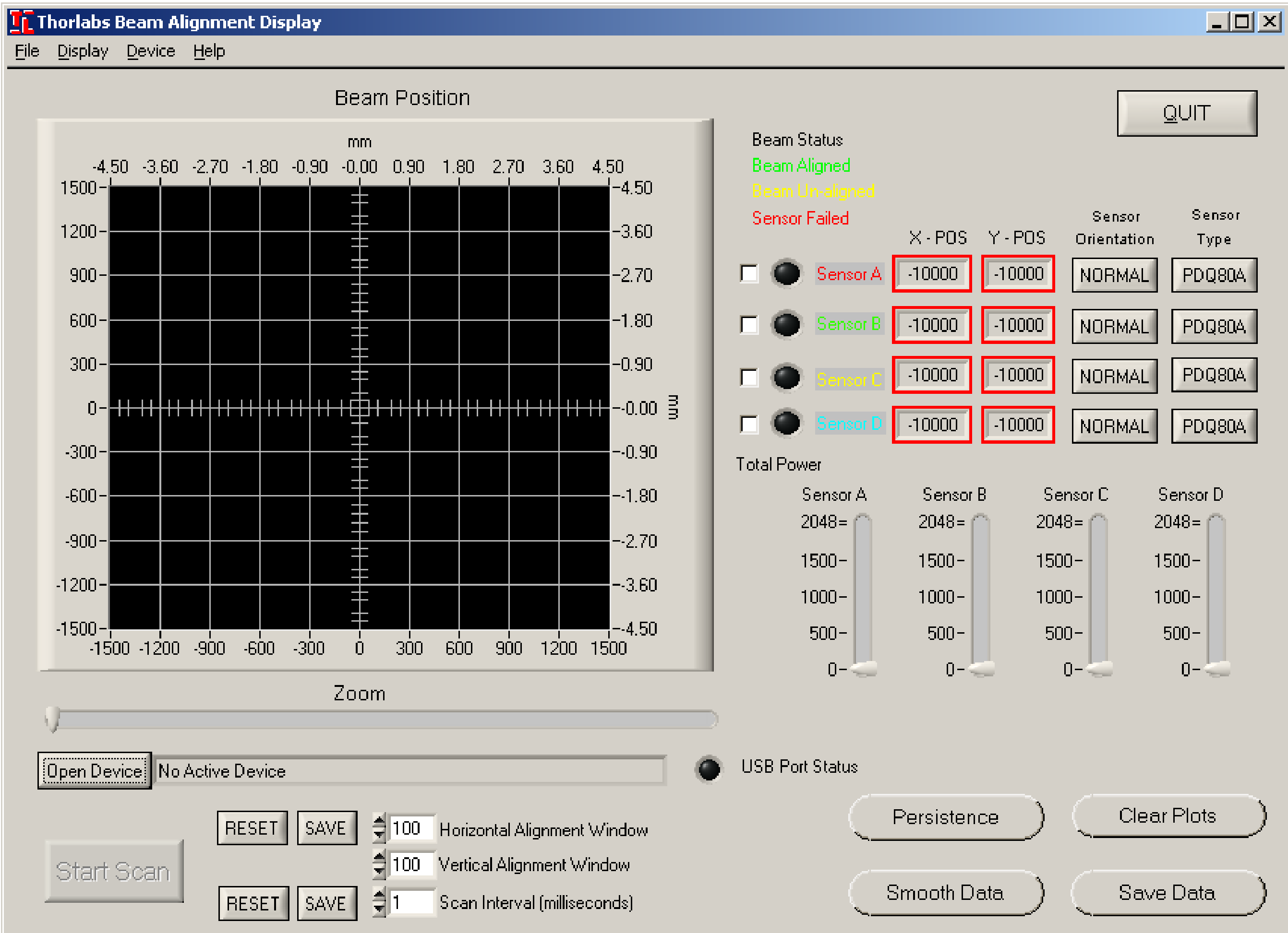

Figure 2: Beam Alignment Detector Program Main Screen

For each sensor input, the display program panel consists of a graphic XY plot display of beam position, a digital display of its X and Y coordinates, sensor status indication, and display control functions.

Once the hub has been connected to a USB port, open the "Device" pull-down menu and select "Open Device."

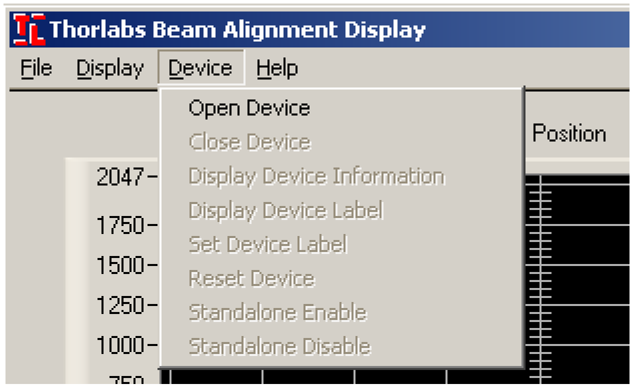

Figure 3: Open Device Menu

The display should resemble the screen in Figure 4:

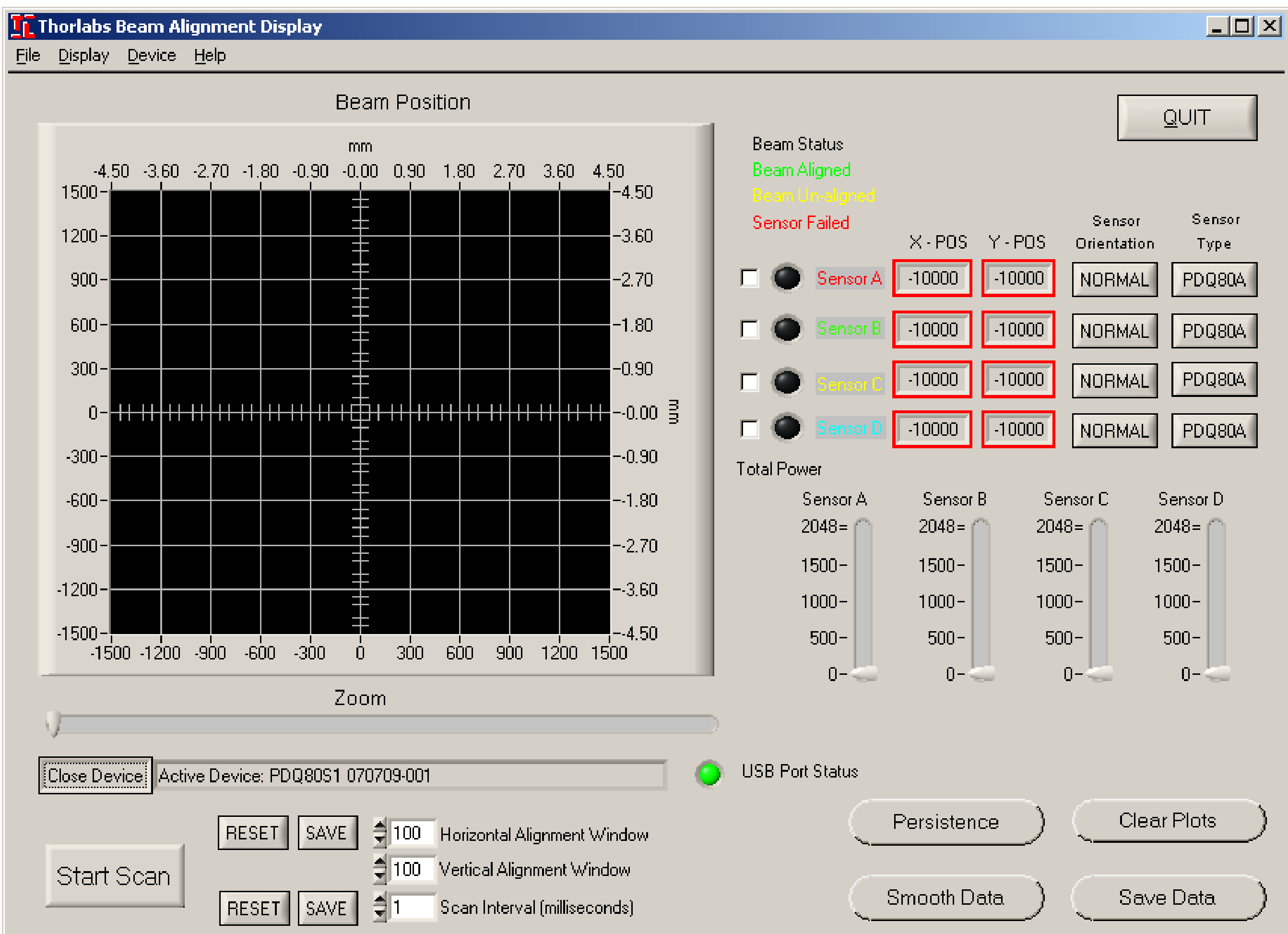

Figure 4: Main Display with Activated Device

Note that the device status bar displays the device label of the PDQ80S1 device that has just been opened. The "Start Scan" button is no longer dimmed and the "USB Port Status" indicator lamp is lit green. This indicates the display program is communicating properly with the PDQ80S1 hub device over the USB port. Also note that the Horizontal and Vertical Alignment Window settings have changed. Upon opening the device, the Beam Alignment Display program's initial settings are read from the Horizontal and Vertical Alignment Window settings and the Scan Interval setting from the device to be used.

Once the device has been opened, click the "Start Scan" button. The display should resemble Figure 5.

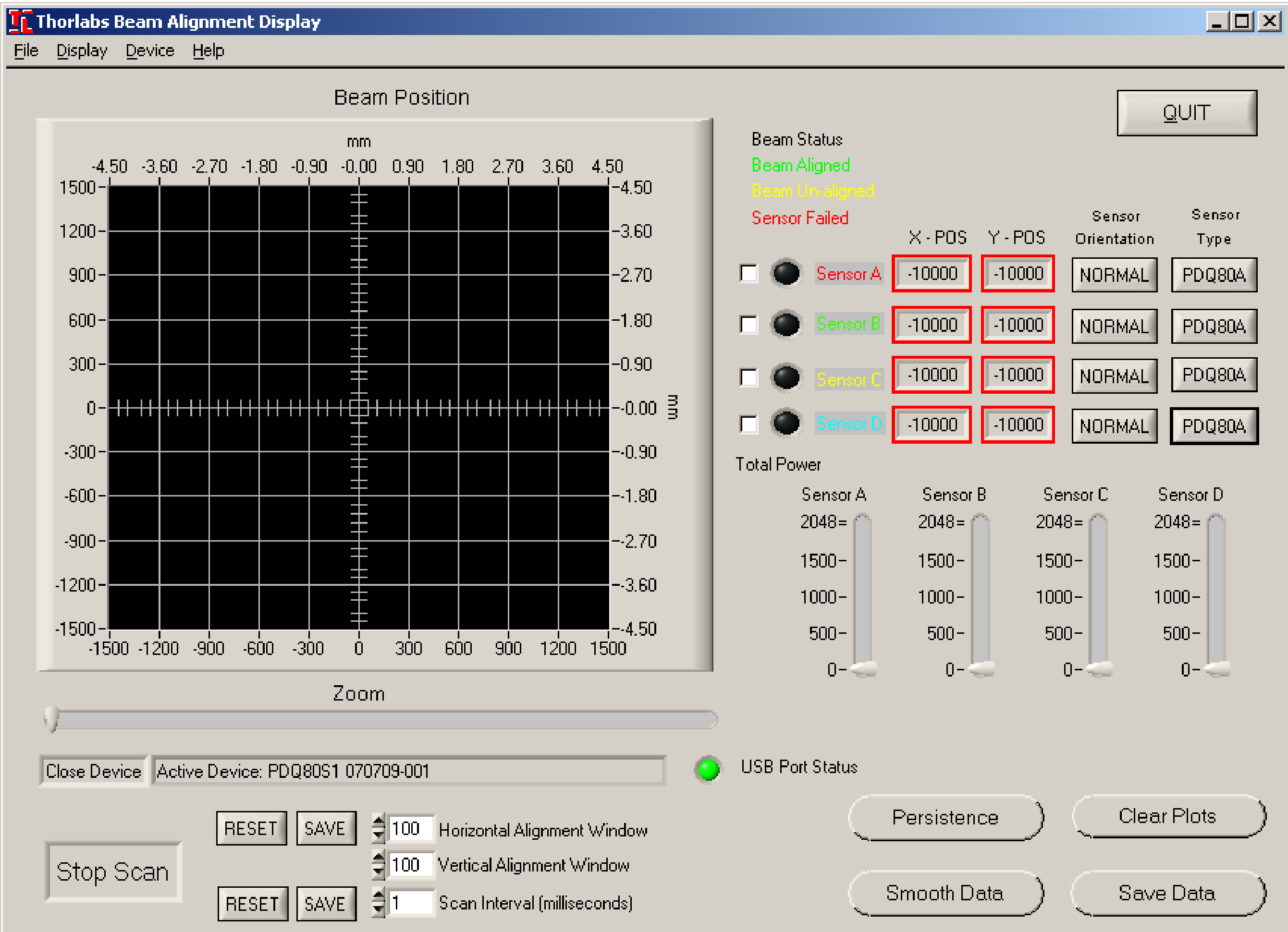

Figure 5: Main Screen with the Start Scan Button Depressed

The Red borders around the X-POS and Y-POS digital display indicate that the beam position values are out of range. The sensor's Beam Status indicator LEDs are toggled off, which indicates the sensors have not been enabled.

To enable the sensor display, click the checkbox located to the left of the sensor indicator. See page 22 Sensor Display Enable/Disable Controls for mode details.

In the display figure on the following page, sensors A and B have been enabled.

Note that the beam position for each enabled sensor is indicated in the graphic display by a color coded icon matching the sensor label adjacent to its Beam Status LED. The Beam Status LED for Sensor A is lit green indicating the beam is aligned. The Beam Status LED for Sensor B is lit yellow indicating the sensor display is enabled but beam alignment has not been detected.

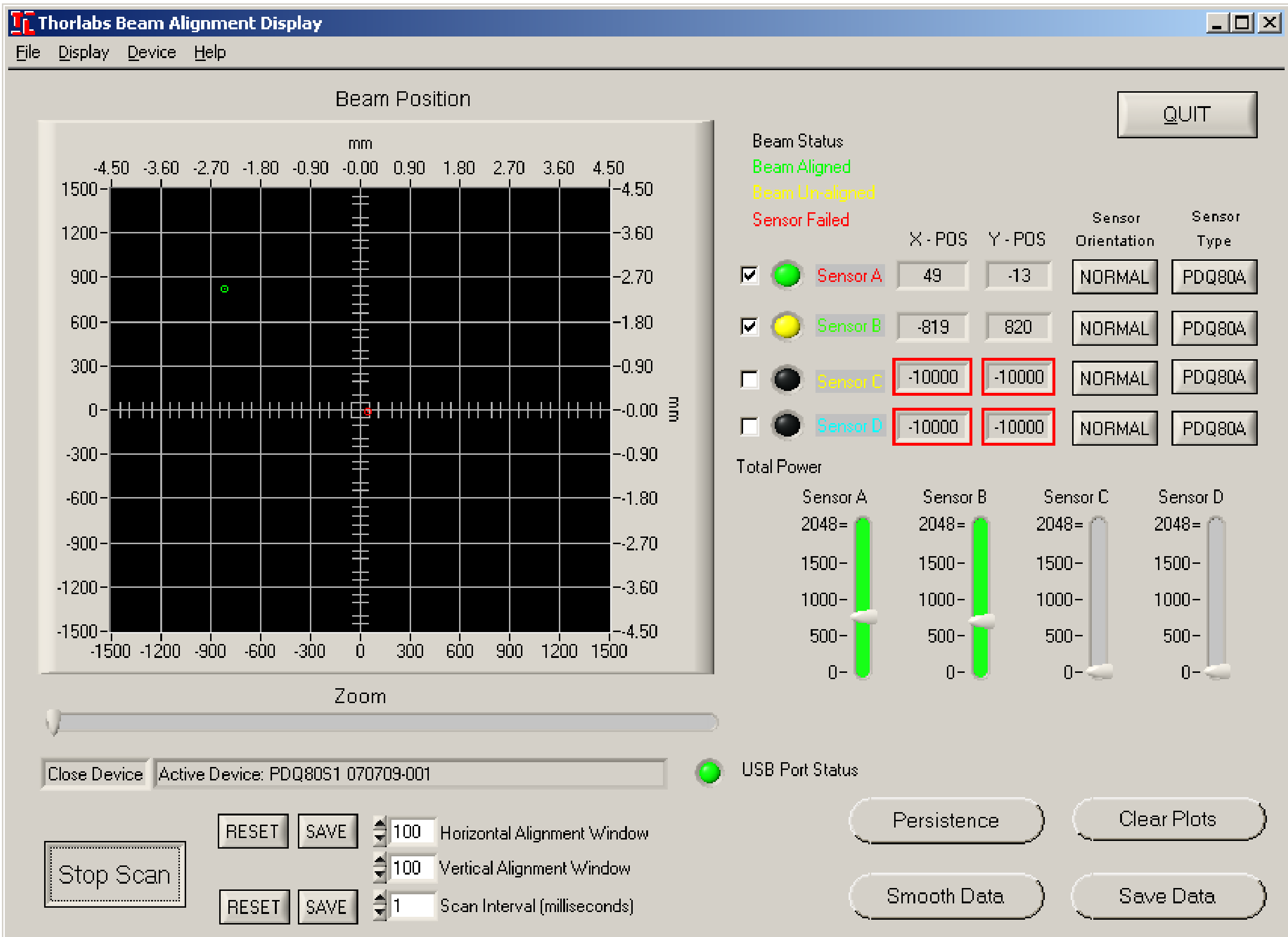

Figure 6: Beam Alignment Display with A and B in Use

Clicking the “Stop Scan” button causes the Beam Alignment Display program to stop retrieving sensor information from the hub device. The display is no longer updated and the sensors’ Beam Status indicator LEDs are toggled off. Note that the “Beam Position” graphic display and numeric position indicators are frozen but not cleared. To clear the display, click on the “Clear Plots” button.

Always click the “Stop Scan” button before unplugging an open device from the USB port.

To close the device, open the “Display” pull-down menu and select “Close Device”. The program ceases communication with the device and the “USB Port Status” LED is toggled off.

It is always advisable to close the device before unplugging it from the USB port. If the hub is unplugged from the USB port while still open, the USB Port Status indicator will be lit red, indicating that the Beam Alignment Display program has lost communication with the device and the device status bar displays “No Active Device.” If the device is re-attached, the Beam Alignment Display program will automatically detect the device, the USB Port Status indicator will light green and the device label will reappear in the device status bar.

Clicking on the "QUIT" button or selecting Exit in the File pull-down menu causes the application to exit. If a device is open, it is automatically closed.

Display Panel Indicators and Controls

The display panel for the Beam Alignment Display is as shown in Figure 7 on the following page. It consists of the following indicators and controls:

- Open/Close Device Control
- Start/Stop Scan Control
- USB Port Status Indicator
- Beam Position Graphical Display
- Beam Position Graphical Display Clear Plots Control
- Beam Position Graphical Display Persistence Control
- Beam Position Graphical Display Smooth Data Control
- Sensor Beam Alignment Status Indicator Lamps
- Sensor X-Axis & Y-Axis Beam Position Numeric Indicators
- Sensor Enable/Disable Controls
- Sensor Orientation Controls
- Sensor Type Controls
- Total Power Indicators
- Horizontal Alignment Window Adjustment Control
- Vertical Alignment Window Adjustment Control
- Scan Interval Adjustment Control

In addition, the display panel provides pull-down menus for configuring the display and controlling the device.

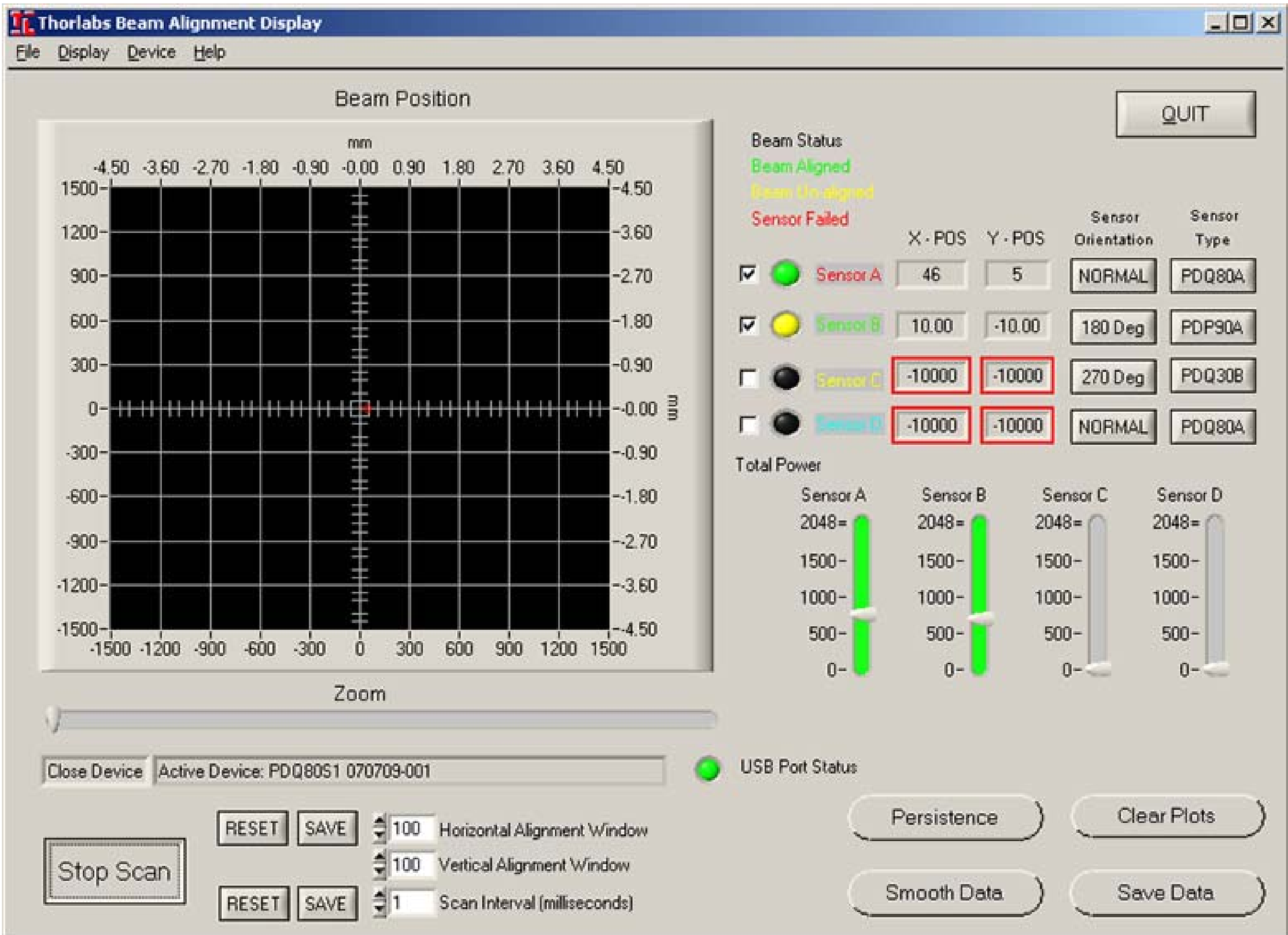

Figure 7: Beam Alignment Display Panel

As shown in Figure 7, the beam alignment status and control indicators to the right of the "Beam Position" display are arranged in a matrix with each row corresponding to a sensor: A, B, C, or D. As stated earlier, only one sensor is provided with the either the PDQ80S1 or PDP90S1, although additional sensors can be purchased separately and operated using one device hub.

Start/Stop Scan Control

Clicking the "Start Scan" button causes the Beam Alignment Display program panel to begin displaying scan data from the sensor. Once activated, the button label changes to "Stop Scan," and the hub begins sending the sensor data parameters described on page 8 for all four sensors to the Beam Alignment Display program via the USB interface. To stop the hub from sending data to the host, click "Stop Scan." Once deactivated, the button label changes to "Start Scan."

USB Port Status Indicator

The USB Port Status Indicator indicates the state of the USB communication path between the display program and the hub. It indicates the following states:

Color	Indication
Black (Off)	No communication initiated between the Beam Alignment Display program and the PDQ80S1. The USB connection is not initialized.
Yellow	The Beam Alignment Display program is attempting to initialize USB communications with the PDQ80S1.
Red	The Beam Alignment Display program has failed following several attempts to initialize USB communications with the PDQ80S1.
Green	The Beam Alignment Display program has successfully initialized USB communications with the PDQ80S1.

Beam Position Graphical Display

The Beam Position Graphical Display indicator is an XY plot of the beam position relative to the center of the sensor. The display range is ±2047 corresponding to the maximum range of digital conversion performed by the PSD as described on page 8. However, as indicated on page 9, the maximum XY position range provided by the sensor is limited to approximately ±1500.

The Beam Position display is capable of plotting the beam position of up to four sensors simultaneously. As a visual aid, the beam positions are plotted as color-coded icons.

Color	Sensor
Red	Sensor A
Green	Sensor B
Yellow	Sensor C
Cyan	Sensor D

As may be seen in the table above, the icon color for each sensor is coordinated with the Beam Status indicator labels.

Display Clear Plots Control

The "Clear Plots" control allows the user to completely clear the Beam Position display and the X-POS and Y-POS numeric indicators for all sensors. Clicking on the "Stop Scan" button does not clear the Beam Position display and the X-POS and Y-POS numeric indicators. To completely clear the display, the "Clear Plots" control must be selected. Clicking on the "Clear Plots" control does not prevent the display indicators from being updated if the display program is actively receiving and processing sensor data from the PDQ80S1.

Persistence Control

The "Persistence" control allows the user to overlay successive plots as a visual aid for tracking movement of the beam position. Normally, each time a beam position is plotted for a sensor, the previous point is deleted. Upon activating the "Persistence" control, the control label changes to "No

Persistence" and successive points are overlaid for each active sensor. Upon de-activation, the control label changes from "No Persistence" to "Persistence", the path points are deleted form the screen, and for each active sensor, the current beam position is deleted from the "Beam Position" display each time a new beam position is plotted.

Smooth Data Control

The "Smooth Data" control helps reduce the affects of display anomalies in the event of errors in the sensor parameters. Normally, each time a beam position is plotted for a sensor, the position is determined from the most recently received sample for each sensor parameter. Upon activating the "Smooth Data" control, the control label changes to "No Smoothing" and successive samples are averaged using a rolling average for each sensor parameter. Upon de-activation, the control label changes from "No Smoothing" to "Smooth Data" and each plot position is determined exclusively from the most recently received sensor data.

Sensor Beam Alignment Status Indicators

The "Beam Status" indicator lamps display the beam alignment status for each of the four sensors as follows.

Color	Indication
Black (Off)	The beam alignment display for the sensor is turned off. The sensor parameters for this sensor are not currently being processed and displayed. This indicates that either the display program is not currently requesting sensor data from the hub or display updates for this sensor have not been enabled via the Sensor Enable/Disable Controls.
Yellow	The beam alignment display for the sensor is turned on and the sensor parameters indicate the beam is not aligned based on the criteria described on page 9.
Red	The beam alignment display for the sensor is turned on and the sensor parameters indicate the sensor has failed.
Greed	The beam alignment display for the sensor is turned on and the sensor parameters indicate the beam is aligned based on the criteria described on page 9.

Sensor X-Axis and Y-Axis Beam Position Numeric Indicators

The X-POS and Y-POS numeric indicators display the X and Y coordinates of the beam position for each sensor enabled. For sensors not enabled, the numeric indicators display the value -10000 framed by a red border indicating the values are invalid (out of range).

Note that the value displayed for X-POS is the negative of the X-Axis Difference value but the value displayed for Y-POS is the actual value of the Y-Axis Difference value as described on page 8.

Sensor Display Enable/Disable Controls

The Sensor Display "ENABLE/DISABLE" checkbox either enables or disables the beam alignment display and status indicators for each of the four sensors. When the display program panel is initially launched, the beam alignment display and status indicators are disabled for all four sensors. Mouse clicking a checkbox enables the beam alignment display indicators for the corresponding sensor. When enabled, the checkbox will have a check. All beam alignment display information for the sensor is continuously displayed and updated. Clicking the checkbox again causes the check to be cleared and the display program to cease updating the beam alignment display information.

Sensor Orientation Controls

The Sensor Orientation controls permit some flexibility in the mounting of the sensors. Depending on the application, there may be a need to mount the sensors either on the side or inverted. The Sensor Orientation control allows the user to rotate the display for each sensor independently to compensate for a sensor that has been mounted counter clockwise 90, 180, or 270 degrees with respect to the normal up-right position. When the display program panel is initially launched, the Sensor Orientation button for each sensor displays the "Normal" label. In this state, the display is configured for sensors mounted in the up-right position. If a button is pressed once, the label changes to "90 Deg" and the display is reconfigured to display the beam position properly if the corresponding sensor is rotated by 90 degrees counter clockwise. If the button is pressed again, the label changes to "180 Deg" then 270 degrees, and finally pressing the button a fourth time returns the orientation to "Normal", 0 degrees.

Sensor Type Controls

The Sensor Type control allows the user to specify the type of sensor connected to each port of the hub. When the display program is initially launched, the Sensor Type is set to PDQ80A for the 4 ports. Pressing the Sensor Type control button will step the user through the list of supported detectors until the desired detector is selected. Once the user gets to the end of the list, pressing the control button again will bring the display back to the PDQ80A. Note that the quadrant detectors (PDQ80A) will use the bottom and left axis for the X/Y alignment, while the position sensors (PDP90A) will use the top and right for the X/Y alignment.

Total Power Indicators

The Total Power Indicators gives a visual representation of the Total Beam Power read from the sensors. These indicators can be used to optimize the power to achieve the highest accuracy. If the beam power is too great and the sensor saturates, the indicator will turn red. If the power is too low, the indicator will turn yellow alerting the user that the accuracy will be decreased. The bars indicate the full input range of the hub, and set the upper and lower limits based on the sensor chosen. The levels are determined by the maximum possible signal voltages for the sensors (±2 V for the PDQ80A and ±4 V for the PDP90A).

Horizontal and Vertical Alignment Window Adjustment Controls

The Horizontal Alignment Window and Vertical Alignment Window adjustment controls allow the user to set the size of the region about the origin within which the beam may be considered aligned with acceptable error for a particular application. The default factory setting for each parameter is 100. By clicking on the respective numeric edit scroll box the user may set each parameter to any value between 1 and 254. Note that changing the alignment window in the display program panel also changes the alignment window on the device so that the sensor alignment LED on the device and the alignment indicator on the display program panel should always indicate the same state. The current settings may be stored on the device by pressing the adjacent "Save" button. Once saved, the current settings become the new default settings used when the device is reset or used in Stand-Alone Mode. The default settings stored on the device may be retrieved as the current settings by pressing the adjacent "Reset" button.

Note that the "Save" and "Reset" buttons act on both the Horizontal Alignment Window and the Vertical Alignment Window adjustment controls. Although the horizontal and vertical settings may be adjusted independently, they may only be stored on the device and retrieved from the device simultaneously.

Scan Interval Adjustment Control

The Scan Interval adjustment control allows the user to set the rate at which data is collected from the four sensors and sent to the PC for display and alignment computation as described on page 8. The default factory setting is 1ms for a scan rate of 1000 scans per second. By clicking on the numeric edit scroll box the user may adjust the scan interval in one millisecond increments up to 254 ms for a minimum scan rate of approximately four scans per second. The current scan interval setting may be stored on the device by pressing the adjacent "Save" button. Once saved, the current setting becomes the new default setting used when the PDQ80S1 device is reset or used in Stand-Alone Mode. The default setting stored on the device may be retrieved as the current setting by pressing the adjacent "Reset" button.

Note: It is not advisable to use the Beam Alignment Display program to scan a PDQ80S1 device with a Scan Interval setting greater than 50 ms. The display program permits the user to set the Scan Interval over its full range for use in Stand-Alone Mode or in conjunction with the device Application Program Interface for applications requiring very low scan rates.

Display Pull-Down Menu

The display pull-down menu on the menu bar allows the user to hide some of the controls on the display program panel. This is useful for hiding controls that rarely need to be adjusted once set.

In the figure below, the Sensor Orientation control, Scan Parameter Settings, and Device Connect button have been hidden from view. Scan Parameter Settings refer to the Horizontal and Vertical Alignment Window adjustment controls and the Scan Interval adjustment controls. Other controls that may be hidden include the Persistence button, the Smooth Data button, and the Sensor Display Enable/Disable button controls.

Selecting the "Save Settings" sub-menu performs two functions. It saves the settings in the "Hide Controls" sub-menu and it also saves the state of the Sensor Display and Sensor Orientation buttons. The settings are stored in the file *QDdisplay.ini* in the working directory of the Beam Alignment Display program: *QDdisplay.exe*. If the file *QDdisplay.ini* is present, the previously stored settings are automatically restored the next time the Beam Alignment Display program is launched. Otherwise, default settings are invoked.

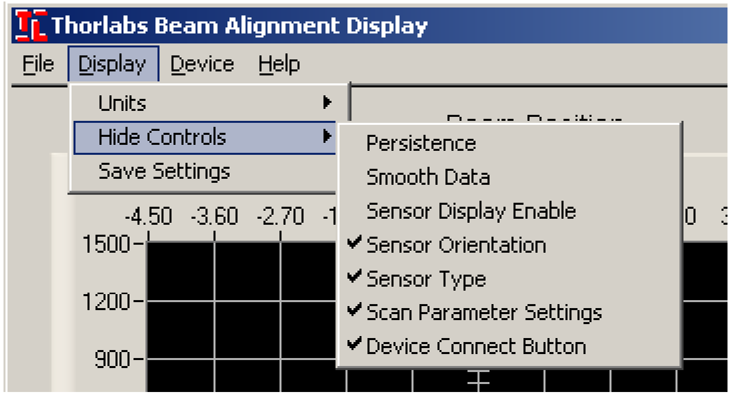

Figure 8: Display Pull Down Menu

Device Pull-Down Menu

The Device pull-down menu on the menu bar provides the user with device status and control functions.

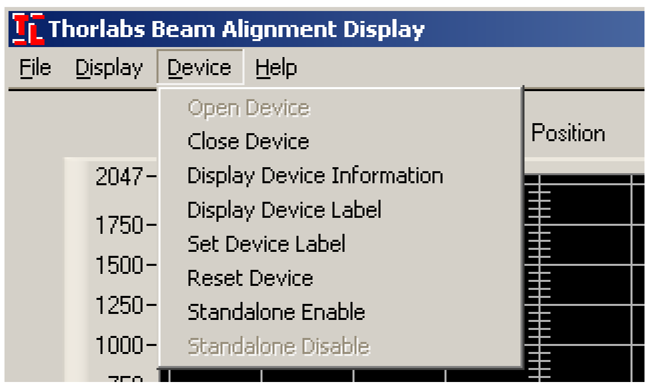

Figure 9: Device Pull Down Menu

Open Device and Close Device perform the same function as the Open/Close Device Control. Note that Open Device is dimmed when there is already a device open. Likewise, Close Device is dimmed if there is no active device open.

"Display Device Information" when selected, launches a pop-up status widow that displays the device serial number and device firmware revision.

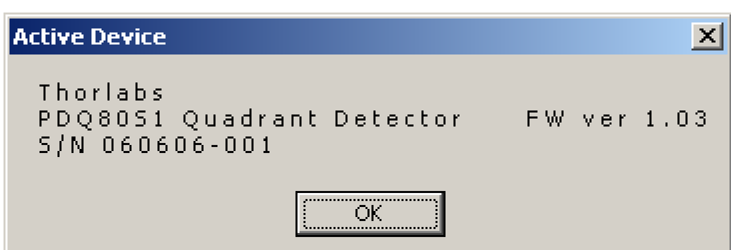

Figure 10: Display Device Information Pop-Up

Display Device Label when selected, launches a pop-up status widow that displays the user selected device label.

Figure 11: Display Device Label Pop-Up

Set Device Label allows the user to change the device label. When selected, the following pop-up dialog window appears:

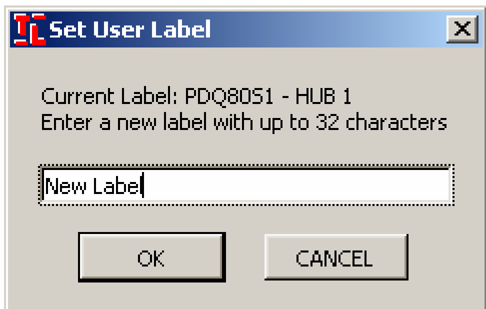

Figure 12: Set Device Label Dialog Box

Upon entering a string of 32 characters or less in the text edit window and clicking on "OK", the string is stored on the device and the new label appears in the device status bar.

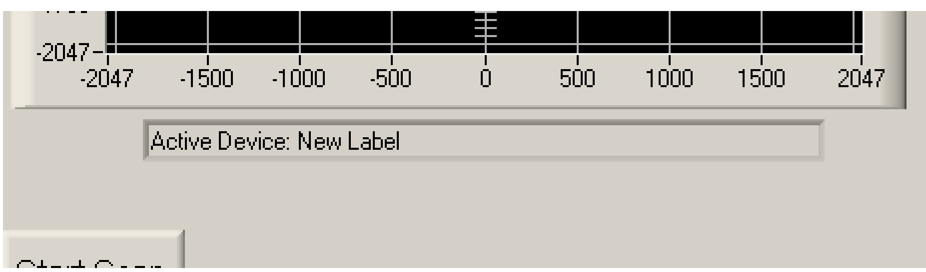

Figure 13: Device Status Bar Label

Reset Device allows the user to force the device to reinitialize into its power up state and restart. Stand-Alone Enable allows the user to configure the device to operate as a stand-alone instrument without a host PC as described on page 11. It is described in greater detail on page 26. Stand-Alone Disable allows the user to configure the device to operate solely as USB instrument connected to a PC.

Configuring the Hub for Stand-Alone Operation

The hub may be configured to either allow or prevent operation as a stand-alone instrument by following the steps below.

1. With the hub connected to a PC, launch the display software and open the device as described on page 14.
2. In the display software window, click the Display pull-down menu.
3. Stand-Alone Mode is disabled if the Stand-Alone Enable menu item is visible and the Stand-Alone Disable menu item is dimmed.
4. To enable Stand-Alone Mode, click on the Stand-Alone Enable menu item. A popup window will appear to allow the user to cancel the operation. Once confirmed, an additional popup window will appear to indicate the operation is complete.

To disable Stand-Alone Mode, click on the Stand-Alone Disable menu item. A popup window will appear to allow the user to cancel the operation. Once confirmed, an additional popup window will appear to indicate the operation is complete. For the change to take affect, the device must be reset. This can be performed by clicking the Reset Device menu item in the Device pull-down menu. Following reset, the device is no longer operable when connected to an AC to USB power supply.

4.3. Save Configuration

The Beam Alignment Display can save and record the position data over time to a file. This action can be controlled from the file drop down menu or direct panel button. One or all of the sensors' data can be saved.

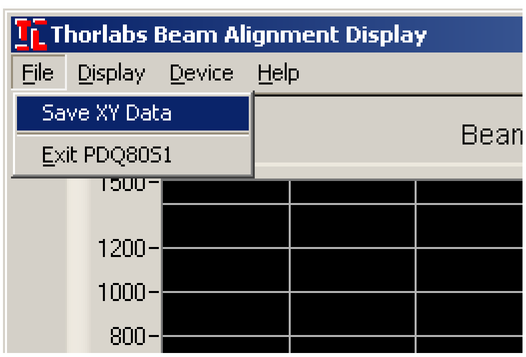

Figure 14: Beam Alignment Display Data Save from File Menu

Figure 15: Beam Alignment Display Data Save from "Save Data" button

Clicking on the "Save XY Data" opens and displays the Save Configuration pop-up window.

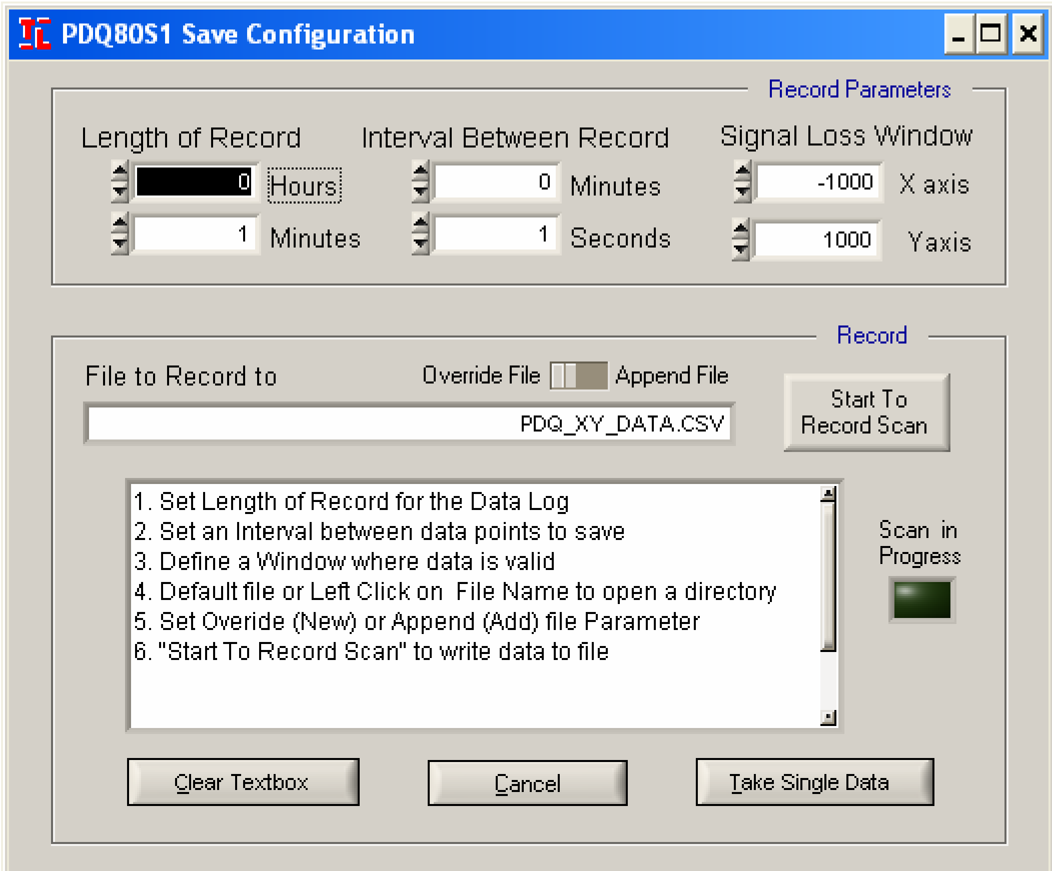

Figure 16: Save Configuration Pop-UP Window

Each data record always displays the data, time and sensor in use, followed by the X and Y coordinate of the incoming beam to the that sensor. There are four categories of variables a user has to configure for a data record scan.

Function	Description
Length of Record	This dialog box sets the length of the recording period from minutes to hours.
Interval Between Record	This input sets the length of the interval before recording the data points. This value must always be less then the Length of Record.
Signal Loss Window	The user has the ability to define an X and Y window. Data inside of this window is considered valid. Data falling outside of these boundaries will not be saved.
File to Record to	The user has an option to name the data file. The default file name is set to *PDQ_XV_DATA.CSV*, a coma delimited ASCII file that can be easily opened with a spreadsheet software program such as Excel. See Figure 17.

	A	B	C	D	E
1	Date	Time	Sensor	X Axis	Y Axis
2	8/14/2006	8:55:38	Sensor A	18	8
3	8/14/2006	8:55:39	Sensor A	18	8
4	8/14/2006	8:55:40	Sensor A	18	8
5	8/14/2006	8:55:41	Sensor A	18	8
6	8/14/2006	8:55:42	Sensor A	18	8
7	8/14/2006	8:55:43	Sensor A	18	8

Figure 17: Sample Data Output Opened in Microsoft Excel

4.4. Recording a Continuous Scan

Function	Description
File Attribute	This button toggles between either having the recorded data be appended to an existing file or overwritten to a file with an existing file name using the "File Attribute" attribute. Selecting "Overwrite File" will destroy any previous recorded data with the same file name. "Append File" will add the new data to the existing data in the file.
Start to Record Scan	This button begins recording the data and selecting it again will stop the recording. While active, the LED will be Green

4.5. Recording a Single Scan

Function	Description
Take Single Data	This button polls the X and Y coordinates and displays the values in the textbox. Data is not recorded to any file.
Clear Text Box	This button clears the entire contents of the text box.
Cancel	This buttons closes the PDQ80S1 Save Configuration of the Panel

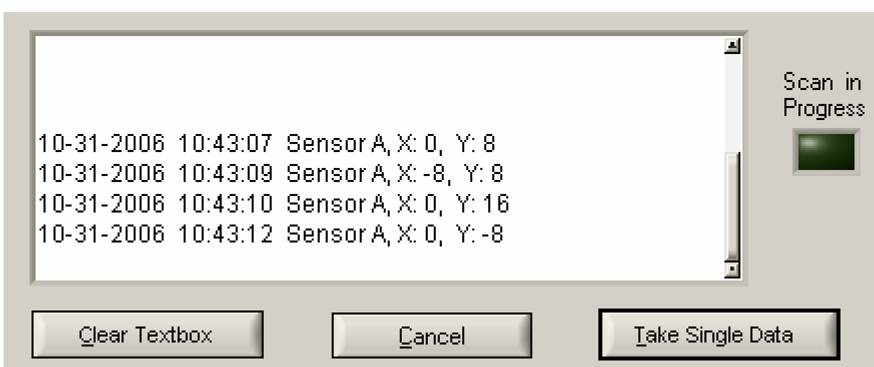

Figure 18: Sample Data Collected in Text Box for Single Scan

4.6. Application Program Interface Description

The Application Program Interface (API) provides a means by which the PSD may be integrated into a software application requiring a beam alignment sensing function. A possible application might be a computer program that continuously monitors beam position via the PSD using the information to acquire and maintain beam position alignment by controlling a mirror mount or translation stage.

The API consists of a Microsoft Visual C++ 6.0 based static library and associated header files providing a C programming language based encapsulation for each of the USB commands used to control and retrieve sensor data from the hub over the USB interface.

API Library Files

The API consists of three files:

File	Description
USB.lib	This is the precompiled static library file that must be linked with an application program that calls functions implemented by the PDQ80S1 API. This file is installed in the API subdirectory of the software installation directory as described on page 13.
USB.h	This is a header file describing the generic USB interface methods for the hub and must be included in C source files that call functions implemented by the PDQ80S1 API. This file is installed in the API subdirectory of the software installation directory as described on page 13.
PDQ.h	This is a header file describing the generic USB interface methods for the PSD and must be included in C source files that call functions implemented by the API. This file is installed in the API subdirectory of the software installation directory as described on page 13.

API Methods Description

The following methods are included in the API. The following methods are not application specific and may be implemented for any Thorlabs USB instruments that are compatible with this API.

API Generic USB Methods	
Command	**Description**
int USBinit(int PreferredDevice)	Initialize the Thorlabs device with USB Product ID PreferredDevice for USB access and return a device handle if successful. Return NULL otherwise. This function must be called and return a non-zero result before any other functions that access the device can be called. This function performs USB initialization for a Thorlabs USB instrument with the USB product ID: PreferredDevice.
void USBUninit(void)	Un-initialize the hub for USB access. This function should be called upon completion of all USB device communication with the hub.
void USBRestartDevice(void)	Force the hub to perform a hardware reset.
char *USBGetDeviceID(void)	Return the PSD USB Product ID as a character string. Upon success, the returned character string is "PDQ80S1 Jungo Device (PID: 0x0020)"
void USBGetProductString (char *pbuffer)	Get the string held by the device USB Product String Descriptor. This function implements the Standard USB (Chapter 9) Request for returning the USB Product String Descriptor in the character buffer pbuffer.
void USBDisplayDevices(void)	Display the device information including the Vendor, Product, and Serial strings in a Windows pop-up message window. This function can only be used in an application incorporating a Windows compatible Graphical User Interface.
void USBDisplayDeviceLabels (void)	Display the user specified device label in a Windows pop-up message window. This function can only be used in an application incorporating a Windows compatible Graphical User Interface.
void USBSetDeviceLabels (unsigned char *cLabel)	Set the user specified device label to the contents of the string argument cLabel. This function allows the user to assign a label string of up to 32 characters to a device. The string may be retrieved to identify the device or distinguish it from other devices.
unsigned char *USBReturnDeviceLabel(void)	Returns the PSD user assigned device label string. A NULL is returned if no device is attached. May be used in conjunction with USBSetDeviceLabels to distinguish between devices.
void USBRetrieveDeviceLabels(void)	Retrieve the user's assigned label string from the hub via the USB interface. Unlike USBReturnDeviceLabel, which reads the string from the host driver, this function reads the device label string from the device over the USB interface if the host driver object contains no copy of the label string.
unsigned char *USBPullDeviceLabel(void)	Updates the driver object device label string with the label string stored on the device. Unlike USBRetrieveDeviceLabels, this method always retrieves the user label string from the device via the USB interface and copies it to the label string stored in the host driver object.

API Generic USB Methods	
Command	**Description**
void* USBGetDevCtx(void)	Returns the attached USB device handle if attached or a NULL pointer if no USB device is attached. This function may be used to test if the hub is initialized for USB access.
unsigned char *USBReturnDevice(void)	Returns the USB device string descriptors concatenated. A NULL pointer is returned if no device is attached. Returns a string consisting of the contents of the USB Vendor String descriptor, the USB Product String descriptor, and the USB Serial String descriptor concatenated into a single string.
unsigned short USBGetDevProductID(void)	Returns the USB device Product ID in 16 bit unsigned format. For the PDQ80S1, it returns the integer 0x0020.
void USBPullDeviceSN(unsigned char *cLabel)	Updates the driver object resident USB serial number string descriptor with the Serial string stored on the USB device.
char *USBGetDeviceSN(void)	Return the USB device serial string descriptor from the host USB driver object. The NULL pointer is returned if no device is attached.

The following definitions are intended to allow the user to create application specific callback functions invoked when a hub is attached or removed from the USB port following USB initialization. An example implementation might be a graphical banner display or popup window that displays the device label of a hub that has just been attached or removed from a USB port.

Command	Description
void SetpDisplayDeviceMethod (PVOIDFUNCPCHAR pFunc)	Initializes PVOIDFUNCPCHAR to a function of type PVOIDFUNCPCHAR. This may be used to initialize a user defined callback function with a single char* parameter. The function is invoked in the event a hub is attached to or removed from a USB port following initialization.
void SetpSystemTypeMethod (PVOIDFUNCSHORT pFunc)	Initializes PVOIDFUNCSHORT to a compatible function. This may be used to initialize a user defined callback function with a single short parameter. The function is invoked in the event a hub is attached to or removed from a USB port following initialization.
typedef void (*PVOIDFUNCPCHAR)(char *cstr)	Defines a pointer to a function with a character string parameter. This may be used to initialize a user defined callback function with a single char* parameter. The function is invoked in the event a hub is attached to or removed from a USB port following initialization.
typedef void (*PVOIDFUNCSHORT)(short spar)	Defines a pointer to a function with a short parameter. This may be used to initialize a user defined callback function with a single short parameter. The function is invoked in the event a hub is attached to or removed from a USB port following initialization.

The following methods are specific to PSD applications.

API PDQ80S1 Application Specific USB Methods	
Command	**Description**
int USBinitPDQ80S1(void)	Initialize the PDQ80S1 for USB access and return a device handle if successful. Return NULL otherwise. This function must be called and return a non-zero result before any other functions that access the PDQ80S1 can be called. This function performs USB initialization for a Thorlabs USB instrument with the PDQ80S1 USB product ID: 0x0020. This function is the equivalent of a call to USBinit(0x0020).
unsigned long PDQStartScan(unsigned char ScanCount)	Commands the attached hub to begin sending sensor data to the host application. If successful, the return value is zero. This function causes the hub to begin sending beam alignment parameters for each of the four sensors to the host over the USB port. All parameters for each sensor are sent in a data packet in the format described on page 8. The packets are sent at a rate of up to 1000 packets per second depending upon the scan interval setting. Upon receiving the command, the device continues to send data packets until either ScanCount packets have been sent or the device receives a PDQStopScan command. If ScanCount is 0, the hub continues to send data packets indefinitely until it receives a PDQStopScan request. Since the host PC may not be able to keep up with the USB send rate, it is possible to begin a scan, collect a sufficiently large number of sensor samples to meet the requirements of the application, and then stop the scan sequence by calling the PDQStopScan function.
unsigned long PDQStopScan(void)	Commands the attached hub to stop sending sensor data to the host application. If successful, the return value is zero. The return value is 1 if the read operation is successful. A return value of -256 indicates the data read is invalid.
int PDQReadScan(short *str, int length, int timeout)	Read a parameter packet from the USB port. The "str" parameter is a pointer to a data buffer where the sensor data is deposited by the USB interface. The data buffer must be compatible with an array of 12 short integers. The length parameter is the number of unsigned short words to be read and the value is always 12. The timeout parameter indicates the maximum latency in milliseconds allowed for the function to return. It should not need to be set greater than 1000. The return value is 1 if the read operation is successful. A return value of -256 indicates the data read is invalid.
int PDQSendScanInterval(unsigned int settime, BOOL WriteToEEPROM)	Commands the attached hub to use settime in milliseconds as the new Scan Interval. If WriteToEEPROM is TRUE, settime is stored on the device as the new default Scan Interval. If the operation is successful, the return value is zero.
int PDQResetScanInterval(void)	Commands the attached hub to reset its Scan Interval setting to the default value stored on the PDQ80S1 device. If the operation is successful, the return value is zero.
int PDQSetAlignmentWindow(unsigned char X, unsigned char Y)	Commands the attached hub to use X and Y as the new settings for Horizontal Alignment Window and the Vertical Alignment Window respectively. If the operation is successful, the return value is zero.
int PDQResetAlignmentWindow(void)	Commands the attached hub to restore the settings for Horizontal Alignment Window and the Vertical Alignment Window to the default values stored on the device. If the operation is successful, the return value is zero.

API PDQ80S1 Application Specific USB Methods	
Command	**Description**
int PDQReadAlignmentWindow(unsigned char *XAlignmentWindow, unsigned char *YAlignmentWindow)	Commands the attached hub to read the default values for the Horizontal Alignment Window and the Vertical Alignment Window stored on the device and return by reference as *XAlignmentWindow and *YAlignmentWindow respectively. If the operation is successful, the return value is zero.
int PDQWriteHAlignmentWindow(unsigned char *XAlignmentWindow)	Commands the attached hub to store the value of *XAlignmentWindow as the default setting for the Horizontal Alignment Window. If the operation is successful, the return value is zero.
int PDQWriteVAlignmentWindow(unsigned char *YAlignmentWindow)	Commands the attached hub to store the value of *YAlignmentWindow as the default setting for the Vertical Alignment Window. If the operation is successful, the return value is zero.
int PDQReadScanInterval(unsigned char *ScanInterval)	Commands the attached hub to read the default value for the Scan Interval stored on the device and return by reference as *ScanInterval. If the operation is successful, the return value is zero.
int PDQRetrieveScanParameters(unsigned char *ScanParameters)	Commands the attached hub to read the currently used values (not stored default values) for the Scan Interval, the Horizontal Alignment Window, and the Vertical Alignment Window in an array referenced by *ScanParameters in the following format: *Scan Interval = ScanParameters[0]* *Horizontal Alignment Window = ScanParameters[1]* *Vertical Alignment Window = ScanParameters[2]* If the operation is successful, the return value is zero.
int PDQReadStandaloneModeFlag(unsigned char *StandaloneModeFlag)	Commands the attached hub to read the default value for the Stand-Alone Mode Flag stored on the device and return by reference as *StandaloneModeFlag. Valid values are 0 for Stand-Alone Mode disabled and 0xFF for Stand-Alone Mode enabled. Returns error status: 0 = success.
int PDQWriteStandaloneModeFlag(unsigned char *StandaloneModeFlag)	Commands the attached hub to store the value of *StandaloneModeFlag as the default setting for the Stand-Alone Mode Flag used the next time the device initializes. Valid values are 0 for Stand-Alone Mode disabled and 0xFF for Stand-Alone Mode enabled. Returns error status: 0 = success.

Guidelines for Implementing a PSD Application Using the API

When using the PSD in a user designed application program, the following guidelines should be followed.

Use only the API library supplied by Thorlabs specifically for the PSD to communicate with this device. Similar API libraries may be supplied for other Thorlabs USB devices. They should not be used with the PSD. The device may respond differently and in an unexpected manner to some USB command methods encapsulated in libraries not intended for the device.

Use only the API methods documented on page 31 to control the device. Never use private methods or undocumented USB commands to communicate with the device. Undocumented private methods or USB commands are not intended for use in end user applications. Use of these methods could corrupt the hub making it permanently unusable. It a violation of the End User License Agreement to reverse engineer the API static library to discover undocumented private methods and USB commands for this USB device.

When retrieving sensor data from the hub, use the following command sequence:

PDQStartScan(0)
for(…)
{
PDQReadScan
}
PDQStopScan

In effect, once a sensor scan is started, the only USB command used should be PDQReadScan. Do not attempt to call other functions that send USB commands to the hub other than PDQStopScan. Sending other USB commands to the hub during a scanning process will result in scan errors and ultimately could require resetting the device and restarting your application.

Example Application Using the PSD API

The following program indicates how to use the API to retrieve scan data from the hub.

```
main()
{
short sensorData[12];
short SensorA_Xdifference;
short SensorA_Ydifference;
short SensorA_TotalPower;
short SensorB_Xdifference;
short SensorB_Ydifference;
short SensorB_TotalPower;
short SensorC_Xdifference;
short SensorC_Ydifference;
short SensorC_TotalPower;
short SensorD_Xdifference;
short SensorD_Ydifference;
short SensorD_TotalPower;
        // The hub must be plugged into a USB port at this point
if (USBinitPDQ80S1() == 0)
{
        // Sucessfully initialized the hub for USB communication
if(PDQStartScan(0) == 0)
{
                        if(PDQReadScan(sensorData, 12, 1000)==0)
                        {
                                SensorA_Xdifference = -sensorData[0];
                                SensorA_Ydifference = sensorData[1];
                                SensorA_TotalPower = sensorData[2];
                                SensorB_Xdifference = -sensorData[3];
                                SensorB_Ydifference = sensorData[4];
                                SensorB_TotalPower = sensorData[5];
                                SensorC_Xdifference = -sensorData[6];
                                SensorC_Ydifference = sensorData[7];
                                SensorC_TotalPower = sensorData[8];
                                SensorD_Xdifference = -sensorData[9];
                                SensorD_Ydifference = sensorData[10];
                                SensorD_TotalPower = sensorData[11];
                        }
                        PDQStopScan();
}
                USBUninit();
}
}
```

Part 5. Troubleshooting

5.1. In the Win2000 and XP Operating System

The USB device driver file WINDRVR6.SYS must be located in *C:\WINNT\system32\drivers*. For Windows 2000 and XP systems, the driver files WINDRVR6.INF and PDQ80S1.INF must be located in *C:\WINNT\INF*. You must have administrator privileges to install and run the application display program.

When the driver software is installed via the CDROM menu program, the driver files should install automatically without operator interaction. In some circumstances, the Installer program may request the path of one or more of the driver files. In this case, browse to the CDROM directory *\USB\driver* and select the file requested. Similarly, following driver software installation, upon connecting the hub to a USB port, the Found New Hardware wizard may launch. In this case, follow the instructions and when prompted, allow the wizard to find the best driver for the device. If the wizard is unable to find the driver files, browse to the CDROM directory *\USB\driver* and select the driver file the wizard is requesting.

Once the software and drivers are properly installed and after launching the Beam Alignment Display program *QDdisplay.exe*, open the "Device" pull-down menu and select "Open Device". With the hub connected to a USB port, you should see the USB Port Status LED lit Green. If the USB Port Status LED fails to light Green and/or after some period of time is Red, make sure the hub is connected to a USB port.

5.2. Running the Display Program

Under normal operating conditions, when running the Beam Alignment Display program with a hub device connected to the USB port, the USB Port Status LED in the display window is lit green upon opening the device. This indicates the Beam Alignment Display program has successfully connected to the device via the USB port and is communicating properly. If the USB Port Status LED is Yellow, the program is attempting to connect with the hub device. If the USB Port Status LED is red, the program has failed to connect with the hub device. In this case, a popup window indicating "Failed to open USB port" may also appear. This usually indicates the hub is not connected to a USB port but in any case, the cause is a failure by the Beam Alignment Display program to communicate with the hub device.

5.3. Troubleshooting FAQ

- **The device is connected to a USB port but the application program cannot communicate with it. What is the problem?**

  In the Windows Control Panel, go to System Properties → Hardware → Device Manager, with the hub connected to a USB port, you should see a device icon called "Jungo" with an additional device icon branching from it called "Thorlabs PDQ80S1". Make sure the green LED adjacent to the USB port on the hub is lit. If you do not see the icons, the drivers are not installed properly or have been corrupted. The best way to proceed is to uninstall the driver and reinstall it. To uninstall the driver, close the application, disconnect your hub, insert the PDQ80S1 CDROM and mouse click the "Uninstall Driver" button. Reconnect your hub to the USB port. Verify the device icon: "Jungo" with "Thorlabs PDQ80S1" is not present. If it is present, left click on "Thorlabs PDQ80S1" and select Uninstall. To re-install the USB driver, insert the CDROM. When the installer menu program launches, click "Install Drivers". Follow the instructions indicated by the popup windows. A popup window will indicate when the driver installation has successfully completed. Do not reconnect the PDQ80S1 before this popup window has appeared. Once your hub has been reconnected, the driver icons should appear in the Device Manager.

- **The device is connected to a USB port, the driver icons appear in the Device Manager, and the application program appears to have opened the device but cannot communicate properly with the device.**

  You may have the device configured for Stand-Alone Mode enabled and the device has initialized as a stand-alone instrument. For a further explanation of Stand-Alone Mode, see pages 11 and 26. This may happen the first time the device is attached to a computer and registers with the host device manager (causing the "Found New Hardware" wizard to launch). It may also happen if the device is used with slower computers. A characteristic of this condition is the display window indicating the device is attached with the USB Port Status LED is Green but the device status bar does not show the proper device label and the indicated values for the Horizontal Alignment Window, Vertical Alignment Window, and Scan Interval are all 254. Unplug the device and reattach it to the USB port. If the display program has properly attached to the device, the device label should appear in the device status bar and the indicated values for the Horizontal Alignment Window, Vertical Alignment Window, and Scan Interval should be the default values read from the device. To prevent this problem from reoccurring, disable Stand-Alone Mode following the instructions on page 11.

- **The device is connected to a USB port with Stand-Alone Mode disabled, the driver icons appear in the Device Manager but the application program still cannot communicate with the device. What is wrong?**

  You may have multiple sessions of the display program open and the hub is attached to another session. Close any additional sessions of the display program, detach, and reattach the hub. In the display program window, open the "Device" pull-down menu and select "Open Device". Verify the USB Port Status LED is green. In addition, if not properly closed, it is possible for a Beam Alignment Display program session to remain running in the background with the hub attached. To see if this is happening, close all Beam Alignment Display program sessions and disconnect the hub. Open the Windows Task Manager and click on the Applications tab. If "USB Quadrant

Detector" appears in the task list, left click on it and click on End Task. Then reconnect the hub and launch a new session of the Beam Alignment Display program.

- **What is needed to install the display software?**

  You must have a PC with a USB port running Windows 2000, or XP. Earlier versions of Windows are not supported and will cause the installation to fail. The software comes on a CDROM. Mount the CDROM, and a menu program should launch. If this fails, run the CDROM program *\CD-Starter\CD-Starter.exe*. In order for you to install and run the software in the Windows 2000 or XP environments you must have administrative privileges.

- **Are there any simple means to tell that communication has been established between the PC and the hub?**

  If the Application and Driver installations are successful, when the hub is connected to the USB port, the Beam Alignment Display program is launched, a device is connected, the and USB Port Status LED is green.

Part 6. PDQ80S1 Specifications

6.1. Hardware Specifications

PSD Hub

Specification	Description
Input Power Requirements[2]	5 V @ <500 mA via USB Port
PC Interface	USB 2.0 High Speed (480 Mb/s)
Maximum Scan Rate	1,000 scans/sec
Number of Sensor Inputs	4
Measurement Resolution	12-bit
Input Voltage Range[3]	±5.6 V

Hub Sensor Alignment lamp for each sensor indicates beam alignment within a range dependent upon the user selectable horizontal and vertical alignment window settings. For the minimum alignment window settings, alignment is indicated for beam position centered within 0.7% sensor range. For the maximum alignment window settings, alignment is indicated for beam position centered within 18% of sensor range. The alignment lamp update rate is (Scan Rate)/4.

PDQ80A Sensor

Specification	Description
Wavelength Range	400 nm to 1050 nm
Sensor Geometry	4 Photodiodes in a Quadrant Configuration
Sensor Active Area	Ø 7.8 mm
Transimpedance Gain	10 kV/A
Detector Bandwidth	150 kHz
Max Photo Current	200 µA
Min Photo Current	27 µA for 1% Accuracy (5.5 µA for 5%)
Recommended Spot Size	Ø1 mm to Ø3.9 mm
Signal Output Voltage Range	±2 V (Min)
Operating Temperature	10 to 40°C

[2] May be powered either by a host PC USB port or by a USB compatible AC power converter connected via the USB port.
[3] Maximum allowable input range per signal. Each channel includes □X, □Y, and SUM signals. SUM signals will always be positive.

PDP90A Sensor

Specification	Description
Wavelength Range	320 nm to 1100 nm
Sensor Geometry	2D Tetra Lateral Pin Cushion Detector
Sensor Active Area	9 mm x 9 mm (0.354″ x 0.354″)
Transimpedance Gain	100 kV/A
Detector Bandwidth	15 kHz
Max Photo Current	40 µA
Min Photo Current	2.5 µA for 1% Accuracy (0.5 µA for 5%)
Recommended Spot Size	Ø0.2 mm to Ø7 mm
Signal Output Voltage Range	±4 V (Min)
Operating Temperature	10 to 40°C

6.2. Software Specifications

The PDQ80S1 Software is compatible with Windows 2000 and XP by means of system device drivers and an Application Program Interface provided via a Microsoft Visual C++ compatible static library. The following files are provided.

- USB Device Driver Files:
  - WINDRVR6.SYS
  - WINDRVR6.INF
  - PDQ80S1.INF
- USB Device Application Program Interface Files:
  - USB.lib
  - USB.h
  - PDQ.h

Display Graphics Specifications

Specification	Description
Sensor Alignment Display Range (X and Y Axes)	
PDQ80A	Up to ±2047 units corresponding to a beam displacement of up to the beam radius given a sensor surface geometry of ±3.9 mm from center
PDP90A	Up to ±2047 units correspond to the distance of the beam centroid from the center of the sensor given a sensor geometry of ±4.5 mm.
Display Update Rate:	~ 5 Hz.

PC System Requirements

- 750 MHz Pentium III or Higher with 128 MB RAM
- Hard-Drive with 16 MB free space
- CD-ROM
- Dedicated USB 2.0 High-Speed Port
- Mouse or Compatible Pointing Device
- Windows 2000 or XP Installed

6.3. PDQ80A Mechanical Drawing

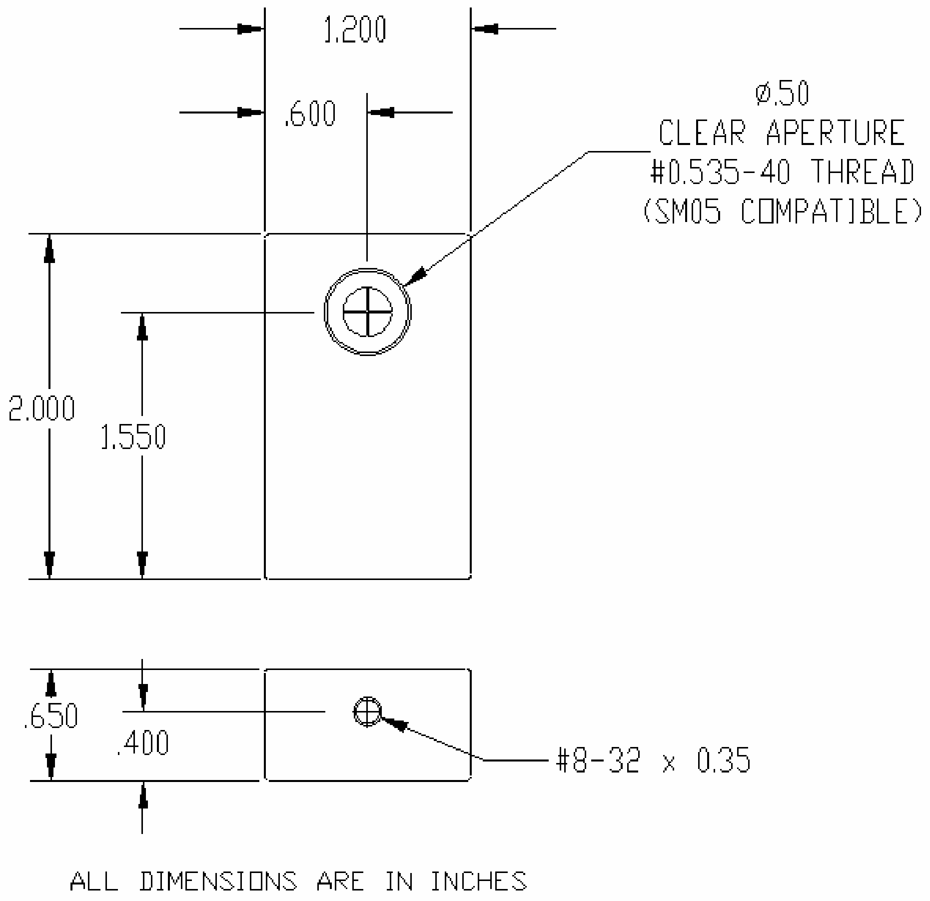

Figure 19: PDQ80A Enclosure Dimensions

Sensor Orientation

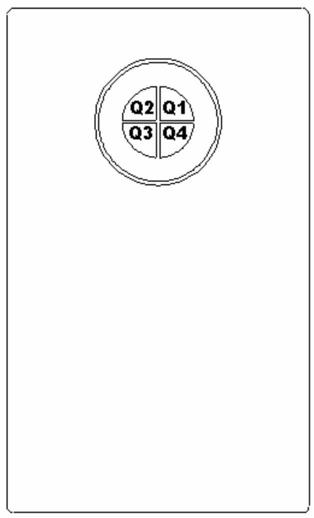

Figure 20: PDQ80A Sensor Orientation

6.4. PDP90A Mechanical Drawing

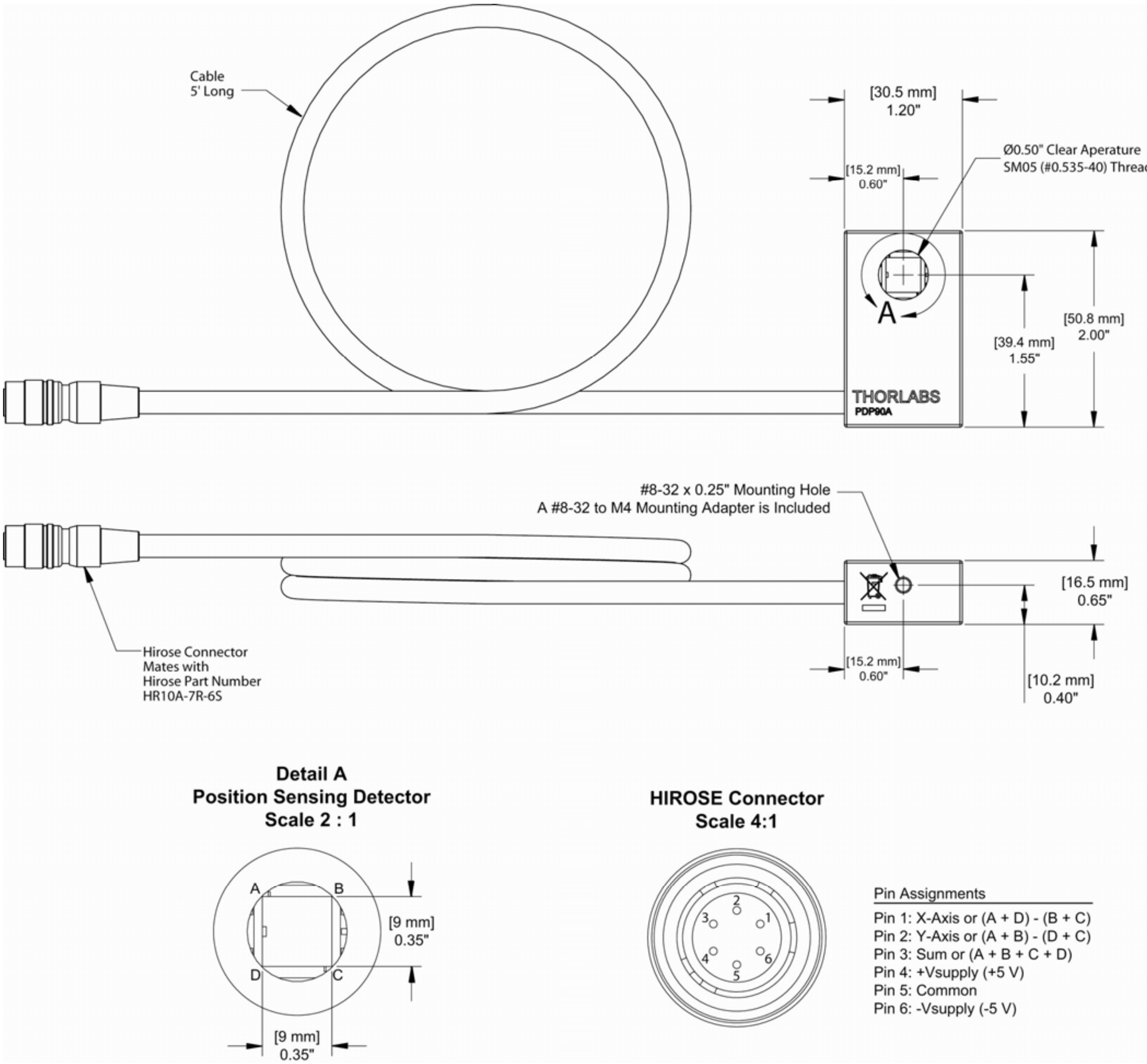

Figure 21: PDP90A Mechanical Drawing

Part 7. Regulatory

As required by the WEEE (Waste Electrical and Electronic Equipment Directive) of the European Community and the corresponding national laws, Thorlabs offers all end users in the EC the possibility to return "end of life" units without incurring disposal charges.

- This offer is valid for Thorlabs electrical and electronic equipment:
- Sold after August 13, 2005
- Marked correspondingly with the crossed out "wheelie bin" logo (see right)
- Sold to a company or institute within the EC
- Currently owned by a company or institute within the EC
- Still complete, not disassembled and not contaminated

Wheelie Bin Logo

As the WEEE directive applies to self contained operational electrical and electronic products, this end of life take back service does not refer to other Thorlabs products, such as:

- Pure OEM products, that means assemblies to be built into a unit by the user (e. g. OEM laser driver cards)
- Components
- Mechanics and optics
- Left over parts of units disassembled by the user (PCB's, housings etc.).

If you wish to return a Thorlabs unit for waste recovery, please contact Thorlabs or your nearest dealer for further information.

7.1. Waste Treatment is Your Own Responsibility

If you do not return an "end of life" unit to Thorlabs, you must hand it to a company specialized in waste recovery. Do not dispose of the unit in a litter bin or at a public waste disposal site.

7.2. Ecological Background

It is well known that WEEE pollutes the environment by releasing toxic products during decomposition. The aim of the European RoHS directive is to reduce the content of toxic substances in electronic products in the future.

The intent of the WEEE directive is to enforce the recycling of WEEE. A controlled recycling of end of live products will thereby avoid negative impacts on the environment.

Part 8. Thorlabs Worldwide Contacts

For technical support or sales inquiries, please visit us at www.thorlabs.com/contact for our most up-to-date contact information.

USA, Canada, and South America
Thorlabs, Inc.
sales@thorlabs.com
techsupport@thorlabs.com

UK and Ireland
Thorlabs Ltd.
sales.uk@thorlabs.com
techsupport.uk@thorlabs.com

Europe
Thorlabs GmbH
europe@thorlabs.com

Scandinavia
Thorlabs Sweden AB
scandinavia@thorlabs.com

France
Thorlabs SAS
sales.fr@thorlabs.com

Brazil
Thorlabs Vendas de Fotônicos Ltda.
brasil@thorlabs.com

Japan
Thorlabs Japan, Inc.
sales@thorlabs.jp

China
Thorlabs China
chinasales@thorlabs.com

UNLEASHED

THORLABS
www.thorlabs.com