SmartSensor™ Interface Module (SSIM) OEM Tools

User Manual

COHERENT®
AUBURN DIVISION

A Member of Coherent Photonics Group

Part Number: 1008005 Rev. A

©2001 Coherent Inc.

Table of Contents

Table of Contents

SSIM OEM Tools

General Description

The SSIM (Smart Sensor Interface Module) OEM Tools combined with the Coherent SmartSensor Interface Module, is a power analysis tool that interfaces with the full line of Coherent “Smart” detector heads. SSIM OEM Tools provides a unique combination of .ocx and .dll tools to provide many custom combinations of Power and Position analysis.

The Coherent Smart Detectors and SSIM module can be used with all CW lasers commonly manufactured today from the UV to the IR, with rated power from nanowatts to kilowatts simply by plugging the appropriate detector head into the SmartSensor Interface Module.

The SmartSensor Interface Module (SSIM) is a unique detector interface. This module will interface with any Coherent Thermal or Semiconductor SmartSensor. The SSIM contains the input amplifiers and the digital converters necessary to accurately analyze CW laser power. When this circuitry is combined with the capability to utilize Coherent SmartSensor technology, you now have a stand alone black box power meter. The USB or serial communication port will transmit the corrected CW laser power and beam position for display and analysis with the SSIM OEM Tools. The serial and USB interface is perfectly suited to communicate with any compatible host computer or OEM application. The SmartSensor Interface Module software can also be updated with the USB port.

Features

Features of the SSIM OEM Tools are described below.

CW Power Measurement

The SSIM OEM Tools supports CW power measurement. (See Appendix A for a list of heads supported.)

Semiconductor and Thermal Detectors

SSIM OEM Tools are compatible with Semiconductor Sensors and the full range of Coherent's patented thermal SmartSensor Heads, providing power measurement capability to 5 KW.

Versatility

More than just a power meter, the SSIM OEM Tools also provide beam position and formatted power output.

Hot Swapping

The SSIM OEM Tools allows swapping of detector heads and SSIM modules while the application is running. The SSIM OEM tools will provide events to detect when a module or detector are connected or disconnected.

SmartSensor™ Technology

SmartSensor detector heads utilize Coherent's SmartSensor™ Technology. Each SmartSensor has an EEPROM that stores the characteristics and calibration data for the detector. The detector information is read by the SmartSensor Interface Module at start-up. This will eliminate the need to make manual changes associated with unique detector settings.

Beam Alignment

Thermal disk CW sensors provide a quadrant display of beam position on the detector head. Centering the beam on the detector head achieves maximum accuracy.

Broad Wavelength Range

Coherent SmartSensors cover the spectrum from 0.19 to 10.6 µmeters.

Ease of Use

The SSIM OEM Tools contain Commands, Events, Methods and Properties to allow for simplified use in a custom application. Automated setup and error condition detection are built into the tools.

Portability

The compact, lightweight SmartSensor Interface Module can be easily mounted in a variety of locations

Reliability

The SmartSensor Interface Module is designed to withstand the rigors of process control or system integration. Coherent's rugged detector head design has been the industry standard for more than 30 years.

Accuracy

The combination of SmartSensor Technology, laser wavelength entry, and accurate beam positioning information create a highly accurate laser measurement system.

Software Installation

Description

The SSIM OEM Tools software installation provides the following items:

Simple Meter and SSIM Terminal sample programs

Source Code for the Simple Meter and SSIM Terminal programs

HTML Help File

SSIM OEM Tools User Manual

SSIM Update (Firmware Update Utility)

SSIM Firmware

Ssimusb.ocx

Ssimusb.dll

NOTE: The Ssimusb.ocx and the Ssimusb.dll will be installed in the Windows System directory. The remaining files will be loaded in the default or user specified installation location.

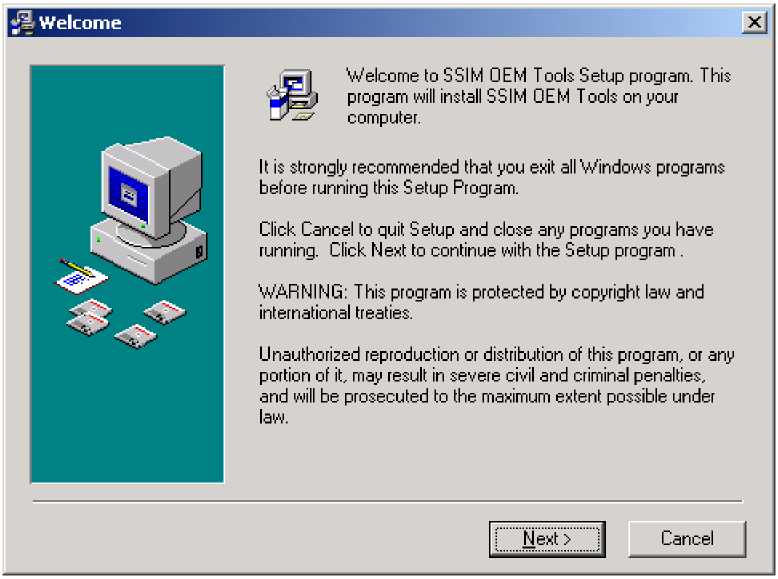
Setup Requirements

The Smart Sensor Interface Module will operate via USB on any USB compatible computer running Windows 98, Windows Me, Windows 2000 or Windows XP. Windows 95 systems will support serial communication. 20 MB of free hard disk space will be required. The hardware required to support your Windows operating system will also support the SSIM and the associated software.

Installation Procedure

1. Insert the SSIM OEM Tools CD in the computer CD drive. The auto-run process should begin the installation automatically. If the auto-run does not start automatically, select "Run" from the Start Menu and type the CD drive letter followed by \setup.exe then press enter.

2. The following Welcome screen will appear.



Welcome to SSIM OEM Tools Setup program. This program will install SSIM OEM Tools on your computer.

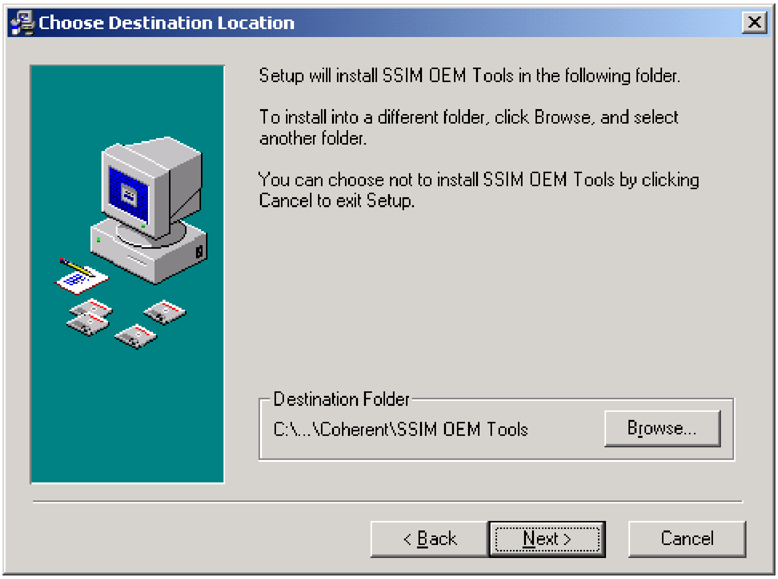
It is strongly recommended that you exit all Windows programs before running this Setup Program.

Click Cancel to quit Setup and close any programs you have running. Click Next to continue with the Setup program .

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

3. The “Next” button will allow you to select the desired path for installation. The default path is C:\Program Files\Coherent\SSIM OEM Tools.



Setup will install SSIM OEM Tools in the following folder.

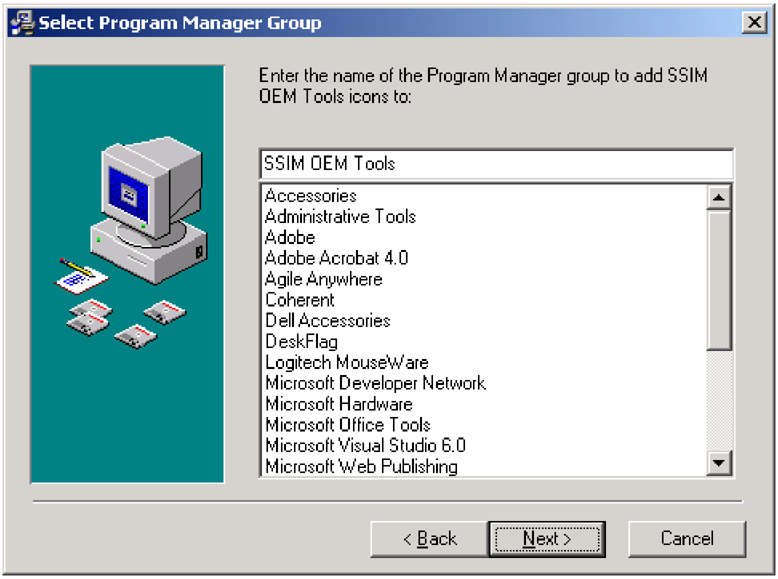
To install into a different folder, click Browse, and select another folder.

You can choose not to install SSIM OEM Tools by clicking Cancel to exit Setup.

Destination Folder

C:\...\Coherent\SSIM OEM Tools

4. The next step will display the name shown by the Windows Program Manager.



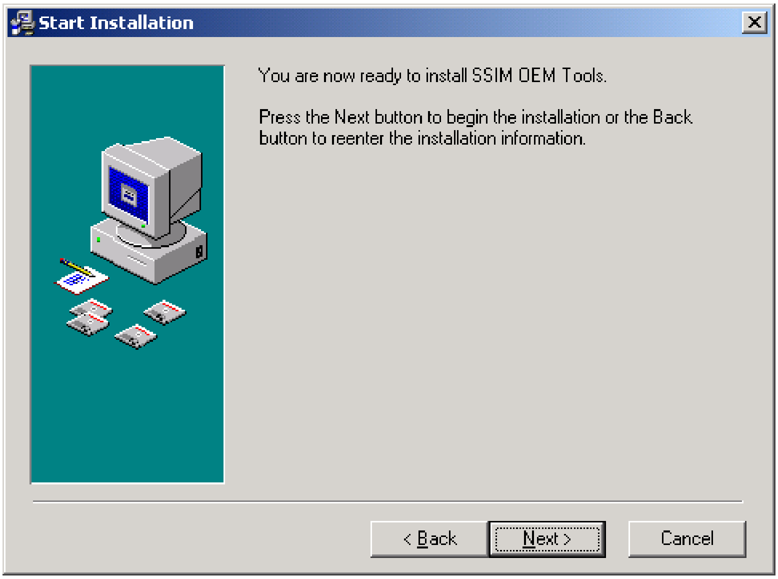
Select Program Manager Group

Enter the name of the Program Manager group to add SSIM OEM Tools icons to:

SSIM OEM Tools

Accessories
Administrative Tools
Adobe
Adobe Acrobat 4.0
Agile Anywhere
Coherent
Dell Accessories
DeskFlag
Logitech MouseWare
Microsoft Developer Network
Microsoft Hardware
Microsoft Office Tools
Microsoft Visual Studio 6.0
Microsoft Web Publishing

5. You are now ready to load the SSIM OEM Tools, supporting files and documentation.

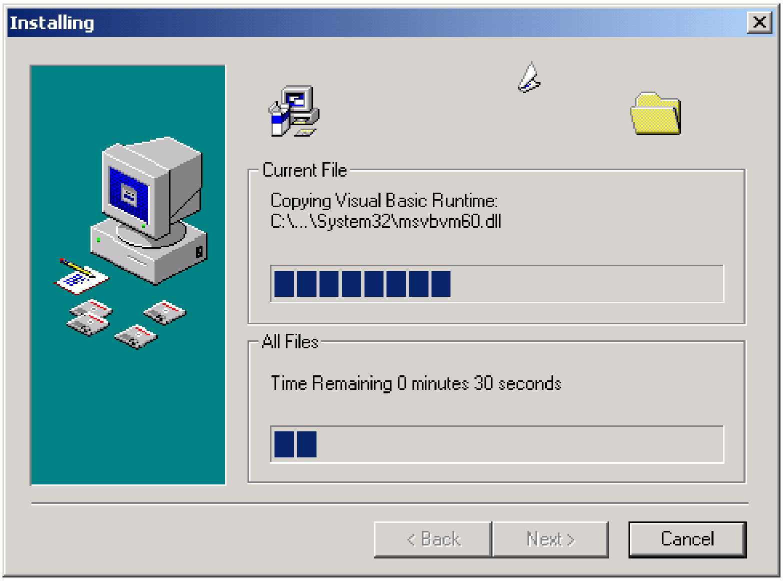


Start Installation

You are now ready to install SSIM OEM Tools.

Press the Next button to begin the installation or the Back button to reenter the installation information.

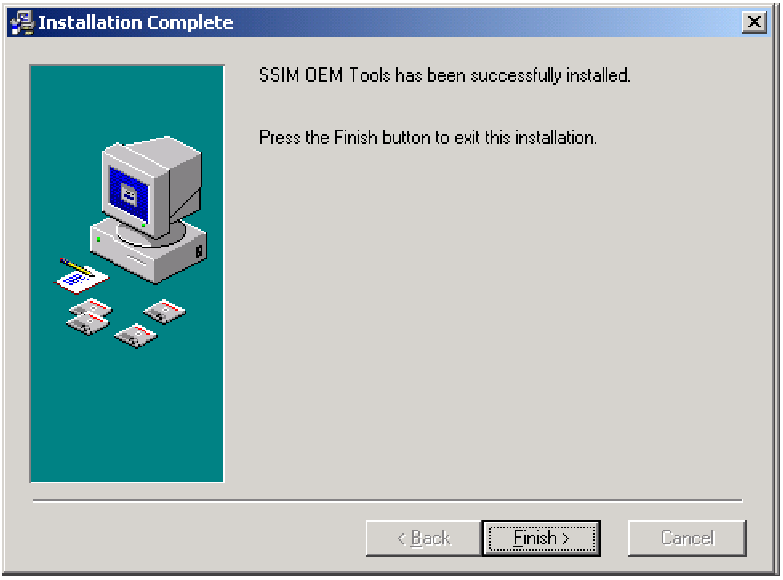
6. The “Next” button will begin the file transfer process.



Copying Visual Basic Runtime:
C:\...\System32\msvbvm60.dll

Time Remaining 0 minutes 30 seconds

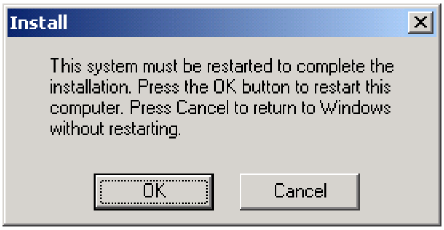
7. When the files are loaded, click the “Finish” button to complete the installation.



SSIM OEM Tools has been successfully installed.

Press the Finish button to exit this installation.

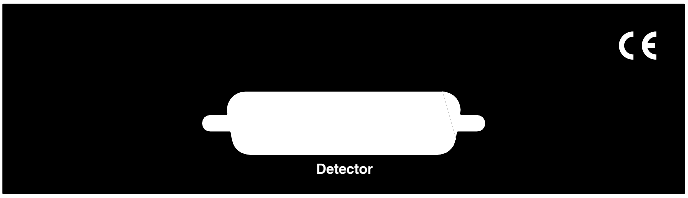
8. You must Restart your computer to complete the installation.



This system must be restarted to complete the installation. Press the OK button to restart this computer. Press Cancel to return to Windows without restarting.

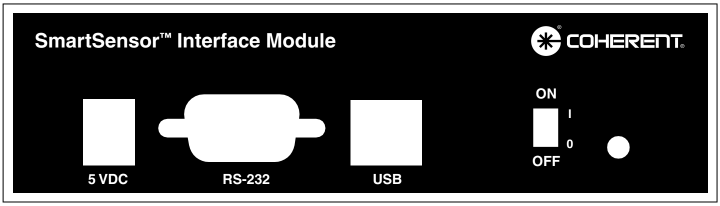
Controls & Connections

Input Panel



Detector Input – Connection to the Coherent SmartSensor Head using a DB-25 connector.

Output Panel



5 VDC Power Input – Connect the provided 5V Power supply. The power supply is designed for the following AC input requirements: 100-240V, ~0.2A, 50-60 Hz. The 5V input also provides battery charging capability for the SmartSensor Interface Module. When the 5V supply is connected to the Interface Module, the battery will be charging regardless of the power switch position. The USB connection does not provide battery charging capability.

NOTE: A fully charged battery will typically operate the SmartSensor Interface Module for 12 hours continuously. Do not exceed this 12 hour cycle without recharging. A low battery condition requires an overnight charging cycle (approximately 14 hours).

RS-232 Input/Output – Connect the Pocket PC serial interface cable to this DB-9 connector.

USB Input/Output – Connection for SmartSensor Interface Module to PC based applications.

Power Off/On – Switches input power on and off to the SmartSensor Interface Module. Utilize this switch to reboot the SmartSensor Interface Module if necessary.

Power Indicator LED – The LED will be illuminated when the 5V power is active and the power switch is in the ON position.

SmartSensor Interface Module Software Update

Description

The SmartSensor Interface Module (SSIM) contains the software required to interface with Coherent SmartSensors. The SSIM software will manage the internal circuitry, SmartSensor calibration information, calculations and data transmission. If any function is changed or a feature is added, the SSIM software can be upgraded by the user. The update software is installed during SSIM OEM Tools software installation.

System Requirements

The host computer requires an active USB port for the SSIM software update. The software update can be accomplished with the following Windows® operating systems: Windows 98, Windows 2000, Windows ME and Windows XP.

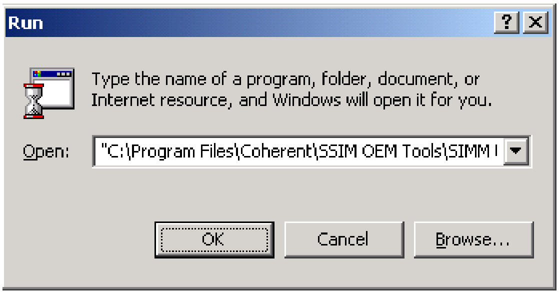
Software Information

The SSIM software update requires the software update file (ssimx.hex). The x in the file name corresponds to the current revision of the file. The ssim.hex file can be obtained from a floppy disk, E-Mail or the Coherent Product Web Site. The location of the update software is specified during the SSIM OEM Tools software installation. The default location is C:\Program Files\Coherent\SSIM OEM Tools.

SSIM Software Update Procedure

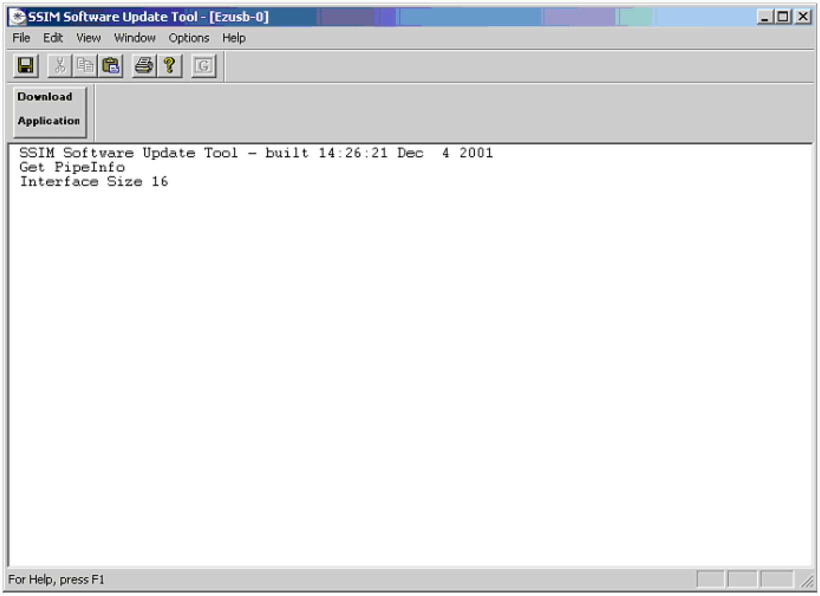
1. With the host PC running, connect a USB cable between the SmartSensor Interface Module (SSIM) and the host PC.
2. Move the SSIM power switch to the ON position. The host PC will detect the SSIM as new hardware. If the proper USB drivers do not reside on the host PC, you may be prompted to load them. The USB drivers are located on the SSIM OEM Tools installation CD. Place the SSIM OEM Tools installation CD in the host computer CD drive. Type the CD drive letter in the Windows dialog box, which is asking for the location of the USB driver files, and click OK.
3. To start the SSIM update software, select “Run” from the Windows Start Menu. Type C:\Program Files\Coherent\SSIM OEM Tools\SSIM Update.exe and then click OK. If the SSIM OEM

Tools were not installed in the default location, alter the C:\Program Files\Coherent\SSIM OEM Tools appropriate to the user specified location.


Run
?
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open:
OK
Cancel
Browse...

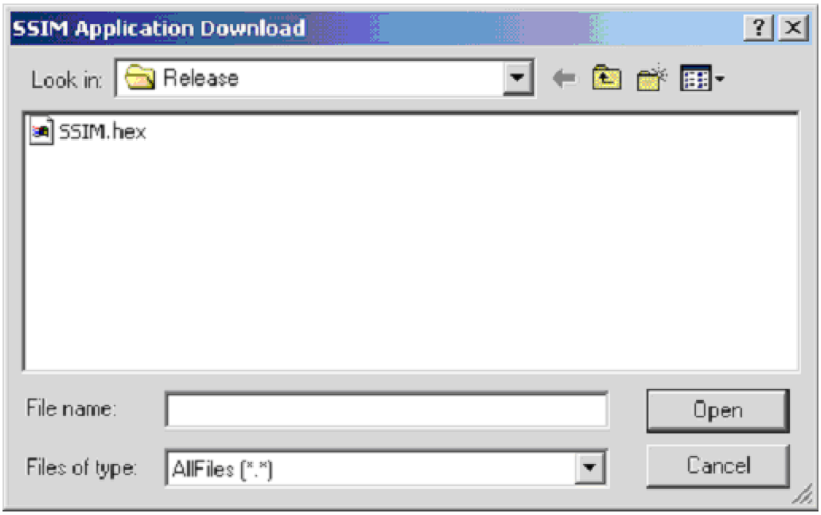
Open: "C:\Program Files\Coherent\SSIM OEM Tools\SIMM I

4. The following screen should appear. Click on the large Download Application button on the screen.



```
SSIM Software Update Tool - built 14:26:21 Dec  4 2001
Get PipeInfo
Interface Size 16
```

5. When the Download Application button is clicked, a standard Windows Open dialog box will appear. Use this box to specify the location of the ssim.hex file. The example shown below would correspond to a floppy disk update.



Look in: Release

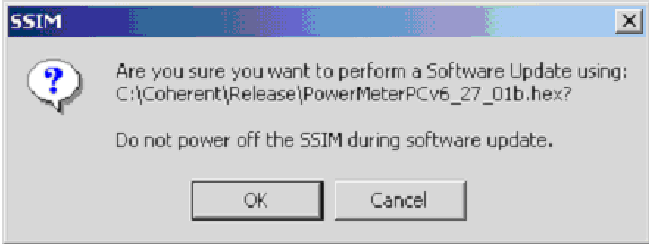
SSIM.hex

File name:

Files of type: AllFiles (*.*)

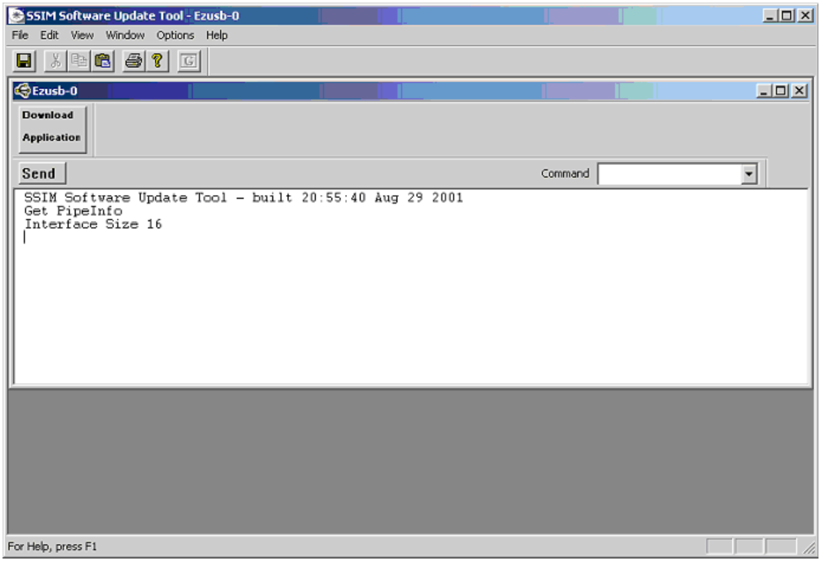
6. When the Open button is clicked, the file will be loaded in the SmartSensor Interface Module. This process will take approximately 30 seconds for the update to complete.

NOTE: Do not turn off power to the SmartSensor Interface Module during the software update process.



Are you sure you want to perform a Software Update using:
C:\Coherent\Release\PowerMeterPCv6_27_01b.hex?

Do not power off the SSIM during software update.



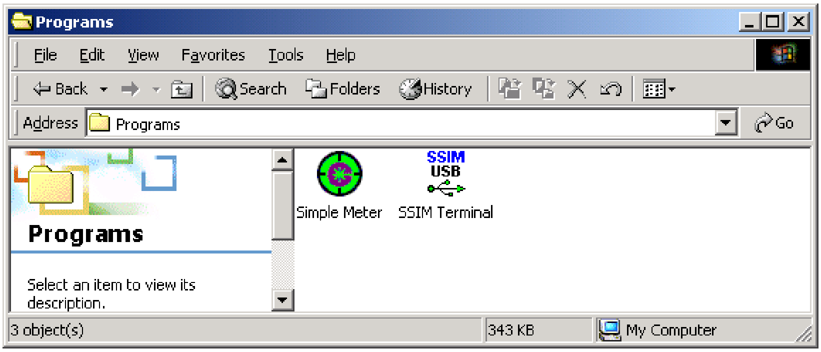
SSIM Software Update Tool - Ezusb-0
File Edit View Window Options Help
Ezusb-0
Download
Application
Send
Command
For Help, press F1

```
SSIM Software Update Tool - built 20:55:40 Aug 29 2001
Get PipeInfo
Interface Size 16
```

7. SSIM Update Complete

Sample Programs

The SSIM OEM Tools installation provides two sample programs. These programs are named "Simple Meter" and "SSIM Terminal". These two programs are located in the Sample Program folder in the specified installation path. The contents of the Sample Programs folder are shown below.



Address: Programs

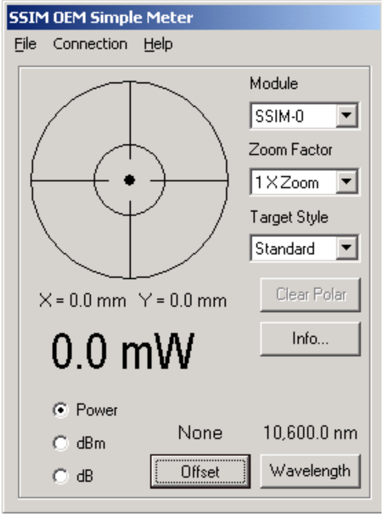
Programs

Select an item to view its description.

3 object(s) | 343 KB | My Computer

Simple Meter

The Simple Meter Program is an example of basic power and position display. This program provides Connection Options, Zoom Factor, Target Styles, Detector Information, Display Units, Offset and Wavelength adjustment. The main screen from the Simple Meter is shown below.



Module: SSIM-0

Zoom Factor: 1 X Zoom

Target Style: Standard

X = 0.0 mm Y = 0.0 mm

0.0 mW

● Power
○ dBm
○ dB

None

10,600.0 nm

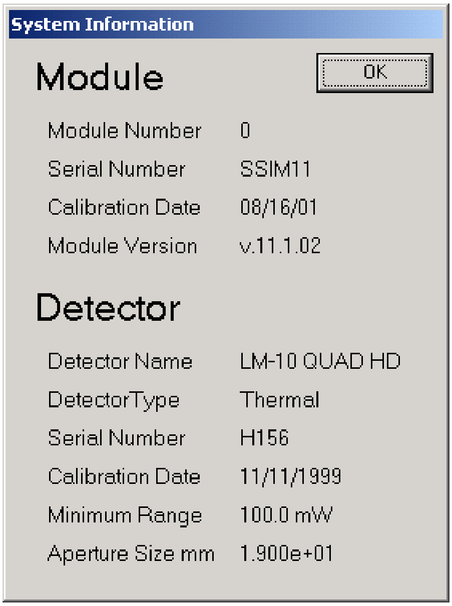
The Module selection box will allow you to select any SSIM that is currently connected.

The Zoom Factor selection box will allow you to select the displayed Alignment sensitivity. The Zoom factor values range from 1x to 32x.

The Target Style selection box will control the style of the displayed target. The target style options consist of the following:

> Standard – Full aperture display with the area of accuracy shown as a central circle.
>
> Alignment – The Alignment setting will remove the central circle and is intended to be used with a Zoom Factor other than 1x.
>
> Polar - The Polar display will show a trace representing the position history. When the Polar selection is running, the “Clear Polar” button will be active. This button will clear all current data.

The “Info…” button will display the current module and detector information. An example of this information screen is shown below.



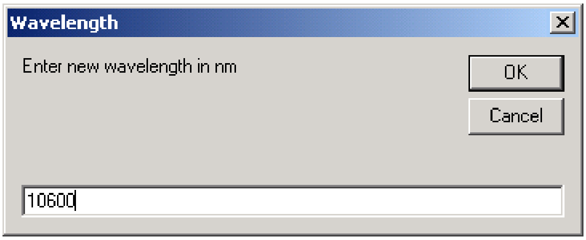
Module

Module Number	0
Serial Number	SSIM11
Calibration Date	08/16/01
Module Version	v.11.1.02

Detector

Detector Name	LM-10 QUAD HD
DetectorType	Thermal
Serial Number	H156
Calibration Date	11/11/1999
Minimum Range	100.0 mW
Aperture Size mm	1.900e+01

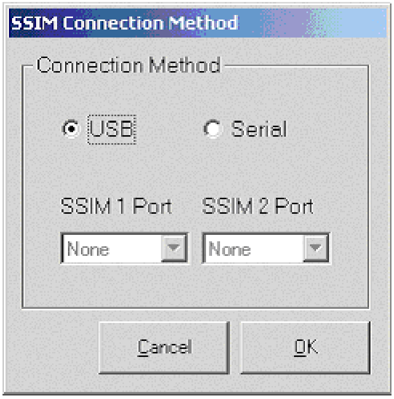
The Wavelength button will allow you to enter the appropriate Wavelength to be utilized in all power calculations. The Wavelength entry dialog is shown below.



Enter new wavelength in nm

10600

The Connection Menu contains a "Communication Method" item to select the type of communication to be utilized between the host computer and the SSIM. The Communication Method dialog is shown below.



Connection Method

● USB ○ Serial

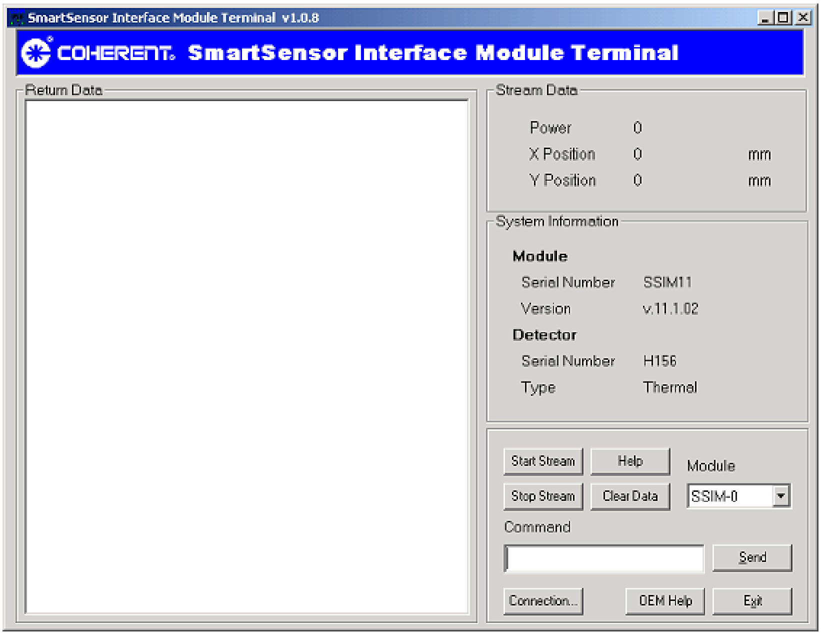
SSIM 1 Port: None

SSIM 2 Port: None

Exit the Simple Meter by selecting "Exit" from the File menu.

SSIM Terminal Program

The SSIM Terminal program will allow you to communicate directly with the SSIM. A complete list of commands begins on page 30. The SSIM Terminal program screen is shown below. The large window on the left will display commands sent and the results returned from the SSIM. When the Start Stream button is clicked, the streaming results will be displayed in the upper right hand portion of the screen. The module and detector information will be displayed below the streaming data display. The Terminal Program controls are located in the lower right hand corner of the screen.



SmartSensor Interface Module Terminal v1.0.8

COHERENT. SmartSensor Interface Module Terminal

Stream Data

Power 0

X Position 0 mm

Y Position 0 mm

System Information

Module

Serial Number SSIM11

Version v.11.1.02

Detector

Serial Number H156

Type Thermal

Module SSIM-0

The Terminal Program controls include the following:

Start Stream – The “Start Stream” button will prompt the selected SSIM to continuously stream data at a rate of 10 Hz. The streaming data will be displayed in the upper right hand corner of the Terminal Program screen.

Stop Stream – The “Stop Stream” button will terminate the data streaming process when clicked.

Help – The “Help” button will send the “h” command to the SSIM and a list of Terminal commands will be displayed in the “Return Data” window.

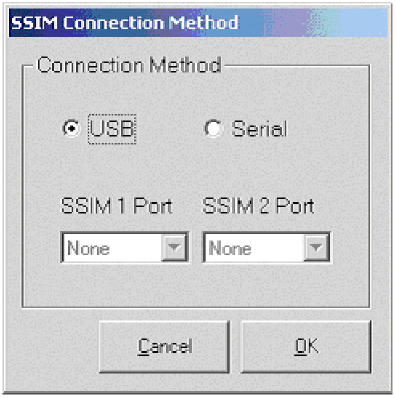
Clear Data – The “Clear Data” button will clear all of the commands and results displayed in the large SSIM Terminal program window.

Module – The Module selection box will allow you to select any SSIM that is currently connected and identified.

Command – The command box will allow you to enter commands to be sent to the SSIM. Type the desired command and click "Send". A complete list of commands is shown on page 30. The "Help" button will also display these commands in the "Return Data" window in the Terminal program.

Send – The "Send" or Enter button will send the command currently displayed in the Command box.

Connection – The "Connection" button will display the "Connection Method" dialog box. This box will allow you to select the desired method to communicate with the connected SSIMs. The Connection Method dialog is shown below.



SSIM Connection Method

Connection Method

● USB ○ Serial

SSIM 1 Port: None

SSIM 2 Port: None

OEM Help – The "OEM Help" button will access the .html help for the SSIM OEM Tools.

Exit – The "Exit" button will exit the terminal program when clicked.

Program Code

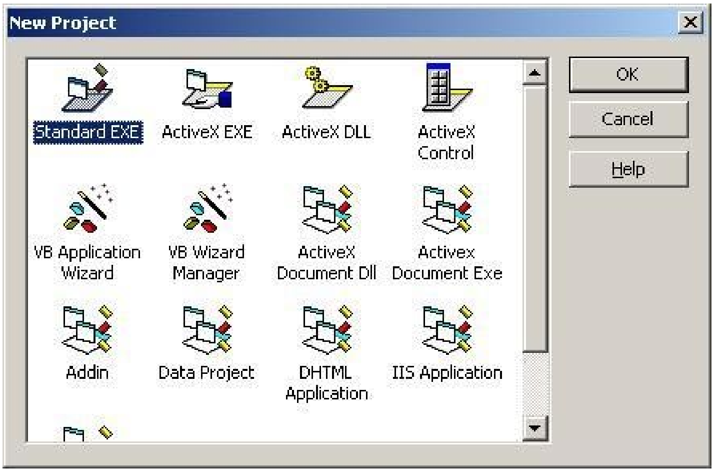
The Program Code for the Simple Meter and the Terminal Program are located in the "Source Code" folder. This folder is in the location specified during the SSIM OEM Tools installation. This source code will provide the information necessary to help develop a custom application utilizing the SSIM.

Tutorial

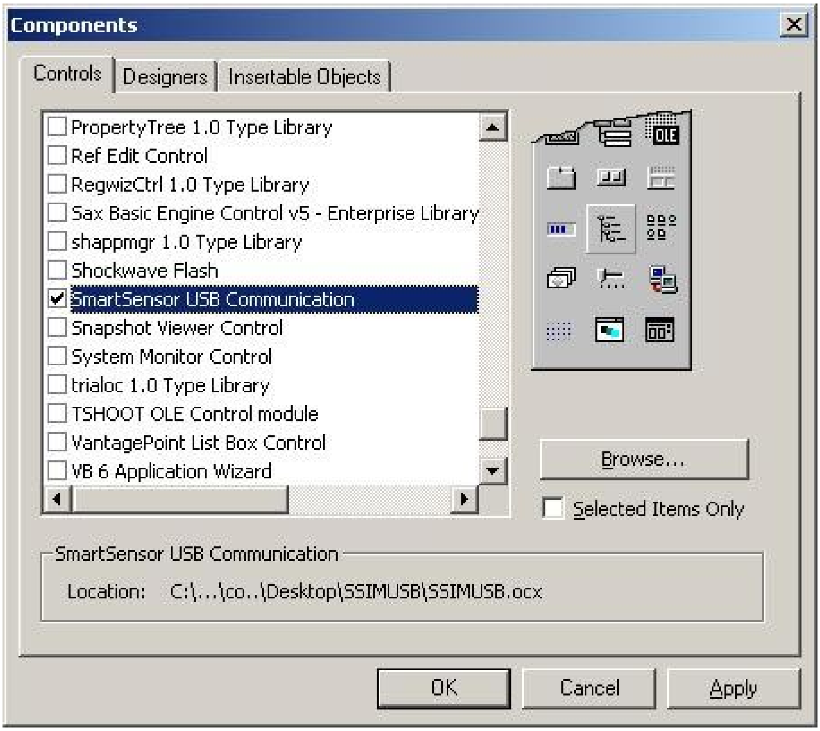
The following is a breif tutorial on the use of the SSIM OEM Tools. The following example illustrates how to link the OEM tools into VisualBasic and establish communications with the SmartSensor Interface Module.

Add SSIM OEM Tools to your Project

1.0 After installation of the SSIM OEM Tools, launch VisualBasic and create a standard EXE.



1.1 Under the Project menu select *Components. Under the Controls Tab,* Select the ***SmartSensor USB Communication*** component.



Components

Controls | Designers | Insertable Objects

- □ PropertyTree 1.0 Type Library
- □ Ref Edit Control
- □ RegwizCtrl 1.0 Type Library
- □ Sax Basic Engine Control v5 - Enterprise Library
- □ shappmgr 1.0 Type Library
- □ Shockwave Flash
- ☑ SmartSensor USB Communication
- □ Snapshot Viewer Control
- □ System Monitor Control
- □ trialoc 1.0 Type Library
- □ TSHOOT OLE Control module
- □ VantagePoint List Box Control
- □ VB 6 Application Wizard

Browse...

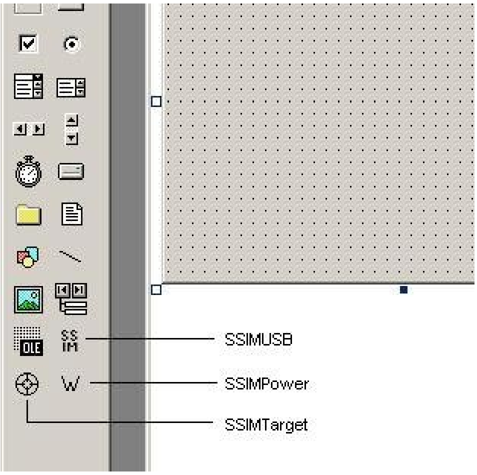
□ Selected Items Only

SmartSensor USB Communication

Location: C:\...\co..\Desktop\SSIMUSB\SSIMUSB.ocx

OK | Cancel | Apply

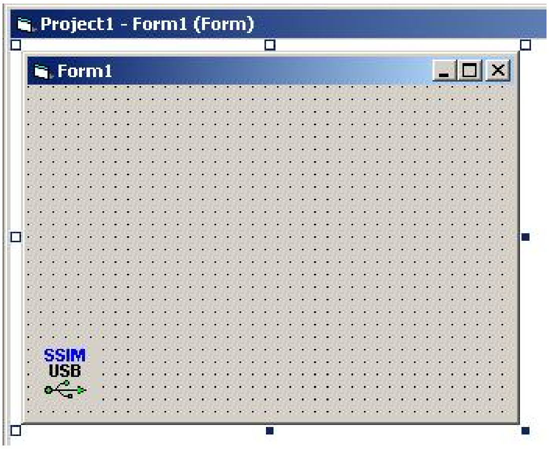
1.2 Three tools will be added to your toolbar (SSIMUSB, SSIMPower and SSIMTarget)



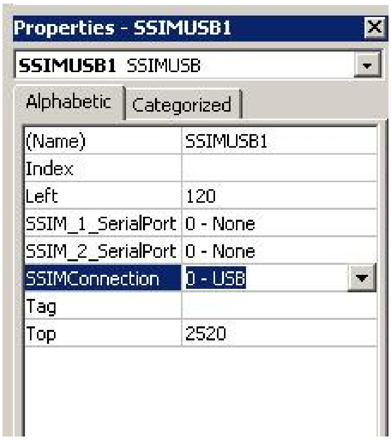
Communicate to the SmartSensor Interface Module

The SSIMUSB object is utilized to communicate to one or more SmartSensor Interface Modules. This object will not be visible at run time.

2.0 Select the **SSIMUSB** object from the toolbar and place on the blank form.



2.1 Select the connection method for your SmartSensor Interface Module (USB or Serial). Please note that a selection of **Serial** will also require you to select the corresponding serial port.



(Name)	SSIMUSB1
Index	
Left	120
SSIM_1_SerialPort	0 - None
SSIM_2_SerialPort	0 - None
SSIMConnection	0 - USB
Tag	
Top	2520

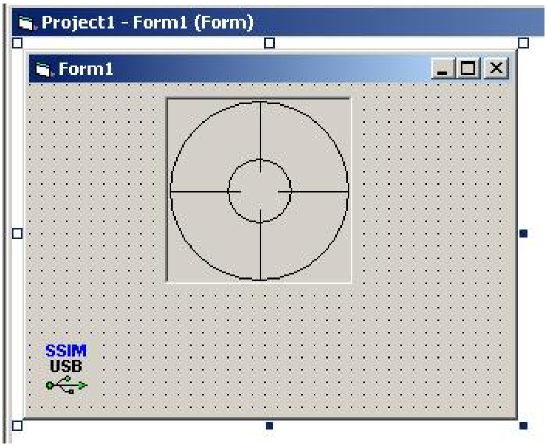
2.2 The **Connection Method** and **Serial Port**

The **Communication Property** can also be set at run time, however, keep in mind that the SSIMUSB object can support only one connection method at a time.

Show position information

The SSIMTarget object is used to display alignment/position information sent back from the SmartSensor Interface Module.

3.0 Select the **SSIMTarget** object from the toolbar and place it on the form.

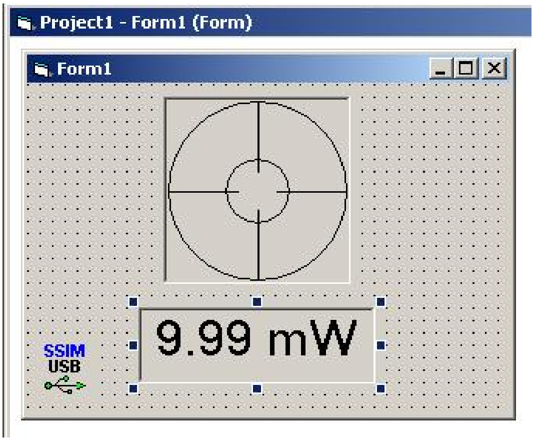


3.1 Select the desired **Target Style**, **Border Style**, **Target Color** and **Dot Color** from the properties window. These properties can also be set at run time.

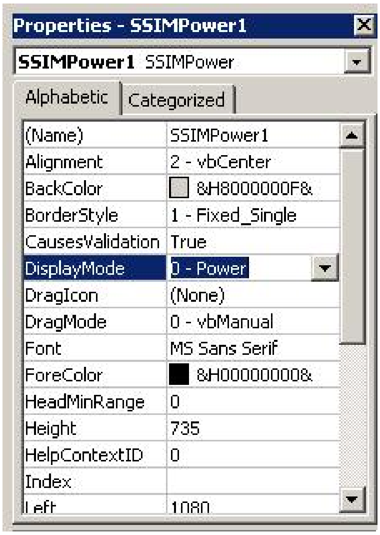
Show Power Information

The SSIMpower object is used to display properly formatted power information in watts, dB or dBm.

4.0 Select the **SSIMPower** object from the toolbar and place on the form.



4.1 Set the SSIMPower object to the desired size. Set the alignment property to the desired alignment. Adjust the font property to the desired font, size and style. Select the desired Display Mode (Power, dB or dBm). These properties can also be set at run time.



(Name)	SSIMPower1
Alignment	2 - vbCenter
BackColor	□ &H8000000F&
BorderStyle	1 - Fixed_Single
CausesValidation	True
DisplayMode	0 - Power
DragIcon	(None)
DragMode	0 - vbManual
Font	MS Sans Serif
ForeColor	■ &H00000000&
HeadMinRange	0
Height	735
HelpContextID	0
Index	
Left	1080

Program code

Now that all the OEM tools are loaded into the project it is time to get them talking.

5.0 Under General Declarations enter the following code.

```
Option Explicit

‘Set a counter to keep track of commands
Dim intCommandNumber    As Integer
’Used to keep track of module number for program exit
Dim intModuleNumber     As String
```

The intCommandNumber variable is used to keep track of commands sent to the SmartSensor Interface Module. All command responses returned from the SmartSensor Interface module enter the program at the **ReturnData** event. The intCommandNumber variable keeps track of the responses.

5.1 Under SSIMUSB1 select the **ModuleAdded** event and enter the following code.

```
Private Sub SSIMUSB1_ModuleAdded(ModuleNumber As String)

    ‘Save module number for program exit
    intModuleNumber = ModuleNumber

End Sub
```

The intModuleNumber variable is used to store the SmartSensor Interface object module number. This number is needed in the program exit portion of the code to stop the data stream.

5.2 Under SSIMUSB1 select the **DetectorConnect** event and add the following code.

```
Private Sub SSIMUSB1_DetectorConnect(ModuleNumber
As String)

    'Send command to get detector aperture size
    SSIMUSB1.SendCommand ModuleNumber, "app"

    'Set command number to 1 or first command
    intCommandNumber = 1

End Sub
```

The above code will send the aperture size command request to the SmartSensor Interface module when the SSIM detects a detector head. Once the command is sent the command number is set to 1 to indicate that the first command has been sent.

5.3 Under SSIMUSB1 select the **ReturnData** event and add the following code.

```
Private Sub SSIMUSB1_ReturnData(ModuleNumber As
String, Data As String)

    Select Case intCommandNumber

        Case 1  'Case for aperture command response

            'Set aperture size property of target object
            SSIMTarget1.Aperture_mm = Data
            'Send command to get detector minimum range
            SSIMUSB1.SendCommand ModuleNumber, "rmi"
            'increment command counter
            intCommandNumber = 2
        Case 2  ‘Case for minimum range command response

            ‘Set head min range property of power object
            SSIMPower1.HeadMinRange = Data
            ‘Send command to start data stream
            SSIMUSB1.SendCommand ModuleNumber, “dst”

        Case Else

    End Select

End Sub
```

The purpose of the Select Case statement is to act as a command decoder for the return data. The first time this code executes, the detector aperture size is returned from the SmartSensor Interface Module (step 5.2). The IntCommandNumber variable is set to 1 for the Case 1 code to be executed and the aperture size will be loaded into the target aperture size property. The detector min range command is then sent to the SSIM and intCommandNumber is set to 2. This is done when the minimum range response is returned and the sub will execute the code in Case 2. The code in Case 2 will load the minimum range response into the power object head min range property and the command to start the data stream will be sent.

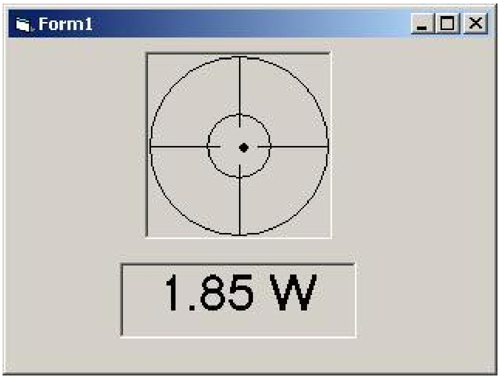
5.4 Under SSIMUSB1 select the **StreamData** event and add the following code.

```
Private Sub SSIMUSB1_StreamData(ModuleNumber As
String, Power As Single, XPosition As Single, YPosition As
Single)

    ‘Send power return to power object
    SSIMPower1.DisplayNewReading Power

    ‘Send position return to target object
```

```
        SSIMTarget1.UpdatePosition XPosition, YPosition

    End Sub
```

All streaming data will be returned in this event, separated into individual variables for power and position. The power return is sent to the DisplayNewReading method of the **SSIMPower** object. The X and Y positions are sent to the **UpdatePosition** method of the **SSIMTarget** object.

5.5 Connect a SmartSensor Interface Module to the computer. Save and run the program.



As streaming data is returned to the program it is automatically sent to the SSIMPower and SSIMTarget objects.

Command Set for SmartSensor Interface Module (SSIM)

The following commands are utilized to communicate directly to the SSIM. The commands can be imbedded into a program or tested with the provided SSIM Terminal program.

*rst	**Description:** Command to reset SSIM. Firmware will be reloaded into system RAM **Return:** None **Note:** This action will cause the SSIM to redetect and load detector information if adetector is present.
*ind v?	**Description:** Request for hardware identification. **Return:** Hardware description Example: "**SmartSensor Interface Module**" **Description:** Request the SSIM firmware version. **Return:** Current firmware version Example: "**v.8.21.01a**"
vb?	**Description:** Request the SSIM firmware boot loader version. **Return:** Current firmware boot loader version Example: "**3**"
vp?	**Description:** Request the SSIM data stream protocol version. **Return:** Current data stream protocol version Example: "**v.8.21.01a**"
msn?	**Description:** Request the SSIM serial number. **Return:** SSIM serial number Example: "**1012F16**"
spd	**Description:** Command SSIM to toggle the noise algorithm on/off. **Return:** New status of noise algorithm "**speedup threshholding is off**" "**speedup threshholding is on**"

	Note:	This command will toggle the current state of the noise algorithm (if current state is on, sending spd command will turn algorithm off). Please refer to manual for more information on the noise algorithm
mcal?	**Description:**	Request for SSIM cal date
	Return:	Current SSIM cal date Example: "**10/1/01**"
spd?	**Description:**	Request the current state of the noise algorithm
	Return:	Current status of noise algorithm "**off**" "**on**"
	Note:	spd? Request will not change the state of the noise algorithm. Please refer to manual for more information on the noise algorithm
h	**Description:**	Request help menu
	Return:	Valid commands: (spaces are significant) *rst - restart system *ind - identify system app - get detector aperture cal - get detector calibration date df? - get detector family dpw - start streaming power only (pocket pc mode) dsp - stop data streaming dst - start streaming all data dt? - get detector name dxy - start streaming position only (pocket pc mode) h - show this help j <password> - enter calibration mode mfg - get date detector was manufactured pos - get x,y position in mm ppc - stream using pocket pc format

	pw?	- get power in watts
	rmi	- get detector minimum range
	rmx	- get detector maximum range
	sn?	- get detector serial number
	tmp	- get uncalibrated thermistor reading
	trm	- stream using terminal format
	v?	- get firmware version
	vp?	- get protocol version
	vb?	- get bootloader version
	wl?	- get detector default wavelength
	wv <float>	- set current wavelength in meters
	wv?	- get current wavelength in meters
	spd	- toggles speedup noise thresholding on and off
	spd?	- queries the status of the speedup noise thresholding
	Commands must be terminated with an ASCII carriage return.	

Detector Command Set

cal	**Description:**	Request the detector calibration date.
	Return:	Detector calibration date from detector EEPROM Example: "**1/12/2000**"
	Note:	Request requires that detector be present.
mfg	**Description:**	Request the detector manufacturing date.
	Return:	Detector manufacturing date from detector EEPROM Example: "**1/10/2000**"
	Note:	Request requires that detector be present.
df?	**Description:**	Request the detector family description.
	Return:	Thermal detector present "**thermal**" Semiconductor detector present "**quantum**" No detector present "**disconnect**"
dt?	**Description:**	Request the detector model name.
	Return:	Detector model name Example: "**LM-10 QD HD /1**"
	Note:	Request requires that detector be present.
wl?	**Description:**	Request the detector default wavelength in meters.
	Return:	Detector default wavelength Example: "**6.328e-07**"
	Note:	Request requires that detector be present.
wv?	**Description:**	Request the currently selected wavelength in meters.
	Return:	Current wavelength Example: "**5.145e-07**"
	Note:	Request requires that detector be present. The current value is stored in the SSIM EEPROM unless changed by "wv <float>" request. The

Detector Command Set

		SSIM will set the current wavelength to the detector default wavelength value if, during initialization of the detector, it is found that the current wavelength is outside the valid wavelength range of the detector.
wv <float>	**Description:**	Request SSIM to set the current wavelength. Example: "**wv 6.328e-7**"for 632.8nm
	Return:	Returns a confirmation of the requested wavelength. Example: "**6.328e-7**"
	Note:	Request requires that detector be present. If requested wavelength is greater than the upper limit, the SSIM will set and return the upper wavelength limit. If the requested wavelength is less than the lower limit, the SSIM will set and return the lower wavelength limit.
sn?	**Description:**	Request the detector serial number.
	Return:	Detector serial number Example: "**25J54**"
	Note:	Request requires that detector be present.
app	**Description:**	Request the detector aperture size in mm.
	Return:	Aperture size (mm) Example: "**1.900e+01**"
	Note:	Request requires that detector be present. Aperture size is a fixed value stored in the detector EEPROM
rmi	**Description:**	Request the detector minimum range in watts
	Return:	Detector minimum range Example LM-10: "**1.000e+00**"
	Note:	Request requires that detector be present. Dividing minimum range of detector by 100 can derive detector minimum detectable power.

Detector Command Set

rmx	**Description:**	Request the detector maximum range in watts
	Return:	Detector maximum range Example LM-10: “**1.000e+01**”
	Note:	Request requires that detector be present. If detector maximum range is exceeded the data stream status flag will change to “r” indicating head is over range. See data streaming commands for more information.
pw?	**Description:**	Request the current detector power reading in watts
	Return:	Current detector power reading Example: “**5.020e+01**”
	Note:	Request requires that detector be present.
pos	**Description:**	Request the current detector coordinate centering information in mm
	Return:	Current detector centering (X,Y) Example: “**0.125e+00,1.004e+00**”
	Note:	Request requires that detector be present. Centering position of 0,0 will be returned for quantum detectors.
tmp	**Description:**	Request the current uncalibrated detector thermistor A/D value (reference only)
	Return:	Current detector thermistor reading Example: “**32704**”
	Note:	Request requires that detector be present.

Data Stream Command Set

trm (Default)	**Description:**	Set terminal mode data stream format. Data stream is coma delimited and uncondensed.
	Return:	None
	Note:	Request may be sent at any time. Thermal detector (X,Y,Power,Status) “*__0.125e+00,1.000e+01,1.500e+00,c__” Quantum detector (Power,Status) “*__1.500e+00,c__” SSIM will return to trm data stream mode upon power up or restart Status: c = Detector is functioning correctly r = Detector is over maximum range t = Detector is over maximum recommended temperature.
ppc	**Description:**	Set condensed mode data stream format. Data stream is condensed. Comas removed and position data condensed.
	Return:	None
	Note:	This command is intended for Pocket PC applications. This command should not be utilized in an OEM environment. Command may be sent at any time. If negative sign is absent then coordinate is positive. First 3 characters are coordinate value, 4th character is exponent. ((-)XXXX(-)YYYYP.PPe-PS) X=X coordinate, Y=Y coordinate, P=Power, S=Status Thermal detector (XYPowerStatus) “**-278478448.38e+00,c**” Quantum detector (PowerStatus) “**8.38e+00c**” Decode for position information section of data stream. (-)XXXX = (-)XXX# = ((-)XXX/100)e-#

Data Stream Command Set

		-2784 = -2.78e-4 = **0.278mm** SSIM will return to trm data stream mode upon power up or restart. See dst command for Status definitions. Status: c = Detector is functioning correctly r = Detector is over maximum range t = Detector is over maximum recommended temperature.
dst	**Description:**	Command start of data stream. Start streaming all data.
	Return:	Data stream See trm and/or ppc command for data stream format.
	Note:	Command requires that detector be present. Power will be displayed in watts. Position will be displayed in mm. Data will stream at 10Hz Data will continue to stream until issue of dsp command or detector not present.
dsp	**Description:**	Request the stop of all streaming data. Stop all data streams.
	Return:	None
dxy	**Description:**	Command start of position data stream. Start streaming position information. (Thermal Detector Only) (PPC Mode Only)
	Return:	Data stream of position data Thermal detector (XYStatus) Example: "**-27847844.3c**"
	Note:	Command requires that detector be present. Position will be displayed in mm. Data will stream at 10Hz Data will continue to stream until issue of dsp command or until detector is unplugged from SSIM.

Data Stream Command Set

	Status: c = Detector is functioning correctly r = Detector is over maximum range t = Detector is over maximum recommended temperature.
dpw	**Description:** Command to start power data stream. Start streaming power information. (PPC Mode Only) **Return**: Data stream of position data (PowerStatus) Example: “**1.000e+01c**” **Note**: Request requires that detector be present. Power will be displayed in watts. Data will stream at 10Hz Data will continue to stream until issue of dsp command or until detector is unplugged from SSIM. Status: c = Detector is functioning correctly r = Detector is over maximum range t = Detector is over maximum recommended temperature.

SSIM OEM Tools

To aid in the development of an OEM application, Coherent is providing a custom ActiveX control. This control will provide your development environment with a set of tools. These tools will enable the programmer to quickly and efficiently communicate with the SmartSensor Interface Module. The SSIM OEM Tools will allow the programmer to easily design a custom application to collect, analyze and display power and position data in any desired method.

SSIM OEM Tools ActiveX control contains the following objects: **SSIMUSB, SSIMPower** and **SSIMTarget**

Objects

SSIMUSB Object

The **SSIMUSB** object is designed to provide a communication interface with one or more SmartSensor Interface Modules.

The **SSIMUSB** object contains the following methods, properties and events.

METHODS: **SendCommand**

PROPERTIES: **SSIMConnection, SSIM 1 SerialPort SSIM 2 Serial Port**

EVENTS: **ModuleAdded, ModuleRemoved, DetectorConnect, DetectorDisconnect, DetectorOverRange, DetectorOvertemp, ReturnData, StreamData**

SSIMPower Object

The **SSIMPower** object is designed to provide a simple and effective method of displaying power readings from a detector head.

The **SSIMPower** object contains the following methods and properties.

METHODS: **ConvertReading, DisplayNewReading, SetNewdBReference**

PROPERTIES: **DisplayMode, Alignment, BorderStyle, HeadMinRange, Offset**

NOTE: The default HeadMinRange value is set to 0. This will cause the object to display readings as low as 1nw for all detector heads. Coherent does not guarantee the accuracy of any readings below the detector's minimum range unless the HeadMinRange property is set to the detector's correct minimum range

SSIMTarget Object

The **SSIMTarget** object is designed to provide a simple and effective method of displaying position readings from a detector head.

The **SSIMTarget** object contains the following methods, properties and events

METHODS: **ClearPolar**, **RefreshTarget**, **UpdatePosition**

PROPERTIES: **Aperturemm**, **Dotcolor**, **PolarPlot**, **RelativeX**, **RelativeY**, **TargetColor**, **TargetStyle**, **ZoomFactor**

Methods

ClearPolar Method

The **ClearPolar** method clears all polar plot graphical data from the target object.

Syntax:

object.**ClearPolar**

The **ClearPolar** method has the following named arguments:

Part	Description
object	A SSIMTarget object.

Remarks:

The ClearPolar method requires the PolarPlot method to be True. Please note that changing the TargetStyle or ZoomFactor will invoke the ClearPolar method.

Code Example:

```
‘Clear all polar plot lines
SSIMTarget1.ClearPolar
```

Also See: **TargetStyle, ZoomFactor**

ConvertReading Method

The **ConvertReading** method is used to convert any value into a formatted reading, based on the current display mode.

Syntax:

object.**ConvertReading (ByVal *InputReading* As Single,**
([*DoNotUseMinRange* As Boolean]) **As String**

The **ConvertReading** function has these named arguments:

Part	Description
object	A SSIMPower object.
InputReading	Single value representing the number you wish to format.
DoNotUseMinRange	Optional boolean used to determine if the return is referenced to the detectors minimum range.

Remarks:

The **ConvertReading** function will return a formatted power reading as it would be displayed in the **SSIMPower** object. This function does not support the offset property value and will format raw power only. Offset must be added as a separate command. This function can be used to display readings outside of the **SSIMPower** object. The optional flag **DoNotUseMinRange**, is a boolean flag that is false by default and is designed to correctly format a reading resolution corresponding to the detector minimum range. For example, an input value of 1.0nw from the SmartSensor Interface Module would be returned as 0.00mW if the **DoNotUseMinRange** flag is set to false and 1.00nW if set to true. Please note that readings returned with this function that are below the detector minimum range and have the **DoNotUseMinRange** flag set to true are not considered accurate based on the detector specifications.

USE CAUTION: The default HeadMinRange value is set to 0. This will cause the object to display readings as low as 1nw for all detector heads. Coherent does not guarantee the accuracy of any readings below the detector's minimum range unless the HeadMinRange property is set to the correct minimum range.

Code Example:

```
‘Convert reading into display format
Private Sub SSIMUSB1_StreamData(ModuleNumber As String,
Power As Single, XPosition As Single, YPosition As Single)
    Text1.Text = ConvertReading (Power, True)
End Sub
```

Also See: **HeadMinRange, Detector Head Information**

SendCommand Method

The **SendCommand** method is used to send a command to a specific SmartSensor Interface Module.

Syntax:

object.**SendCommand ByVal** *ModuleNumber* **As String, ByVal** *Command* **As String**

The **SendCommand** method has these named arguments:

Part	Description
object	A SSIMUSB object.
ModuleNumber	String value representing the object module number assigned to each SmartSensor Interface Module. The ModuleNumber is utilized only when the TrackBySerialNumber Property is set to False.
Command	String value representing command sent to the SmartSensor Interface Module.

Remarks:

The **SendCommand** method is used to send a command to a specific SmartSensor Interface Module. The **ModuleNumber** String will be assigned by the object when a SmartSensor interface Module is first detected. For more information on**ModuleNumber**see **ModuleAdded**. The **Command** string must be sent without a <cr> character. For more information on commands see SmartSensor interface Module **CommandSet** .

Code Example:

```
‘Send request to change wavelength to 514 nm to module 2 ‘
SSIMUSB1.SendCommand  2, “wl 5.14e-6”
```

SetNew_dB_Reference Method

The **SetNew_dB_Reference** method is used to set a specified reference point for dB measurements.

Syntax:

object.**SetNew_dB_Reference**(**ByVal** *New_Reference* **As Single**)

The **SetNew_dB_Reference** function has these named arguments:

Part	Description
object	A SSIMPower object.
New_Reference	Single value for the dB reference point.

Remarks:

The dB reference value will only be used if the **SSIMPower** object **DisplayMode** property is set to dB. If no value is detected, the **DisplayMode** property will use the next power value as a dB reference.

Code Example:

```
‘Set new Reference to 5.12mw
SSIMPower1.SetNew_dB_Reference = 0.00512
```

Also See: **DisplayMode**

UpdatePosition Method

The UpdatePostion Method will update the centering dot location or draws new polar plot line on the SSIMTarget object.

Syntax:

*object.***UpdatePosition**(*X_Coordinate* **As Single**, *Y_Coordinate* **As Single**)

The **UpdatePosition** method has these parts:

Part	Description
Object	A SSIMPower object.
X_Coordinate	Distance from center in mm along the X axis.
Y_Coordinate	Distance from center in mm along the Y axis.

Remarks:

If **PolarPlot** is set to True then the SSIMTarget object will retain the last position entry to correctly draw the polar plot line. This action will continue until the **ClearPolar** method is called or the **TargetStyle** is changed.

Code Example:

```
Private Sub SSIMUSB1_StreamData(ModuleNumber As String, Power As Single, XPosition As Single, YPosition As Single)

        'Show new position on target.
        SSIMTarget1.UpdatePosition XPosition, Yposition

        'Show new power reading.
        SSIMPower1.DisplayNewReading Power

End Sub
```

Also See: **PolarPlot, ClearPolar, TargetStyle**

Properties

Aperture_mm Property

The Aperture_mm Property returns or sets the detector aperture size.

Syntax:

*object.***Aperture_mm** [= *number*]

The **Aperture_mm** property has the following parts:

Part	Description
Object	A SSIMPower object.
Aperture_mm	The Aperture_mm property corresponds to an integer value to specify the detector aperture size in mm.

Remarks:

The detector's aperture size is required by the SSIMTarget object to properly display alignment accuracy. The default setting is 19mm.

For a list of default aperture sizes, please refer to **Detector Head Information**

Code Example:

```
‘Set aperture size to 11 mm
SSIMtarget1.ApertureSize = 11
```

Also See: **Detector Head Information**

BorderStyle Property

The BorderStyle property returns or sets a value corresponding to the border style of the SSIMPower object.

Syntax:

*object.***BorderStyle** [= *number*]

The **BorderStyle** property has these parts:

Part	Description
Object	A SSIMPower object.
Number	The BorderStyle Property corresponds to a integer value to specify the type of border, described in Settings below.
Settings	0 None 1 (Default) Fixed Single.

Remarks:

By default, the Border Style is set to 1 (Fixed).

Code Example:

```
'Set Border style to none
SSIMPower.BorderStyle= 0
```

DisplayMode Property

The DisplayMode Property returns or sets a value that determines the format of the displayed reading in a control.

Syntax:

*object.***DisplayMode** [= *number*]

The **DisplayMode** property has these parts:

Part	Description
Object	A SSIMPower object.
Number	An integer that specifies display unit of measure.

Settings		
	0	(Default) Power Kw, w, mw, uw, nw.
	1	dBm.
	2	dB.

Remarks:

A DisplayMode setting of dB will cause the **SSIMPower** object to use the next power input value as defined in **DisplayNewReading** or **ConvertReading** as the dB reference value. The result will be a displayed in dB utilizing the first value as a reference. The SetNew_dB_Ref will may be used to set a new reference point.

Code Example:

```
‘Set mode to dBm
SSIMPower.DisplayMode = 1
```

DotColor Property

The DotColor Property will set or return the OLE color used to display the alignment dot or polar plot line.

Syntax:

*object.***DotColor**[= ByVal ***New_DotColor*** As OLE_COLOR]

The **DotColor** property has these parts:

Part	Description
Object	A SSIMTarget object.
New_DotColor	OLE color value used to display the alignment dot or polar plot line.

Remarks:

The default OLE color is set to black.

Code Example:

```
‘Set polar plot line to vbRed
SSIMTarget.DotColor = vbRed
```

Also See: **Target Color**

HeadMinRange Property

The HeadMinRange Property will set or return the minimum range value used to properly display readings in the object.

Syntax:

*object.***HeadMinRange** [= ByVal ***New_HeadMinRange*** As Single]

The **HeadMinRange** property has these parts:

Part	Description
Object	A SSIMPower object.
New_HeadMinRange	Singe value representing detector head's minimum range.

Remarks:

Each detector head has an associated minimum range. This range is used to determine maximum resolution of output readings. This property is designed to keep the object from displaying power readings below levels were the detector is determined to be accurate. Please note that a detector minimum range is not the same as minimum detectable power. The minimum detectable power can be calculated by dividing the detector's minimum range by 10. For a list of detector minimum range values please see ***Detector Head Information.***

CAUTION: The default HeadMinRange value is set to 0. This will cause the object to display readings as low as 1 nW for all detector heads. Coherent does not guarantee the accuracy of any readings below the detectro's minimum range unless the HeadMinRange property is set to the detector's correct minimum range. The property will only utilize values listed in the Detector Head Information, other values will display no power value.

Code Example:

```
'Display new reading
SSIMPower1.DisplayNewReading(12.35)
```

Also See: **Detector Head Information**

Offset Property

The Offset Property will set or return the current offset value used to display readings in **ConvertReading** and **DisplayNewReading.**

Syntax:

*object.***Offset**[= ByVal ***New_Offset*** As Single]

The **Offset**property has these parts:

Part	Description
Object	A SSIMPower object.
New_Offset	Return value representing new offset in watts.

Remarks:

By default the offset property is set to 0.

Code Example:

```
‘Set offset
SSIMPower1.Offset = 1.25
```

PolarPlot Property

The PolarPlot Property will set or return the polar plot status.

Syntax:

*object.***PolarPlot** = [ByVal ***New_PolarPlot*** As Boolean]

The **New_PolarPlot** property has these parts:

Part	Description
Object	A SSIMTarget object.
New_PolarPlot	Boolean value of polar plot status as described in **Settings**
Settings	False (Default) target will not plot in Polar form. Target will display the alignment dot. True. Target will display alignment in polar plot form.

Remarks:

The default setting is false

Code Example:

```
'Set target to polar plot
SSIMTarget.PolarPlot = True
```

RelativeX Property

The RelativeX Property sets or returns the X position offset in mm.

Syntax:

*object.***RelativeX** [= ByVal ***New_RelativeX*** As Single]

The **RelativeX** property has these parts:

Part	Description
Object	A SSIMTarget object.
RelativeX	A single value representing the offset value of the X position as displayed on the target object in mm.

Remarks:

The default **RelativeX** value is 0

Code Example:

```
‘Set relative X position a 2.25 mm
SSIMtarget1.RelativeX = 2.25
```

Also See: **RelativeY**

RelativeY Property

The RelativeY Property sets or returns the Y position offset in mm.

Syntax:

*object.***RelativeY** [= ByVal ***New_RelativeY*** As Single]

The **RelativeY** property has these parts:

Part	Description
Object	A SSIMTarget object.
RelativeY	A single value representing the offset value of the Y position as displayed on the target object in mm.

Remarks:

The default **RelativeY** value is 0

Code Example:

```
‘Set relative Y position a 2.25 mm
SSIMtarget1.RelativeY = 2.25
```

Also See: **RelativeX**

SSIM_1_SerialPort Property

The **SSIM_1_SerialPort** Property Sets and returns the communications port number for one of two serial SmartSensor Interface Modules.

Syntax:

object **.SSIM_1_SerialPort**[= *number*]

The **SSIM_1_SerialPort** property has these named arguments:

Part	Description
Object	A SSIMUSB object.
Number	Integer value representing the serial communications port number.

Remarks:

The value must be set to an integer between 0 and 9. A value of 0 indicates SSIMUSB object will not attempt to connect to the SmartSensor Interface Module.

Code Example:

```
‘SSIM 1 will connect to serial port 1
SSIMUSB1.SSIM_1_SerialPort = 1
```

Also See: **SSIM 2 Serial Port**

SSIM_2_SerialPort Property

The **SSIM_2_SerialPort** Property Sets and returns the communications port number for one of two serial SmartSensor Interface Modules.

Syntax:

object **.SSIM_2_SerialPort**[= *number*]

The **SSIM_2_SerialPort** property has these named arguments:

Part	Description
Object	A SSIMUSB object.
Number	Integer value representing the serial communications port number.

Remarks:

The value must be set to an integer between 0 and 9. A value of 0 indicates SSIMUSB object will not attempt to connect to the SmartSensor Interface Module.

Code Example:

```
'SSIM 2 will connect to serial port 2
SSIMUSB1.SSIM_2_SerialPort =2
```

Also See: **SSIM_1_SerialPort**

SSIMConnection Property

The **SSIMConnection** Property Sets and returns the connection method to the SmartSensor Interface Modules.

Syntax:

object.**SSIMConnection**[= *number*]

The **SSIMConnection** property has these named arguments:

Part	Description
Object	A SSIMUSB object.
ModuleNumber	String value representing the object module number assigned to each SmartSensor Interface Module. The ModuleNumber is utilized only when the TrackBySerialNumber Property is set to False.
Settings	0 Connect using USB. (Default)
	1 Connect using Serial port.

Remarks:

The SSIMUSB object supports Serial communication or USB connection but not simultaneously. The connection property can be changed at run-time, however communications will be lost with currently connected SmartSensor Interface modules.

Code Example:

```
'SSIM 1 will connect to serial port 1
SSIMUSB1.SSIMConnection = 1
```

Also See: **SSIM_1_SerialPort, SSIM_2_SerialPort**

TargetColor Property

The TargetColor Property will set or return the OLE color used to display the target graphic.

Syntax:

*object.***TargetColor**[= ByVal ***New_TargetColor*** As OLE_COLOR]

The **TargetColor** property has these parts:

Part	Description
Object	A SSIMTarget object.
New_TargetColor	OLE color value used to display the target graphic.

Remarks:

The default OLE color is set to black.

Code Example:

```
‘Set the target color to vbBlue
SSIMTarget.TargetColor = vbBlue
```

Also See: DotColor

TargetStyle Property

The TargetStyle Property will set or return the graphic style of the displayed target.

Syntax:

*object.***TargetStyle**[= *number*]

The **TargetStyle** property has these parts:

Part	Description
Object	A SSIMTarget object.
Number	The integer value represents the target dsiplay style as described in Settings**.**

Settings

The **TargetStyle** property settings are:

Constant	Setting	Description
None	0	Target will display the outer aperture circle.
Standard	1	Target will display the outer aperture circle, out of alignment circle and quadrent lines
Alignment	2	Target will display the outer aperture circle and quadrent lines
Polar	3	Target will display the outer aperture circle polar plot lines

Remarks:

Changing the target style at run time will clear any displayed polar plot information.

Code Example:

```
‘Set the target style to polar
SSIMTarget.TargetStyle = Polar
```

TrackBySerialNumber Property

The **TrackBySerialNumber** Property Sets and returns the method of referencing a SmartSensor Interface Module.

Syntax:

object .**TrackBySerialNumber** [= *boolean*]

The **TrackBySerialNumber** property has these named arguments:

Part	Description
Object	A SSIMPower object.
Boolean	boolean value representing the method of tracking the SmartSensor Interface Module.
Settings	False. (Default) All references to the SmartSensor Interface Module use a module number. True. All references to the SmartSensor Interface Module use the module serial number.

Remarks:

If the **TrackBySerialNumber** property is set to *true* then the **ModuleAdded** event will return the SmartSensor Interface Module serial number rather than a unique module number. All module specific events and properties will require the use of this serial number.

Code Example:

```
‘SSIM 1 will track by serial number
SSIMUSB1.TrackBySerialNumber = True
```

Also See: **ModuleAdded**

ZoomFactor Property

The ZoomFactor Property will set or return the multiple value used to display position information.

Syntax:

*object.***ZoomFactor**[= *number*]

The **ZoomFactor** property has these parts:

Part	Description
Object	A SSIMTarget object.
Number	Integer value representing the sensitivity multiple.

Settings

The **ZoomFactor** property settings are:

Zoom Level	Setting	Description
x1	1	Target sensitivity will be set at 1:1.
x2	2	Target sensitivity will be set at 1:2.
x4	4	Target sensitivity will be set at 1:4.
x8	8	Target sensitivity will be set at 1:8.
x16	16	Target sensitivity will be set at 1:16.
x32	32	Target sensitivity will be set at 1:32.

Remarks:

If **PolarPlot** is set to True then the **SSIMTarget** object will clear any displayed polar plot information if **ZoomFactor** is changed.

Code Example:

```
‘Set Zoom Factor to 16 X
SSIMTarget.ZoomFactor = X16
```

Also See: **PolarPlot**

Events

DetectorConnect Event

The **DetectorConnect** event is generated when a SmartSensor Interface Module detects detector head connection.

Syntax:

Sub *object*_**DetectorConnect** *(ModuleNumber* **as String)**

The **DetectorConnect** event has these named arguments:

Part	Description
Object	A SSIMUSB object.
ModuleNumber	String value representing the object module number assigned to each SmartSensor Interface Module. The ModuleNumber is utilized only when the TrackBySerialNumber Property is set to False.

Remarks:

The **DetectorConnect** event will only be raised upon the initial detector head detection. The **ModuleNumber** String will be assigned by the object when a SmartSensor interface Module is first detected. For more information on **ModuleNumber** see **ModuleAdded** and **TrackBySerialNumber**

Code Example:

```
Private Sub SSIMUSB1_DetectorConnect(ModuleNumber As String)
MsgBox "New Detector Found by module " & ModuleNumber,
    vbOKOnly, "Detector"
End Sub
```

Also See: **Detector Disconnect**

DetectorDisconnect Event

The **DetectorDisconnect** event is generated when a SmartSensor Interface Module detects the removal of a detector head.

Syntax:

Sub *object* **_DetectorConnect (***ModuleNumber* **as String)**

The **DetectorDisconnect** event has these named arguments:

Part	Description
Object	A SSIMUSB object.
ModuleNumber	String value representing the object module number assigned to each SmartSensor Interface Module. The ModuleNumber is utilized only when the TrackBySerialNumber Property is set to False.

Remarks:

The **DetectorDisonnect** event will be raised once upon the initial detection of a detector head. The ModuleNumber String will be assigned by the object when a SmartSensor interface Module is first detected by the object. For more information on **ModuleNumber** see **ModuleAdded** and **TrackBySerialNumber.**

Code Example:

```
Private Sub SSIMUSB1_DetectorDisconnect(ModuleNumber
    As String)
    MsgBox “Detector removal detected by module “ &
    ModuleNumber, vbOKOnly, “Detector”
End Sub
```

Also See: **DetectorConnect**

DetectorOverRange Event

The **DetectorOverRange** event is generated when a SmartSensor Interface Module detects when the laser input level exceeds the maximum rated power of the detector.

Syntax:

Sub *object* **_DetectorOverRange (***ModuleNumber* **as String)**

The **DetectorOverRange** event has these named arguments:

Part	Description
Object	A SSIMUSB object.
ModuleNumber	String value representing the object module number assigned to each SmartSensor Interface Module. The ModuleNumber is utilized only when the TrackBySerialNumber Property is set to False.

Remarks:

The **DetectorOverRange** event will continue to be raised as long as the attached detector is over the maximum rated power level. The **ModuleNumber** string will be assigned by the object when a SmartSensor interface Module is first detected. For more information on **ModuleNumber** see **ModuleAdded** and **TrackBySerialNumber.**

Code Example:

```
Private Sub SSIMUSB1_Detector OverRange(ModuleNumber As
    String)
    MsgBox “Warning. Detector attached to “ & ModuleNumber &
    “ is over the maximum rated power level”, vbOKOnly,
    “Detector”
End Sub
```

Also See: **DetectorOverTemp**

DetectorOverTemp Event

The **DetectorOverTemp** event is generated when a SmartSensor Interface Module detects a detector temperature exceeding the maximum rated temperature (100 deg C)

Syntax:

Sub *object* **_DetectorOverTemp (***ModuleNumber* **as String)**

The **DetectorOverRange** event has these named arguments:

Part	Description
Object	A SSIMUSB object.
ModuleNumber	String value representing the object module number assigned to each SmartSensor Interface Module. The ModuleNumber is utilized only when the TrackBySerialNumber Property is set to False.

Remarks:

The **DetectorOverTemp** event will continue to be raised for the duration of the OverTemp condition. Normal operation will resume when the detector temperature is within range. The **ModuleNumber** String will be assigned by the object when a SmartSensor interface Module is first detected. For more information on **ModuleNumber** see **ModuleAdded** and **TrackBySerialNumber.**

Code Example:

```
Private Sub SSIMUSB1_Detector OverTemp(ModuleNumber As
    String)
    MsgBox “Warning. Detector attached to “ & ModuleNumber &
    “ is over temperature”, vbOKOnly, “Detector”
End Sub
```

Also See: **DetectorOverRange**

ModuleAdded Event

The **ModuleAdded** event is generated when the host system detects a new SmartSensor Interface Module.

Syntax:

Sub *object* **_ModuleAdded (*ModuleNumber* as String)**

The **ModuleAdded** event has these named arguments:

Part	**Description**
Object	A SSIMUSB object.
ModuleNumber	String value representing the object module number assigned to each SmartSensor Interface Module. The ModuleNumber is utilized only when the TrackBySerialNumber Property is set to False.

Remarks:

The **ModuleAdded** event will be raised only upon the initial detection of a SmartSensor Interface Module. When the host system detects a new connection a unique identifier number is generated (***ModuleNumber)***. That number is retained by the SSIMUSB object for future reference to the SmartSensor Interface Module. All *module specific* commands and methods will require this number to identify the target device. All *module specific* events and returns will also include this unique number to identify the specific device that returned the response or raised the event. This is the only event that will identify each module and return the unique identifier number.

CAUTION: *It is not recommended to hard code module numbers. The module numbers are assigned in the order that they are detected. The TrackBySerialNumber Property can be utilized to identify each SSIM by serial number instead of the assigned Module Number.*

Code Example:

```
Private Sub SSIMUSB1_ModuleAdded(ModuleNumber As String)
    ‘Load module into master array
    abolMasterArray(ModuleNumber) = True
End Sub
```

Also See: **ModuleRemoved**

ModuleRemoved Event

The **ModuleRemoved** event is generated when the host system detects the removal of a SmartSensor Interface Module.

Syntax:

Sub *object* **_ModuleRemoved** (*ModuleNumber* **as String**)

The **ModuleRemoved** event has these named arguments:

Part	**Description**
Object	A SSIMUSB object.
ModuleNumber	String value representing the object module number assigned to each SmartSensor Interface Module. The ModuleNumber is utilized only when the TrackBySerialNumber Property is set to False.

Remarks:

The **ModuleRemoved** event will be raised upon the initial detection of a SmartSensor Interface Module removal from the host system. The **ModuleNumber** String will be assigned by the object when a SmartSensor interface Module is first detected by the object. For more information on **ModuleNumber** see **ModuleAdded** and **TrackBySerialNumber.**

Code Example:

```
Private Sub SSIMUSB1_ModuleRemoved(ModuleNumber As String)
    'Remove module from master array
    abolMasterArray(ModuleNumber) = False
End Sub
```

Also See: **ModuleAdded**

ReturnData Event

The **ReturnData** event is generated whenever SSIMUSB object returns data that is not streaming data.

Syntax:

Sub *object* _ReturnData(*ModuleNumber* as String, *Data* as String)

The **ReturnData** event has these named arguments:

Part	Description
Object	A SSIMUSB object.
ModuleNumber	String value representing the object module number assigned to each SmartSensor Interface Module. The ModuleNumber is utilized only when the TrackBySerialNumber Property is set to False.
Data	String value representing the data returned from the SmartSensor Interface Module.

Remarks:

The **ReturnData** event will be raised when data is returned from the SmartSensor Interface Module. The **Data** string return will represent a single response from the SmartSensor Interface Module, if multiple commands are sent each response will return as a separate event. The **ModuleNumber** String will be assigned by the object when a SmartSensor interface Module is first detected. For more information on **ModuleNumber** see **ModuleAdded** and **TrackBySerialNumber.**

Code Example:

```
Dim lngWavelength as long
‘Send request to change wavelength to 514 nm to module 2
SSIMUSB1.SendCommand  2, “wl 5.14e-6”

Private Sub SSIMUSB1_ReturnData(ModuleNumber As String, Data
    as String)
    ‘Display new wavelength
    lblWavelength(ModuleNumber).Caption = Data
    lngWavelength = Data
End Sub
```

Also See: **StreamData**

StreamData Event

The **StreamData** event is generated when the SSIMUSB object returns streaming data.

Syntax:

Sub *object* **_StreamData(*ModuleNumber* As String, *Power* As Single, *XPosition* As Single, *YPosition* As Single)**

The **StreamData** event has these named arguments:

Part	Description
Object	A SSIMUSB object.
ModuleNumber	String value representing the object module number assigned to each SmartSensor Interface Module. The ModuleNumber is utilized only when the TrackBySerialNumber Property is set to False.
Power	Single value representing the current detector power input.
XPosition	Single value representing the current X position of input beam (mm from center of detector).
YPosition	Single value representing the current Y position of input beam (mm from center of detector).

Remarks:

SmartSensor Interface Modules with data streaming turned on will return data at a 10Hz rate. This event will continue to be raised when data streaming is turned on. ***Xposition*** and ***YPosition***will be set to 0 for Semiconductor detectors.Semiconductordetectors are not ableto provide centering information.The **ModuleNumber** String will be assigned by the object when a SmartSensor interface Module is first detected. For more information on **ModuleNumber**see **ModuleAdded** and **TrackBySerialNumber.**

Code Example:

```
Dim intCounter as Integer
Dim asngPowerLog(99) as single
Dim asngPosition(99, 1)
intCounter = 0
Private Sub SSIMUSB1_StreamData(ModuleNumber As String, Power
    As Single, XPosition As Single, YPosition As Single)
    ‘Save 100 power readings
    asngPowerLog(intCounter) = Power
    ‘Save 100 position measurements
    asngPosition(intCounter, 0) = XPosition    ‘Save X
    asngPosition(intCounter, 1) = YPosition    ‘Save Y
    ‘Incrament counter
    intCounter = intCounter + 1
End Sub
```

Also See: **ReturnData**

Specifications

Ranges: Microprocessor controlled. Ranges determined by EEPROM in SmartSensor heads.

Communication: USB and RS-232 Communication between Host Computer and SmartSensor Interface Module.

NOTE: The USB Standard supports up to 127 USB devices. Coherent has designed the OEM Tools to support many modules simultaneously and testing has been conducted with 14. Communication with greater than 14 modules can be verified by Coherent for specific customer requirements.

Data Stream: The SSIM will stream data at a rate of 10Hz.

CW Thermal Detector: Provides CW Power and Beam Position Data. See Appendix A for accuracy specifications and input beam limitations.

Semiconductor Sensor: Provides CW Power Data. See Appendix A for accuracy specifications and input beam limitations.

Memory: The SSIM has the capability to buffer multiple commands to process in a serial order.

NOTE: Data streaming occurs at a rate of 10 Hz, commands may be processed at a slower rate, based on the host system and the protocol required to process the commands.

Operating Temperature: 5° C to 40° C

Safety Features: Sensor over-temperature, alignment and over-range events.

Size: 5.25" long x 5" wide x 1.26" thick (13.3 cm x 12.7 cm x 3.2 cm).

Weight: 0.88 lbs. (0.400 Kg), with batteries.

DC Input Requirements: 5V Center Positive Jack Input. Wall Plug and Rechargeable Battery included.

> **AC wall plug input requirements:** 100-240V,
> ~0.2 A, 50-60 Hz
>
> **6V Internal rechargeable battery pack:** Recharged automatically during AC operation. Battery life: 12 hours typical. Complete recharge cycle is 14 hours.

Appendix A: Supported Coherent SmartSensors™

Thermal SmartSensors

Catalog Number	Name	Power			Aperture (mm)	Notes	Spectral Range (µm)	Accuracy (± %)	Sensor Coating (note m)
		Max (W)	Min (W) (note f)	Resolution (W)					
Standard Thermal Sensors: 6000 W/cm^2 maximum CW power density									
33-7832	LM-200XL HTD	200	1.0	0.1	55	c	0.25-10.6	4	HTD
33-7857	LM-200 HTD	200	0.2, 0.1	0.01	19	k	0.25-10.6	5	HTD
33-7840	LM-200 HTD	200	0.2, 0.1	0.01	19	t	0.25-10.6	5	HTD
33-7865	LM-200B HTD	200	0.2, 0.1	0.01	19	c, w	0.25-10.6	5	HTD
33-7873	LM-150 FS HTD	150	0.1	0.01	19	r	0.25-10.6	5	HTD
33-7881	LM-100XL HTD	100	1.0	0.1	40		0.25-10.6	4	HTD
33-7899	LM-100 HTD	100	0.1	0.01	19		0.25-10.6	2	HTD
33-0472	LM-100B HTD	100	0.1	0.01	38	c, w	0.25-10.6	4	HTD
33-7915	LM-45 HTD	45	0.1	0.01	19		0.25-10.6	2	HTD
33-7923	LM-10 HTD	10	0.01	0.001	16		0.25-10.6	2	HTD
33-7931	LM-3 HTD	3	0.01	0.001	19		0.25-10.6	2	HTD
High Power Thermal Sensors: 2500 W/cm^2 maximum CW power density									
33-1132	LM-5000	5000	300	1.0	55	a,e	0.25-10.6	5	H
33-1124	LM-2500	2500	150	1.0	55	a,e	0.25-10.6	5	H
33-1116	LM-1000	1000	2.0,10	0.1, 1.0	38	b	0.25-10.6	5	H
Volume Absorbers for High Peak Power: 1 GW/cm^2 at 10 ns maximum peak power density									
33-1017	LM-80V	80	0.5, 0.1	0.01	38	h	0.19-6.0	4	V
33-0993	LM-30V	30	0.1	0.01	19	h	0.19-6.0	3.5	V
Excimer Special Coated Sensors: 2000 W/cm^2 maximum power density									
33-1108	LM-200XLE	200	1.0	0.1	55	c	0.19-0.35	4	UV
33-1033	LM-100E	100	0.5, 0.1	0.01	40		0.19-0.35	5	UV
33-4508	LM-100EB	100	0.5, 0.1	0.01	38	c, w	0.19-0.35	5	UV
High Sensitivity Thermal Sensors: 400 W/cm^2 maximum power density									
33-0928	LM-1	1.0	0.001	0.0001	9		0.25-2.5	5	P
33-7956	LM-1 IR	1.0	0.001	0.0001	9		1.0-10.6	5	P
Special Purpose Sensors: 2500 W/cm^2 maximum CW power density									
33-1215	Beam Finder	1000	2.0	0.1	35	b,d,e	0.3-10.6	5	H
33-7907	LM-20 HTD	20	0.1	0.01	19	p	0.25-10.6	2	HTD

All SmartSensors are supplied with a 1.8 m (6 ft) cable (unless otherwise noted), stand and NIST traceable Calibration Certificate.

(a) 7.5 liter/min cooling water required. Maximum temperature deviation 3%/min. Maximum flow deviation 2%/min.

(b) 4 liter/min cooling water required. Maximum temperature deviation 3%/min. Maximum flow deviation 2%/min.

(c) 1 liter/min cooling water required. Maximum temperature deviation 3%/min. Maximum flow deviation 2%/min.

(d) Designed for system integration for measurement of power and position, for use on the Ultima only (supplied with 6 m (20 ft) cable and cooling hoses).

(e) Supplied with 6 m (20 ft) cable.

(h) Damage limit: Pulse 1 GW/cm^2, measured at 1.06 µm, (10 ns pulse). Varies with wavelength, derate by 50% 0.26-0.35 µm, by 90% below 0.26 µm, Maximum Average Power Density 30 W/cm^2.

(k) Must be used with 110 VAC power for internal fan cooling for powers above 50 Watts.

(m) See coating specifications and maximum power density limits on page 22 of the Power & Energy Meters Catalog.

(p) Designed for system integration, must be mounted to a heat sink.

(r) Intermittent duty (2 min @ 150 W; 5 min @ 100 W; 20 min @ 40 W). See page 24 of the Power & Energy Meters Catalog for further details.

(t) Must be used with 220 VAC power for internal fan cooling for powers above 50 Watts.

(w) These sensors require water cooling to operate at the maximum specified power ratings. They can also be operated using convection and conduction cooling methods. See details on cooling methods and associated power ratings in the OEM sensor section, pages 28-31 of the Power & Energy Meters Catalog.

Semiconductor SmartSensors

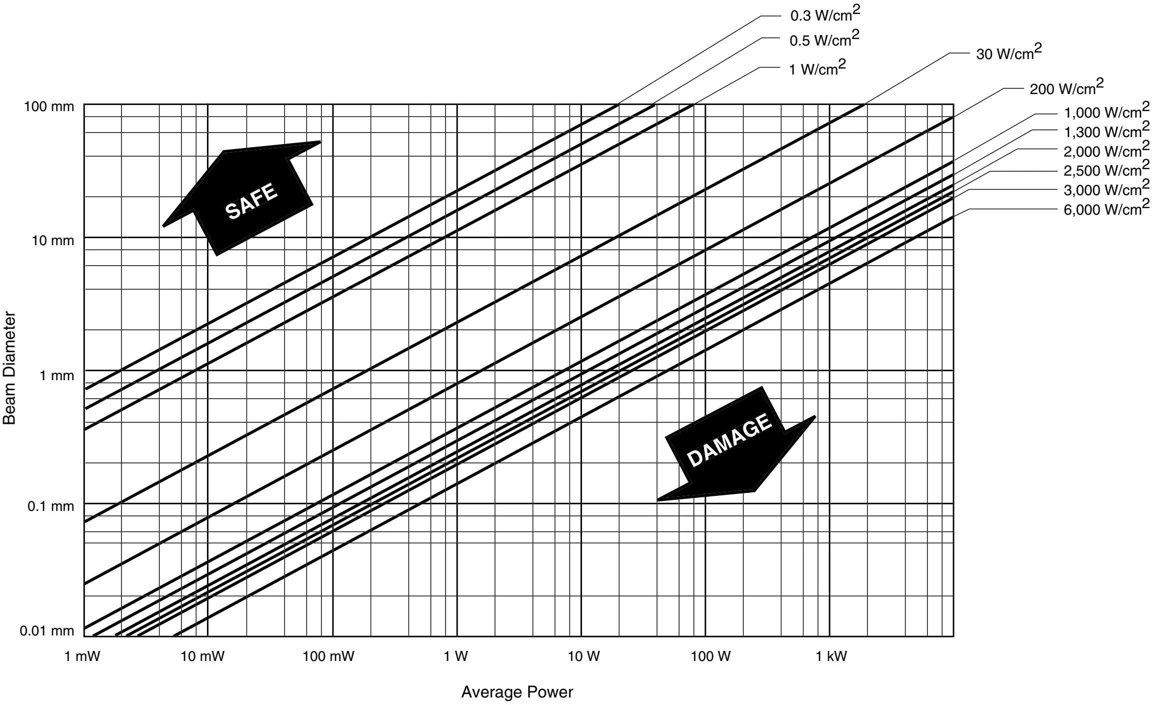
Catalog Number	Name	Power			Aperture (mm)	Spectral Range (µm)	Accuracy (±%) (note f)	Maximum Power Density (W/cm^2)	Max Powe with Attenuator (W)
		Max (mW)	Min (µW)	Resolution (nW)					
33-0951	LM-2 UV	30	0.01	1.00	6.0	0.25-0.4	8, 5	0.3	NA
33-0936	LM-2 VIS	50	0.01	1.00	7.9	0.4-1.064	5, 3	1.0	5.0
33-0944	LM-2 IR	10	0.5 z	1.00	5.0	0.8-1.55	4.5, 2	0.5	5.0

All SmartSensors are supplied with a 1.8 m (6 ft) cable, stand and NIST traceable Calibration Certificate.

(f) Value to the left is for Ultima LabMaster and FieldMaster GS, value to the right is for FieldMaster.

(z) 1 µW for FieldMaster.

Appendix B: Safe Detector Operating Region



Variation of beam diameter and power for average power densities of thermal and semiconductor sensors. Safe operating limit above damage threshold lines.

Appendix C

SmartSensor™ Interface Module Circuit and Operation Description

Each power (detector) head contains a temperature sensor, electrically erasable programmable read only memory (EEPROM) and either a photodiode or thermopile detector. At power On, the firmware interrogates the EEPROM to determine which type of detector is present.

Amplifier gains and analog switches (mux) are adjusted to match the information received from the EEPROM. The 16-bit analog-to-digital converter (A/D) switches between channels 1 and 2 to measure either sensor output or temperature. This data is then sent through the Interface to the Microprocessor.

The program in the Microprocessor controls the mux, amplifier gains, and A/D converter channels. The memory and serial communication are selectively reached by the Microprocessor through the Decoder. The Decoder interprets the Microprocessor address lines to determine which device the Microprocessor wants to talk to.

The thermoelectric detector heads have four outputs that determine beam position and laser power. The Microprocessor scans the mux through the four outputs and the temperature signal to arrive at a power reading (with the Gain Amplifier set to provide maximum signal strength). The A/D then converts the outputs for the Microprocessor.

Semiconductor detector heads have one output. They use a current-to-voltage converter to precondition the output before it reaches the Gain Amplifier. The A/D then converts the output for the Microprocessor.

The Microprocessor reads the A/D output and computes the power based on the data loaded from the SmartSensor EEPROM. The quadrant output from a Thermal SmartSensor provides position information to the Microprocessor. The Microprocessor organizes the power and position information for RS-232 serial transmission. The SSIM OEM Tools receive the power and position data stream at a rate of 10 Hz.

Appendix D: Troubleshooting Tips

PROBLEM	PROBABLE CAUSE	SOLUTION
Software does not detect Smart Sensor Interface Module or Module Removed Event occurs.	Dead Battery in Smart Sensor Interface Module	Connect 5V wall plug to the Smart Sensor Interface Module for a 14 hour complete charging cycle.
	Serial Cable or USB is not properly connected to Smart Sensor Interface	Verify proper cable connection between the host computer and the Smart Sensor Interface Module.
Software does not detect the Smart Sensor or Detector Disconnect Event occurs.	Smart Detector is not connected properly to the Smart Sensor Interface Module	Secure the DB 25 head connector at the Smart Sensor Interface Module.
	Smart Detector EEPROM is faulty	Contact Coherent for Service.
	Smart Sensor Interface Module is faulty	Contact Coherent for Service.
Over Temp Event Occurs	The maximum operating temperature for the Thermal Smart Detector has been exceeded.	Reduce the Laser input power or attenuate the laser output beam. When the Smart Detector is within the normal temperature range, the Event will automatically clear.
Over Range Event Occurs	The maximum input power to the Smart Detector has been exceeded.	Reduce the Laser input power or attenuate the laser output beam. When the Smart Detector is within the specified input power range, the Event will automatically clear. See Appendix A for the specified input power range for Coherent Smart Detectors.
Batteries do not hold a charge	Rechargeable battery pack is faulty	Contact Coherent for Replacement Battery Pack.
Semiconductor Sensor Power Readings are not correct.	The current wavelength settings do not match the laser source.	Verify the Wavelength value utilized in the software matches the Laser Source.
Semiconductor Sensor Power Readings do not increase as the laser power increases.	The Semiconductor Sensor is saturated.	Verify that the laser input is within the proper range specified in Appendix B of this manual.
Unexpected Reading or Repeated Readings	Readings are requested before processing is complete.	Add delays before reading requests to allow for proper setup and processing. This delay is dependent on the host system and the number of commands currently buffered by the SSIM.

Warranty

The seller warrants to the original Buyer each item manufactured by it to be free from defects in material and workmanship for a period of time and under such conditions as specified in the Seller's warranty for the individual product, or for twelve (12) months from delivery if a warranty for the individual product is not specified. Major sub-systems manufactured by other firms but integrated into the Seller's systems are covered by the original Manufacturer's warranty. The Seller's liability under valid warranty claims is limited to repair or replacement at the Seller's plant or the Buyer's location, all at the option of the Seller.

THE FOREGOING WARRANTY IS EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, WHETHER WRITTEN, ORAL OR IMPLIED AND SHALL BE THE BUYER'S SOLE REMEDY AND THE SELLER'S SOLE LIABILITY ON CONTRACT OR WARRANTY OR OTHERWISE FOR THE PRODUCT. THE SELLER DISCLAIMS ANY IMPLIED WARRANTY OR MERCHANTABILITY OR FITNESS FOR PURPOSE.

All claims under warranty must be made promptly after occurrence of circumstances giving rise thereto, must be received within the applicable warranty period by the Seller, and shall be subject to the terms and conditions stated herein. Such claims should include the product serial number, the date of shipment, and a full description of the circumstances giving rise to the claim. Before any products are returned for repair and/or adjustment, authorization for the return and instructions as to how and where these Products should be shipped must be obtained from the Seller. Any product returned to the Seller for examination and/or warranty repair shall be sent prepaid via the means of transportation indicated as acceptable by the Seller. The Seller reserves the right to reject any warranty claim on any item that has been shipped by non-acceptable means of transportation. When any product is returned for examination and inspection, or for any other reason, the Buyer and its shipping agency shall be responsible for all damage resulting from improper packing or handling, and for loss in transit, notwithstanding any defect of non-conformity in the Product. In all cases, the Seller has sole responsibility for determining the cause and nature of failure, and the Seller's determination with regard thereto shall be final.

If it is found the Seller's Product has been returned without cause and is still serviceable, the Buyer will be notified and the Product returned at the Buyer's expense. A charge for testing and examination may, in the Seller's sole discretion, be made on products so returned.

Sales and Service Information

Thirty years of innovation

Coherent has been the world leader in the manufacture of lasers, and the instruments to measure them, for over thirty years. Coherent Auburn Division offers complete solutions for laser characterization, process control, and system design.

You may contact your local representative or Coherent Auburn Division for further information regarding these items. Call and ask for one of our customer care specialists. Purchase orders may be placed by phone, fax or mail. Online orders can be placed at **http://catalog.CoherentInc.com**

United States
2303 Lindbergh Street
Auburn, CA 95602
Sales Tel: 1-800-343-4912
Sales Fax: 530-889-5366
Service Tel: (530) 888-5062
Service Fax: (530) 889-5262

United Kingdom
15 Greycaine Road
Watford, Herts WD24 7GW
England
Free Phone: 0800 515801
Tel: +44 (0) 1923 206900
Fax: +44 (0) 1923 244461

Germany
Dieselstraße 5b
D-64807
Dieburg
Germany
Tel: +49-6071-968-303
Fax: +49-6071-968-499

France
Domaine Technologique
de Saclay
Bâtiment AZUR
4, rue René Razel
91892 Orsay Cedex
France
Tel: +33-1-60 19 40 40
Fax: +33-1-60 19 40 00

Japan
Toyo MK Building
7-2-14 Toyo
Koto-ku
Tokyo 135
Japan
Tel: +81 (0) 3 5635 8680
Fax: +81 (0) 3 5635 8681