



User Guide | Revision 2017.09.30

Optical Sensing Instrumentation and Software



Micron Optics Inc.
1852 Century Place NE
Atlanta, GA 30345 USA

phone 404 325 0005
fax 404 325 4082
www.micronoptics.com



Table of Contents

- 1. SAFETY AND COMPLIANCE 8**
 - 1.1. Safety 8**
 - 1.2. Line Power 9**
 - 1.2.1. Laser Safety 9
 - 1.2.2. Use in Potentially Explosive Atmospheres 10
 - 1.3. Compliance 10**
 - 1.3.1. Preventative Maintenance 10
 - 1.3.2. Regulatory Testing Information 11
 - 1.3.3. Warranty and Calibration Information 11
 - 1.3.4. WEEE 11
- 2. INTRODUCTION 12**
 - 2.1. Core Technology 12**
 - 2.1.1. x25 Core 12
 - 2.1.2. x41 Multiplexer Technology 12
 - 2.1.3. x55 Core 12
 - 2.1.3.1. High Speed, Wide Wavelength Range Laser 12
 - 2.1.3.2. High Speed Full Spectrum Optical Measurement 13
 - 2.1.3.3. Absolute Wavelength Referencing 13
 - 2.1.3.4. Flexible On-Board Digital Peak Detection 13
 - 2.1.3.5. Full Spectrum Data Sets 13
 - 2.1.3.6. High Dynamic Range, No Configuration Required 14
 - 2.1.3.7. Parallel Multiplexing 14
 - 2.2. Form Factors 14**
 - 2.2.1. 1xx Series – Ruggedized Field Form Factor 14
 - 2.2.2. 2xx Series - Ruggedized Industrial Form Factor (19” Rack Compatible) 14
 - 2.3. Use of Interrogator Naming Convention to Navigate Reference Materials 15**
- 3. QUICK START GUIDES 16**
 - 3.1. sm125 Quick Start 17**
 - 3.2. sm225 Quick Start 19**
 - 3.3. HYPERION Quick Start 21**
- 4. INTERROGATOR CORES 22**
 - 4.1. “x25” Swept Laser Core Technology 22**
 - 4.2. Full Spectrum Measurement 22**
 - 4.3. Internal Peak Detection Application 22**
 - 4.4. Multiplexing 23**
 - 4.5. Network Settings 23**
 - 4.6. Optimizing Peak Detection Parameters 23**
 - 4.7. “x41” Multiplexer Technology 32**
 - 4.7.1. Remote Control Interface 33
 - 4.8. “x55” Swept Laser Core Technology 33**
 - 4.8.1. Sweep Rates and Acquisition Speeds 33
 - 4.8.2. Basic Peak Data Acquisition 33



- 4.8.2.1. Peak Wavelength Data..... 33
- 4.8.2.2. Command/Reply and Streaming Data Transfers..... 33
- 4.8.3. Full Spectrum/Spectral Diagnostic View 33
 - 4.8.3.1. Diagnosing Connector Issues 33
 - 4.8.3.2. Measuring FBG Profile Detail 34
 - 4.8.3.3. Detecting Sensor Debonding or Delamination 34
 - 4.8.3.4. Full Spectrum with Averaging for Deep Dynamic Range 36
- 4.8.4. Internal Peak Detection Capability 37
 - 4.8.4.1. Programmable Hardware Peak Detection Algorithm – High Level Theory of Operation 37
 - 4.8.4.2. MOI Fabry-Pérot Peak Detection:..... 38
 - 4.8.4.3. Other Custom Peak Detection Development and Deployment 38
- 4.8.5. Parallel Multiplexing 39
- 5. “X25” PROGRAMMING INTERFACE..... 40**
 - 5.1. TCP-IP Remote Control Interface 40**
 - 5.1.1. Configuration (applies to smx25 modules) 40
 - 5.1.2. Communication Protocol 40
 - 5.1.2.1. LabVIEW Implementation of x25 TCP-IP Protocol 40
 - 5.1.2.2. Visual Basic Implementation of x25 TCP-IP Protocol 40
 - 5.1.3. Data Retrieval 41
 - 5.1.4. Data Interpretation 41
 - 5.1.5. Data Interpretation for Multiplexed Channels 42
 - 5.2. User Command Set 43**
 - 5.2.1. LabVIEW TCP-IP Examples 48
 - 5.2.1.1. Four Channel Complete Interface Example 49
 - 5.2.1.2. Channel Acquisition Example (with sm041)..... 56
 - 5.2.1.3. Internal Peak Detection Example..... 56
- 6. “X55” PROGRAMMING INTERFACE..... 58**
 - 6.1. Introduction 58**
 - 6.2. Communication Protocols 58**
 - 6.2.1. TCP/IP 58
 - 6.2.1.1. Write 58
 - 6.2.1.2. Read 59
 - 6.2.1.3. Streaming Responses from the Interrogator 60
 - 6.2.2. UDP 60
 - 6.3. HYPERION Remote Command Interface 60**
 - 6.4. x55 APIs and Examples 90**
 - 6.4.1. x55 LabVIEW API 90
 - 6.4.2. six55 TCP/IP Python 91
 - 6.4.3. x55 TCP/IP MATLAB 91
 - 6.4.4. x55 TCP/IP C# 91
- 7. DEPLOYMENT CLASS VARIANT FUNCTIONS 93**
 - 7.1. “si”s – Sensing Instruments 93**
 - 7.1.1. Measurement Time-bases 93
 - 7.1.1.1. Instrument Timestamp 93
 - 7.1.1.2. Network Time Protocol (NTP) 93



- 7.1.1.3. Precision Time Protocol (PTP, IEEE 1588) 93
- 7.1.2. IP Network and Configuration Management..... 93
- 7.1.3. Firmware and FPGA Updates..... 95
 - 7.1.3.1. Scope..... 95
 - 7.1.3.2. Required Resources 95
 - 7.1.3.3. IP Address Management..... 97
- 7.1.4. Front Panel LCD Screen and Other Diagnostics 100
 - 7.1.4.1. Front Panel Tools and Diagnostics 100
 - 7.1.4.1.1 Information 100
 - 7.1.4.1.2 IP Settings..... 100
- 8. FORM FACTOR SPECIFICS..... 101**
 - 8.1. Field Modules, Small Form Factor 101**
 - 8.1.1. si155..... 101
 - 8.1.1.1. Compliance 101
 - 8.1.1.2. Mechanicals 102
 - 8.2. Field Modules, Large Form Factor 104**
 - 8.2.1. si255..... 104
 - 8.2.1.1. Compliance 104
 - 8.2.1.2. Mechanical..... 105
 - 8.2.1.2.1 Ports and Indicators 105
- 9. PLATFORM SPECIFICS 107**
 - 9.1. sm Sensing Modules 107**
 - 9.1.1. Module IP Address Management 107
 - 9.1.1.1. Host PC and module are not on the same logical network..... 107
 - 9.1.1.2. Host PC and module are already on the same logical network. 107
 - 9.1.2. Module Keyboard/VGA Configuration Utilities..... 107
 - 9.1.3. Module Web Interface Utilities 109
 - 9.1.3.1. Home Page..... 109
 - 9.1.3.2. Module Command Console Window 109
 - 9.1.3.3. Module Firmware Upgrade Page 110
 - 9.2. si Sensing Instruments..... 112**
 - 9.2.1. Instrument ENLIGHT Environment Upgrade..... 112
 - 9.2.2. Instrument Integrated ENLIGHT Environment Connection Settings..... 112
 - 9.2.2.1. Host IP Address Management 112
 - 9.2.2.2. Instrument Machine Naming Control 112
 - 9.2.3. Instrument Remote Desktop Instrument Control..... 112
 - 9.2.4. Instrument Local Data Storage and Remote Data Access..... 113
 - 9.2.5. Ethernet Pass-through 114
 - 9.2.6. Safe Instrument Shutdown Procedures. 115
 - 9.3. sp Sensing Processors 115**
 - 9.3.1. Features and Options..... 115
 - 9.3.2. Support Materials 116
- 10. FORM FACTOR SPECIFICS..... 116**
 - 10.1. Field Modules..... 116**
 - 10.1.1. sm125..... 116

10.1.1.1.	Compliance	116
10.1.1.2.	Mechanical.....	117
10.1.2.	sm041.....	120
10.1.2.1.	Compliance	120
10.1.2.2.	Mechanical.....	120
10.1.3.	sp125.....	122
10.1.3.1.	Compliance	122
10.1.3.2.	Mechanical.....	122
10.2.	Rack Mount Units	125
10.2.1.	sm225/si225.....	125
10.2.1.1.	Compliance	125
10.2.1.2.	Mechanical.....	125
10.3.	Cabinet Installation Considerations	130
11.	ENLIGHT SENSING ANALYSIS SOFTWARE	131
11.1.	ENLIGHT Introduction	131
11.1.1.	Background and Motivation.....	131
11.1.2.	New and Different in ENLIGHT Version 1.5.....	131
11.1.2.1.	Features	131
11.1.2.2.	Considerations	132
11.1.3.	Scope and Support	132
11.1.3.1.	Supported Instruments.....	132
11.1.3.2.	Supported Sensors.....	133
11.1.3.3.	ENLIGHT Deployment Versions.....	133
11.1.3.4.	Deployment Configurations.....	133
11.1.4.	Help and References	134
11.1.4.1.	User Manual and White Papers.....	134
11.1.4.2.	Tool Tips.....	135
11.1.5.	Supplementary Programs and Plug-ins	135
11.2.	Conceptual Overview	135
11.2.1.	Concepts.....	135
11.2.1.1.	x25 and x30 Interrogator Cores	135
11.2.1.2.	Defining Measured Elements.....	135
11.2.2.	Structure	136
11.2.2.1.	Acquisition Rate	137
11.2.2.2.	Global Controls	137
11.2.2.3.	Control States	141
11.2.2.4.	Error Indicators	141
11.3.	Operational Overview	142
11.3.1.	Module Selection and IP Settings.....	142
11.3.1.1.	Choosing Module and Configuring IP Settings.....	142
11.3.1.2.	Connection Splash Screen.....	142
11.3.2.	Acquisition tab	143
11.3.2.1.	Data Acquisition.....	143
11.3.2.2.	Spectrum View.....	144
11.3.2.3.	Peak Detection.....	147
11.3.2.4.	Table View	152
11.3.2.5.	Chart View	153




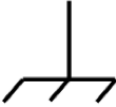


11.3.2.6.	FFT View (x30 interrogator cores only).....	153
11.3.2.7.	Multiplexing.....	154
11.3.3.	Sensors tab.....	156
11.3.3.1.	FBGs.....	156
11.3.3.2.	Sensors.....	162
11.3.3.3.	Sensor Operating Temperature Range: A Manually Created Macro-Sensor Example.....	178
11.3.3.4.	Creating Sensors from MOI Definitions (*.mst) Files – Multiple Examples.....	182
11.3.3.5.	Referencing Sensors (independent FBG zeroing).....	192
11.3.4.	Charts Tab.....	195
11.3.4.1.	Show/Hide Chart.....	196
11.3.4.2.	Reset All.....	196
11.3.4.3.	Show/Hide Chart Setup.....	196
11.3.5.	Image tab.....	198
11.3.5.1.	Select Image.....	198
11.3.5.2.	Hide/Show Image Setup.....	198
11.3.5.3.	Indicator types.....	199
11.3.6.	Archive Tab.....	201
11.3.6.1.	Data Archive Management Controls.....	201
11.3.6.2.	ENLIGHT Data Archive Display Parameters.....	201
11.3.6.3.	ENLIGHT Data Archive Retrieval and Archive Chart Display Examples.....	204
11.3.7.	Save Tab.....	209
11.3.7.1.	Configuration.....	210
11.3.7.2.	Local Data Saving.....	211
11.3.8.	Alerts Tab.....	227
11.3.8.1.	Message Notifications.....	227
11.3.8.2.	Event Log.....	231
11.3.9.	Settings Tab.....	232
11.3.9.1.	Hardware.....	232
11.3.9.2.	Notification.....	235
11.3.9.3.	Configuration Notes.....	236
11.3.9.4.	Application Settings.....	237
11.3.9.5.	Module Diagnostics.....	238
11.3.9.6.	Data Archiving.....	238
11.3.9.7.	Firmware Upgrade.....	244
11.3.9.8.	News & Updates.....	244
11.4.	ENLIGHT Remote Command Interface.....	245
11.4.1.	TCP/IP Remote Control Interface.....	245
11.4.1.1.	Overview.....	245
11.4.1.2.	Communication Protocol.....	245
11.4.1.3.	Data Retrieval (polling vs. streaming).....	246
11.4.1.4.	User Command Set.....	246
11.4.2.	ENLIGHT Remote Command Interface LabVIEW Example.....	255
11.4.2.1.	Introduction.....	255
11.4.2.2.	Command.....	255
11.4.2.3.	Get Sensors.....	255
11.4.2.4.	Stream Sensors.....	255
11.4.2.5.	Get Data Archives.....	256
11.4.2.6.	Advanced.....	256
11.1.	ENLIGHT Web Interface.....	257
12.	APPENDICES.....	258

12.1.	Using Legacy Notation for MOI Multiplexers	258
12.2.	Configuring TCP-IP Settings for Interrogator Instrument Connections	258
12.2.1.	Selecting Settings to Connect to the Module.....	258
12.2.1.1.	Setting the Netmask and IP Address on the Host PC.....	259
12.2.1.1.1	Windows XP	259
12.2.1.1.2	Windows 7/8.....	260
12.3.	Speed of Light Effects on Swept Laser based Wavelength Measurements	263
12.3.1.	Introduction	263
12.3.2.	Unique Benefits of a Swept Laser based System.....	263
12.3.2.1.	Broadband source, dispersive element, diode array	263
12.3.2.2.	Broadband source, Optical Spectrum Analyzer/Multi-line wavelength meter	263
12.3.2.3.	OTDR/TDM systems	263
12.3.2.4.	External Cavity Tunable Laser, Power Meter, Wavelength Meter.....	263
12.3.2.5.	Swept Wavelength Fiber Laser Source	264
12.3.3.	Swept Laser Calibration Methodology	265
12.3.3.1.	Case 1: Distance to sensor = ~0m.	265
12.3.3.2.	Case 2: Distance to sensor = 100m.	266
12.3.3.3.	8.3.3.3 Case 3: Distance to sensor = 20km.....	267
12.3.3.4.	Estimation of Speed-of-Light Induced Wavelength Shift.....	267
12.3.4.	sm130 Speed-of-Light Effect Correction Algorithm and Software	267
12.3.4.1.	Speed-of-light Induced Shift Effects and Correction.....	267
12.3.4.2.	Calibration Measurements for Correction Algorithm and Results.....	268
12.3.4.2.1	Known or assumed index, OTDR measurement for length.....	268
12.3.4.2.2	Known or assumed index, known wavelength artifact at end of fiber.	268
12.4.	Specification Definitions and Test Methodology	269
12.4.1.	Wavelength Accuracy.....	269
12.4.2.	Repeatability	269
12.4.3.	Stability.....	270
12.4.4.	Resolution	270
12.4.5.	Reproducibility	270
12.5.	Optical Strain Gage Temperature Compensation Techniques.....	270
12.6.	International Certificate for Fiber Optic Sensing	271
12.7.	ATEX Declaration of EC Conformity.....	272

1. SAFETY AND COMPLIANCE

1.1. Safety

The following symbols and messages may be marked on the unit. The purpose of safety symbols is to alert the user to possible dangers. The safety symbols and the explanations with them deserve your careful attention and understanding. The safety WARNING's do not by themselves eliminate any danger. The instructions or WARNING's they give are not substitutes for proper accident prevention measures.

Symbol	Description
	Laser Safety. Refer to user's manual for safety instructions for use and handling.
	Refer to user's manual for safety instructions for use and handling.
	Caution. Risk of electric shock.
	Frame or chassis terminal for electrical earth ground.
	Electrostatic discharge (ESD). Refer to user's manual for safety instructions in use and handling.
	Protective conductor terminal for electrical earth ground.
WARNING	This sign indicates a procedure with the potential to cause serious injury or loss of life to the user if not performed with strict adherence to all safety instructions. Ensure that all conditions are fully understood and met before proceeding.
CAUTION	This sign indicates a procedure with the potential to cause serious damage to or destruction of the unit if not performed with strict adherence to the all safety instructions. Ensure that all conditions are fully understood and met before proceeding.

1.2. Line Power

WARNING: If equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be lessened or eliminated. There are no internal user-serviceable parts.

The x55 (HYPERION) and x25 models of instruments covered in this document can operate from any DC power supply that outputs 7 – 36 V and 80 W of power. All instruments are supplied with an external AC adapter which accepts an input voltage ranging from 100 - 240 VAC within a frequency range of 47 - 63 Hz. Please see the product data sheets at www.micronoptics.com for power consumption specifications for specific modules.

WARNING: To avoid the possibility of injury or death, do not operate any electrical device with visible damage to power supply, line cord, or outer enclosure.

CAUTION: Do not expose the unit directly to rain or other excessive moisture.

CAUTION: IMPORTANT NOTICE REGARDING POWER CONNECTORS: All instruments covered in this document are equipped with a panel mounted four terminal block removable connector that is intended for fixed wiring applications. It is important that the power be applied in the proper sequence:

1. Begin with the power switch in the OFF position.
2. Connect the power supply leads to the terminal block.
3. Plug the power connector block into the panel of the interrogator unit.
4. Tighten the two (2) retaining screws.
5. Move the power switch to the ON position.

If the panel mounted power terminal block needs to be removed, the power supply MUST be turned OFF BEFORE it is disconnected or reconnected. This is extremely important. These connectors are not intended to be used as switches.

Micron Optics recommends the use of a qualified surge protector to prevent damage from unexpected power transients.

1.2.1. Laser Safety

This section applies to the following MOI interrogators: all x55 models, all x25 models.

Laser Type	Fiber Laser
Laser Class (IEC 60825-1, 21 CFR 1040.10)	1
Max Output Power	0.25 mW
Min Output Power	0.06 mW
Mode Field Diameter	~ 10.4 μm
Numerical Aperture	0.14
Wavelength	1460 – 1620 nm

NOTE:



These laser safety WARNING labels are fixed on the instrument.

WARNING: Malfunctioning lasers MUST be returned to Micron Optics for repair and calibration.

WARNING: Use of controls or adjustment of performance or procedures other than those specified for the laser source may result in hazardous radiation exposure.

Refer servicing only to qualified and authorized personnel. There are no internal user-serviceable parts.



WARNING: Do not enable the laser (turn on instrument) when neither an optical connector nor LC/APC connector cover is connected to the optical output connector.

The laser is enabled when power is supplied to the instrument and the instrument is powered. The red LED on the instrument from panel indicates power is present.

WARNING: Under no circumstances look into the end of an optical cable attached to the optical output when the device is operational.

WARNING: The laser radiation is not visible to the human eye, but it can seriously damage your eyesight.

CAUTION: Connecting damaged or dirty fibers to the unit will damage the optical connectors on the unit. Always clean optical connectors before making a connection.

CAUTION: Never force-fit an optical connector. A ferrule may break and damage the unit.

CAUTION: Do not attempt to connect a PC (physical contact) optical connector to any of the panel mounted APC (angle polished) connectors of the module. Such mismatched connections can lead to malfunction or damage to the module.

Follow standard guidelines for cleaning optical fibers.

1. Fold a lint-free wipe into a compress.
2. Moisten the compress with isopropyl alcohol.
3. Remove the connector's protective cover.
4. Press the connector end face firmly to the moistened section of the compress for a moment then forcefully wipe the end with a twisting motion towards the edge of the compress, finishing in a clean, dry section of the compress.
5. Repeat process, making sure not to reuse dirty sections of the compress.
6. Discard the used compress.

NOTE: In order to maintain optimal measurement performance of the Micron Optics interrogator, the panel mounted optical connectors must be kept clean. Use of an In-Adapter Ferrule Cleaner is recommended periodically. Please contact Micron Optics or visit www.micronoptics.com to purchase or receive a recommendation for an appropriate ferrule cleaner.

Rather than direct connection of optical sensors to the interrogator, Micron Optics recommends the use of short patch cords to a fiber patch panel for the following reasons:

1. Reduced handling and connection/disconnection cycles directly on the interrogator, reducing wear on the optical connections of instrument.
2. Hybrid LC to FC/APC (or other) jumpers provide for maximum connector flexibility.
3. Maintaining a constant LC connection to the interrogator minimizes opportunities for dust ingress.
4. Short, non-armored fiber jumpers can ensure a non-conductive path from any stainless steel armored cables to module.

1.2.2. Use in Potentially Explosive Atmospheres

All Micron Optics (MOI) Interrogators MUST be located and operated in a safe zone. With ATEX certification the passive components of an optical sensing system (optical fiber, interconnects, fiber Bragg grating sensors, etc.) that are optically coupled to an ATEX certified MOI interrogator may be installed and operated in Zone 0, 1 or 2 environments. Contact Micron Optics for more information regarding purchasing an ATEX certified interrogator.

1.3. Compliance

1.3.1. Preventative Maintenance

The HYPERION (or x55) family of optical sensor interrogators has been designed to operate without need for a fan, eliminating filter maintenance concerns. Good practices for power conditioning and environmental control to meet module temperature and conduction requirements will ensure extended, trouble-free operation.

1.3.2. Regulatory Testing Information

Each optical interrogator module is tested to a specific set of industry qualification and regulatory standards. See Section 10 for more hardware specific information.

1.3.3. Warranty and Calibration Information

Micron Optics offers excellent warranty protection for all components, modules and instruments sold. Details regarding the warranty are available in the Terms and Conditions of Sale distributed at the time of purchase.

The Micron Optics Optical Sensing Interrogators are designed with long-term field deployment in mind. Continual on-board calibration procedures using Telcordia qualified optical referencing components and/or NIST Standard Reference Materials ensure that sensor wavelength measurements remain within specification over the life of the product.

1.3.4. WEEE

Micron Optics is committed to meeting the requirements of the European Union (EU) Waste Electrical and Electronic Equipment (WEEE) Directive. This Directive requires producers of electrical and electronic equipment to provide for the take-back, for reuse or recycling, of their products placed on the EU market after August 13, 2005. This commitment is indicated by the following label affixed to all Micron Optics interrogators.



Micron Optics products that are within the scope of the Directive are labeled with a crossed-out "wheeled-bin" symbol, as required by the Directive. It indicates that the product was placed on the market after August 13, 2005 and that end users should segregate the product from other waste at end-of-life.


The WEEE Directive is being implemented in each of the 27 EU countries through national legislation. Norway and Switzerland have also implemented similar pieces of legislation. As a result, the detailed requirements vary considerably throughout the EU, and the Micron Optics WEEE compliance approach varies among countries. In each country affected by the WEEE legislation:

- Micron Optics offers, where required, a product take-back service to its business customers.
- Micron Optics will seek to register as a producer wherever local legislation requires.

In addition:

- The equipment that you bought has required the extraction and use of natural resources for its production. It may contain hazardous substances that could impact health and the environment.
- In order to avoid the dissemination of those substances in our environment and to diminish the pressure on the natural resources, we encourage you to use the appropriate take-back systems. Those systems will reuse or recycle most of the materials of your end life equipment in a sound way.



- The crossed-out wheeled bin symbol () invites you to use those systems.
- If you need more information on the collection, reuse and recycling systems, please contact your local or regional waste administration.
- You can also contact Micron Optics for more information on the environmental performances of our products.

2. INTRODUCTION

HYPERION (or x55) is a completely new optical sensing platform based upon the mature x25 core technology. Over time, the platform will support a multitude of measurement modalities, processing capabilities, and communication protocols from within a single software and user interface framework.

Each Micron Optics interrogator product within the HYPERION platform shall be defined by three principal elements. First, the core technology defines the types of measurements the interrogator can make, in terms of measurement methodology, data rate, dynamic range, wavelength accuracy, repeatability, stability, and other key performance parameters. Second, the software Deployment Class on which the interrogator core technology is deployed determines the on-board data processing and storage capabilities of the interrogator. Lastly, the physical form factor determines the size, volume, user interface options, and power supply considerations of the interrogator. The following sections introduce the reader to the available options for core technology, platform, and form factor within the MOI Optical Sensing Interrogator product line.

2.1. Core Technology

Two core measurement technologies are covered in this user manual.

2.1.1. x25 Core

The x25 optical interrogator core is based upon swept wavelength laser based measurement technology that is enabled by full spectral scanning and data acquisition. The x25 core technology is focused on providing reflected spectral measurements with high absolute accuracy, software post-processing flexibility, and loss budget/dynamic range measurement performance. The x25 core technology based interrogators are typically characterized by lower acquisition rates and higher dynamic ranges, and also support continuous on-board NIST traceable wavelength reference components. The x25 based interrogators are also ideally suited to measure many different optical sensor types, including FBG's, long period gratings, extrinsic Fabry-Pérot sensors, and many others. Interrogators based upon x25 technology are typically best suited for "static" measurement applications, requiring data acquisition rates in the low singles or tens of Hz.

2.1.2. x41 Multiplexer Technology

The x41 multiplexers employ a network of optical switches or passive coupler components and are used to expand the physical or spectral distribution of sensor measurements on the x25 platform. The x41 switch modules feature high speed, solid state optical switches that provide years of reliable and durable optical switching functionality.

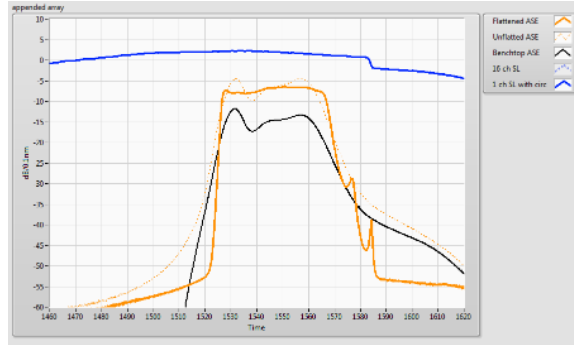
This feature is **only** available with the x25 technology.

2.1.3. x55 Core

The x55 optical interrogator core is comprised of a high speed capable, wide wavelength swept laser source, a high speed/full optical spectrum data acquisition engine, and traceable absolute wavelength referencing artifacts that combine to yield a combination of features and specifications previously un-attainable in a single instrument.

2.1.3.1. High Speed, Wide Wavelength Range Laser

Central to the x55 core is a new generation of swept wavelength fiber laser. The x55 laser is capable of scanning a wide 1460 to 1620 nm wavelength range at the dynamic speeds of up to 1 kHz (or 80 nm scan at 5 kHz), with a low degree of noise and jitter to support measurements with wavelength accuracies of 1 pm. In addition to high speed and wide range, the x55 laser support scan rates from 2 Hz to 5 kHz to extract the optimum combination of speed, repeatability, and range for the measurement.



The figure above shows the high output power and wide wavelength range of the si255 swept wavelength laser (blue) in comparison to traditional Erbium ASE sources (orange and black).

2.1.3.2. High Speed Full Spectrum Optical Measurement

Previously relegated only to static (1 – 10 Hz) speeds, the x55 core brings the clarity and precision of optical full spectrum measurement to dynamic laser sweep and data acquisition rates. Powered by the latest generation of high speed digitizers and signal processors, the x55 digitizes and processes a full high resolution digital spectrum for each measurement channel during each high-speed acquisition cycle. Collecting the full sensor spectrum on each sweep of the laser for each channel affords the benefits described in the following sections.

2.1.3.3. Absolute Wavelength Referencing

Previously limited to only static interrogators, the x55 core has the high-speed sampling resolution necessary to measure NIST certified absolute wavelength reference materials concurrent with every sweep, ensuring both short and long term absolute accuracy and stability of the measurements. The x55 samples atomic absorption spectra, detecting and analyzing multiple discrete absolute wavelength lines in real time on every scan. No other dynamic optical sensing interrogator technology on the market can match this feature.

2.1.3.4. Flexible On-Board Digital Peak Detection

Before the x55, high speed optical sensor analysis relied upon specific, rigid peak detection parameters, which were optimized for a particular type of sensor and/or detection methodology. For example, a high-speed diode array will require a relatively coarse center of gravity calculation or Gaussian fit to find peaks from its limited number of detection points. An edge filter system will always look at variations in total integrated power. Previous high speed swept laser systems have relied upon a fixed analog detection circuit which, although both high performance and reliable, left little room for modifications to accommodate sensors of varying spectral design (such as FBG's of different widths, apodization profiles, reflectivity, etc.) or device type (such as LPG's, chemical composition gratings, or Fabry-Pérot sensors). At its most basic level of functionality, the x55 core simply the laser and digitizes the full optical spectrum at high resolution, thus offering a wealth of possibilities in processing the spectrum and deriving optimal meaning from the measured sensor.

Even with this degree of flexibility, the user is not tasked with converting spectrum to peak information at high speed. The x55 technology core is loaded with a standard set of highly reliable, high performance programmable hardware-driven peak detection presets that converts the measured full spectrum to a robust set of measured fiber Bragg gratings (FBG), long-period gratings (LPPG), or Fabry-Pérot FP sensor peaks or valleys in real time over a wide variety of sensor variations and differential loss conditions. Not only DUT peak detection but reference comparisons and wavelength calibration conversions are all handled deterministically in hardware to ensure the highest performance and most reliable sensor measurements available today. Moving forward, and in collaboration with our industry partners, *x55 will grow to support a collection of peak detection presets* that take advantage of the core high-speed, high accuracy full spectrum measurement.

2.1.3.5. Full Spectrum Data Sets

Lastly, the obvious reason to digitize the full optical spectrum of a signal is to equip the user with that critical data to ensure robust and reliable system setup, configuration, and ongoing sensor and system diagnostic support. The x55 core



down-samples and averages the fast full spectrum to an optimized, user-consumable approximately 10 Hz rate for delivery to client applications. Depending on the application, a host of information can be gleaned from the full spectrum measurements, including connector losses, connector back-reflections, sensor peak distortion, bonding delamination, etc. Additionally, for custom sensor types, custom peak detection algorithms can be developed by externally monitoring and processing the full optical spectrum of the signals. All while internal peak detection operates simultaneously at the full acquisition rate.

Calibrated full spectrum traces from the x55 instruments can also be used as the measurement data by the customer for software-driven spectral detection algorithms exercised in client code. The x55 core instruments can serve up to 16 channels of simultaneous full spectrum channels simultaneously over Ethernet at rates of up to 10 Hz.

2.1.3.6. High Dynamic Range, No Configuration Required

The x55 optical interrogator core brings to market a new receiver circuit technology that eliminates the need for receiver side gain adjustments, either automatic or manual. Previously relegated to static measurement systems, the x55 core brings to dynamic sensor measurements a continuous high dynamic range and even higher full spectrum dynamic range, available to each sensor on each channel, independent of differential system losses. For the user, this means more optical measurement dynamic range with less configuration. In conjunction with the high performance on-board DSP peak detection algorithms, the x55 receiver is truly “plug-and-play”. There are no gains to set and no ranges to set for “automatic” gain control, making for a truly simple and straightforward user experience with this dynamic interrogator.

NOTE: in certain application circumstances, different receiver technologies may still offer specific advantages. Please refer to the datasheets for the various options and models at www.micronoptics.com.

2.1.3.7. Parallel Multiplexing

Lastly, all of the advantages described by the features above can be realized on up to 16 parallel channels simultaneously. The x55 line of instruments can read 20,000 full spectrum data points at the full acquisition rate (at 1kHz) over 160 nm on each of 16 parallel channels simultaneously, eliminating the need for optical switches or other modes of multiplexing that otherwise degrade performance. Because of the high power from the swept wavelength laser source, there is sufficient optical power on each of the 16 channels to overcome appreciable transmission losses that the measurement system is likely to incur. Through parallel detections, all sensors on all channels receive the full benefit of fast acquisition speed, absolute accuracy, full spectrum data, and full independent loss budget or dynamic range.

2.2. Form Factors

In addition to core technology and firmware platform options, an optical interrogator is defined by its physical form factor. Micron Optics offers sensor interrogation instrumentation into several distinct form factors

2.2.1. 1xx Series – Ruggedized Field Form Factor

Examples: si155, sm125, sm041

The 1xx series ruggedized field form factor is designed to deploy smaller sensor counts implementations (up to 4 channels) in a minimum overall volume, while supporting wide operating temperature ranges and stable operation over rigorous shock and vibration conditions. The small physical footprint of the 1XX series field form factor makes it an ideal physical fit for many installation scenarios. The field unit form factor also has the lower average power consumption of all of the form factors and is the frequent choice for power, weight, or volume restricted installations.

2.2.2. 2xx Series - Ruggedized Industrial Form Factor (19" Rack Compatible)

Examples: si255, sm225

The 2xx series ruggedized industrial form factor is designed to facilitate larger sensor counts implementations (from 4 to 16 parallel channels) in a minimal overall volume, while supporting wide operating temperature ranges and stable operation over rigorous shock and vibration conditions. The 2xx Series can also be made compatible with standard 19" rack mount enclosure installation through the use of instrument specific mounting hardware.



The combination of form factor, platform, and core technology are reflected in the model number for all Micron Optics Interrogation instrumentation. For example, an sm125 module is comprised of:

- sm125. => **Sensing Module** Platform.
- sm125. => **1xx** Field Form Factor.
- sm125. => **x25** Swept Laser Core Technology.

Similarly, an si255 Instrument is comprised of:

- si255. => **Sensing Instrument** Platform.
- si255. => **2xx** Rack Mount Form Factor.
- si255. => **x55** Swept Laser Core Technology.

2.3. Use of Interrogator Naming Convention to Navigate Reference Materials

The above defined terms and alphanumeric product model designations will be used to indicate applicability of each section of documentation to a particular subset of the MOI interrogation system line. An “x” or “X” will be used to represent universal applicability as it pertains to that portion of the module naming convention.

For example,

- a section that refers to an “x55 unit”, “x55 interrogator” or “x55 core” will apply universally to si155 and si255 instruments and NOT (for example) to any x25 units (not the sm125, sm225).

The following section of Quick Start guides will make use of the guidelines set forth above to ease the navigation of this manual.

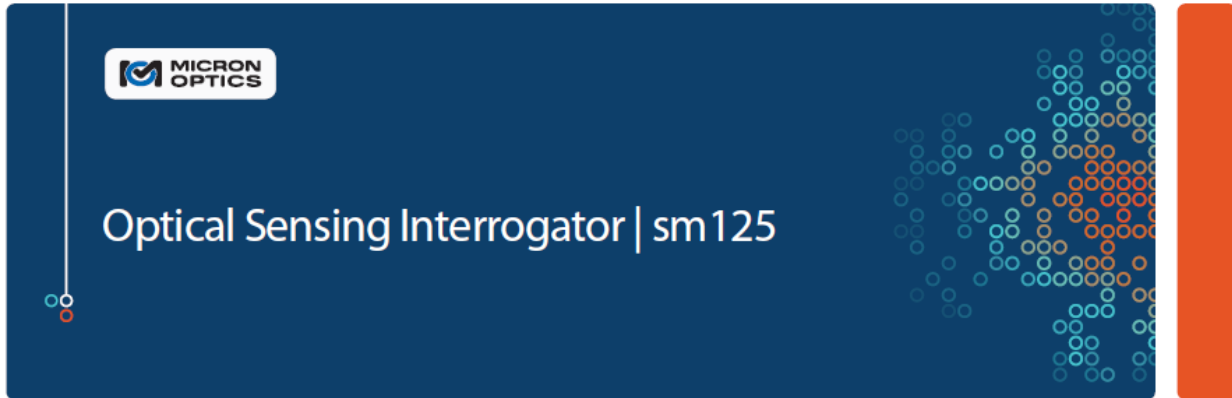


3. QUICK START GUIDES

The following section is a collection of Quick Start guides for each Micron Optics optical sensor interrogator. The purpose of the Quick Start guides is to assist the user with hardware and software setup so that sensor measurements can be made quickly and simply with each interrogator type.

This page is intentionally left almost blank.

3.1. sm125 Quick Start



sm125 Optical Sensing Interrogator Module

The interrogator module monitors the optical sensors and outputs wavelength data via Ethernet port and custom MOI protocol.

External PC running ENLIGHT

LabVIEW examples, or other custom client code. All module settings, sensor calculations, data visualization, storage, and alarming tasks are run on external PC or sp125 module.

Navigating User Guide Resources

sm125. => **1xx** Field Form Factor.

Refer to section 8 for information on mechanical ports, drawings, and power consumption, as well as regulatory and reliability testing.

sm125. => **x25** Swept Laser Core Technology.

Refer to section 2.1.1 for information on interrogator core technology, including: functions and features, TCP/IP remote control interface, and peak detection parameter optimization.

ENLIGHT

Refer to section 11 of the manual for information on ENLIGHT Sensing Analysis Software.

sm125 Quick Start Instructions

Follow this quick set of instructions to begin making sensor measurements with the sm125 Module.

What's in the box?

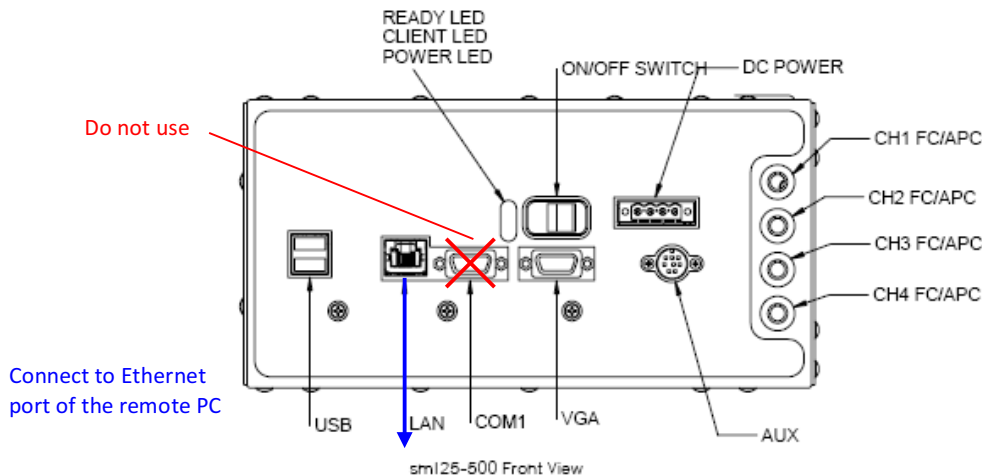
1. To get started, locate and identify the sm125, the included Ethernet crossover cable, AC/DC converter, and AC line cord.

What else is needed?

2. Have an FBG sensor available to measure and a networked PC for remote access to the instrument.
3. Configure the remote PC NIC Card with a static IP address of 10.0.0.121 and Netmask 255.255.255.0.
4. Download and Install Micron Optics ENLIGHT Software via the link at <http://www.micronoptics.com/products/sensing-solutions/software/>.
5. Unzip the downloaded file and run "setup.exe". Follow the on-screen instructions to complete the installation.

Prepare the Instrument Hardware:

6. Plug the AC line cord into AC adapter and then AC adapter to sm125 Module and move power switch to the ON position.
7. Connect test FBG to Channel 1 Optical Connector.
8. Connect the Ethernet crossover cable between the port labeled "LAN" on the sm125 Module and the Ethernet port of the remote PC.



Connect to the Instrument:

9. On the remote computer, click Start, point to All Programs, and then point to Micron Optics.
10. In the Micron Optics menu, point to ENLIGHT, and then click MOI-ENLIGHT
11. In the IP Connection Parameters box shown in the following figure, choose the x25 Swept Laser Core and enter the default sm125 IP address of 10.0.0.122.
12. Click "OK" and the ENLIGHT software will connect to the module.
13. For additional assistance, refer to the Micron Optics Optical Sensing Instrumentation and Software User Guide.

NOTE: If the IP address of the module has been changed from the default value of 10.0.0.126, the updated value will be displayed on the VGA output of the module. Verification of successful IP configurations can be made by typing "ping 10.0.0.122" into a command prompt (Start->Run->type "cmd <Enter>"). A successful ping reply will be of the form "Reply from 10.0.0.122: bytes=32 time=1ms TTL=63".

3.2. sm225 Quick Start



sm225 Optical Sensing Interrogator Module

The interrogator module monitors the optical sensors and outputs wavelength data via Ethernet port and custom MOI protocol.

External PC running ENLIGHT

LabVIEW examples, or other custom client code. All module settings, sensor calculations, data visualization, storage, and alarming tasks are run on external PC, external sp125 module, or optional internal processor module.

Navigating User Guide Resources

sm225. => **2xx** Rack Mount Form Factor.

Refer to section 8 for information on mechanical ports, drawings, and power consumption, as well as regulatory and reliability testing.

sm225. => **x25** Swept Laser Core Technology.

Refer to section 2.1.1 for information on interrogator core technology, including: functions and features, TCP/IP remote control interface, and peak detection parameter optimization.

ENLIGHT

Refer to section 11 for information on ENLIGHT Sensing Analysis Software.

sm225 Quick Start Instructions

Follow this quick set of instructions to begin making sensor measurements with the sm225 Module.

What's in the box?

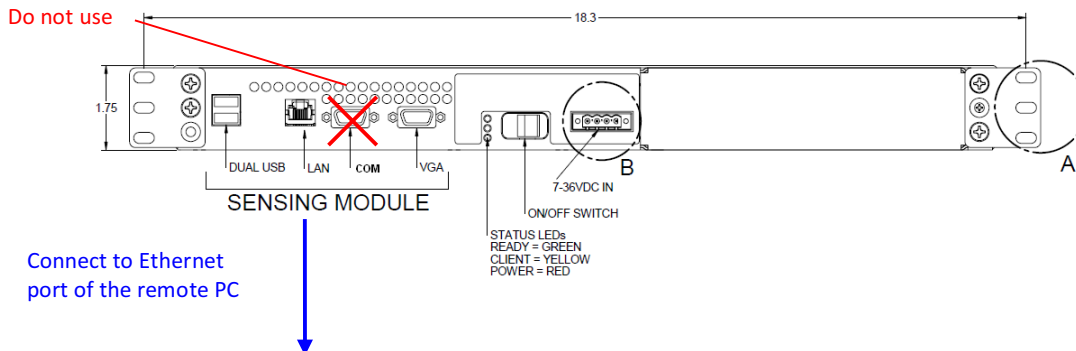
1. To get started, locate and identify the sm225, the included Ethernet crossover cable, AC/DC converter, and AC line cord.

What else is needed?

2. Have an FBG sensor available to measure and a networked PC for remote access to the instrument.
3. Configure the remote PC NIC Card with a static IP address of 10.0.0.121 and Netmask 255.255.255.0.
4. Download and Install Micron Optics ENLIGHT Software via the link at <http://www.micronoptics.com/products/sensing-solutions/software/>.
5. Unzip the downloaded file and run "setup.exe". Follow the on-screen instructions to complete the installation.

Prepare the Instrument Hardware:

6. Plug the AC line cord into AC adapter and then AC adapter to sm225 Module and move power switch to the ON position.
7. Connect test FBG to Channel 1 Optical Connector (not pictured).
8. Connect the Ethernet crossover cable between the port labeled "LAN" on the sm225 Module and the Ethernet port of the remote PC.



Connect to the Instrument:

9. On the remote computer, click Start, point to All Programs, and then point to Micron Optics.
10. In the Micron Optics menu, point to ENLIGHT, and then click MOI-ENLIGHT
11. In the IP Connection Parameters box shown in the following figure, choose the x25 Swept Laser Core and enter the default sm225 IP address of 10.0.0.122.
12. Click "OK" and the ENLIGHT software will connect to the module.
13. For additional assistance, refer to the Micron Optics Optical Sensing Instrumentation and Software User Guide.

NOTE: If the IP address of the module has been changed from the default value of 10.0.0.122, the updated value will be displayed on the VGA output of the module. Verification of successful IP configurations can be made by typing "ping 10.0.0.122" into a command prompt (Start->Run->type "cmd <Enter>"). A successful ping reply will be of the form "Reply from 10.0.0.122: bytes=32 time=1ms TTL=63".

3.3. HYPERION Quick Start



Optical Sensing Interrogator | HYPERION

What's in the box?

- Micron Optics HYPERION Optical Sensing Interrogator
- Power Supply
- AC Line Cord
- Ethernet Crossover Cable

What else is needed?

- Windows 7 or Windows 10 PC
- An LC/APC connectorized 0.25 nm FBG sensor
- Ethernet cable (used when NOT directly connecting interrogator to PC)

Getting ready

1. Download and install ENLIGHT from the product page: <http://www.micronoptics.com/product/optical-sensings-instrument-si155>
2. Power the HYPERION Optical Sensing Interrogator by connecting the power supply and the AC line cord to the instrument. Toggle the power switch until the red LED illuminates. The instrument is booted and ready when the green LED is illuminated.
3. Connect your LC/APC connectorized 0.25 nm FBG to channel #1 of the HYPERION Optical Sensing Interrogator.
4. Connect your HYPERION Optical Sensing Interrogator to your Windows PC (2 ways):
 - a. Direct Connection
 - i. Configure your PC's Ethernet NIC to a static IP address of 10.0.0.2, Subnet Mask of 255.255.255.0, and Default Gateway of 10.0.0.1.
 - ii. Connect the si155 Optical Sensing Interrogator to your PC via the included Ethernet crossover cable.
 - iii. Ensure the si155 Optical Sensing Interrogator is set to the **static** IP address 10.0.0.55. Press the front panel CTRL button until the IP address is displayed. If the IP address screen does display "10.0.0.55 (S)", press the CTRL button for 15 seconds to reset the IP address.
 - b. Network Connection (via network switch or router)
 - i. Connect the HYPERION Optical Sensing Interrogator to the same network as your PC via an Ethernet cable (not included).
 - ii. Ensure the HYPERION Optical Sensing Interrogator is set to acquire a **dynamic** IP address. Press the front panel CTRL button until the IP address is displayed. If the IP address does not end in a "(D)", press the CTRL button for 5 seconds to enable a dynamic IP address.
5. Open ENLIGHT.
6. Select "HYPERION" in the Swept Laser Core drop down menu.
7. Enter in the IP address displayed on the LCD screen of the instrument.
8. Click OK
9. Start making measurements!

4. INTERROGATOR CORES

4.1. “x25” Swept Laser Core Technology

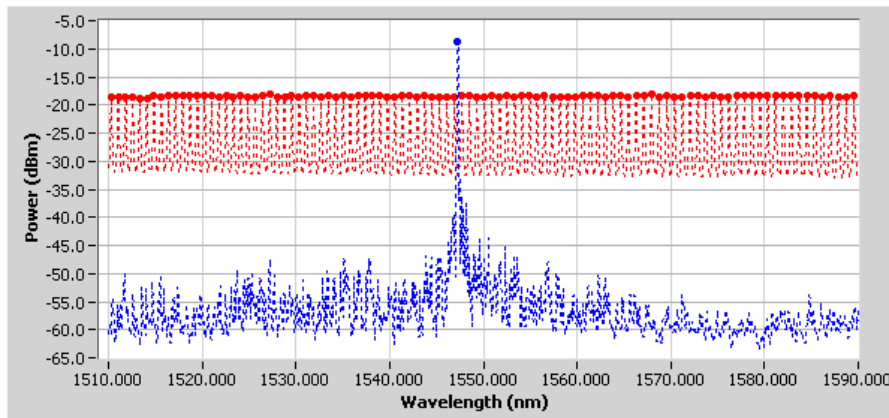
4.2. Full Spectrum Measurement

The principal measurement mode for the x25 interrogators is a Full Spectrum Measurement. When polled, the unit returns the full optical spectrum profile for each of the enabled channels. As many as four simultaneous channels of data can be collected. Use of a multiplexer module will increase the channel count to as high as 16. The module scanning rate and spectral sampling density varies with module model.

4.3. Internal Peak Detection Application

With the full optical spectrum signal available from the sensors, the x25 units offer great flexibility in the area of peak detection and data interpretation. Custom sensors can be analyzed with any type of user-developed custom peak detection.

The x25 interrogator is most frequently used to measure the reflection spectrum of Fiber Bragg Grating sensors. As such, MOI has developed a flexible, yet robust peak detection algorithm for use with the x25 and FBG sensors. This peak detection algorithm can be accessed on port 55000 of a version 1.x x25 unit and on both ports 50000 and 55000 of a version 2.x x25 unit.



x25 full spectrum trace with peak detection results

The x25 peak detection algorithm makes use of the following parameters:

Rel. Thresh (dB):

This control sets the relative threshold for the peak detection algorithm in units of dB. The relative threshold will set the effective threshold for the peak detection algorithm at the prescribed level below the point of highest power for peak detection, or above the point of lowest power for valley detection. For example, if a relative threshold of -3 dB is set for a channel with highest optical power at -15 dBm, the effective threshold will be set at -18 dBm, provided that the Threshold value is set below -18 dBm.

For example, assume that a single FBG is to be monitored on a particular channel, like the FBG seen in the figure above on Channel 1. It can be seen that the peak power value of the FBG is \sim -8 dBm. Use of the Rel. Thresh parameter with a value of -3 dB will set the effective threshold for the peak detection at -11 dBm, which is well above the noise floor of this instrument. Should the level of the FBG change, the relative nature of the peak detection will allow the unit to continue to find the FBG center location. For example, if the round trip path loss for the FBG should increase by 10 dB, the peak power value of the FBG would be -18 dBm, and Rel. Thresh would still be -3 dB, and so the new effective threshold would be -21 dBm.

Threshold (dBm):

This control sets the absolute minimum threshold that can be used by the peak detection algorithm. Use of this parameter prevents the relative threshold from setting the effective threshold below a known noise floor limit. If the value of threshold set

by the Rel. Thresh parameter is lower than the Threshold value for peak detection (or higher, for valley detection), then the value set by Threshold will be used.

Continuing the example from above, should the path loss of the FBG were to increase substantially, say by 40 to 50 dB, it is possible that the reflected signal could be dominated by noise, either in form of system back reflection or instrument noise floor. In that event, it would not be optimal for the peak detection algorithm to continue to set the effective threshold 3 dB below the point of highest power: doing so could result in peak detection of noise, not signal. To prevent this from happening, the Threshold parameter is used. The value set by Threshold defines the lowest effective threshold that will be used by the peak detection algorithm, despite a value which might be set lower by the relative threshold.

Width Level (dB):

This parameter sets the relative level for each peak at which the width will be compared to the width requirement, seen in the next parameter description. Typically, this value might be set to 3 dB, indicating that the sensor's 3 dB bandwidth will be evaluated against the width requirement in determining its peak value. This parameter can be modified in order to isolate desired signals in the presence of unwanted side lobes. It can also be modified to track signals of smaller than normal contrast in a highly back reflective environment.

Width (nm):

This parameter is used to set the width requirement for the peak detection, in conjunction with the Width Level parameter described above. For a given pair of parameters, only spectral features with a bandwidth evaluated at a level of Width Level down from the peak of value greater than Width will be considered to be a peak.

For example, assume Width Level and Width values of 3 dB and 0.1 nm, respectively. A sensor feature will only be captured a peak if the width of the signal at 3 dB below the peak exceeds 0.1 nm. These parameter values will detect signals with 3 dB bandwidths of 0.200 nm, for example, but will ignore any side lobes adjacent to the signal that might have 3dB bandwidths of 0.07nm.

Please see section 4.6 of this document for tips on optimizing peak detection parameters.

4.4. Multiplexing

The x25 units supports many tens of sensors over a wide common wavelength range on each of four simultaneous, parallel measurement channels. Should an application call for a larger number of sensors or a wider effective spectral range, the number of channels can be functionally increased through the use of an sm041 Channel Multiplexer. The sm041 Channel Multiplexer consists of a series of high speed, highly reliable optical switches that are seamlessly integrated with the x25 data acquisition process. The sm041 can expand a four-channel interrogator to either eight or sixteen sub-channel operation, while maintaining the same wavelength repeatability and stability, as well as real-time, deterministic data acquisition.

4.5. Network Settings

The smx25 modules support a command set for all basic TCP-IP configuration settings, such as IP address, Netmask, and default gateway. Additionally, DHCP (Dynamic Host Protocol) is supported and can be configured via remote commands or configuration with a monitor and keyboard during boot up.

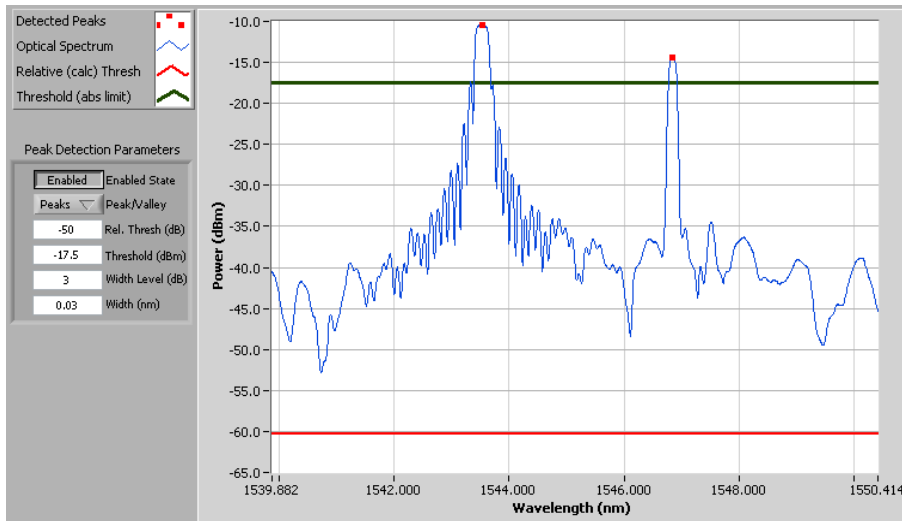
All x25 core Ethernet settings and configurations are managed automatically on the six25 instrument platform.

4.6. Optimizing Peak Detection Parameters

The basic x25 peak detection application functions in a relatively intuitive manner, using a unique threshold level (derived from both Rel. Thresh and Threshold parameters) and peak width (derived from both Width Level and Width parameters) for each channel. The algorithm will identify the central wavelength value and peak amplitude of any spectral peak feature that exceeds the specified threshold value in a symmetric manner with a spectral width greater than the specified peak width.

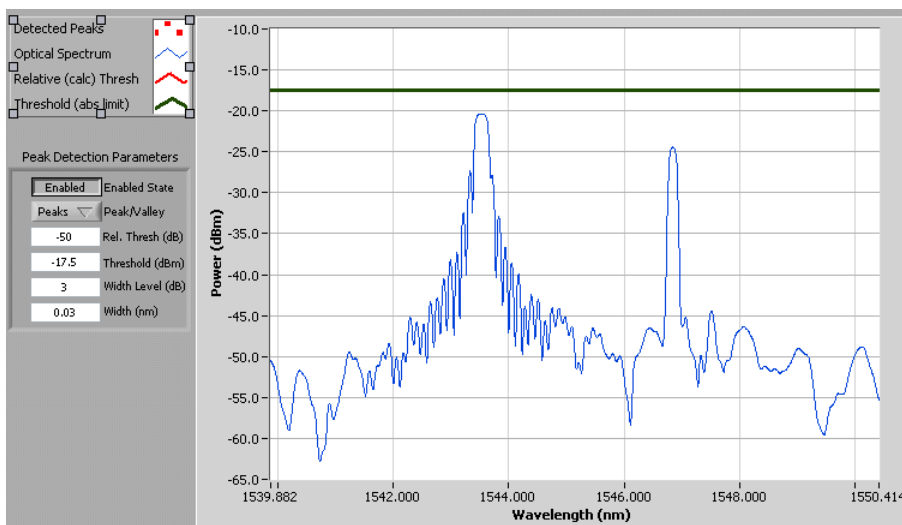
The width and threshold parameters can be optimized to selectively detect peaks of desired width and amplitude, while ignoring other spurious peak features. This section will show how the controls can be used to derive desired results.

The following sixteen images show how to optimize the peak detection parameters of the sm125 for different signal conditions and sensor orientations. A starting point configuration of the peak detection parameters is shown in Case A below.



Case A: Starting point for peak detection explanation. Peak detection is enabled, peaks selected (as opposed to valleys), Rel. Thresh = -50, Threshold = -17.5, Width Level = 3dB, and Width = 0.03nm.

In Case A above, simple peak detection is performed on two sensor signals of relatively low insertion loss and similar peak power values. Here, the Rel. Threshold parameter is set to -50 dB and the Threshold parameter is set to -17.5 dB. As the peak power for the channel is ~ -10 dBm, the effective threshold calculated by the relative threshold parameter would be at $-10 + (-50) = -60$ dBm. Since the Threshold parameter value of -17.5 is greater than the calculated value of -60 dBm, the value set by the Threshold parameter is used by the PD algorithm, and the relative threshold calculated value is effectively ignored. The width level parameter is set to 3 dB, meaning that the 3 dB bandwidth of each peak will be compared to the minimum width requirement of 0.03 nm. This value is lower than what practically makes sense for most FBG sensors, but it will help to illustrate some of the features of the peak detection algorithm.



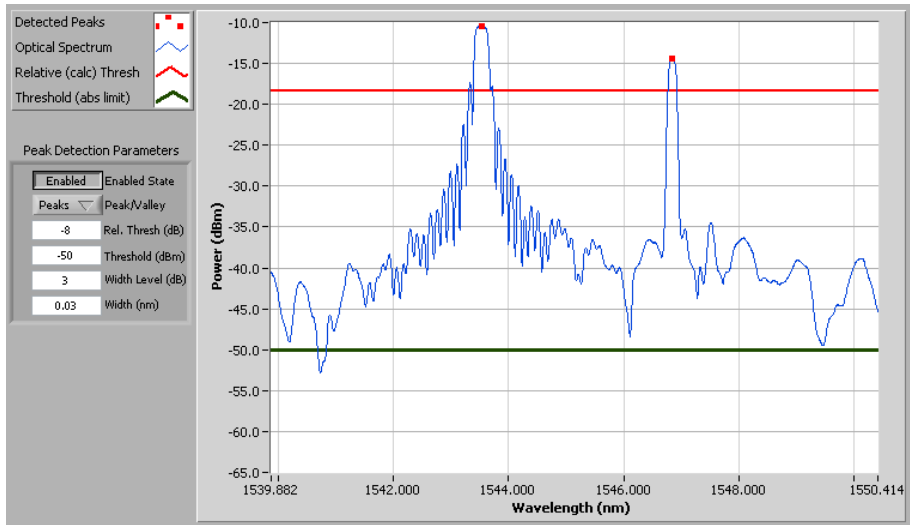
Case B: An additional 10 dB of insertion loss causes the spectrum to fall below the fixed Threshold value of -17.5 dBm.

Case B illustrates the limitation of using the Threshold parameter on its own, without utilizing the dynamic power tracking capabilities of the Rel. Thresh parameter. As the insertion loss of the two sensors was increased by 10 dB, both of the peaks fell below the Threshold value of -17.5 dBm, and neither of the FBG peaks were successfully identified as peaks.



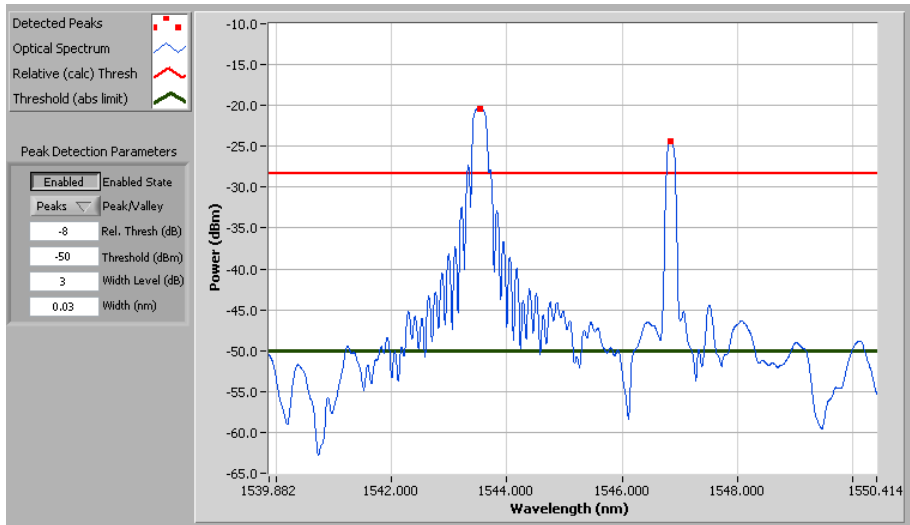
The solution to this problem is the use of the Rel. Threshold parameter. By setting the Rel. Thresh parameter to a higher number and the Threshold value to a lower number, like the relative values of -8 dBm and -50 dBm shown in case C, the peak detection algorithm is able to accommodate for channel-wide changes in insertion loss.

Case C shows the optical spectrum similar to that of Case A, but the peak detection algorithm is fed the parameter values described above. With a channel peak power of -10 dBm and a Rel. Thresh value of -8 dB, the calculated relative threshold is $-10 + (-8) = -18$ dBm. As -18 dBm is higher than the Threshold value of -50 dBm, the effective threshold used for the peak detection is -18 dBm and both peaks are effectively detected by the algorithm. The results can be seen in the figure labeled Case C.



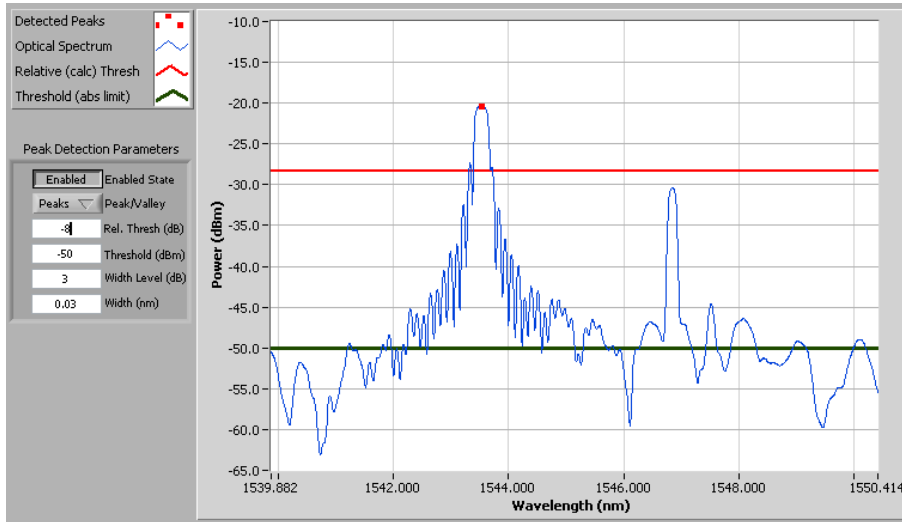
Case C: Relative thresholding used with a low-loss signal.

After the two threshold parameters have been set as described above, the peak detection algorithm is ready to accommodate for changes in channel insertion loss. Case D shows that, unlike Case B, an additional 10 dB of insertion loss will not cause the loss of any peak data from the peak detection algorithm.



Case D: Use of relative thresholding allows for changes in insertion loss without the need to change peak detection parameters.

It has been here shown that the use of relative thresholding can accommodate for channel-wide changes insertion loss. What if, though, there is a change in the relative power levels between two sensors on the same device channel? The next several cases examine how to use the peak detection parameters to contend with such a situation.

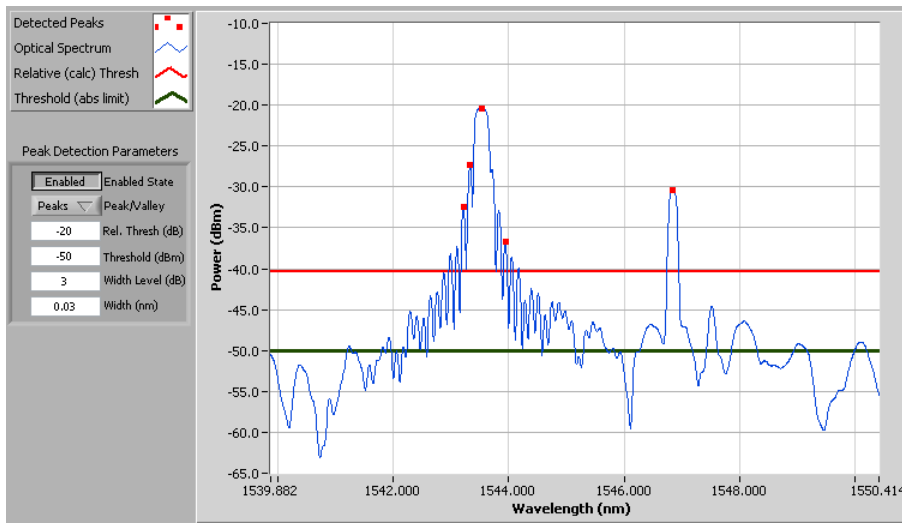


Case E: Insertion loss on a single FBG loss leads to a missing peak

The figure for Case E shows the same two sensors as the previous cases, but with an additional 10dB of insertion loss which has been locally applied to the 2nd of the two FBG peaks. It can be seen that with a Rel. Threshold value of -8 dB and a peak channel power of -20dBm, the resulting -28 dBm effective threshold is too high to effectively locate and identify both of the FBG peaks. In order to detect the 2nd peak, the Rel. Thresh value needs to be lowered to reduce the level of the effective threshold. Case F shows the effect of such a change.

Once the Rel. Thresh parameter has been changed to -20 dB, making the effective threshold -40 dBm, it can be seen in Case F that the 2nd FBG peak has been recovered by the peak detection algorithm. Unfortunately, this step places the threshold below some of the side modes of the 1st FBG peak, and those undesired side modes have been detected as signals as well.

This can be a common situation in optical sensor systems and is particularly problematic when non-apodized FBGs like FBG 1 in these figures are used.

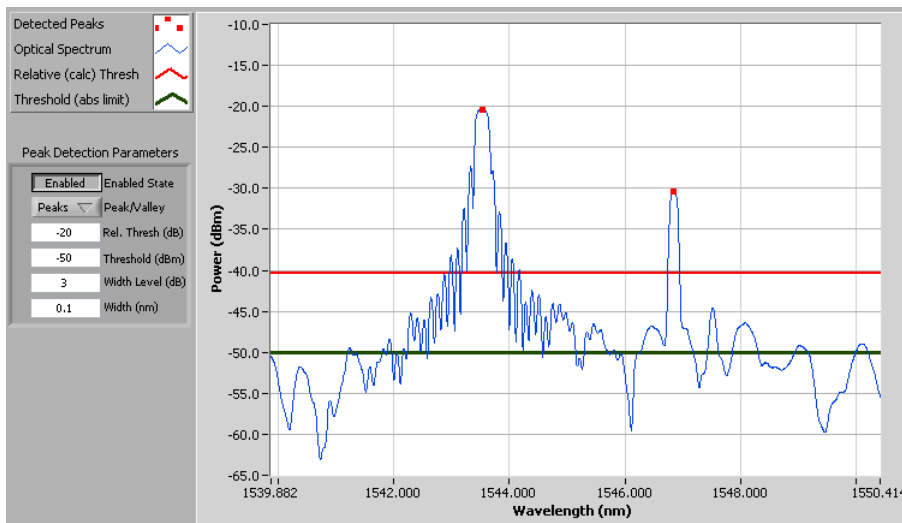


Case F: Reduction of the Rel. Thresh parameter allows for FBG 2 detection, but also includes several side modes from FBG 1.

In order to successfully peak detect the attenuated 2nd FBG while avoiding detection of any side modes, two approaches can be used with the parameters and controls of the x25 peak detection algorithm

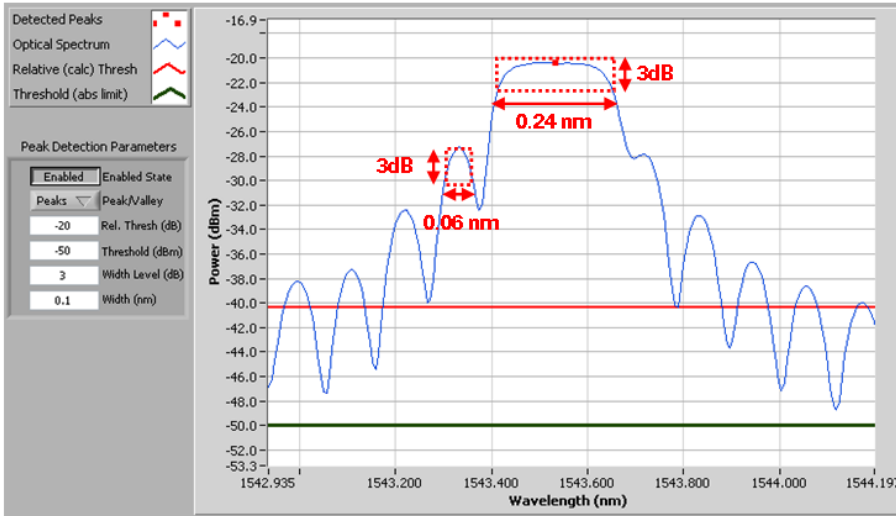
To effectively use the peak detection controls, it is useful to understand some of the typical spectral characteristics of FBG sensors. The first useful property is the relative width of FBG side modes as compared to the FBG fundamental mode. Typically, the lowest order or fundamental mode of the FBG is not only the strongest signal, but also tends to be the widest. Thus, when setting width requirements for the sensors, information about sensor and side mode bandwidths can be used to isolate the fundamental modes from the side modes. In addition to width differences, there are also difference in height and symmetry of the fundamental modes, as compared to the higher order side modes. As will be seen in the coming cases, side modes of higher optical power will often have an asymmetrical shape that can be exploited for exclusion by the peak detection algorithm.

Case F shows a collection of peak detection control settings that cause both FBGs and some side modes to be detected. Specifically, the effective threshold is at -40 dBm, and the peak requirement is set such that each prospective peak must have a 3 dB bandwidth of 0.03 nm or more. As we mentioned before, a width value of 0.03 nm is usually impractically low for FBG sensors and Case F shows exactly why. From the last paragraph and from visual inspection of the Case F figure, we can see that the 3 dB bandwidths of the two fundamental modes are much higher than the 3 dB band width of the detected side modes. Thus, the first technique used to isolate the fundamentals modes from the higher order modes is the use of the Width parameter. So, with the Threshold and Rel. Threshold parameters set to the same values as those of Case F, the width requirement for the 3 dB Width Level has been increased from 0.03 nm to 0.1 nm, as is seen in figure for Case G(1). By increasing the width requirement, none of the narrow side modes then qualify as peaks according to the detection algorithm, and only the two desired fundamental modes are reported as peaks.



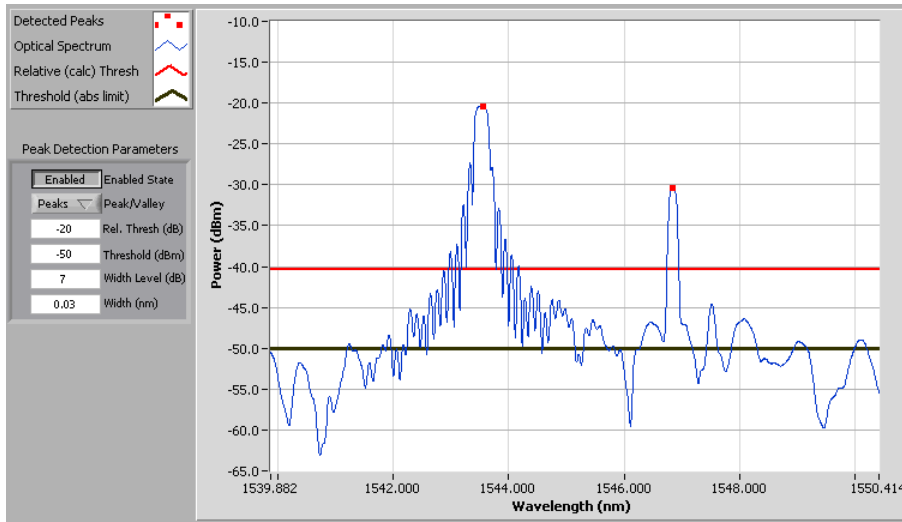
Case G(1). First method for isolating signals from side modes. Keep evaluation of peak width at 3 dB bandwidth and change the required bandwidth, such that narrower side modes do not meet the criteria, but fundamental FBG modes do.

The figure for Case G(2) offers a more in depth explanation of how the width parameter was used to discriminate side modes from the fundamental FBG mode. As is seen in the parameters summary to the left side of the plot, the Width Level parameter is still set to 3 dB. That means that the 3 dB bandwidth of each potential peak will be evaluated and compared against the Width requirement. For illustration purposes, two peaks on the first FBG have been evaluated for width at that isolation value, and the results graphically depicted in the figure. It is clear from the image that the 3 dB bandwidth of the fundamental mode is 0.24 nm, which does exceed the 0.1nm minimum Width requirement. As such, the peak passes through the peak detection algorithm, and the image is so noted with a red dot atop the peak. The side mode to the blue side of the fundamental mode has a 3 dB bandwidth of only 0.06 nm, which is less than the 0.1 nm requirement set. As such, the peak is rejected by the peak detection algorithm and is not adorned with a red dot.



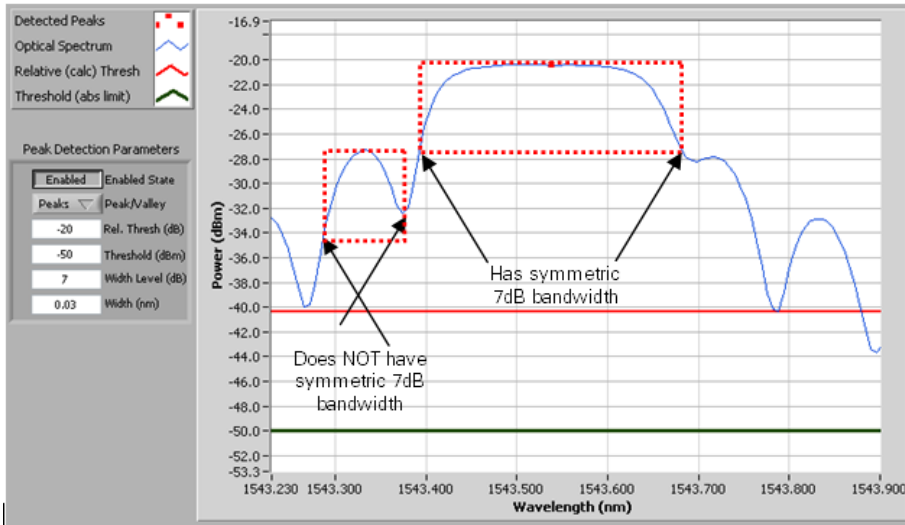
Case G(2). A close up reveals that the 3 dB bandwidth of the fundamental mode is greater than the required 0.1 nm, while the 3dB bandwidth of the side mode is not.

A second way to eliminate side mode detection in the peak algorithm is to exploit the frequent asymmetry of the higher order modes, relative to the fundamentals. The next two figures show how this method might be employed. To illustrate the method, the Width parameter has been set back to the previous value of 0.3 nm. This time, the Width Level has been set to 7 dB. The results can be seen in the Case H(1) figure that follows. In brief, the side modes have been eliminated, and only the two FBG fundamental modes have been reported by the peak detection algorithm.



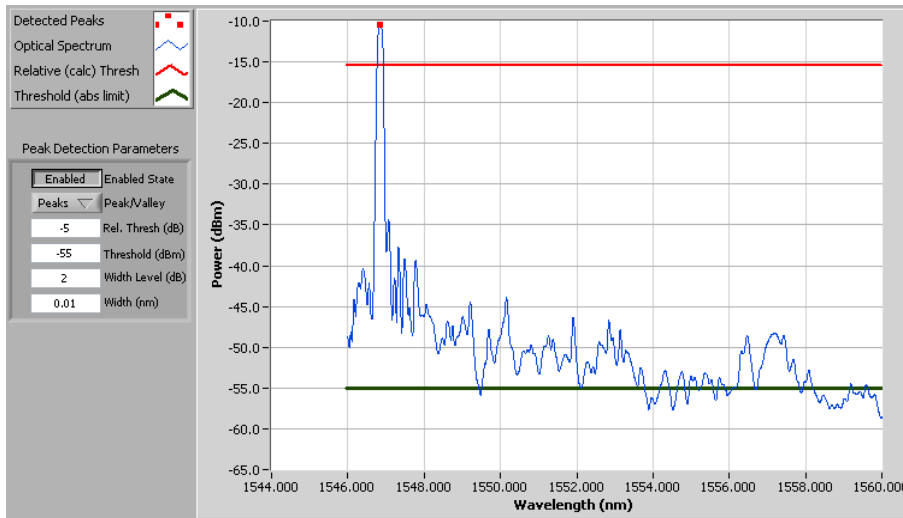
Case H(1). Keep original PD width parameter, but change the Width Level to take advantage of spectral symmetry of fundamental FBG modes

The figure labeled Case H(2) will offer a more in depth explanation of how the Width Level parameter was used to isolate the two fundamental peaks. Note the red dotted box drawn about the fundamental mode in figure H(2). At 7 dB attenuation beneath the top of the peak, the fundamental mode exhibits a symmetrical spectrum with both sides crossing the 7 dB attenuation point, allowing for the calculation of a 7 dB bandwidth. Compare that condition to the one about the first side mode left of the fundamental. Here, tracing down the curve to the left of the side mode peak allows for the identification of a 7 dB attenuation point. Try to look to the right side of the peak for a comparable 7 dB attenuation point, though, and one cannot be found. There is a point on the short wavelength side that meets the definition, but on the long wavelength side, the isolation from the next higher mode does not ever go past 5 dB or so. The higher order mode does not have a symmetric 7 dB bandwidth about the peak value, and as such, no 7 dB bandwidth can be computed for comparison against the Width requirement. As such, the peak is disqualified by the PD application and not recorded among the qualified sensor signals.



Case H(2). The fundamental mode has a symmetric 7 dB bandwidth, but the side lobe does not.

Thus far, there has been explanation of when and how to use the Rel. Thresh, Width, and Width Level parameters to optimize peak detection with the x25 default peak detection application. Next, there will be a brief explanation of how to make use of the Threshold parameter most effectively to set an absolute limit for the effective threshold.



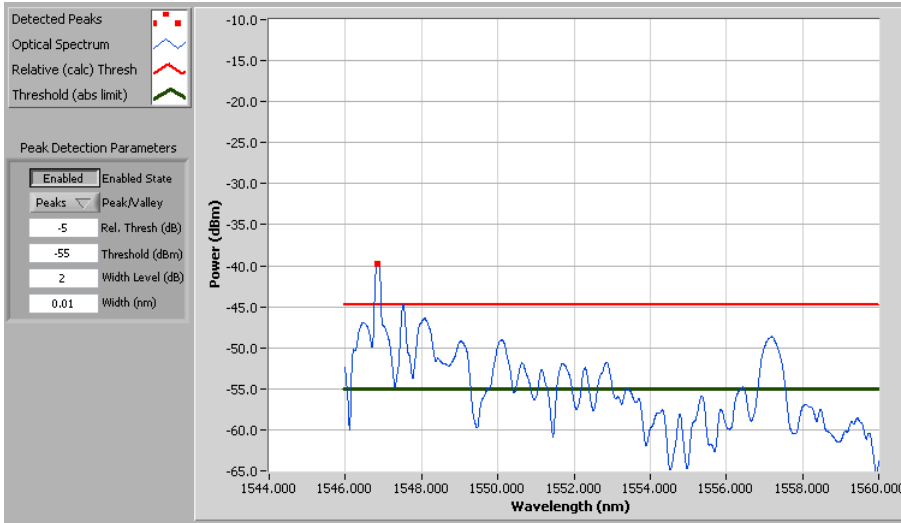
Case I: Single FBG, Rel. Thresh sets effective threshold far above noise floor.

Consider the case presented in the preceding figure. Here a single FBG is seen with low loss, high side mode suppression, and high signal to noise ratio. The parameters are set with a 2dB Width Level, a 0.01 nm width requirement (again, lower than might be practical, but useful for illustrative purposes), and a Rel. Thresh of -5 dB, setting the effective threshold 5 dB below the peak power of -10 dBm. With this low degree of insertion loss, the peak detection is clean and clear.

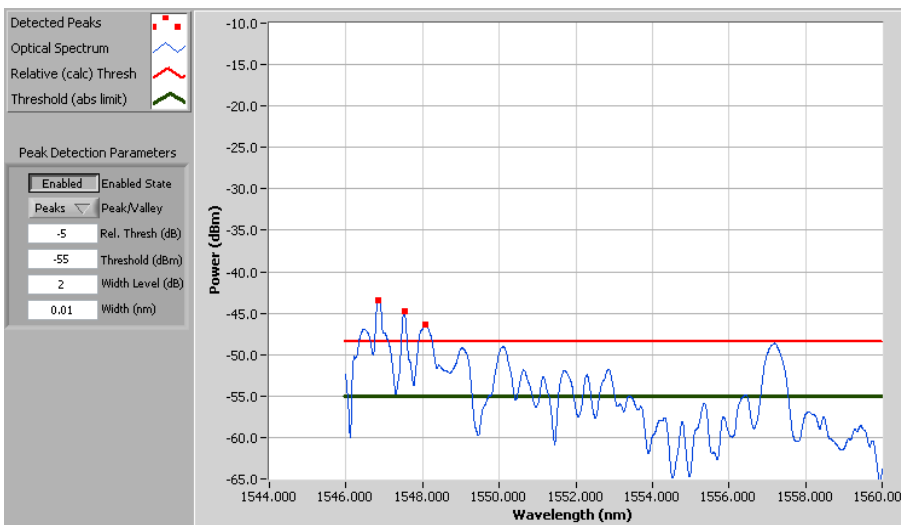
What might result from the peak detection algorithm with these settings if the insertion loss were to increase dramatically, like by 30 dB or so? The figure for Case J shows the result. With the current settings taking advantage of the Rel. Thresh value of -5 dB, the peak detection continues to track the signal without complication through an insertion loss change of 30 dB.

What then if the signal was to decrease again by a few more dB? As can be seen in Case K, when the signal is further reduced in intensity it begins to appear at a similar power level as the noise floor of this measurement. The noise floor may be dominated by back reflections from the fiber end face, or the aggregate signal from all other sensors on the fiber. Either way, it is not desirable to have the effective threshold continue to track the signal all the way to the noise floor, such that false signals are generated out of that noise.

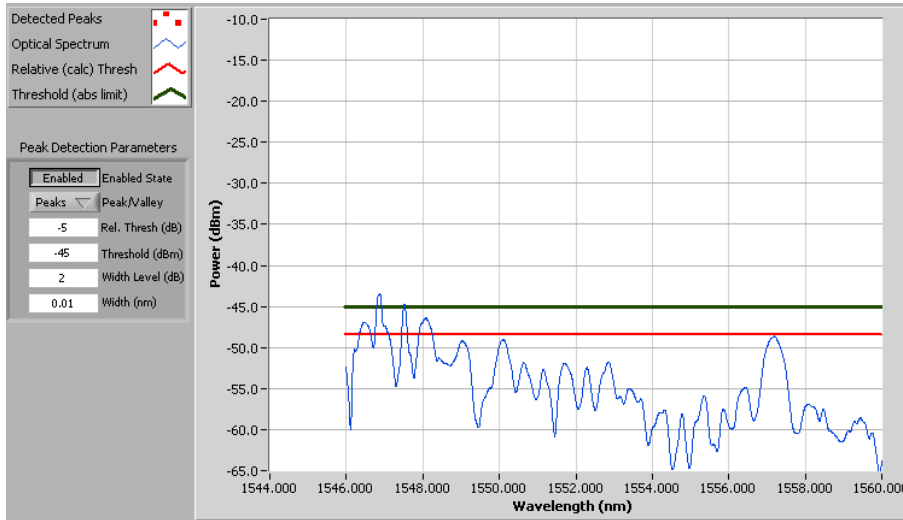
The solution to this problem is the judicious use of the Threshold parameter, represented in these plots by the black horizontal line. If the Threshold parameter is ever higher than the red line, dictated by the peak level and relative threshold, then it becomes the gating factor in determining the effective threshold. This effect can be seen in Case L. The Threshold parameter is set to -45 dBm, which is higher than the value determined by the peak and relative threshold, which is $44 \text{ dBm} + (-5 \text{ dB}) = -49 \text{ dBm}$. Thus, the effective threshold is set to -45 dBm, and no signal can meet the width and width level requirements.



Case J: By using the relative threshold parameter, 30 dB of signal attenuation is accommodated in the PD algorithm. This is a good thing. Effective threshold is still dictated by the peak level and relative threshold, as they are above the limiting “threshold” parameter.

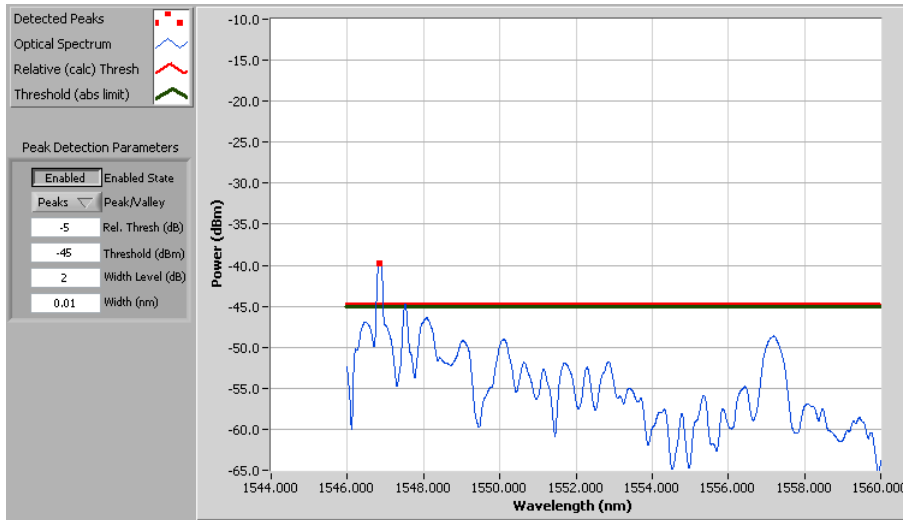


Case K: With several more dB of attenuation and still using relative threshold, the PD algorithm begins to get false positives, by identifying noise structure as valid peaks. This is why the absolute “threshold” parameter is included. It should be used to prevent massive attenuations from resulting in false positive numbers.

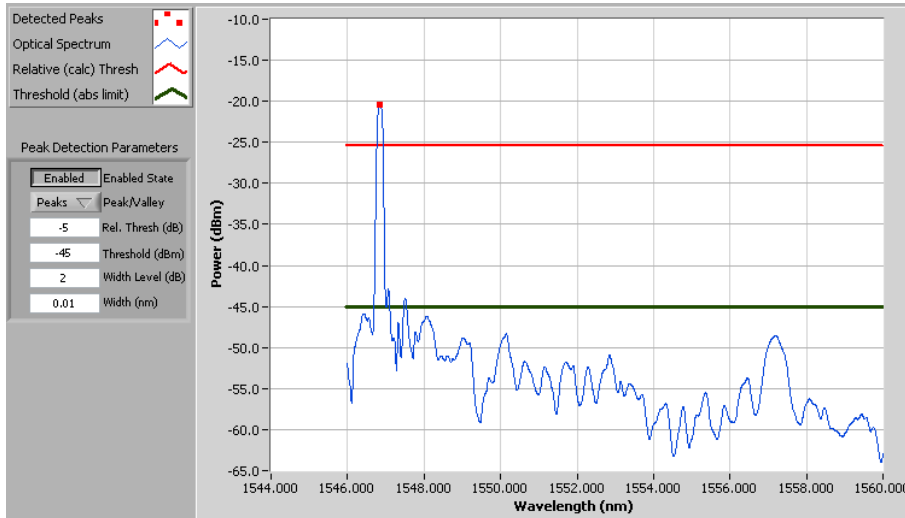


Case L: By increasing the limiting “Threshold” parameter, we prevent noise floor feature from being falsely identified as valid peak structure.

Cases M and N show how these new settings, designed to prevent false positive signals near the noise floor, continue to function well as the insertion loss on the FBG signal is reduced.



Case M: With “Threshold” set at -45, attenuation of the FBG is decreased, and the relative calculated threshold increases above the “threshold” limit...the higher of the two lines is applied as the effective threshold.



Case N: With attenuation returning towards zero, the calculated threshold continues to track 5dB below the peak amplitude. Since its value of 25dBm is greater than the limiting “threshold” values of -55dBm, the calculated relative threshold continues to function as the effective threshold, used in the PD. The updated “threshold” value of -45dBm remains to protection against false positive findings in the event of a future increases in insertion loss.

4.7. “x41” Multiplexer Technology

Additional measurement channels can be conveniently and economically added to an x25 core through the use of x41 Multiplexer Technology. The switch-type sm041 multiplexer is based upon the latest generation of solid state optical switches that redirect the optical path without moving parts. These switches feature low insertion loss, fast response time, high extinction ratio, and extremely high reliability and repeatability. It is designed to meet the most demanding switching requirements of continuous operation without wear-out, longevity without fail, and live operation under vibration/shock.

Optical multiplexing can be achieved through an external multiplexer module like the sm041, or through internal multiplexing, as is the case for the sm225.

The sm041 module does not have a power switch of its own. Rather, it is powered on when connected to a live sm125 host interrogator. Power can be verified on the sm041 by a red LED on the lower right side of the front panel.

The sm041 module has four optical inputs and 8 or 16 optical outputs, depending upon model. The input connectors are aligned vertically on the right side of the module, from 1 to 4, from top to bottom. The output connectors are arranged in 2 horizontal groupings of 8, with Multiplexed Channels numbering 1.1, 2.1, 3.1, 4.1, 1.2, 2.2, 3.2 and 4.2 on the top row, and 1.3, 2.3, 3.3, 4.3, 1.4, 2.4, 3.4 and 4.4 on the bottom row, in the case of a 16-channel unit.

Each of the 4 sm041 optical inputs should be connected to the corresponding detector channel on the host sm1XX interrogator. In other words, the sm1XX Physical Detector Channel 1 would connect to the sm041 Input Channel 1 with an APC to APC jumper. Sm1XX Physical Detector Channel 2 would connect to the sm041 Input Channel 2 with an APC to APC jumper, and so on through channels 3 and 4.

The sm041 Channel Expansion Module multiplexes either 2 or 4 channels from the module to a single channel on the host Sensing Interrogator Instrument. As such, each “Multiplexed Channel” in the two or four channel group is ported to the same “Physical Channel” on the host instrument. The Multiplexed Channels are grouped to Physical Channel in this manner:

NOTE: Legacy users may prefer the previous “all integer” standard for multiplexer notation. Please see the appendix entitled “Using Legacy Notation for MOI Multiplexers” in section 12.1 for details.

For an 8 sub-channel expansion module, the multiplexing is as follows:

Expansion Module Sub-Channels x25 or x30 Interrogator Core



1.1, 1.2	1
2.1, 2.2	2
3.1, 3.2	3
4.1, 4.2	4

For a 16 sub-channel expansion module, the multiplexing is as follows:

Expansion Module Sub-Channels	x25 or x30 Interrogator Core
1.1, 1.2, 1.3, 1.4	1
2.1, 2.2, 2.3, 2.4	2
3.1, 3.2, 3.3, 3.4	3
4.1, 4.2, 4.3, 4.4	4

4.7.1. Remote Control Interface

Control of the sm041 module is facilitated by the host x25. Once the host module is configured to interact with the sm041, no additional control of the module is necessary. All multiplexed data generated by the sm041 and host instrument pair is available exclusively through the remote Ethernet utility of the host instrument.

4.8. “x55” Swept Laser Core Technology

4.8.1. Sweep Rates and Acquisition Speeds

The x55 swept laser core has been designed to operate at a number of sweep speeds. Individual performance parameters may vary as a function of sweep rate. Details and specifications for the various models can be found at www.micronoptics.com.

4.8.2. Basic Peak Data Acquisition

The primary function of the x55 core is to rapidly and repeatedly make center wavelength measurements on 0.25 nm FBGs over an array of input conditions and varying sensor signal losses, without the need to manage gain settings or peak detection parameters. As such, the bulk of the data transfer from the module is comprised of wavelength measurement data.

4.8.2.1. Peak Wavelength Data

Wavelength data from the x55 core can be queried for either a single channel or multiple channels.

4.8.2.2. Command/Reply and Streaming Data Transfers

There are two primary modes of acquiring wavelength data from the x55 core. Most commands are implemented via a “command and response” method, where a command requesting data is sent from the client PC to the module instigating a data transfer from the module to the client. Additionally, though, there is a data streaming mode implemented that reduces the total communications overhead, easing the transmission load of large measures of data from the module. Wavelength data can either be “polled” or “streamed” to best fit the use case.

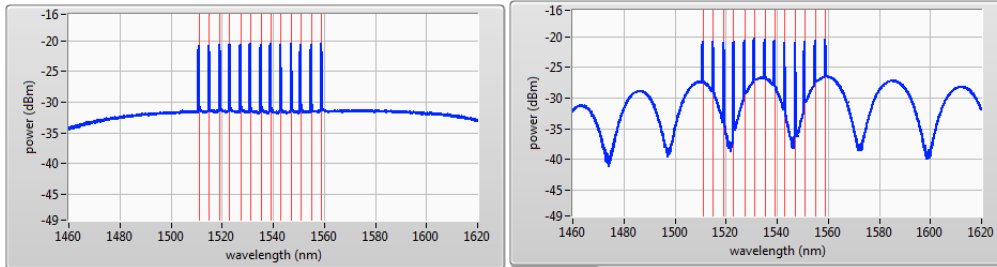
4.8.3. Full Spectrum/Spectral Diagnostic View

In addition to making peak wavelength measurements, the x55 interrogator also provides amplitude information in the form of Full Spectrum Measurements. The Full Spectrum measurement transfers a complete, contiguous acquisition of a single laser spectral sweep, and is actually the exact data that is fed into the internal peak detection algorithm. In contrast to the continuous 1kHz peak data, full spectrum is reduced down to a 10 Hz rate for ease of data transfer, plotting and storage. The Full Spectrum Measurement capability can provide a wealth of information about the health and operational condition of a sensor measurement system, as well as assist in making optimal choices with regard to sensor parameters and placement.

4.8.3.1. Diagnosing Connector Issues

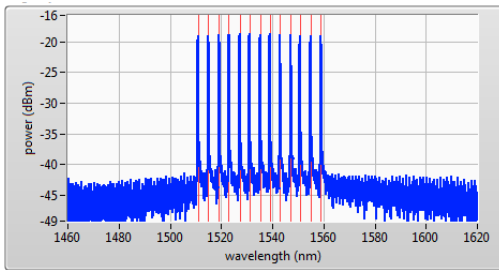


The next two figures show example situations in which the Full Spectrum Measurement of the x55 interrogators can be used to avoid potential data acquisition issues. The blue trace in the next figures shows a series of FBG sensor reflection profiles riding atop two broad spectrum noise sources: a uniform back reflection on the left, and a Fabry-Pérot etalon on the right, both of which can be the result of poor, reflective PC connections. As is seen in the figures below, the signal level from the unwanted reflection spectra degrades the overall loss budget of the system and should be corrected.



Issues like back reflections and poor connections lead to degraded optical signal to noise ratio and should be corrected.

With the insight granted by the Full Spectrum Measurement as to the nature of the issue, the obvious best choice is to locate and correct the poor optical connection, restoring the desired low, flat noise floor of an ideal setup.

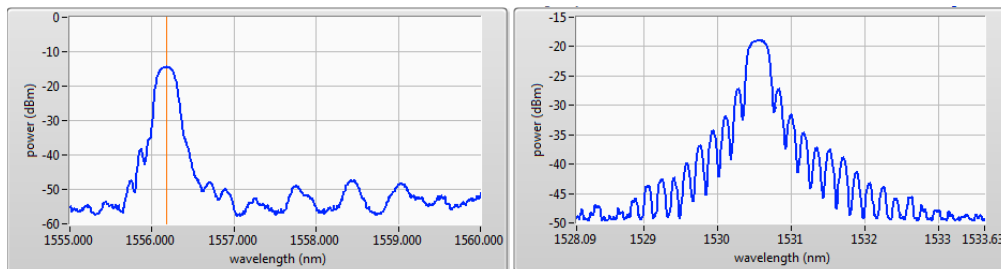


With sensor connector issues addressed, sensor signal-to-noise ratio is returned to optimal.

In addition to diagnosing connector issues, the Full Spectrum measurement can be used to measure the quality and features of FBGs or other optical sensors.

4.8.3.2. Measuring FBG Profile Detail

The following two images show Full Spectrum measurements of two different FBG designs. With a high optical resolution, deep dynamic range, and high sampling resolution, the x55 interrogator core offers the user a great deal of insight with regarding to such a shape and health.



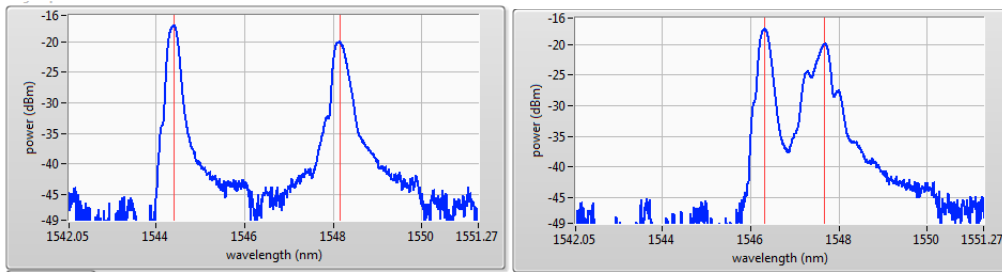
Full spectrum measurements show clear and distinct differences in the spectral responses of these two FBG sensors.

4.8.3.3. Detecting Sensor Debonding or Delamination

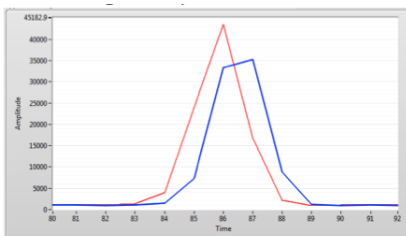
Far more than just distinguishing between grating designs, the full spectrum measurement can aid the user in early detection of sensor bonding or delamination issues. The following two images show the changes in sensor spectral responses with applied strain. The image on the right exhibits clear spectral shape distortion resulting from non-uniform strain transfer from the substrate into the FBG. By monitoring the Full Spectrum measurements from the x55 core, the



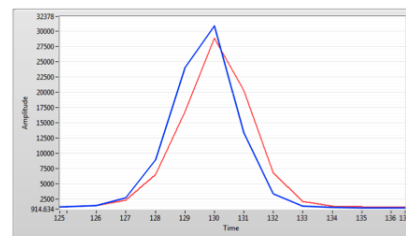
user can be certain that the high-speed peak detected data is being derived from the spectra of healthy and well-formed sensor signals.



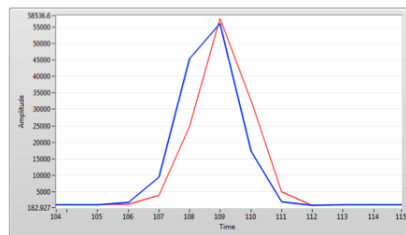
For point of comparison, consider the three FBGs as measured by an alternate optical interrogator technology. From looking, is it obvious which sensor has become partially detached from its substrate?



FBG 1



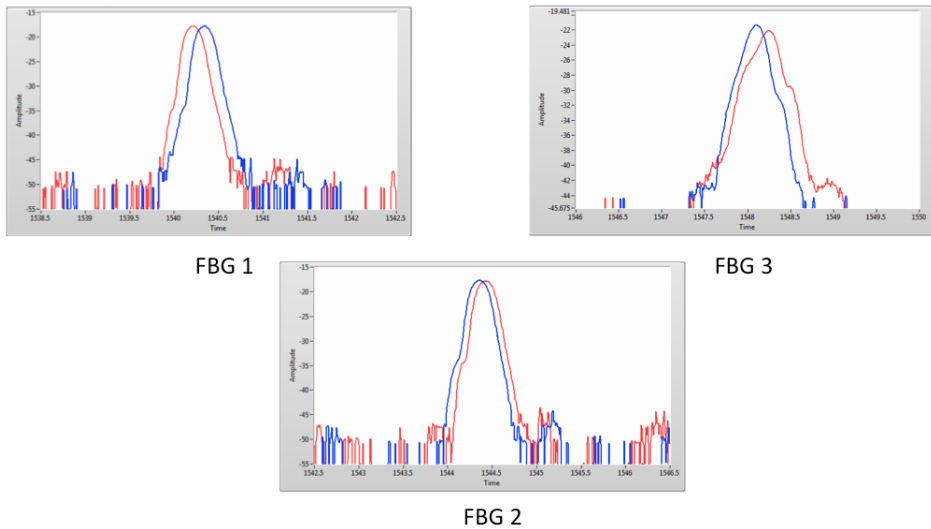
FBG 3



FBG 2



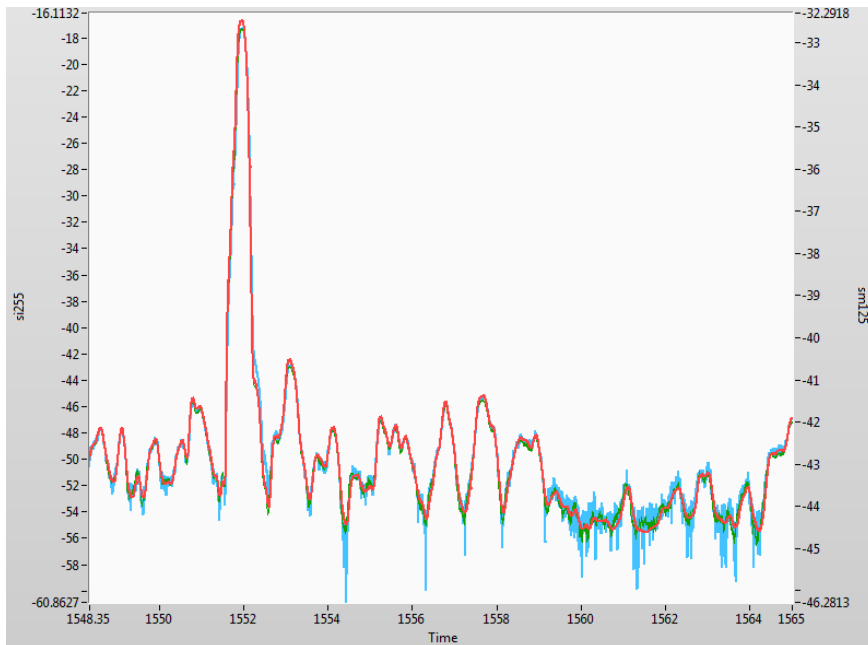
Now looking at the same FBGs with an x55 core, it is clear that the sensor that experienced delamination with applied strain is FBG 3. Without the Full Spectrum Measurement, how would the user know with certainty whether or not the sensor data was valid?



4.8.3.4. Full Spectrum with Averaging for Deep Dynamic Range

It is also worth noting that, although the noise floor of the Full Spectrum Measurement of the dynamic x55 core may carry more noise than some high-end static interrogators, the rapid acquisition rates of full spectrum allow for averaging of those signals to yield very useful, deep dynamic range Full Spectrum Measurements, comparable to those of the best static-only interrogators.

The following trace shows a spectral measurement of an apodized FBG with a Micron Optics 2 Hz sm125 (red trace) and a Micron Optics si255 (green and blue traces). Note that for data delivered at rates of 10 Hz to the user (green trace) the differences in measurement capability are quite small, affirming excellent full spectrum dynamic range for the si255 interrogator.



4.8.4. Internal Peak Detection Capability

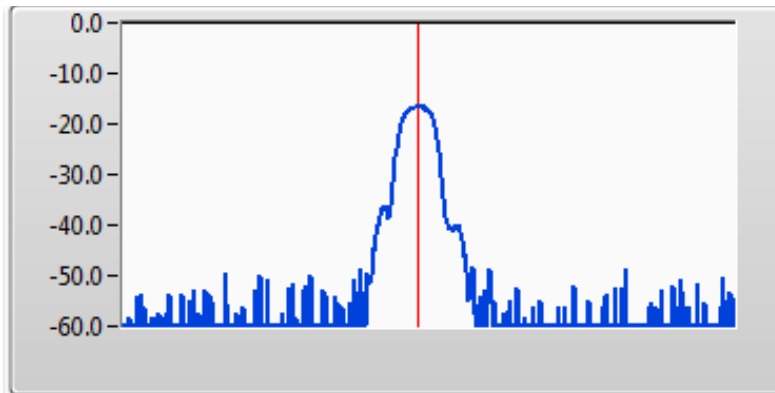
With the full optical spectrum signal available from the sensors, the x55 units offer great flexibility in the area of peak detection and data interpretation. Unlike any other optical spectrum interrogator on the market, dynamic peak data is actually calculated from full spectrum data collected on the si255 HYPERION Interrogator on every high-speed scan.

The x55 interrogator is most frequently used to measure the reflection spectrum of Fiber Bragg Grating and Fabry-Pérot sensors. As such, MOI has developed a series of flexible, yet robust internal peak detection algorithm for use with the x55 and those types of sensors. This algorithm is implemented within the fabric of the on-board FPGA in the x55 core to derive accurate, robust, repeatable FBG peak location data on each 1ms set of 16 full spectrum traces captured by the instrument.

The internal peak detection capabilities of the x55 core can be adapted to measure any of a large number of optical sensor types. Since the interrogator collects 1,000s of full spectral data points for each channel at each 1 kHz sweep, the onboard FPGA resources can be programmed to run any of a number of peak/null detection algorithms. The default peak detection algorithm works as follows:

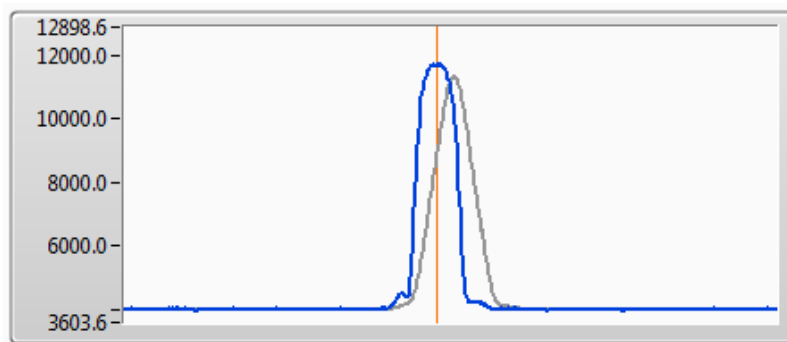
4.8.4.1. Programmable Hardware Peak Detection Algorithm – High Level Theory of Operation

On each 1 ms sweep of the x55 laser, a full optical spectral trace is collected for each of the up to 16 parallel channels resulting in a trace as depicted in the image below:



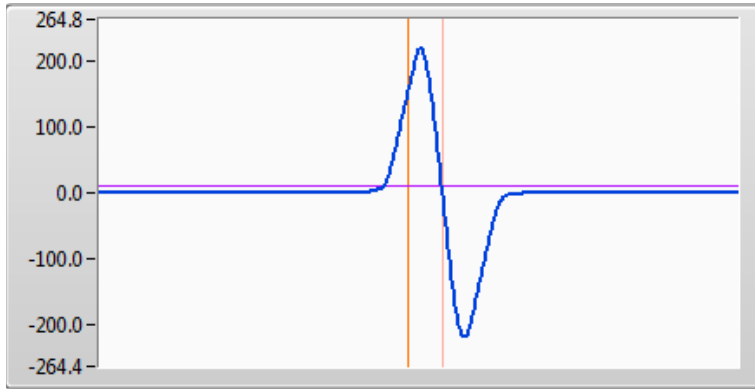
Direct output of Optical Receiver, as digitized by x55 ADCs. “Raw” signal, power calibrated for visualization.

The “raw” digitized spectrum of the sensor is then digitally filtered, eliminating unwanted high frequency sensor spectrum and/or system noise and minimizing the influence of unwanted side modes.



“Raw” signal, not scaled for power calibration (blue). Low pass “Box Car Filter” output of the first DSP block (gray).

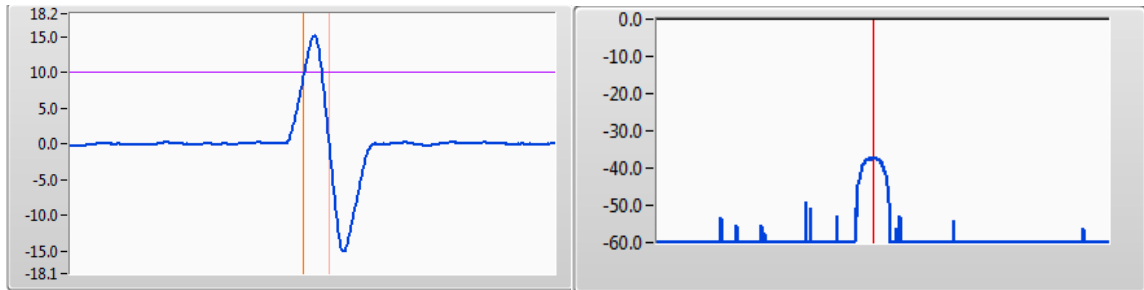
Once filtered, the signal is passed through a Differential Delay block, which serves to convert the peak amplitude of the filtered FBG signal into a “zero crossing” which will be used for determining the signal peak location.



Differential Delay, where the zero crossing serves as an indicator of the peak location.

A delay from the two DSP blocks of exactly the number of box car averages and differential delay counts is imparted to the differentiated signal and can thus be precisely adjusted to determine the exact spectral location of the peak/null of the optical signal.

An internal threshold (no user setting required) ensures that only zero crossings that exceed the instrument noise floor are considered to be valid peak signals. This threshold is set at the factory and does not require any user intervention under normal operating circumstances.



The Internal noise threshold ensures that only valid peaks are identified as such, even under low SNR conditions.

For standard FBG and FP sensors operating under normal conditions, no user optimization of peak detection parameters is required.

4.8.4.2. MOI Fabry-Pérot Peak Detection:

Micron Optics has developed a line of Fabry-Pérot sensors that are also x55 core compatible. Ultimately, detection and conversion of the Fabry-Pérot spectrum into calibrated engineering units will be handled completely by the x55 interrogator firmware. However, in the near term, the standard FBG peak (null) detection algorithm can be set with FP optimized parameters to ensure robust null detection of these unique sensors. Additional Fabry-Pérot order tracking and parameter conversions are facilitated by external software modules, available for both LabVIEW and Python in the HYPERION API distribution.

4.8.4.3. Other Custom Peak Detection Development and Deployment

To date, x55 peak detection algorithms have been developed and validated for 0.25 nm FBGs, hydrogen cyanide and acetylene gas cell absorption lines, and Fabry-Pérot sensor types. However, at its core, the x55 interrogator is truly optical sensor technology agnostic. If a sensor can be accurately profiled in the spectral domain by the interrogator, then an algorithm can likely be developed and implemented in DSP blocks within the FPGA to deliver optimally detected high speed data from that sensor type.

It is suggested that the Full Spectrum Measurement of the x55 core can supply complete data sets to a prospective volume application client for algorithm development. By evaluating the fully calibrated power and wavelength traces



delivered by the x55 core on a custom sensor type, the user can develop and validate a peak detection algorithm quickly and flexibly in software, which can then be implemented into the hardware in partnership with Micron Optics.

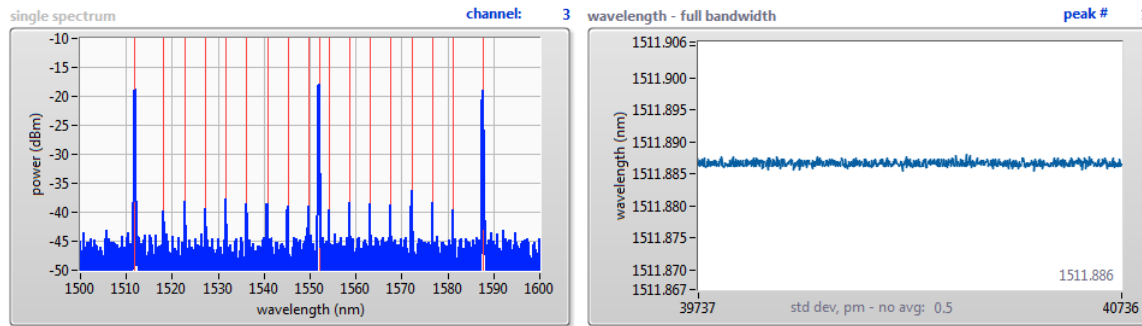
4.8.5. Parallel Multiplexing

The x55 units supports many tens of sensors over a wide 160nm wavelength range on each of 16 simultaneous, parallel measurement channels. In contrast to other interrogator technologies that required switch or coupler multiplexing to achieve higher fiber counts, the x55 measures all 16 channels of 160 nm spectrum truly simultaneously at a rate of 1 kHz. Assuming a nominal sensor spacing of 4 nm, the x55 interrogator can support 640,000 sensor measurements per second!

In addition to the benefit of large simultaneous sensor counts, the true parallel nature of the x55 core eases the user burden of power budget management. Consider a case where 4 fibers may be optically multiplexed into a single receiver channel. Should the receiver channel have a functional dynamic range of ~10 dB (as do many interrogators with linear receiver electronics), all sensors from all 4 fibers may need to be power matched within that limited loss budget. Experience shows that maintaining 13 dB over long run of sensors, such as that supported by a 160 nm wavelength range can be difficult. Compounded with maintaining uniform loss profiles over 4 switch-multiplexed channels, the practical field challenge of multi-fiber power budgeting can really limit the theoretical promise of large optical sensor networks.

For the x55 platform, the combination of independent, parallel channel detections with continuous, “no-gain-settings-required” detection for each channel allows users to fully realize the potential of large scale sensor network deployments.

The following image shows an array of sensors with large sensor-to-sensor variations in returned optical power. Unlike other dynamic interrogators with linear receiver circuitry, the x55 core can simultaneously detect all of the sensors without any user changes in peak detection parameters. The user has this same range of measurement support on all of the 16 parallel, simultaneously detected channels.



5. “X25” PROGRAMMING INTERFACE

5.1. TCP-IP Remote Control Interface

5.1.1. Configuration (applies to smx25 modules)

Data transfer to and from the x25 core is through a 100 Mbit/S Ethernet port on the front of the unit. The unit can be connected to an existing network through a hub or it can be connected directly to a host PC using a crossover Ethernet cable. Both of these arrangements require that the smx25 unit and the host be on the same logical network. The network administrator can assure this by providing an appropriate static IP address and netmask for the smx25 unit and/or host PC. These values can be configured using the programming interface described in this section.

The x25 core supports bi-directional communication through a TCP/IP data socket on port 50000.

NOTE: The default IP address and netmask for the sm125/sm225 modules are 10.0.0.122 and 255.255.255.0, respectively.

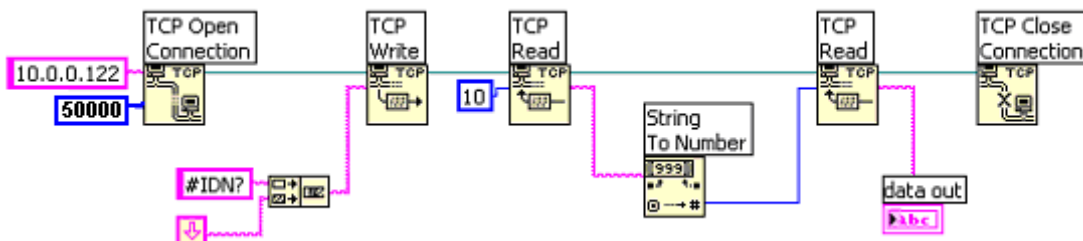
5.1.2. Communication Protocol

Commands and queries to the x25 core are in the form of ASCII strings that begin with a # character and end with a linefeed (ASCII char 10 [0x0a in hex, '\n' in C, vbCrLf in Visual Basic]). All characters sent to the x25 are internally buffered until a linefeed character is detected. Once the x25 receives the linefeed, it then interprets all the buffered data that preceded the linefeed. In response to the received data, the x25 will first return a 10-byte ASCII string. This 10-byte string is not a response to the submitted command or inquiry. Rather, the 10-byte string representing the number bytes contained in the response to the command. This is the number of bytes that the host is expected to read in addition to the first 10 bytes. The example below illustrates the point:

```
host:      #IDN?<LF>
sm125:    0000000032
sm125:    SM125, Rev. 0.00001, Micron Optics, Inc.
```

In this example, the host has issued the #IDN? command to an sm125. This command is used to retrieve the sm125 software version number. In response to this command, the sm125 first submits the 10-byte string “0000000032” indicating that the response to the #IDN? command is 32 bytes long. The sm125 then submits the 32-byte string “SM125, Rev. 0.00001, Micron Optics, Inc.” to the host. Thus, for every command written to the sm125 by the host, the host must make 2 reads. The first read is always 10 bytes long. The second read has a variable length dictated by the first. Note that this protocol is followed whether or not the data sent to the sm125 is a valid command. A LabVIEW implementation of the above example appears in the following figure. A Visual Basic implementation appears thereafter.

5.1.2.1. LabVIEW Implementation of x25 TCP-IP Protocol



5.1.2.2. Visual Basic Implementation of x25 TCP-IP Protocol

A Winsock control allows user to communicate with x25 core via Transmission Control Protocol (TCP)



1. To establish the Winsock control in Visual Basic, select Microsoft Winsock Control 6.0 from the Component dialog box of the Project Menu.
2. Set Protocol property of the Winsock as sckTCPProtocol. Set the RemoteHost and RemotePort properties of the Winsock as x25's IP address and 50000, respectively.
3. Give a name to the Winsock control (e.g. tcpClient).
4. Establish connection with x25 core with the command **Call tcpClient.Connect.**
5. Once the connection is established (check connection status with tcpClient.State) send command to x25 core to request data size.
6. **Call tcpClient.SendData("#IDN?" & vbCrLf)**
7. Wait until 10 bytes are received and get data size as string
 - a. **While tcpClient.BytesReceived < 10**
 - b. **DoEvents**
 - c. **Wend**
 - d. **Call tcpClient.GetData(strSize, vbString, 10)**
8. Convert data size format from string into integer.
9. **intSize = Cint(strSize)**
10. Then, read data for data size.

5.1.3. Data Retrieval

Sensor wavelength data can be retrieved from the x25 core using the #GET_DATA command. The response of a #GET_DATA command starts with a Main Header, which gives information about current version of the protocol, number of DUTs contained in the response and a sequence counter for the data. After the Main Header comes the data for all of the DUTs. The number of DUT channel entries on the table is a function of how many DUT channels are enabled. This data structure can carry from 1 – 4 channels for an x25 core and expands to as many as 16 channels for an sm125 with an sm041 Channel Expansion Module or a 16 channel sm225. The actual wavelength data for each DUT is preceded with a Sub Header, which gives the size of the Sub Header, the starting wavelength point of the data, the wavelength difference between each data point, number of data points and the DUT number.

5.1.4. Data Interpretation

#GET_DATA command

The data returned by the #GET_DATA command is a binary string that encodes a mixed set of data. First comes a 20-byte long Main Header, which contains five 4-byte fields. Each field represents a 32-bit unsigned integer with Least-Significant-Byte (LSB) first.

After the Main Header comes the wavelength data for every configured DUT. Every array of actual wavelength data for each channel is preceded by a 20-byte long Sub Header. The Sub Header consists of five 4-bytes fields. Each field represents a 32-bit unsigned integer with the LSB first. The "Min Wavelength" field and the "Wavelength Increment" field are both multiplied by a factor of 10,000 before they are transmitted from the unit. To convert the values back to floating point wavelength numbers, in nm, one must divide the corresponding 32-bit values by 10,000.

Each wavelength data point value is represented by a 16-bit signed integer, with the LSB first. The values are multiplied by a factor of 100 before they are transmitted from the unit. To convert the values back to floating point numbers the two decimals precision, one must divide each 16-bit integer with 100. The response structure is at most 128108 (20+4*20+4*2*16001) bytes long for a four-channel unit.

The byte offsets along with a brief description of each quantity are summarized in the following table. This table shows a portion of the structure to the first DUT. As additional DUTs are activated, additional sub headers and data blocks are concatenated to the response.

Byte Offset	Size (bytes)	Type	Description
0	4	u32	The size of the Main Header. In this version, it is set to 20.
4	4	u32	Protocol Version - Current version of the definition of this response structure.
8	4	u32	Number of DUTs this response contains



12	4	u32	Reserved for future use.
16	4	u32	Counter - A sequential counter for the data set that the instrument has processed internally.
20	4	u32	The size of the following Sub Header
24	4	u32	Minimum Wavelength - Wavelength of the first data point, multiplied by 10000. Default value is either 15100000 (1510.0 nm) or 15200000 (1520.0 nm).
28	4	u32	Wavelength Increment - Wavelength distance between each data point, multiplied by 10000. Default value is 50 (5 pm).
32	4	u32	nPoints - Number of data points for this DUT. Default value is either 10001 or 16001.
36	4	u32	DUT Number - The number of the DUT that this data belongs to (1 – 16).
40	2*nPoints	u16	Spectrum Data - The actual data for each wavelength sampled. Every data point value is multiplied by 100.
...	Additional sub headers and data blocks will be concatenated to this structure as more channels are activated.

Description of #GET_DATA response structure

#GET_PEAKS_AND_LEVELS command

The data returned by #GET_PEAKS_AND_LEVELS is a binary string that encodes a mixed set of data. The first 4 bytes (LSB first, MSB last) form a 32-bit unsigned integer that represents the seconds portion of the timestamp. The second 4 bytes (bytes 4-7) is also a 32-bit unsigned integer that represents the micro-seconds portion of the timestamp. Thus:

$$\begin{aligned} \text{seconds} &= \text{byte } 0 + \text{byte } 1*(2^8) + \text{byte } 2*(2^{16}) + \text{byte } 3*(2^{24}) \\ \text{microseconds} &= \text{byte } 4 + \text{byte } 5*(2^8) + \text{byte } 6*(2^{16}) + \text{byte } 7*(2^{24}) \end{aligned}$$

$$\text{timestamp} = \text{seconds} + \text{microseconds}/1000000.0$$

The data set serial number is also a 32-bit unsigned number with a byte offset of 8 (bytes 8-11). The serial_number = byte 8 + byte 9*(2^8) + byte 10*(2^16) + byte 11*(2^24)

The number of sensors detected on each of the channels is represented by an unsigned 16-bit integer with the following byte offsets:

channel 1:	byte offset 12
channel 2:	byte offset 14
channel 3:	byte offset 16
channel 4:	byte offset 18

For x25 models with less than 4 channels, the structure remains the same and empty channels will return data as zero values.

The wavelength data for channel 1 is represented with signed 32-bit numbers with a byte offset of 32. Subsequent byte offsets scale by the number of signals detected.

Amplitude data is represented with signed 16-bit numbers. The byte offset is a function of the number of detected signals on each channel. The details of the data structure are presented in the following table.

5.1.5. Data Interpretation for Multiplexed Channels

For sm125/sm041 combinations, or open operating systems sm225 16 channel configurations.

NOTE: Use of an optional sm041 Channel Expansion Module or internal multiplexer will facilitate as many as 16 optical sub-channels. The #GET_DATA data structure will expand for all available channels that have been activated. Additional channels enabled by the multiplexer can be activated using the #SET_DUT[Ch]_STATE command outline in the next section.

NOTE: Legacy users may prefer the previous “all integer” standard for multiplexer notation. Please see appendix 12.1 “Using Legacy Notation for MOI Multiplexers” in section 8 for details.



Byte Offset	Size (bytes)	Type	Description
0	4	u32	Timestamp (seconds) - Seconds portion of the timestamp. The timestamp, which is the absolute time in microseconds elapsed since 00:00:00 of January 1, 1970, must be reconstructed by combining the seconds and the microseconds portions as described below.
4	4	u32	Timestamp (ns) - Microseconds portion of the timestamp. See the note above on the seconds portion. The microseconds portion must be combined with the seconds portion to reconstruct the full timestamp as follows: <i>timestamp = sec portion + (msec portion)/1000000</i> The computed timestamp is a double-precision float.
8	4	u32	Serial number of the data set. Data points are sequentially numbered. In situations where the possibility of data loss or corruption exists during transfers, the serial number can be used to re-order and/or determine when data loss has occurred.
12	2	u16	Number of spectral peaks detected on the 1 st channel.
14	2	u16	Number of spectral peaks detected on the 2 nd channel.
16	2	u16	Number of spectral peaks detected on the 3 rd channel.
18	2	u16	Number of spectral peaks detected on the 4 th channel.
20	2	u16	Thermal Stability Flag. This value is non-zero when the system is in warm-up stage.
22	2	u16	MUX State - Communicates switch position when channel expansion module used. Values 0, 1 for 8 channels; 0,1,2,3 for 16 channel system.
24	8	U8	---
32	2*NS1	s32	Ch1 data
32+4*(NS1)	2*NS2	s32	Ch2 data
30+4*(NS1+NS2)	2*NS3	s32	Ch3 data
32+4*(NS1+NS2+NS3)	2*NS4	s32	Ch4 data
32+4*(NS1+NS2+NS3+NS4)	NS1	s16	Ch1 levels
32+4*(NS1+NS2+NS3+NS4) + 2*(NS1)	NS2	s16	Ch2 levels
32+4*(NS1+NS2+NS3+NS4) + 2*(NS1+NS2)	NS3	s16	Ch3 levels
32+4*(NS1+NS2+NS3+NS4) + 2*(NS1+NS2+NS3)	NS4	s16	Ch4 levels

Description of #GET_PEAKS_AND_LEVELS structure

5.2. User Command Set

Other available user commands are listed below. The use of these commands follows the same communication and data retrieval protocols outlined in section 5.1.2. Unless otherwise noted, all x25 commands are followed by a response from the unit, indicating either the successful completion of a command or a syntax error.

#HELP

Description:

The help command returns a complete alphabetized listing of x25 core user commands.

Syntax:

#HELP

Example:

#HELP



Response:
#GET_DHCP
#GET_DUTx_STATE (x >= 1, x <= 16)
...

#IDN?

Description: This command returns a string containing the name and revision of the x25 core.
Syntax: #IDN?
Example: #IDN?
Response: Micron Optics sm125 Optical Sensing Interrogator, Rev 2.0

#GET_SYSTEM_IMAGE_ID

Description: This command returns a string containing the revision identification of the firmware currently installed on x25 core.
Syntax: #GET_SYSTEM_IMAGE_ID
Example: #GET_SYSTEM_IMAGE_ID
Response: #SYSTEM_IMAGE_ID Image: Sm125, Rev 0.6, Created: Mon Sep 20 15:41:41 2004

#GET_PRODUCT_SN

Description: This command returns a string containing the serial number of the connected x25 core.
Syntax: #GET_PRODUCT_SN
Example: #GET_PRODUCT_SN
Response: #PRODUCT_SN SIA3XX

#SET_IP_ADDRESS

Description: This command changes the IP address of the x25 core. NOTE: The default IP address for the sm125 version 0.9 or higher is 10.0.0.122.
Syntax: #SET_IP_ADDRESS [IP Address]
Example: #SET_IP_ADDRESS 10.0.0.122
Response: #IP_ADDRESS 10.0.0.122

#SET_IP_NETMASK

Description: This command changes the IP Netmask of the x25 core. NOTE: The default IP Netmask for the sm125 and sm225 is 255.255.255.0
Syntax: #SET_IP_NETMASK [Netmask]

**Example:**

#SET_IP_NETMASK 255.255.255.0

Response:

#IP_NETMASK 255.255.255.0

#GET_IP_ADDRESS**Description:**

This command gets the currently configured IP address of the x25 core.

Syntax:

#GET_IP_ADDRESS

Example:

#GET_IP_ADDRESS

Response:

#IP_ADDRESS 10.0.0.122

#GET_IP_NETMASK**Description:**

This command gets the currently configured IP Netmask of the x25 core.

Syntax:

#GET_IP_NETMASK

Example:

#GET_IP_NETMASK

Response:

#IP_NETMASK 255.255.255.0

#GET_DUT[Ch]_STATE**Description:**

This command queries the x25 core for the acquisition state for the particular device under test (DUT) channel specified by the command name. As shown in this example, the x25 core would respond with the acquisition state for Channel 1. In order to query the acquisition state for sub-channel 4.2, the command would simply be modified to reflect the changed channel to #GET_DUT4.2_STATE.

The acquisition state signifies if a particular channel's data acquisition is active (parameter = 1) or inactive (parameter = 0).

Syntax:

#GET_DUT[Ch]_STATE

Where, Ch = Ch# (1-4) or sub-channel # (1.1 – 4.4)

Example:

#GET_DUT1_STATE

Response:

#DUT1_STATE 1

#SET_DUT[Ch]_STATE**Description:**

This command sets the acquisition state for the particular channel specified by the command name. As shown in this example, the x25 core would set the acquisition state for Channel 1. In order to set the acquisition state for sub-channel 4.2, the command would simply be modified to reflect the changed channel to #SET_DUT4.2_STATE.

Setting the acquisition state for a channel to 1 will cause the x25 core to collect data for that channel and include a corresponding entry in the data transfer command responses. Inclusion of the parameter 1 will cause the specified channel to become active; parameter 0 will cause the specified channel or sub-channel to become inactive.



NOTE: This command is used to trigger the use of an optional sm041 channel expansion unit, if connected to an sm125 module. If a sub-channel (or a channel above Channel 4) is activated, the sm125 will initiate switch drive commands to the attached sm041 module and port the data accordingly through the data transfer command responses.

Syntax:

```
#SET_DUT[Ch]_STATE [Val]
```

Where, Ch = Ch# (1-4) or sub-channel # (1.1 – 4.4) and Val = 0 for disabled and a 1 for enabled.

Example:

```
#SET_DUT1_STATE 1
```

Response:

```
#DUT1_STATE 1
```

#REBOOT_SYSTEM**Description:**

This command will cause the x25 core to reboot.

Syntax:

```
#REBOOT_SYSTEM
```

Example:

```
#REBOOT_SYSTEM
```

Response:

```
N/A
```

NOTE: The following peak detection algorithm commands can be accessed on port 55000 of a version 1.x x25 unit and on both ports 50000 and 55000 of a version 2.x x25 unit. Port 55000 commands support only the legacy integer channel notation, while port 50000 commands support both integer and decimal fraction sub-channel notation. See the appendix entitled “Using Legacy Notation for MOI Multiplexers” in the appendices section of this document for additional details.

#SET_PEAK_THRESHOLD_CH[Ch]**Description:**

This command sets the threshold level in dBm for the peak detection algorithm for the particular channel specified by the command name. As shown in this example, the channel whose level would be set by the command is Channel 1. In order to set the threshold level for sub-channel 4.2, the command would simply be modified to reflect the changed channel to #SET_PEAK_THRESHOLD_CH4.2.

Syntax:

```
#SET_PEAK_THRESHOLD_CH[Ch] [Val]
```

Where:

Ch = Ch# (1 – 4) or sub-channel # (1.1 – 4.4)

Val = the threshold level (in dBm) for use with the specified channel or sub-channel.

Example:

```
#SET_PEAK_THRESHOLD_CH1 -50
```

Response:

```
#PEAK_THRESHOLD_CH1 -50.00
```

#GET_PEAK_THRESHOLD_CH[Ch]**Description:**

This command gets the threshold level in dBm for the peak detection algorithm for the particular channel specified by the command name. As shown in this example, the channel whose level would be retrieved by the command is Channel 1. In order to get the threshold level for sub-channel 4.2, the command would simply be modified to reflect the changed channel to #GET_PEAK_THRESHOLD_CH4.2.

Syntax:

```
#GET_PEAK_THRESHOLD_CH[Ch]
```



Where:
Ch = Ch# (1 – 4) or sub-channel # (1.1 – 4.4)

Example:

```
#GET_PEAK_THRESHOLD_CH1
```

Response:

```
#PEAK_THRESHOLD_CH1 -50.00
```

#SET_REL_PEAK_THRESHOLD_CH[Ch]**Description:**

This command sets the relative threshold level in dBm for the peak detection algorithm for the particular channel specified by the command name. As shown in this example, the channel whose level would be set by the command is Channel 1. In order to set the relative threshold level for sub-channel 4.2, the command would simply be modified to reflect the changed channel to #SET_REL_PEAK_THRESHOLD_CH4.2.

Syntax:

```
#SET_REL_PEAK_THRESHOLD_CH[Ch] [Val]
```

Where:
Ch = Ch# (1 – 4) or sub-channel # (1.1 – 4.4)
Val = the relative threshold (in dB) for use with the specified channel

Example:

```
#SET_REL_PEAK_THRESHOLD_CH1 -6.00
```

Response:

```
#REL_PEAK_THRESHOLD_CH1 -6.00
```

#GET_REL_PEAK_THRESHOLD_CH[Ch]**Description:**

This command gets the relative threshold level in dBm for the peak detection algorithm for the particular channel specified by the command name. As shown in this example, the channel whose level would be retrieved by the command is Channel 1. In order to get the relative threshold level for sub-channel 4.2, the command would simply be modified to reflect the changed channel to #GET_REL_PEAK_THRESHOLD_CH4.2.

Syntax:

```
#GET_REL_PEAK_THRESHOLD[Ch]
```

Where:
Ch = Ch# (1 – 4) or sub-channel # (1.1 – 4.4)

Example:

```
#GET_REL_PEAK_THRESHOLD_CH1
```

Response:

```
#REL_PEAK_THRESHOLD_CH1 -6.00
```

#SET_PEAK_WIDTH_CH[Ch]**Description:**

This command sets the peak width in nanometers for the peak detection algorithm for the particular channel specified by the command name. As shown in this example, the channel whose width would be set by the command is Channel 1. In order to set the peak width for sub-channel 4.2, the command would simply be modified to reflect the changed channel to #SET_PEAK_WIDTH_CH4.2.

#Syntax:

```
#SET_PEAK_WIDTH_CH[Ch] [Val]
```

Where:
Ch = Ch# (1 – 4) or sub-channel # (1.1 – 4.4)
Val = the peak width in nm for use with the specified channel.

Example:

```
#SET_PEAK_WIDTH_CH1 0.25
```

Response:



```
#PEAK_WIDTH_CH1 0.25
```

#GET_PEAK_WIDTH_CH[Ch]

Description:

This command gets the peak width in nanometers for the peak detection algorithm for the particular channel specified by the command name. As shown in this example, the channel whose width would be retrieved by the command is Channel 1. In order to get the peak width for sub-channel 4.2, the command would simply be modified to reflect the changed channel to #GET_PEAK_WIDTH_CH4.2.

Syntax:

```
#GET_PEAK_WIDTH_CH[Ch]
```

Where:

Ch = Ch# (1 – 4) or sub-channel # (1.1 – 4.4)

Example:

```
#GET_PEAK_WIDTH_CH1
```

Response:

```
#PEAK_WIDTH_CH1 0.25
```

#SET_PEAK_WIDTH_LEVEL_CH[Ch]

Description:

This command sets the peak width level in dB for the peak detection algorithm for the particular channel specified by the command name. As shown in this example, the channel whose width level would be set by the command is Channel 1. In order to set the peak width level for sub-channel 4.2, the command would simply be modified to reflect the changed channel to #SET_PEAK_WIDTH_LEVEL_CH4.2.

Syntax:

```
#SET_PEAK_WIDTH_LEVEL_CH[Ch] [Val]
```

Where:

Ch = Ch# (1 – 4) or sub-channel # (1.1 – 4.4)

Val = the peak width level (in dB) for use with the specified channel.

Example:

```
#SET_PEAK_WIDTH_LEVEL_CH1 3.0
```

Response:

```
#PEAK_WIDTH_LEVEL_CH1 3.0
```

#GET_PEAK_WIDTH_LEVEL_CH[Ch]

Description:

This command gets the peak width level in dB for the peak detection algorithm for the particular channel specified by the command name. As shown in this example, the channel whose width level would be retrieved by the command is Channel 1. In order to get the peak width level for sub-channel 4.2, the command would simply be modified to reflect the changed channel to #GET_PEAK_WIDTH_LEVEL_CH4.2.

#Syntax:

```
#GET_PEAK_WIDTH_LEVEL_CH[Ch]
```

Where:

Ch = Ch# (1 – 4) or sub-channel # (1.1 – 4.4)

Example:

```
#GET_PEAK_WIDTH_LEVEL_CH1
```

Response:

```
#PEAK_WIDTH_LEVEL_CH1 3.0
```

5.2.1. LabVIEW TCP-IP Examples

NOTE: For the simplest and most feature rich interface to the modules, Micron Optics recommends the use of the MOI ENLIGHT Sensing Analysis Software, available as a free download for at www.micronoptics.com.

NOTE: The following examples were written for and reference the sm125 Field Module specifically, but are equally applicable to any Ethernet accessible x25 core. See section 5 for platform specific networking details.

5.2.1.1. Four Channel Complete Interface Example

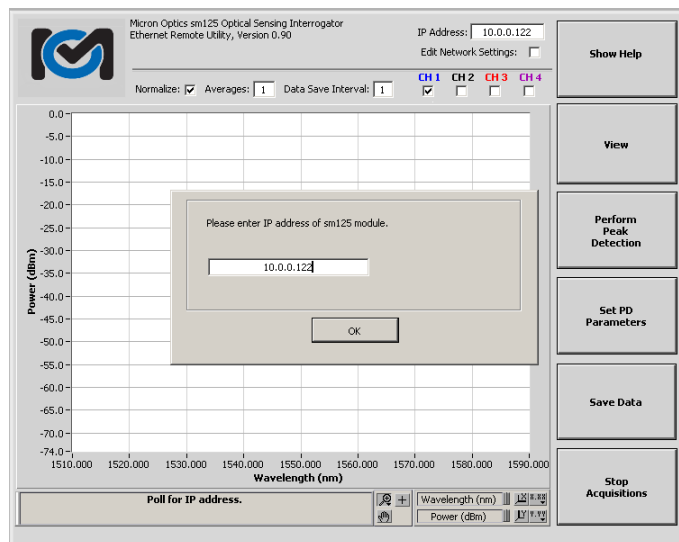
Included with the instrumentation support materials is a full four channel example program that uses the TCP-IP communication protocol. The example is written in National Instruments LabVIEW, and is available as an open-source LabVIEW library and as a stand-alone executable. The library files are compatible with the LabVIEW 7.1 Full Development System.

The included sm125 remote LabVIEW examples are compatible with Windows 98, NT, 2000, XP, Vista and Windows 7. An 800MHz Pentium III or better is recommended to run the utilities.

This LabVIEW Ethernet Remote Utility example, shown in the figures of this section, uses the command set and protocol as described in sections 4.1.2.1 – 4.1.2.6 of this document. The data can be retrieved, parsed and plotted at the full acquisition speed of the sm125 module.

i. Initialization and Connection

The following figure shows a screen shot of the example immediately after initialization. A pop-up window polls the user for the IP address of the sm125.



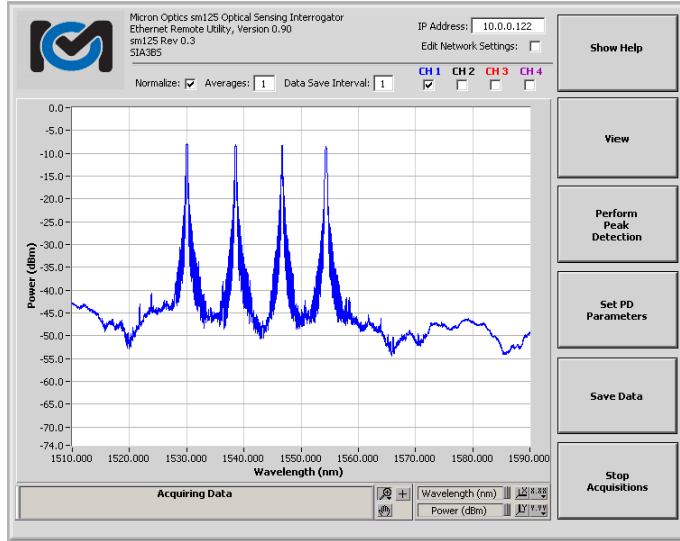
Screen Shot from included LabVIEW Ethernet Remote Utility example

Once the sm125 LabVIEW Ethernet Remote Utility example is connected and running, it can be used to set the IP address and Netmask settings of the sm125 unit. Beneath the IP Address control on the remote utility is a checkbox control labeled “Edit Network Settings.” Selecting this box will initiate a sequence to change those settings. The user will be asked to choose between wired and wireless settings, and then polled to enter the desired IP address and Netmask, with the current values offered as the default. When the values have been entered, the settings are automatically saved to the sm125 and will take effect upon sm125 reboot.

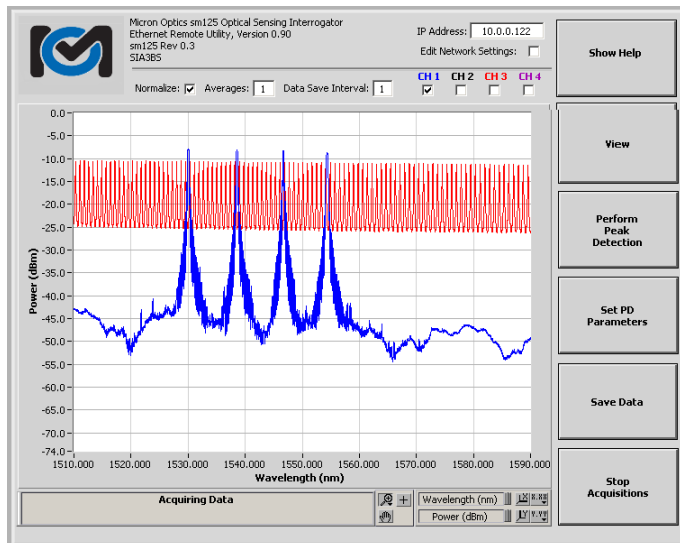
ii. Spectrum Graph Mode



The following figure shows a screen shot of the LabVIEW remote example in Spectrum Graph Mode with Channel 1 active. The array of checkboxes above the graph on the right allows the user to activate and deactivate each of the available sensor or device under test (DUT) channels, up to four. The color of the checkbox control labels corresponds to the display colors for each of the channels on the graph.



This next figure shows how these colors allow the user to visually discriminate between data from the multiple DUT channels.



The preceding figure shows the sm125 measuring a mixed collection of FBG and Fabry-Pérot Sensors. The FBG sensors can be seen on Channel 1 represented by the blue trace, and selected by the blue-labeled checkbox control “CH 1.” The FP sensor can be seen on Channel 3, represented by the red trace and selected by the red-labeled checkbox control “CH 3.”

- iii. Other front panel controls visible on the LabVIEW Ethernet Remote Utility example are:
 - i. Normalize



This checkbox control will toggle the power normalization feature. This feature normalizes the DUT traces to an internal monitor of the optical output power as a function of wavelength. This tool is intended to assist in flattening out the resultant optical spectrum of optical sensors such that peak detection is simplified.

ii. Peak Avgs.

This control will allow the user to set a number of peak detected values to be used in a boxcar average. Data will continue to be reported at the acquisition speed of the unit, but the peak detected values will reflect the average value over the number of averages set. This tool can be used to minimize the effects of system and peak detection noise to increase measurement repeatability on signals that are relatively stable for peak detection, but may change value relatively rapidly.

iii. Spectrum Avgs.

This control will allow the user to set a number of optical spectrum traces to be used in a boxcar average. Data will continue to be reported at the acquisition speed of the unit, but the spectrum will reflect the average optical signal over the number of averages set. This tool can be used to minimize the effects of system noise to increase measurement repeatability on signals that may be difficult to peak detect.

iv. Data Stream Interval

As will be explained later in this section, the LabVIEW Ethernet Remote Utility example can save data in multiple formats. For "Streaming Data", peak wavelength data will be continuously streamed to disk at a rate determined by the acquisition rate divided by the Data Stream Interval. For example, at an acquisition rate of 1 Hz and a Data Stream Interval of 10, data will be saved to disk at 1/10 Hz, or every 10s.

v. Show Help

This control activates the on-screen help feature, which will be covered in more detail in a later entry of this section.

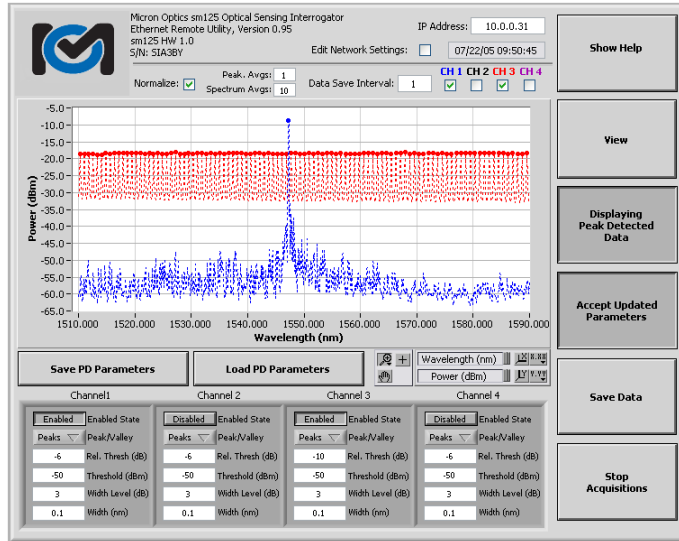
vi. Perform Peak Detection

This control activates the remote utility's peak detection algorithm, which is used to distill full spectral data into useful center wavelength values for sensor signals. The behavior of the peak detection algorithm will be set by the PD parameters, as explained in section 4.1.1.2 of this document.

vii. Set PD Parameters

The next two functional buttons are used to perform peak detection on the acquired full spectral data sets. The following figure shows a view of the LabVIEW Ethernet Remote Utility example with the Perform Peak Detection and Set PD parameters buttons activated. The "Perform Peak Detection" button activates the peak detection algorithms, which will operate based on the PD Parameters as defined by the functions described in the next section. When "Perform Peak Detection" is set to on, the spectral data plots will change from solid lines to dashed lines, and the peak detected values will be indicated by a solid round dot of identical color to the spectrum plot.

The "Set Peak Detect Parameters" button will open a window allowing the user to customize peak detection parameters for each of the available channels, up to four. When this feature is active, the graph display will be reduced in size to accommodate display of both the graph and the parameters control, as seen in this next figure.



sm125 Ethernet Remote Utility with Set PD parameters selected

viii. Save PD Parameters

Once peak detection parameters are set, they can be stored to a file location of the user's choice. The peak detection algorithm is saved as a flattened string data file that can be saved as a simple text file.

ix. Load PD Parameters

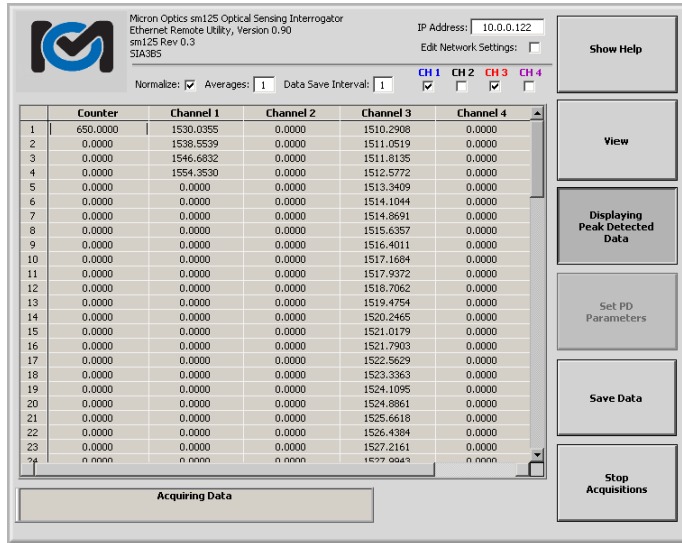
Once peak detection parameters have been saved, they can be loaded from the file. The LV utility will unflatten the string file and restore the parameters.

When desired PD parameters have been selected, toggling the "Accept Update Parameters" will return the utility to its ordinary functions.

In addition to Spectral Graph View, two other modes of data visualization are available in the four channel LabVIEW Ethernet Remote Utility example. These other modes can be accessed by pressing the "View" button, which can be found on the right side of the GUI, second button from the top. Clicking the "View" button repeatedly causes the GUI to toggle among the three available views. The next two sections explain these views.

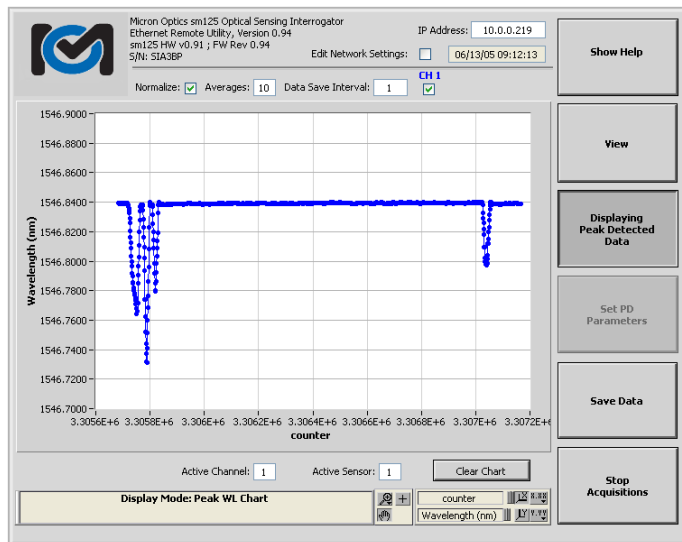
iv. Peak Table Mode

In Peak Table mode, the resultant values from the peak detection algorithm will be displayed on the table. The first column will contain a single value reflecting the counter, or relative timestamp, of the data set collected by the sm125. The next four columns will contain wavelengths that represent the peak values as derived by the PD algorithms for each of the present channels, up to four. The image below shows a view of the sm125 Ethernet Remote Utility in Peak Table display mode.



sm125 Ethernet Remote Utility in Peak Table display mode.

v. Peak Tracking Mode

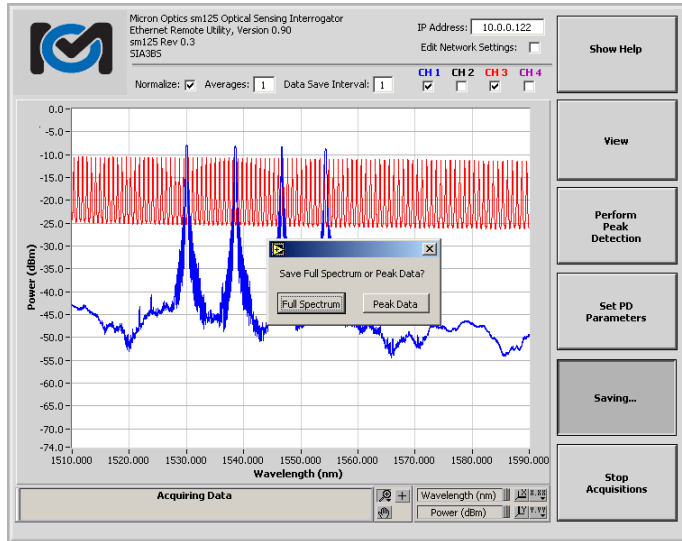


sm125 Ethernet Remote Utility in Peak Tracking Mode.

Peak tracking mode displays a single FBG value in a strip chart versus timestamp. This view is a convenient way to visually track changes in a sensor value. Controls for this mode include “Active Channel” and “Active Sensor”, which together select among the available channels and sensors to get the appropriate sensor value for display. The preceding figure shows the sm125 remote utility in Peak Tracking mode.

vi. Data Saving Features

The sm125 LabVIEW Ethernet Remote Utility example also provides a mechanism to save the spectral data collected by the unit to a file. Note on the figure below, a control button located on the right side labeled “Save Data.” By clicking on that button, the user will initiate a save data sequence. Immediately following the activation of the button, the user will be prompted to choose what type of data to save, either Full Spectrum or Peak Data. The following figure shows a screen shot of the utility at this phase of the Save Data sequence.



Ethernet Remote Utility screen shot during Data Save sequence.

Should the user choose "Full Spectrum", he will be prompted for a filename and path to which the data should be saved. Data is stored in following format: Each entry consists of five columns, delimited by tabs. Each column contains a number of elements equal to the number of spectral samples collected by the sm125, commonly 3201, 5001, 10001 or 16001. The first column represents the X-axis or Wavelength scale. Each of the next four columns represent the corresponding optical power array for each of the present channels 1-4, in that order. Any channels either not present or without data will be represented by a column of zeros. An example of the format for the saved data is seen in the table below. These might be representative of the first few lines of a system with Channels 1 and 3 active:

1520.000	-32.400	0.000	-2.980	0.000
1520.005	-32.390	0.000	-3.370	0.000
1520.010	-32.450	0.000	-3.650	0.000
1520.015	-32.410	0.000	-3.930	0.000
1520.020	-32.380	0.000	-4.170	0.000
1520.025	-32.360	0.000	-4.360	0.000
1520.030	-32.340	0.000	-4.570	0.000
1520.035	-32.350	0.000	-4.810	0.000
1520.040	-32.370	0.000	-5.140	0.000
1520.045	-32.460	0.000	-5.430	0.000
1520.050	-32.510	0.000	-5.550	0.000
1520.055	-32.420	0.000	-5.690	0.000
1520.060	-32.450	0.000	-5.950	0.000
1520.065	-32.370	0.000	-6.190	0.000

Example of Full Spectrum data file format from LabVIEW Ethernet Remote Utility example

Should the user choose "Peak Data", he will be prompted to choose between "Single Save" and "Streaming Data." Single save will save the peak detected data one time to disk, after a selection of filename and path are selected by the user. If streaming data is selected, the PD data will be saved to the selected file at a rate determined by the acquisition rate divided by the Data Stream Interval (see earlier explanation of the Data Stream Interval control for details).

The user will then be prompted whether or not to "Reference time base locally for these measurements." The time base included in the datasets reflects the total number of acquisitions of the sm125 unit since its last boot up. For a 1Hz unit, the time base can then be interpreted as the number of seconds that the unit has been on. For a 10 Hz unit, the counts reflect 1/10th second acquisitions. The "reference time base locally for these measurements" will cause the saved dataset to start the time base over again at zero.

The peak data will be stored according to the following format:

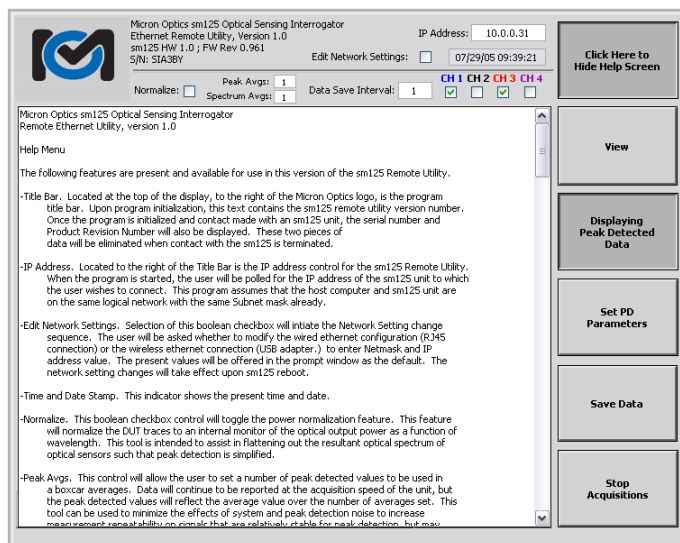
Each entry consists of a single row, delimited by tabs. The first entry of the row is the timestamp for the data set. The next four entries are the number of sensors for the four channels, in order. Following the number of sensor are all of the read sensor wavelengths and amplitudes, with each channels wavelength and power reading grouped together. In the example shown in the table below, there is one sensor on channel 1, two sensors on channel 3 and zero sensors on channels 2 and 4.

TIMEBASE	CH1	CH2	CH3	CH4	DATA
10421.000	1	0	2	0	1547.2300 -8.9100 1534.3432 1544.1429 -8.5200 -8.8100
10422.000	1	0	2	0	1547.2300 -8.9100 1534.3432 1544.1429 -8.5200 -8.8100
10423.000	1	0	2	0	1547.2297 -8.9100 1534.3435 1544.1422 -8.5700 -8.8500
10424.000	1	0	2	0	1547.2297 -8.9100 1534.3435 1544.1422 -8.5700 -8.8500
10425.000	1	0	2	0	1547.2300 -8.8300 1534.3444 1544.1407 -8.5700 -8.7700
10426.000	1	0	2	0	1547.2308 -8.9200 1534.3442 1544.1423 -8.5600 -8.9000
10427.000	1	0	2	0	1547.2308 -8.9200 1534.3442 1544.1423 -8.5600 -8.9000
10428.000	1	0	2	0	1547.2302 -8.9100 1534.3444 1544.1430 -8.5600 -8.8400
10429.000	1	0	2	0	1547.2300 -8.8600 1534.3432 1544.1415 -8.5500 -8.8100
10432.000	1	0	2	0	1547.2296 -8.8600 1534.3435 1544.1423 -8.5700 -8.7700
10433.000	1	0	2	0	1547.2303 -8.9600 1534.3429 1544.1414 -8.5300 -8.8000
10436.000	1	0	2	0	1547.2299 -8.8600 1534.3444 1544.1435 -8.5700 -8.8200
10439.000	1	0	2	0	1547.2293 -8.9200 1534.3428 1544.1400 -8.5700 -8.8400

Example of Peak Data file format from sm125 LabVIEW Ethernet Remote Utility example

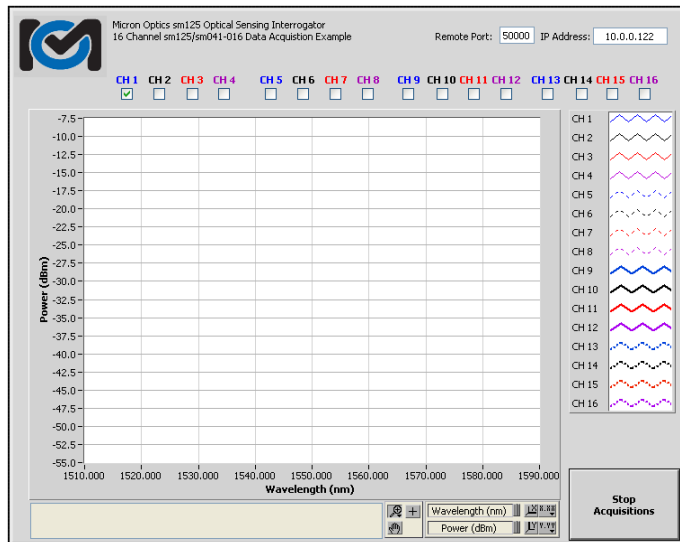
vii. On-screen Help Menu

Note the button on the upper right side of the remote utility labeled “Show Help.” By clicking on this control, the user will activate a help screen, which will guide the user through the operation of the LabVIEW Ethernet Remote Utility example. Any changes to the remote utility example following the release of this manual will be captured and sufficiently explained using this on-screen Help feature. The following figure shows the sm125 remote utility with the help screen active. To hide the help screen, simply click on the upper right button again, now labeled “Click Here to Hide Help Screen.”



sm125 Ethernet Remote Utility with the Help Screen active.

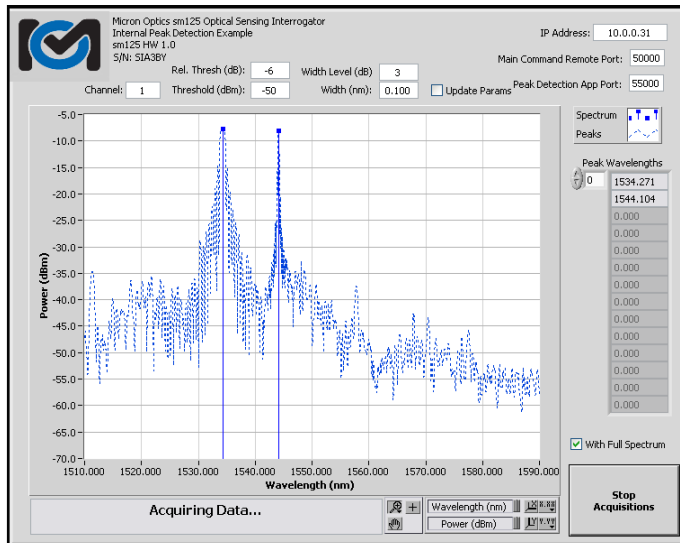
5.2.1.2. Channel Acquisition Example (with sm041)



sm125 Data Acquisition Example

Also included with the sm125 support materials is a 16 Channel Acquisition Example. This example program utilizes the commands required to operate the sm125 module with the optional sm041 Channel Expansion Module. Each of the checkbox controls seen on the top of the interface enables/disables its corresponding acquisition channel on the sm125/sm041 combination. Additional details regarding the physical connections between the two devices can be found in section 6.1.3.2 of this document.

5.2.1.3. Internal Peak Detection Example



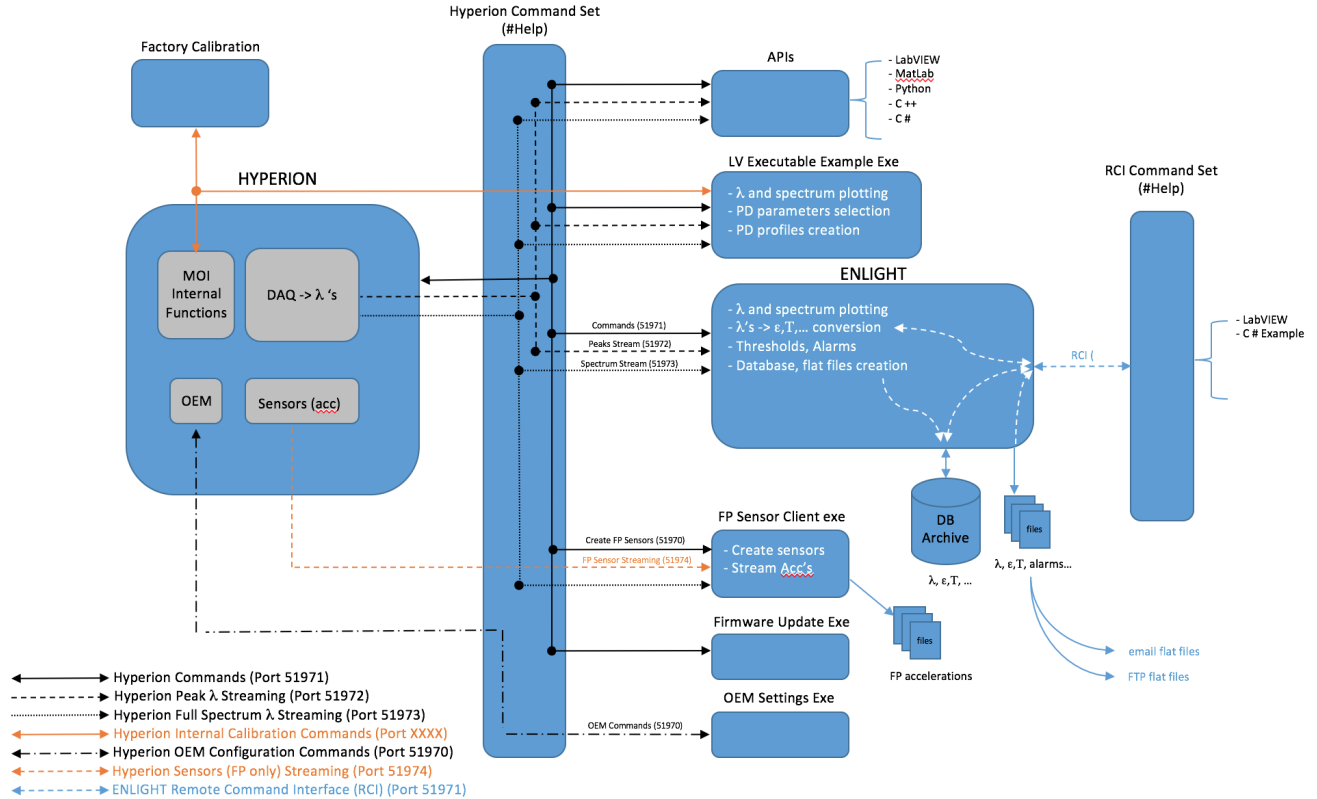
sm125 Internal Peak Detection Example

The preceding figure shows a screen shot of the included Internal Peak Detection Example. This example uses the peak detection command set outlined in section 4.1.2.6 of this document to perform peak detection on the collected spectrum inside the sm125 module itself. The purpose for this example is to demonstrate the internal peak detection functionality and to allow the user to become familiar with the features of the peak detection algorithm.



Controls available in this example include: Channel selection, Threshold and Relative Threshold controls, Width Level and Width controls, as well as IP address and port selectors. The example can either display peaks only, or simultaneous spectrum and peak data, for purpose of illustration.

6. "X55" PROGRAMMING INTERFACE



6.1. Introduction

As mentioned in Section 2 of this document, the HYPERION Optical Sensing Platform supports multiple measurement technologies and communication. The HYPERION platform supports a range of interface commands that govern system settings, network and/or communication settings, as well as data acquisition commands and queries. The following sections will convey the way in which commands are organized with respect to platform, measurement core, and communication protocol.

6.2. Communication Protocols

6.2.1. TCP/IP

Ethernet-based communication to HYPERION interrogators is achieved through a TCP socket interface on ports 51971, 51972, and 51973, and 51974. Port 51971 is the standard port used to exchange commands and responses to and from the interrogator. Ports 51972 - 51974 are used for data streaming which will be discussed in a later section.

Communication with the interrogator is achieved by submitting a valid command to the device and subsequently reading back a reply. The list of valid commands makes up the interrogator's command set and are detailed in the section 6.3 (HYPERION Remote Command Interface) of this manual.

Binary transfers of multi-byte quantities (integers, floating points, data structures, etc.) to and from the interrogator use x86 little-endian byte ordering

6.2.1.1. Write



Commands sent to a HYPERION interrogator consists of an 8-byte header followed by the command string and optional arguments. The format of the header and command/argument payload is as follows:

Byte Offset	Size (bytes)	Type	Description
0	1	u8	Request Option. 0-None, 1-Supress Message, 2-Supress Content, 4-Compress
1	1	N/A	Reserved
2	2	u16	Size of command, N_c
4	4	u32	Size of argument, N_a
8	N_c	ASCII	Command
8+ N_c	N_a	ASCII	Arguments

Commands written to a HYPERION interrogator have a total size of $8 + N_c + N_a$, where N_c is the number of bytes in the command string and N_a is the number of bytes in the command argument. All HYPERION commands are ASCII strings. However, the argument can be either ASCII readable or binary data. See specific command for more details.

The 1st byte of the header contains a *Request Option*. As described in the next section, responses to submitted commands produce two outputs, a human-readable ASCII *Message* and a machine-readable *Content*, which may or may not be ASCII formatted. The *Request Option* consists of bit fields that permit the caller to optionally suppress one or both of these responses. Further, the *Response Option* gives us the option of compressing the data returned by the interrogator. *Response Option* values are additive, meaning that multiple options can be selected by combining (adding or OR-ing) the value associated with each option. Setting the *Request Option* to 0 is the standard mode of operation and causes the interrogator to respond with uncompressed versions of both the human-readable *Message* and the machine-readable *Content*. Setting the *Request Option* to 1 causes the interrogator to suppress the human readable *Message* and respond only with the machine-readable *Content*. Setting the *Request Option* to 2 causes the interrogator to suppress the machine-readable *Content* and respond only with the human readable *Message*. Setting the *Request Option* to 3 suppresses both the *Message* and *Content*. In the same way, adding a value of 4 to the above *Request Option* causes all un-suppressed outputs to be compressed.

Compression of the responses from the interrogator is performed using the Deflate algorithm (gzip data format, RFC 1951) when the compress option is requested.

6.2.1.2. Read

HYPERION interrogators respond to commands with an 8-byte header optionally followed by the requested information formatted as a human-readable ASCII string, as a machine-readable non-ASCII block of data or both. The format of the header and response payload is as follows:

Byte Offset	Size (bytes)	Type	Description
0	1 bytes	u8	Status
1	1 bytes	u8	Option (Request Option echo)
2	2 bytes	u16	Message length, N_m
4	4 bytes	u32	Content length, N_c
8	N_m bytes	ASCII	Message string
8 + N_m	N_c bytes	variable	Content

The human-readable ASCII message, has a byte length of N_m while the machine-readable output *Content* has a byte length of N_c .

The returned *Status* byte can have one of the following values:

Value	Description
0 – Success	The requested command completed successfully
1 – Failed	The requested action is invalid



Should the status bit return as a 1, additional error information will be found in the “message” field returned from the instrument.

6.2.1.3. Streaming Responses from the Interrogator

The command/response method described in the previous sections can be used to retrieve data and other information from HYPERION interrogators. However, for high speed, continuous acquisition, the recommended method of data transfer from the interrogator is through the use of a data stream. Data streams are accessed on TCP socket ports 51972 – 51974 on HYPERION interrogators: 51972 for peak data, 51974 for spectrum data, and 51975 for sensor data.

The single difference between accessing this data on port the command port 51971 versus on a streaming port is the command does not have to be repeatedly issued for after each acquisition on the streaming ports. If using the command port for data acquisition, the command must be issued at the acquisition rate. For example, on a 1khz instrument, the command needs to be issued 1000 per second. The streaming port eliminates this requirement by internally generating and sending the command response. As a result, a client consuming data from a streaming port does not need to issue a command. It pulls data from the stream without needing to send a specific request. This decreases network traffic and significantly improves overall network performance and while maintaining continuity in the data.

Peak data streaming begins when a network client opens port 51972. The format of the data streaming on port 51972 is identical to the *read* response to the #GetPeaks command issued on the standard 51971 port.

Spectrum data streaming begins when a network client opens port 51973. The format of the data streaming on port 51973 is identical to the *read* response to a #GetSpectrum command issued on the standard 51971 port.

Sensor data streaming begins when a network client opens port 51974. The format of the data streaming on port 51974 is identical to the *read* response to a #GetSensorData command issued on the standard 51971 port.

6.2.2. UDP

A streaming socket can be created on a UDP socket for streaming peak data. This communication stream results in a more deterministic delivery of data for use in, for example, control environments where data timing is critical. See section 6.3 (HYPERION Remote Command Interface) for more information on specifying this socket.

The format of the data streaming on a UDP socket is identical to the *read* response to the #GetPeaks command issued on the standard 51971 port.

6.3. HYPERION Remote Command Interface

A summary of all HYPERION commands can be obtained from the instrument by issuing the #Help command. A listing of the different command groups can be obtained by issuing the command #Help with a single argument specifying a command group.

The following convention is used in the description of the commands:

#MyCommand

Description:

Here is a description of each command with information regarding arguments (if any) and message and content results. N/A may be used to indicate that are arguments, messages, and/or content associated with this command.

Syntax:

Command: #MyCommand
Arguments: [description of required argument #1] . . . <description of optional argument #1> . . .

Example:

Command: #MyCommand
Arguments: [required argument #1] . . . <optional argument #1> . . .

Message:

Typical message for the example usage above.

**Content:**

Length: Description of the length and type of data of the content returned by this command.
 Content: Example content from the example usage above.

In some cases, a more detailed content response is tabulated. The following symbols are used to describe the different data types used in the description of commands. All numbers are transferred in little endian format.

Abbreviation	Data Type Description
float	Represents a 32-bit IEEE 754 encoded single precision floating-point number (e.g., a C/C++ float)
double	Represents a 64-bit IEEE 754 encoded double precision floating-point number (e.g., a C/C++ double)
s8	Represents a signed 8-bit integer
s16	Represents a signed 16-bit integer
s24	Represents a signed 24-bit integer
s32	Represents a signed 32-bit integer
s64	Represents a signed 64-bit integer
struct	Represents a data structure
string	Represents an un-terminated ASCII string
stringZ	Represents a zero-terminated ASCII string
u8	Represents an 8-bit unsigned integer
u16	Represents a 16-bit unsigned integer
u24	Represents a 24-bit unsigned integer
u32	Represents a 32-bit unsigned integer
u64	Represents a 64-bit unsigned integer

#Help**Description:**

Returns the list of available commands. The list can be limited to a specific group using '#Help System'. Groups can be System, Detection, Acquisition, Laser, Network, Calibration, Host, or Sensors.

Syntax:

Command: #Help
 Arguments: <Group Name>

Example:

Command: #Help
 Arguments: System

Message:

SYSTEM (17)

#Help - Returns the list of the available commands. The list can be limited to a specific group using '#Help System'.
 #GetCommandNames - Get a list of command names that are available from a terminal (non-binary input/output).
 #GetEnabledFeatures - Get the instruments features that are enabled and exposed. This is a bit field with
 ...

Content:

N/A

#GetCommandNames**Description:**

Get a list of command names that are available from a terminal (non-binary input/output). This will not include those commands such as **#GetPeaks** that require parsing the returned content.

Syntax:

Command: #GetCommandNames
 Arguments: N/A



Example:
Command: #GetCommandNames
Arguments: N/A

Message:
#AddDetectionSetting
#AddSensor
#AddSshPublicKey
#ClearAllPeakOffsets
#ClearPeakOffsets
#ConvertCountToWavelength
#ConvertWavelengthToCount
...

Content:
N/A

#GetEnabledFeatures

Description:
Get the instruments features that are enabled and exposed. This is a bit field with the current entries: 0x01 - Channel Delay, OEM information - 0x02. The content is returned as a UInt64 where each enabled feature value is OR'ed together.

Syntax:
Command: #GetEnabledFeatures
Arguments: N/A

Example:
Command: #GetEnabledFeatures
Arguments: N/A

Message:
The following features are enabled: ChannelDelay.

Content:
Length: 8 bytes (u64)
Content: 0x 01 00 00 00 00 00 00 00

#GetFirmwareVersion

Description:
Get the version of the installed firmware. This version is considered the "System Version" that indicates the version of all internal software versions within the instrument.

Syntax:
Command: #GetFirmwareVersion
Arguments: N/A

Example:
Command: #GetFirmwareVersion
Arguments: N/A

Message:
Firmware Version: 12.12.1.20099.

Content:
Length: Variable length ASCII string
Content: 0x 31 32 2E 31 32 2E 31 2E 32 30 30 39 39

#GetFpgaVersion

Description:
Get the version of the installed FPGA code. It is recommended to use #GetFirmwareVersion to keep track of "System Version" of the instrument.

Syntax:
Command: #GetFpgaVersion
Arguments: N/A



Example:

Command: #GetFpgaVersion
Arguments: N/A

Message:

FPGA Version: 12.0.13.0.

Content:

Length: Variable length ASCII string
Content: 0x 31 32 2E 30 2E 31 33 2E 30

#GetProductPlatform

Description:

Get the product platform for the instrument (e.g. HYPERION).

Syntax:

Command: #GetProductPlatform
Arguments: N/A

Example:

Command: #GetProductPlatform
Arguments: N/A

Message:

The product platform is HYPERION.

Content:

Length: Variable length ASCII string
Content: 0x 48 79 70 65 72 69 6F 6E

#GetSerialNumber

Description:

Get the instrument serial number.

Syntax:

Command: #GetSerialNumber
Arguments: N/A

Example:

Command: #GetSerialNumber
Arguments: N/A

Message:

The instrument serial number is HIAD8G.

Content:

Length: Variable length ASCII string
Content: 0x 48 49 41 44 38 37

#GetHardwareInformation

Description:

Get the information that describes the hardware and its capabilities. The information is returned as a string with each field delimited by the '#' character. The fields are 'info_version#model#board_version#board_revision#num_physical_ch#num_active_ch#amp_type#polar_type#redundancy_type'. This command can be used to return more information about the specific configuration of the instrument.

Syntax:

Command: #GetHardwareInformation
Arguments: N/A

Example:

Command: #GetHardwareInformation
Arguments: N/A

Message:

The hardware information key is 1.0#SI255#0#ZZ#16#16#Log#Unknown#None.

Content:



Length: Variable length ASCII string
Content: 0x 31 2E 30 . . .

#GetInstrumentName

Description:

Get the name of the instrument (Lab-1). The maximum length is 20 characters.

Syntax:

Command: #GetInstrumentName
Arguments: N/A

Example:

Command: #GetInstrumentName
Arguments: N/A

Message:

The instrument name is Lab-1.

Content:

Length: Variable length ASCII string
Content: 0x 4C 61 62 2D 31

#SetInstrumentName

Description:

Set the name of the instrument (Lab-1). The maximum length is 20 characters. Names with spaces should be passed in single quotation marks.

Syntax:

Command: #SetInstrumentName
Arguments: [Name]

Example:

Command: #SetInstrumentName
Arguments: 'Lab-1'

Message:

The instrument name has been set to Lab-1.

Content:

N/A.

#IsReady

Description:

Determine if the system is able to provide data. An illuminated green LED on the front panel of the instrument is a visual indicator of this function. 0x1 is returned when ready, 0x0 is returned when not ready.

Syntax:

Command: #IsReady
Arguments: N/A

Example:

Command: #IsReady
Arguments: N/A

Message:

The System is ready

Content:

Length: 1 byte
Content: 0x 01

#Reboot

Description:

Rebooting will begin in 2 seconds.

Syntax:

Command: #Reboot



Example: Arguments: N/A

Message: Command: #Reboot
Arguments: N/A

Content: Rebooting in 2 seconds.

#GetUserData

Description: Get the user data from the specified location. Valid locations are 1-8. The maximum size is 65534 bytes.

Syntax: Command: #GetUserData
Arguments: [User Data Location 1-8]

Example: Command: #GetUserData
Arguments: 1

Message: 1024 bytes of user returned from storage.

Content: Length - Variable (up to 65534 bytes)
Content - Variable

#SetUserData

Description: Set user data for the specified location. Valid locations are 1-8. The maximum size is 65534 bytes. The data is automatically saved when set. The first byte is used to determine the location. THE PAYLOAD FOR THIS COMMAND IS BINARY AND CANNOT BE EASILY ACCESSED THROUGH A TEXT BASED INTERFACE.

Syntax: Command: #SetUserData
Arguments: See table below

Byte Offset	Size (Bytes)	Type	Description
0	1	U8	User data location 1-8
2	variable	variable	Payload

Example: Command: #SetUserData
Arguments: See table above

Message: 1024 bytes of user data were received and stored.

Content: N/A

#AddSshPublicKey

Description: Add a new public ssh-rsa key to allow SSH access to the instrument using the username 'hyperion'. The first argument is a key name (used as description only) followed by the contents of the id_rsa.pub file generated by ssh-keygen.

Syntax: Command: #AddSshPublicKey
Arguments: [Key Name] [Contents of public key]

Example:

Command: #AddSshPublicKey
 Arguments: MyPublicKey ssh-rsa AAAAB3Nz . . .

Message: The key was successfully added.
Content: N/A

#RemoveSshPublicKey

Description: Remove an existing public ssh-rsa key used to allow SSH access to the instrument using the username 'hyperion'. The first argument and only argument is the key name.

Syntax: Command: #RemoveSshPublicKey
 Arguments: [Key Name]

Example: Command: #RemoveSshPublicKey
 Arguments: MyPublicKey

Message: The key was successfully removed.
Content: N/A

#GetSshPublicKeyNames

Description: Get a list of the public RSA SSH keys currently installed for the 'hyperion' user.

Syntax: Command: #GetSshPublicKeyNames
 Arguments: N/A

Example: Command: #GetSshPublicKeyNames
 Arguments: N/A

Message: Key1
 Key2

Content: Length: Variable length ASCII string
 Content: 0x 4B 65 79 . . .

#GetDutChannelCount

Description: Get the number of DUT channels. Valid values are 4, 8, 12, 16. The value is returned as a Int32.

Syntax: Command: #GetDutChannelCount
 Arguments: N/A

Example: Command: #GetDutChannelCount
 Arguments: N/A

Message: The number of DUT channels is 16.

Content: Length: 4 bytes (u32)
 Content: 0x 00 00 00 10

#SetPeakOffsets



Description:

Set the distance related time of flight peak offsets (in ns) for the specified channel. The first parameter is the channel number, the second is the number of count/delay pairs, and the remainder of the data are the offset zone. Each zone is defined by the count in the scan and the offset in nanoseconds to apply. Any DUT peak that falls within the zone will be adjusted by the time offset before it is converted to a wavelength value.

Syntax:

Command: #SetPeakOffsets
Arguments: [Channel Number] [Number of Zones] [Zone 1 Delay (ns)] [Zone 1 Boundary] . . .

Example:

Command: #SetPeakOffsets
Arguments: 1 1 1000 10000

Message:

The peak offsets for Channel 1 have been updated.

Content:

N/A

#GetPeakOffsets

Description:

Get the distance related time of flight peak offsets (in ns) for the specified channel. The first two bytes are an UInt16 with the number of pairs. This is followed by each time/boundary offset as UInt32 then UInt16.

Syntax:

Command: #GetPeakOffsets
Arguments: [Channel Number]

Example:

Command: #GetPeakOffsets
Arguments: 1

Message:

1 peak offsets: (1000, 10000).

Content:

Content: 0x 01 00 E8 03 00 00 10 27

Byte Offset	Size (Bytes)	Type	Description
0	2	u16	Number of pairs of offsets and zone boundaries
2	2	u16	1 st time of flight delay (ns)
4	4	u32	1 st zone boundary (counts)
8	2	u16	2 nd time of flight delay (ns)
...

#ClearPeakOffsets

Description:

Clear the peak offsets for the specified channel.

Syntax:

Command: #ClearPeakOffsets
Arguments: [Channel Number]

Example:

Command: #ClearPeakOffsets
Arguments: 1

Message:

Peak offsets for channel 1 have been cleared.

Content:

N/A

**#ClearAllPeakOffsets**

Description: Clear the peak offsets for all channels.

Syntax: Command: #ClearAllPeakOffsets
Arguments: N/A

Example: Command: #ClearAllPeakOffsets
Arguments: N/A

Message: Peak offsets for all channels have been cleared.

Content: N/A.

#SetChannelDelay

Description: Set the time of flight delay (in ns) for the specified channel. This command is used to correct the full spectrum for the time of flight of light through the fiber. Applicable when ALL sensors on a channel are co-located.

Syntax: Command: #SetChannelDelay
Arguments: [Channel Number] [Channel Delay in ns]

Example: Command: #SetChannelDelay
Arguments: 1 1000

Message: The channel delay (in ns) for Channel 1 has been set to 1000.

Content: N/A

#GetChannelDelay

Description: Get the time of flight delay (in ns) for the specified channel.

Syntax: Command: #GetChannelDelay
Arguments: [Channel Number]

Example: Command: #GetChannelDelay
Arguments: 1

Message: The channel delay (in ns) for Channel 1 is 1000.

Content: Length: 4 bytes (u32)
Content: 0x E8 03 00 00

#GetMaximumPeakCountPerDutChannel

Description: Get the maximum number of DUT peaks per optical channel.

Syntax: Command: #GetMaximumPeakCountPerDutChannel
Arguments: N/A

Example: Command: #GetMaximumPeakCountPerDutChannel
Arguments: N/A

Message:



Content:
 The DUT peak count maximum is 256.
 Length: 4 bytes (u32)
 Content: 0x 00 01 00 00

#GetDutPeakCountMaximum

Description:
 Get the maximum number of DUT peaks per optical channel.

Syntax:
 Command: #GetDutPeakCountMaximum
 Arguments: N/A

Example:
 Command: #GetDutPeakCountMaximum
 Arguments: N/A

Message:
 The DUT peak count maximum is 256.

Content:
 Length: 4 bytes (u32)
 Content: 0x 00 01 00 00

#GetAvailableDetectionSettings

Description:
 Get the available user and preset peak detection settings that can be assigned to channels to determine the location of peak/valley features. IDs from 1-127 are user definable and 128-255 are fixed presets.

Syntax:
 Command: #GetAvailableDetectionSettings
 Arguments: N/A

Example:
 Command: #GetAvailableDetectionSettings
 Arguments: N/A

Message:
 ID: 1
 Name: 0.25 nm Peak (EV)
 Description: 0.25 nm Peak Default (strip/recoat FBGs and DTGs) for use with a "EV" type instrument
 Boxcar (pm): 350
 Diff (pm): 350 . . .

Content:

Byte Offset	Size (Bytes)	Type	Description
0	1	u8	1 st setting ID number
1	1	u8	Length of name for 1 st setting (<i>Nn1</i>)
2	<i>Nn1</i>	ASCII	Name for 1 st setting
<i>Nn1+2</i>	1	u8	Length of description for 1 st setting (<i>Nd1</i>)
<i>Nn1+3</i>	<i>Nd1</i>	ASCII	Description for 1 st setting
<i>Nn1+Nd1+3</i>	2	u16	Boxcar for 1 st setting
<i>Nn1+Nd1+5</i>	2	u16	Diff delay for 1 st setting
<i>Nn1+Nd1+7</i>	2	u16	Lockout for 1 st setting
<i>Nn1+Nd1+9</i>	2	u16	NTV for 1 st setting
<i>Nn1+Nd1+11</i>	4	u32	Threshold for 1 st setting
<i>Nn1+Nd1+15</i>	1	u8	Detection mode, Peak - 1, Valley - 0
<i>Nn1+Nd1+16</i>	1	u8	2 nd setting ID Number
...

#GetDetectionSetting



Description: Get the peak detection settings for a specified ID (1-127 are user, 128-255 are preset).

Syntax: Command: #GetDetectionSetting
Arguments: [Detection ID]

Example: Command: #GetDetectionSetting
Arguments: 1

Message: ID: 1
Name: 0.25 nm Peak (EV)
Description: 0.25 nm Peak Default (strip/recoat FBGs and DTGs) for use with a "EV" type instrument
Boxcar (pm): 350
Diff (pm): 350 . . .

Content: See the content description for #GetAvailableDetectionSettings.

#AddDetectionSetting

Description: Add a new detection setting. The following parameters are required: ID (valid IDs for user created settings are 1-127), name (length <= 32), description (length <= 128), boxcar filter length (in pm), difference filter length (in pm), lockout period (in pm), NTV period (in pm), threshold, and detection mode (0 => Valley, 1 => Peak). If the ID is already in use an error is returned.

Syntax: Command: #AddDetectionSetting
Arguments: [Detection ID Number] [Name between single quotes] [Description between single quotes] [Box Car Length] [Diff Filter Length] [Lockout Period] [NTV Period] [Threshold] [Detection Mode]

Example: Command: #AddDetectionSetting
Arguments: 1 'My New ID' 'My New Detection Description' 250 250 0 100 10000 1

Message: Detection Setting with ID = 1 has been added:
ID: 1
Name: My Name
Description: My Description
Boxcar (pm): 250
Diff (pm): 250
Lockout (pm): 0
NTV (pm): 1000
Threshold: 1000
Mode: Peak

Content: N/A

#RemoveDetectionSetting

Description: Remove the user detection setting with the specified ID (1-127). Settings currently assigned to a channel cannot be removed.

Syntax: Command: #RemoveDetectionSetting
Arguments: [ID Number]

Example: Command: #RemoveDetectionSetting
Arguments: 1

Message: The detection setting with ID = 1 has been removed.



Content:
N/A

#UpdateDetectionSetting

Description: Update existing user detection setting with the specified ID (1-127). Any channel assigned to the preset immediately begins using the new values.

Syntax:
Command: #UpdateDetectionSetting
Arguments: See #AddDetectionSetting

Example:
Command: #UpdateDetectionSetting
Arguments: 1 'My Name' 'My Description' 250 330 0 1000 1000 0

Message:
Detection Setting updated (0 channels updated):

ID: 8
Name: My Name
Description: My Description
Boxcar (pm): 250
Diff (pm): 330
Lockout (pm): 0
NTV (pm): 1000
Threshold: 1000
Mode: Valley.

Content:
N/A

#SetChannelDetectionSettingId

Description: Set the ID of the peak detection settings used to determine the location of peak/valley features.

Syntax:
Command: #SetChannelDetectionSettingId
Arguments: [Channel Number] [Detection ID]

Example:
Command: #SetChannelDetectionSettingId
Arguments: 1 128

Message:
The detection setting ID for Channel 1 has been set to '0.25 nm Peak (Preset)' (ID = 128).

Content:
N/A

#GetChannelDetectionSettingId

Description: Get the ID of the peak detection settings used to determine the location of peak/valley features.

Syntax:
Command: #GetChannelDetectionSettingId
Arguments: [Channel Number]

Example:
Command: #GetChannelDetectionSettingId
Arguments: 1

Message:
The detection setting ID for Channel 1 is 128 - 0.25 nm Peak (Preset).

Content:
Length – 2 bytes (u16)



Content – 0x 80 00

#GetAllChannelDetectionSettingIds

Description:

Get the IDs of the peak detection settings used to determine the location of peak/valley features for all of the channels.

Syntax:

Command: #GetAllChannelDetectionSettingIds
Arguments: N/A

Example:

Command: #GetAllChannelDetectionSettingIds
Arguments: N/A

Message:

The channel detection setting IDs are:

- 1 - 128 - 0.25 nm Peak (Preset)
- 2 - 128 - 0.25 nm Peak (Preset)
- 3 - 128 - 0.25 nm Peak (Preset)
- 4 - 128 - 0.25 nm Peak (Preset)
- 5 - 128 - 0.25 nm Peak (Preset)
- 6 - 128 - 0.25 nm Peak (Preset)
- 7 - 128 - 0.25 nm Peak (Preset)
- 8 - 128 - 0.25 nm Peak (Preset)
- 9 - 128 - 0.25 nm Peak (Preset)
- 10 - 128 - 0.25 nm Peak (Preset)
- 11 - 128 - 0.25 nm Peak (Preset)
- 12 - 128 - 0.25 nm Peak (Preset)
- 13 - 128 - 0.25 nm Peak (Preset)
- 14 - 128 - 0.25 nm Peak (Preset)
- 15 - 128 - 0.25 nm Peak (Preset)
- 16 - 128 - 0.25 nm Peak (Preset)

Content:

Length – Number of Channels bytes (e.g. 16 bytes for a 16 channel instrument)
Content – 0x 80 80 80 80 . . .

#GetUserWavelengthStart

Description:

Get the starting wavelength (in nm).

Syntax:

Command: #GetUserWavelengthStart
Arguments: N/A

Example:

Command: #GetUserWavelengthStart
Arguments: N/A

Message:

The Starting Wavelength (nm) is currently set to 1460.000.

Content:

Length – 8 bytes (double)
Content – 0x 00 00 00 00 00 00 D0 96 40

#GetUserWavelengthStop

Description:

Get the stop wavelength (in nm).

Syntax:

Command: #GetUserWavelengthStop



Arguments: N/A

Example:

Command: #GetUserWavelengthStop
Arguments: N/A

Message:

The Ending Wavelength (nm) is currently set to 1620.000.

Content:

Length – 8 bytes (double)
Content – 0x 00 00 00 00 00 50 99 40

#GetUserWavelengthDelta

Description:

Get the wavelength difference (in nm) between full spectrum amplitudes.

Syntax:

Command: #GetUserWavelengthDelta
Arguments: N/A

Example:

Command: #GetUserWavelengthDelta
Arguments: N/A

Message:

The Delta Wavelength (nm) is currently set to 0.008.

Content:

Length – 8 bytes (double)
Content – 0x FC A9 F1 D2 4D 62 80 3F

#GetUserWavelengthNumberOfPoints

Description:

Get the number of points returned in the full spectrum.

Syntax:

Command: #GetUserWavelengthNumberOfPoints
Arguments: N/A

Example:

Command: #GetUserWavelengthNumberOfPoints
Arguments: N/A

Message:

The Number Of Points is currently set to 20000.

Content:

Length – 4 bytes (u23)
Content – 0x 20 4E 00 00

#ConvertWavelengthToCount

Description:

Get the count (for the current speed) value that corresponds to the specified wavelength (in nanometers) and offset (in nanoseconds).

Syntax:

Command: #ConvertWavelengthToCount
Arguments: [Wavelength (nm)] [Time of Flight Offset (ns)]

Example:

Command: #ConvertWavelengthToCount
Arguments: 1550 10000

Message:

The count corresponding to the wavelength is 16972.5913.

Content:

Length – 8 bytes (double)
Content – 0x 54 0F A5 D8 25 93 D0 40



#ConvertCountToWavelength

Description: Get the wavelength (in nm) that corresponds to the specified count.

Syntax: Command: #ConvertCountToWavelength
Arguments: [Count Value]

Example: Command: #ConvertCountToWavelength
Arguments: 16972

Message: The wavelength (in nm) corresponding to the count is 1550.0419.

Content: Length – 8 bytes (double)
Content – 0x D7 D4 17 EB 2A 38 98 40

#ShiftWavelengthByOffset

Description: Get the wavelength resulting from a time of flight offset. The offset is in nanoseconds and can be positive or negative.

Syntax: Command: #ShiftWavelengthByOffset
Arguments: [Wavelength (nm)] [Time of Flight Offset (ns)]

Example: Command: #ShiftWavelengthByOffset
Arguments: 1550 100000

Message: The shifted wavelength is 1550.4180.

Content: Length – 8 bytes (double)
Content – 0x 93 B7 2C 02 AC 39 98 40

#GetPeaks

Description: Get the detected optical peaks and valleys.

Syntax: Command: #GetPeaks
Arguments: N/A

Example: Command: #GetPeaks
Arguments: N/A

Message: N/A

Content: See table below for more information. This table is used to parse the content of the #GetPeaks command on the command port 51971 as well as the peak streaming port 51972 and any UDP socket.

Byte Offset	Size (bytes)	Data Type	Description
0	2	u16	<i>Header Length</i> - The size of the header in bytes. This is also the offset to the beginning of any data transferred with the header.
2	2	u16	<i>Header Version</i> - The version in place to manage software-firmware compatibility issues that may arise as the firmware of the device evolves.



4	4	N/A	Reserved
8	8	u64	<i>Serial Number</i> - Each acquisition performed by the instrument is serialized beginning with serial number 0 whenever the instrument is powered up. Successive acquisitions have successive serial numbers. The data set serial numbers can be used to verify no loss of data during transfer.
16	4	u32	<i>Timestamp (seconds)</i> - The timestamp is the absolute time since 00:00:00 of January 1, 1970. More concisely, <i>Timestamp</i> represents the elapsed time since 1969. The timestamp is provided in two parts: an integer part in seconds and a fractional part in nanoseconds (10^{-9} seconds). The timestamp, in seconds, is full reconstructed as: $\text{timestamp} = \text{Timestamp (seconds)} + \text{Timestamp (nanoseconds)} * 10^{-9}$
20	4	u32	<i>Timestamp (nanoseconds)</i> - See above.
24	32	u16	<i>NumPeaks</i> - A 16-element array of u16 values. Each element in the array corresponds to a DUT channel on the interrogator. The value of the array is the number of peaks detected on the corresponding channel. Thus, if the 6 th element has a value of 11, then 11 optical peaks were detected on the 6 th channel (Channel 6) of the interrogator.
56	Variable	double array	<i>Array of Peaks (nm)</i> - An array of doubles in which the length is equal to the sum of the elements in the <i>NumPeaks</i> array.

#GetSpectrum

Description:

Get detected optical spectrum. If no parameter is specified all available channels are returned. A single channel number may be specified to return data for only that channel.

Syntax:

Command: #GetSpectrum
Arguments: N/A

Example:

Command: #GetSpectrum
Arguments: N/A

Message:

N/A

Content:

See table below for more information. This table is used to parse the content of the **#GetSpectrum** command on the command port 51971 as well as the spectrum streaming port 51973.

Byte Offset	Size (bytes)	Type	Description
0	2	u16	<i>Header Length</i> - The size of the header in bytes. This is also the offset to the beginning of any data transferred with the header.
2	2	u16	<i>Header Version</i> - In place to manage software-firmware compatibility issues that may arise as the firmware of the device evolves.
4	4	N/A	Reserved
8	8	u64	<i>Serial Number (seconds)</i> - Each acquisition performed by the instrument is serialized beginning with serial number 0 whenever the instrument is powered up. Successive acquisitions have successive serial numbers. For every acquired spectrum, there is a corresponding Peaks data set. Use the <i>Serial Number</i> field in this header to match the spectral data to a corresponding peaks data obtained with #GetPeaks.
16	4	u32	<i>Timestamp (seconds)</i> - The <i>Timestamp</i> is the absolute time since 00:00:00



			of January 1, 1970. More concisely, <i>Timestamp</i> represents the elapsed time since 1969. The timestamp is provided in two parts: an integer part in seconds and a fractional part in nanoseconds (10^{-9} seconds). The timestamp, in seconds, is full reconstructed as:
			$\text{timestamp} = \text{Timestamp (seconds)} + \text{Timestamp (nanoseconds)} * 10^{-9}.$
20	4	u32	<i>Timestamp (nanoseconds)</i> - See description of <i>Timestamp (seconds)</i> portion.
24	8	double	<i>StartWvl</i> - Start wavelength is a double precision number indicating the corresponding wavelength in nm of the first power level returned by #GetSpectrum.
32	8	double	<i>WvlInc</i> - Wavelength Increment is the wavelength increment between successive data points. Thus, if data point <i>N</i> has a corresponding wavelength of <i>W</i> , data point <i>N+1</i> will have a wavelength of <i>W + WvlInc</i> .
40	4	u32	<i>NumPoints</i> - The number of data points contained in the spectrum. If we designate the wavelength of the final data point (the <i>NumPoints</i> th data point) as <i>EndWvl</i> , then, we can calculate <i>EndWvl</i> as: $\text{EndWvl} = \text{StartWvl} + \text{WvlInc} * (\text{NumPoints} - 1).$
44	2	u16	<i>Channels</i> - A bit-field value where each bit corresponds to one of 16 optical channels. When a bit is set, then spectral data for the corresponding channel is included in the current data set. If all 16-bits are set, then spectral data for all 16 channels are included in the current data set. Power data from different channels are organized in order based on channel numbers beginning with the lowest channel number and proceeding to the highest. Note that the first channel (channel 1) corresponds to the first bit (bit 0). Thus, channels 1-16 are represented by bits 0-15.
46	2	N/A	Reserved
48	Variable	u16 array	<i>Channel Spectra</i> – Each channel’s spectrum is encoded as an array of u16’s with length <i>NumPoints</i> . Each active full spectrum (indicated by the <i>Channels</i> bit field) spectrum is concatenated together here. The values must be scaled via the power calibration information for that table. See #GetPowerCalibrationInfo.

#GetActiveFullSpectrumDutChannelNumbers

Description:

Get the list of active full spectrum channels.

Syntax:

Command: #GetActiveFullSpectrumDutChannelNumbers
Arguments: N/A

Example:

Command: #GetActiveFullSpectrumDutChannelNumbers
Arguments: N/A

Message:

The active DUT channel numbers are 1 5 9 13.

Content:

Length: Array of u32’s (Length of array is equal to number of active channels)
Content: 0x 01 00 00 00 05 00 00 00 09 00 00 00 0D 00 00 00

#SetActiveFullSpectrumDutChannelNumbers

Description:

Set the list of active full spectrum channels.

Syntax:

Command: #SetActiveFullSpectrumDutChannelNumbers
Arguments: [Channel Number] <Channel Number> <Channel Number> . . .



Example:
Command: #SetActiveFullSpectrumDutChannelNumbers
Arguments: 1 16

Message:
The active full spectrum channels have been successfully updated.

Content:
N/A

#EnablePeakDataStreaming

Description:
Enable receiving data on the peak data streaming socket.

Syntax:
Command: #EnablePeakDataStreaming
Arguments: N/A

Example:
Command: #EnablePeakDataStreaming
Arguments: N/A

Message:
The data streaming socket has been enabled.

Content:
N/A

#DisablePeakDataStreaming

Description:
Disable receiving data on the peak data streaming socket (enabled by default). Newly received peak data will be skipped until streaming is reenabled.

Syntax:
Command: #DisablePeakDataStreaming
Arguments: N/A

Example:
Command: #DisablePeakDataStreaming
Arguments: N/A

Message:
The data streaming socket has been disabled.

Content:
N/A

#GetPeakDataStreamingStatus

Description:
Determine if peak data is currently being received on the streaming socket. A value of 0 indicates that the streaming output is disabled. A value of 1 indicates that the streaming output is enabled.

Syntax:
Command: #GetPeakDataStreamingStatus
Arguments: N/A

Example:
Command: #GetPeakDataStreamingStatus
Arguments: N/A

Message:
The data streaming is currently disabled.

Content:
Length – 4 bytes (u32)
Content – 0x 00 00 00 00

#GetPeakDataStreamingDivider



Description: Get the current value of the divider. For example, a value of 5, will return every fifth value.

Syntax: Command: #GetPeakDataStreamingDivider
Arguments: N/A

Example: Command: #GetPeakDataStreamingDivider
Arguments: N/A

Message: The data streaming divider is currently 1.

Content: Length – 4 bytes (u32)
Content - 0x 01 00 00 00

#SetPeakDataStreamingDivider

Description: Set the value of the divider. For example, a value of 5, will return every fifth value.

Syntax: Command: #SetPeakDataStreamingDivider
Arguments: [Divider]

Example: Command: #SetPeakDataStreamingDivider
Arguments: 2

Message: The data streaming divider has been set to 2.

Content: N/A

#GetPeakDataStreamingAvailableBuffer

Description: Get that remaining available peak data streaming buffer as a percentage of the total capacity. A value of 100% indicates that the data retrieved is current and that all of the client-side streaming buffer is not being utilized. The value only has meaning when a data streaming is connected to a client.

Syntax: Command: #GetPeakDataStreamingAvailableBuffer
Arguments: N/A

Example: Command: #GetPeakDataStreamingAvailableBuffer
Arguments: N/A

Message: The available streaming data buffer is currently 100%.

Content: Length – 4 bytes (u32)
Content - 0x 64 00 00 00

#EnableFullSpectrumDataStreaming

Description: Enable receiving data on the full spectrum data streaming socket.

Syntax: Command: #EnableFullSpectrumDataStreaming
Arguments: N/A

Example: Command: #EnableFullSpectrumDataStreaming
Arguments: N/A

Message:



Content: The data streaming socket has been enabled.
N/A

#DisableFullSpectrumDataStreaming

Description: Disable receiving data on the full spectrum data streaming socket (enabled by default). Newly received full spectrum data will be skipped until streaming is reenabled.

Syntax: Command: #DisableFullSpectrumDataStreaming
Arguments: N/A

Example: Command: #DisableFullSpectrumDataStreaming
Arguments: N/A

Message: The data streaming socket has been disabled.

Content: N/A

#GetFullSpectrumDataStreamingStatus

Description: Determine if full spectrum data is currently being received on the streaming socket. A value of 0 indicates that the streaming output is disabled. A value of 1 indicates that the streaming output is enabled.

Syntax: Command: #GetFullSpectrumDataStreamingStatus
Arguments: N/A

Example: Command: #GetFullSpectrumDataStreamingStatus
Arguments: N/A

Message: The data streaming is currently disabled.

Content: Length – 4 bytes (u32)
Content - 0x 00 00 00 00

#GetFullSpectrumDataStreamingDivider

Description: Get the current value of the divider. For example, a value of 5, will return every fifth value.

Syntax: Command: #GetFullSpectrumDataStreamingDivider
Arguments: N/A

Example: Command: #GetFullSpectrumDataStreamingDivider
Arguments: N/A

Message: The data streaming divider is currently 1.

Content: Length – 4 bytes (u32)
Content - 0x 01 00 00 00

#SetFullSpectrumDataStreamingDivider

Description: Set the value of the divider. For example, a value of 5, will return every fifth value.

Syntax:



Example: Command: #SetFullSpectrumDataStreamingDivider
Arguments: [Divider]

Message: Command: #SetFullSpectrumDataStreamingDivider
Arguments: 3

Content: The data streaming divider has been set to 3.

N/A

#GetFullSpectrumDataStreamingAvailableBuffer

Description: Get that remaining available full spectrum streaming buffer as a percentage of the total capacity. A value of 100% indicates that the data retrieved is current and that all of the client side streaming buffer is not being utilized. The value only has meaning when a data streaming is connected to a client.

Syntax: Command: #GetFullSpectrumDataStreamingAvailableBuffer
Arguments: N/A

Example: Command: #GetFullSpectrumDataStreamingAvailableBuffer
Arguments: N/A

Message: The available streaming data buffer is currently 100%.

Content: Length – 4 bytes (u32)
Content - 0x 64 00 00 00

#StreamFullSpectrumToStorage

Description: Write full spectrum data to internal storage for a specified number of seconds.

Syntax: Command: #StreamFullSpectrumToStorage
Arguments: [Time (s)]

Example: Command: #StreamFullSpectrumToStorage
Arguments: 5

Message: Full spectrum will be internally acquired for 5 seconds.

Content: N/A

#StreamFullSpectrumFromStorage

Description: Read full spectrum data from internal storage. After issuing this command, continue to perform response reads until a zero length content is received. This will indicate the end of the internally stored data.

Syntax: Command: #StreamFullSpectrumFromStorage
Arguments: N/A

Example: Command: #StreamFullSpectrumFromStorage
Arguments: N/A

Message: N/A

Content:



See **#GetSpectrum**. Use the content description to continually issue command read sequences until a zero length content is received.

#SetFullSpectrumStreamToStorageDivider

Description:

Set the value of the divider used when streaming full spectrum data to storage. For example, a value of 5, will store every fifth value.

Syntax:

Command: #SetFullSpectrumStreamToStorageDivider
Arguments: [Divider]

Example:

Command: #SetFullSpectrumStreamToStorageDivider
Arguments: 5

Message:

The stream to storage divider has been set to 5.

Content:

N.A

#GetFullSpectrumStreamToStorageDivider

Description:

Get the value of the divider used when streaming full spectrum data to storage. For example, a value of 5, will store every fifth value.

Syntax:

Command: #GetFullSpectrumStreamToStorageDivider
Arguments: N/A

Example:

Command: #GetFullSpectrumStreamToStorageDivider
Arguments: N/A

Message:

The stream to storage divider is currently 5.

Content:

Length – 4 bytes (32)
Content - 0x 05 00 00 00

#SetFullSpectrumStreamToStorageAverage

Description:

Set the number of averages used when streaming full spectrum data to storage. For example, a value of 5, will average 5 consecutive spectra and write the result to storage.

Syntax:

Command: #SetFullSpectrumStreamToStorageAverage
Arguments: [Number of Averages]

Example:

Command: #SetFullSpectrumStreamToStorageAverage
Arguments: 3

Message:

The stream to storage average has been set to 3.

Content:

N/A

#GetFullSpectrumStreamToStorageAverage

Description:

Get the number of averages used when streaming full spectrum data to storage. For example, a value of 5, will average 5 consecutive spectra and write the result to storage.

Syntax:

Example: Command: #GetFullSpectrumStreamToStorageAverage
Arguments: N/A

Message: Command: #GetFullSpectrumStreamToStorageAverage
Arguments: N/A

Content: The stream to storage average is currently 3.

Length – 4 bytes (u32)
Content - 0x 03 00 00 00

#GetFullSpectrumStreamToStoragePercentComplete

Description: Get the status of the current stream to storage request as a percentage. Once the operation is complete 100 will be returned. The value will hold near 99% after the acquisition completes while the storage media is unmounted.

Syntax: Command: #GetFullSpectrumStreamToStoragePercentComplete
Arguments: N/A

Example: Command: #GetFullSpectrumStreamToStoragePercentComplete
Arguments: N/A

Message: The current operation is 100% complete.

Content: Length – 4 bytes (u32)
Content - 0x 64 00 00 00

#EnableUdpPeakDatagrams

Description: Enable transmitting UDP peak datagrams to the specified IP Address and Port. If the network is disconnected, transmission will resume when connectivity is restored.

Syntax: Command: #EnableUdpPeakDatagrams
Arguments: [IP Address] [Port]

Example: Command: #EnableUdpPeakDatagrams
Arguments: 192.168.0.10 1000

Message: The UDP Datagram Server has been started.

Content: N/A

#DisableUdpPeakDatagrams

Description: Disable transmitting UDP peak data grams.

Syntax: Command: #DisableUdpPeakDatagrams
Arguments: N/A

Example: Command: #DisableUdpPeakDatagrams
Arguments: N/A

Message: The UDP datagram transmission has been disabled.

Content:



N/A

#GetLaserScanSpeed

Description: Get the current scan rate (in Hz) of the swept laser source.

Syntax: Command: #GetLaserScanSpeed
Arguments: N/A

Example: Command: #GetLaserScanSpeed
Arguments: N/A

Message: The current scan speed is 10 Hz.

Content: Length – 4 bytes
Content - 0x 0A 00 00 00

#SetLaserScanSpeed

Description: Set the scan rate (in Hz) of the swept laser source.

Syntax: Command: #SetLaserScanSpeed
Arguments: [Scan Speed (Hz)]

Example: Command: #SetLaserScanSpeed
Arguments: 10

Message: Laser scan speed set to 10 Hz.

Content: N/A

#GetAvailableLaserScanSpeeds

Description: Get the available scan rates (in Hz) supported by the instrument.

Syntax: Command: #GetAvailableLaserScanSpeeds
Arguments: N/A

Example: Command: #GetAvailableLaserScanSpeeds
Arguments: N/A

Message: The available scan speed are 2 10 in Hz.

Content: Length – Number of Available Speeds u32's
Content - 0x 02 00 00 00 0A 00 00 00

#GetNetworkIpMode

Description: Determine if the current network configuration is static or dynamic (DHCP).

Syntax: Command: #GetNetworkIpMode
Arguments: N/A

Example: Command: #GetNetworkIpMode



Arguments: N/A
Message:
The network configuration mode is: DHCP
Content:
Length – Variable Length ASCII
Content - 0x 44 48 43 50

#EnableDynamicIpMode

Description:
Use DHCP to configure the network interface dynamically. ** THIS VALUE IS NOT PERSISTENT UNLESS YOU USE #SaveInstrumentSettings. **.

Syntax:
Command: #EnableDynamicIpMode
Arguments: N/A

Example:
Command: #EnableDynamicIpMode
Arguments: N/A

Message:
The network interface has been configured to use DHCP.

Content:
N/A

#EnableStaticIpMode

Description:
Use the static IP address, netmask, and default gateway setting to configure the network interface. ** THIS VALUE IS NOT PERSISTENT UNLESS YOU USE #SaveInstrumentSettings. **. Use care when issuing this command. The instrument, following the reply message, will change its Ethernet communication settings.

Syntax:
Command: #EnableStaticIpMode
Arguments: N/A

Example:
Command: #EnableStaticIpMode
Arguments: N/A

Message:
The following static network configuration has been applied:
IP Address: 10.0.0.55
Netmask: 255.255.255.0
Gateway: 10.0.0.1.

Content:
N/A

#GetActiveNetworkSettings

Description:
Get the active (current) network configuration settings (IP, Netmask, Gateway).

Syntax:
Command: #GetActiveNetworkSettings
Arguments: N/A

Example:
Command: #GetActiveNetworkSettings
Arguments: N/A

Message:
The current network settings are:
IP Address: 10.0.10.56
Netmask: 255.255.0.0
Gateway: 10.0.0.1



Mode: Dynamic

Content:

Byte Offset	Size (Bytes)	Type	Description
0	1	u8	IP Address - MSB
1	1	u8	IP Address
2	1	u8	IP Address
3	1	u8	IP Address - LSB
4	1	u8	Netmask - MSB
5	1	u8	Netmask
6	1	u8	Netmask
8	1	u8	Netmask - LSB
9	1	u8	Gateway - MSB
10	1	u8	Gateway
11	1	u8	Gateway
12	1	u8	Gateway - LSB
13	1	u8	DHCP – 1, Static - 0

Content - 0x 0A 00 0A 38 FF FF 00 00 0A 00 00 01 01

#GetStaticNetworkSettings

Description:

Get the static network configuration settings (IP, Netmask, Gateway).

Syntax:

Command: #GetStaticNetworkSettings
Arguments: N/A

Example:

Command: #GetStaticNetworkSettings
Arguments: N/A

Message:

The static network configuration settings are:
IP Address: 10.0.0.55
Netmask: 255.255.255.0
Gateway: 10.0.0.1

Content:

Byte Offset	Size (Bytes)	Type	Description
0	1	u8	IP Address - MSB
1	1	u8	IP Address
2	1	u8	IP Address
3	1	u8	IP Address - LSB
4	1	u8	Netmask - MSB
5	1	u8	Netmask
6	1	u8	Netmask
8	1	u8	Netmask - LSB
9	1	u8	Gateway - MSB
10	1	u8	Gateway
11	1	u8	Gateway
12	1	u8	Gateway - LSB

Content - 0x 0A 00 00 37 FF FF FF 00 0A 00 00 01

#SetStaticNetworkSettings

Description:

Configure the static network configuration settings. For example, #SetStaticNetworkSettings 192.168.1.100 255.255.255.0 192.168.1.1. Passing only the IP Address to a static configuration will automatically set the



netmask to 255.255.255.0 and a default gateway to X.X.X.1 where X.X.X are taken from the specified static IP Address.

Syntax:
 Command: #SetStaticNetworkSettings
 Arguments: [IP Address] <Netmask> <Gateway>

Example:
 Command: #SetStaticNetworkSettings
 Arguments: 10.0.0.55 255.255.255.0 10.0.0.1

Message:
 The static network configuration settings have been updated:
 IP Address: 10.0.0.55
 Netmask: 255.255.255.0
 Gateway: 10.0.0.1

Content:
 N/A

#GetPowerCalibrationInfo

Description:
 Get the minimum output power and fixed point scaling for the available channels. When power calibration is enabled, the output values can be converted to dBm using the following equation: PdBm = value / scale + offset where PdBm is the recorded received optical power in dBm. offset is Int32 and scale is UInt32.

Syntax:
 Command: #GetPowerCalibrationInfo
 Arguments: N/A

Example:
 Command: #GetPowerCalibrationInfo
 Arguments: N/A

Message:
 -65 1092

Content:
 Length – Number of Channel pairs of offset (s32) and scale (u32)
 Content – 0x BF FF FF FF 44 04 00 00 BF FF FF FF ...

#SetInstrumentUtcDateTime

Description:
 Set the instrument date/time in UTC. You must ensure that the instrument booted with the NTP server disabled or the date/time will be overwritten. The required parameters are YEAR MONTH DAY HOUR MIN SEC. HOUR is 24-hour format so am/pm is not required or supported.

Syntax:
 Command: #SetInstrumentUtcDateTime
 Arguments: [Year] [Month] [Day] [Hour] [Minute] [Second]

Example:
 Command: #SetInstrumentUtcDateTime
 Arguments: 2017 01 01 00 00 00

Message:
 The instrument date/time has been set to '2017-01-01 00:00:00'.

Content:
 N/A

#GetInstrumentUtcDateTime

Description:
 Get the instrument date/time in UTC. The value is returned as six unsigned 16-bit integers as follows: YEAR MONTH DAY HOUR MIN SEC. HOUR is 24-hour format so no am/pm is returned.

Syntax:



Example: Command: #GetInstrumentUtcDateTime
Arguments: N/A

Message: Command: #GetInstrumentUtcDateTime
Arguments: N/A

The instrument date/time is 2017-01-01 00:00:00.

Content: Length – 2 bytes each (u32) – Year, Month, Day, Hour, Minute, Second
Content - 0x E1 07 01 00 01 00 00 00 00 00 00 0.

#SetNtpEnabled

Description: Set whether or not a remote Network Time Protocol (NTP) server is used to keep the system time synchronized.

Syntax: Command: #SetNtpEnabled
Arguments: [1 – Enable, 0 – Disable]

Example: Command: #SetNtpEnabled
Arguments: 1

Message: The NTP Server has been enabled.

Content: N/A

#GetNtpEnabled

Description: Get whether or not a remote Network Time Protocol (NTP) server is used to keep the system time synchronized.

Syntax: Command: #GetNtpEnabled
Arguments: N/A

Example: Command: #GetNtpEnabled
Arguments: N/A

Message: NTP server is currently enabled.

Content: Length – u32 (0x1 – Enabled, 0x0 – Disabled)
Content - 0x 01 00 00 00

#SetNtpServer

Description: Set the IP address of the remote Network Time Protocol (NTP) server used to keep the system time synchronized.

Syntax: Command: #SetNtpServer
Arguments: [IP Address of NTP Server]

Example: Command: #SetNtpServer
Arguments: 98.175.203.200

Message: The NTP Server has been set to '98.175.203.200'.

Content:



N/A

#GetNtpServer

Description:

Get the IP address of the remote Network Time Protocol (NTP) server used to keep the system time synchronized.

Syntax:

Command: #GetNtpServer
Arguments: N/A

Example:

Command: #GetNtpServer
Arguments: N/A

Message:

NTP server is currently set to '98.175.203.200'.

Content:

Length – Variable length ASCII
Content - 0x 39 38 2E 31 37 35 2E 32 30 33 2E 32 30 30

#SetPtpEnabled

Description:

Set whether or not a Precision Time Protocol (PTP) client is used to keep the system time synchronized to a local clock.

Syntax:

Command: #SetPtpEnabled
Arguments: [1 – Enable, 0 – Disable]

Example:

Command: #SetPtpEnabled
Arguments: 1

Message:

The PTP Server has been enabled.

Content:

N/A

#GetPtpEnabled

Description:

Get whether or not a Precision Time Protocol (PTP) client is used to keep the system time synchronized to a local clock.

Syntax:

Command: #GetPtpEnabled
Arguments: N/A

Example:

Command: #GetPtpEnabled
Arguments: N/A

Message:

PTP server is currently enabled.

Content:

Length – 4 bytes (u32)
Content – 0x 01 00 00 00

#UpdateSystem

Description:

Update the system firmware to a specific version. A power cycle is required for the changes to take effect.

Syntax:

Command: #UpdateSystem



Example: Arguments: [Contents of firmware file]
Command: #UpdateSystem
Arguments: [Contents of firmware file]
Message: The system has been updated. Please reboot.
Content: N/A

#ExportSensors

Description: Export the sensors. The message contains readable descriptions of each sensor. This feature is currently being developed.
Syntax: Command: #ExportSensors
Arguments: N/A
Example: Command: #ExportSensors
Arguments: N/A
Message: N/A
Content: N/A

#SaveSensors

Description: Save the defined sensors. This feature is currently being developed.
Syntax: Command: #SaveSensors
Arguments: N/A
Example: Command: #SaveSensors
Arguments: N/A
Message: The sensors have been successfully saved.
Content: N/A

#GetSensorNames

Description: Get the names of the sensors. The order of the names corresponds to the order of the data values on the streaming sensor data port. The names are space delimited in both the message and the content fields of the response. This feature is currently being developed.
Syntax: Command: #GetSensorNames
Arguments: N/A
Example: Command: #GetSensorNames
Arguments: N/A
Message: N/A
Content: N/A

#AddSensor

Description:

Add a new sensor. The first field is the name, the second is the model (i.e. os7510), the third is the sensor channel number, the fourth is distance (in meters) and any additional parameters are sensor specific (i.e. the wavelength band for the os7510). This feature is currently being developed.

Syntax:

Command: #AddSensor

Arguments: N/A

Example:

Command: #AddSensor

Arguments: N/A

Message:

N/A

Content:

N/A

#RemoveSensor**Description:**

Remove a new sensor by name. This feature is currently being developed.

Syntax:

Command: #RemoveSensor

Arguments: N/A

Example:

Command: #RemoveSensor

Arguments: N/A

Message:

N/A

Content:

N/A

#GetSensorValueByName**Description:**

Get the current value of a sensor by name. This feature is currently being developed.

Syntax:

Command: #GetSensorValueByName

Arguments: N/A

Example:

Command: #GetSensorValueByName

Arguments: N/A

Message:

N/A

Content:

N/A

6.4. x55 APIs and Examples

The latest Micron Optics HYPERION API is available for download at the product pages at www.micronoptics.com.

6.4.1. x55 LabVIEW API

Micron Optics offers an API written in LabVIEW available at www.micronoptics.com. Usage of the API requires a license for LabVIEW Development Suite 2014 or later.

As with all of the provided API's, the complete command set is described in Section 6.3.

The download package contains the API plus a number of simple examples that illustrate usage of the API.



Each LabVIEW vi is self-documenting and information about the input and output arguments can be found in the context help.

6.4.2. six55 TCP/IP Python

Micron Optics offers an API written in Python available at www.micronoptics.com. The python API was written for python 2.7.

As with all of the provided API's, the complete command set is described in Section 6.3.

The download package contains the API plus a number of simple examples that illustrate usage of the API. These examples are a great starting point for gaining familiarity with the API.

The python API (hyperion.py) is self-documenting and information about the input and output arguments can be found in the comments.

To install and use the python API, at the command prompt, navigate to the folder that contains 'setup.py' and run:

```
sudo python setup.py install
```

This will run and install the API to your system.

With the API installed all of the simple examples can be run via command line, e.g.:

```
python simpleconsole.py
```

Note some of the python scripts require changing a hardcoded IP address to run without error. Some of the examples utilized **matplotlib**, **scipy**, and **numpy** libraries for additional processing and plotting functionality. Please see these the installation instructions for these packages at their respective websites. For distributions with pre-compiled binaries, these can often be installed with the following commands:

```
sudo apt-get install python-numpy  
sudo apt-get install python-scipy  
sudo apt-get install python-matplotlib
```

6.4.3. x55 TCP/IP MATLAB

Micron Optics offers an API written in MATLAB available at www.micronoptics.com. Usage of the API requires a license for MATLAB license and the MathWorks Instrument Control Toolbox for TCP/IP communication.

As with all of the provided API's, the complete command set is described in Section 6.3.

The download package contains the API plus a number of simple examples that illustrate usage of the API. To use the API, add the main folder and its subfolders the working path within MATLAB.

Each MATLAB file is self-documenting and information about the input and output arguments can be found in the context help.

6.4.4. x55 TCP/IP C#

A C# .NET library has been posted to the NuGet package repository and can be found within Visual Studio or at the link below by searching for either "Micron Optics" or "HYPERION".

<https://www.nuget.org>





7. Deployment Class Variant Functions

7.1. “si”s – Sensing Instruments

As stated previously in this manual, the majority of HYPERION Interrogators will be of the Instrument deployment class. That implies that independent of the specifics of the measurement core capabilities, there will be commonality in many of the system level functions, including things like time base, data logging, network management, and more. This section will endeavor to outline the specifics and mechanics of those elements that are common to all HYPERION Instruments.

7.1.1. Measurement Time-bases

The measurement time base for HYPERION instrumentation will evolve as the platform matures.

7.1.1.1. Instrument Timestamp

To first order, as the initial HYPERION deployments come in the form of the 16 channel si255 instrument, simultaneity of measurements are assured by the simultaneous, parallel sampling of all sensors on all 16 parallel channels. All measurements taken within a single acquisition of the si255 will share a common timestamp.

Further synchronization of these measurements to other HYPERION interrogators or external instrumentation will most likely be implemented by one or more of the following two networked time protocols.

7.1.1.2. Network Time Protocol (NTP)

Network Time Protocol, or NTP, will soon be implemented on the HYPERION Sensing Instrument Class deployments, including the si155 and si255. Depending on network architecture and availability of a local NTP server, timestamp synchronization between NTP networked instrumentation can approach 1 ms.

7.1.1.3. Precision Time Protocol (PTP, IEEE 1588)

A longer term, more accurate timing/synchronization standard for the HYPERION platform may be IEEE 1588, “Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.” For questions or to convey interest in an IEEE 1588 synchronization solution on the HYPERION platform, please contact Micron Optics.

7.1.2. IP Network and Configuration Management

The six55 instruments are equipped with a front mounted LCD screen and CTRL (formerly labeled IP ADDR RESET) button that can assist in assigning and/or determining the IP settings for the unit. Both Static and Dynamic Host Configuration Protocols are supported by the instrument. See section 7.1.4.1 of this document for details.

Once the existing state of the IP settings are gleaned from the LCD screen, any of the supported APIs (see Section 6.4 of this user guide for details) or the HYPERION IP Network Configuration Tool can be used to change those settings.

1a) HYPERION Tools can be downloaded from:

<http://www.micronoptics.com/products/sensing-solutions/support/> under the si255 HYPERION subheading.

1b) Tools are distributed as LabVIEW 2014 executables and require that the 32 bit National Instruments 2014 LabVIEW Runtime engine be installed, which is available for free download at:

<http://www.ni.com/download/labview-run-time-engine-2014/4887/en/>

2) Upon extraction, open the HYPERION Tools folder and locate the Network and Configuration Management directory.

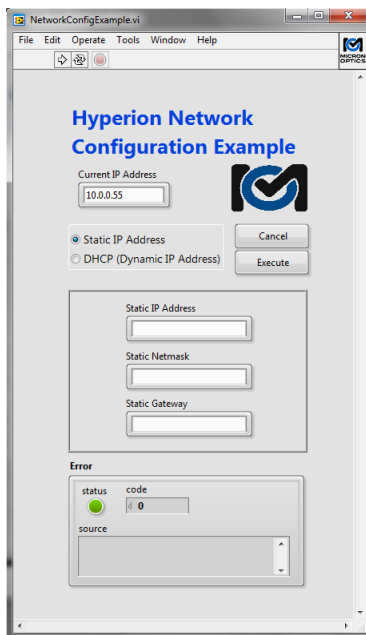
Firmware and FPGA Updates	10/15/2014 9:08 AM	File folder
Network and Configuration Management	10/15/2014 9:08 AM	File folder
si255 Research Tools	10/13/2014 11:53 AM	File folder

3) Contained in that directory are two directories that house a simple NetworkConfigExample tool and a SimpleConsoleExample tool. Browse to the NetworkConfigExample folder.

4) In that folder is located a NetworkConfigExample.exe program. Open NetworkConfigExample.exe.

NetworkConfigExample.aliases	8/12/2014 10:19 AM	ALIASES File	1 KB
NetworkConfigExample.exe	8/12/2014 10:19 AM	Application	512 KB
NetworkConfigExample.ini	8/12/2014 10:19 AM	Configuration sett...	1 KB

Launching this program will yield an interface window as seen below:



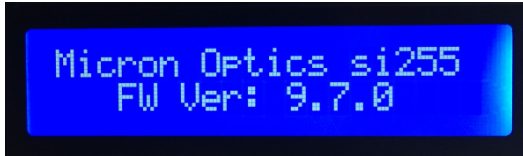
To change the network Settings on the module, the user would enter the target HYPERION's current IP address into the **Current IP Address** control and click run. Then by making selections among **Static IP Address/DHCP (Dynamic IP Address)**, setting the pertinent specific parameters, and clicking the **Execute** button on the UI, the HYPERION Instrument will be updated accordingly. The updated settings will then be reflected on the panel mounted LCD screen of the unit.



7.1.3. Firmware and FPGA Updates

7.1.3.1. Scope

This tool and process can be used to upgrade any x55 HYPERION unit with firmware of version 8.x or greater. The firmware version can be viewed on the LCD screen of the unit by cycling through screens using the front panel button.



If you are not sure of the FW version of your instrument, please contact Micron Optics for assistance.

NOTE: it is critically important to allow Firmware and FPGA updates to complete before turning off a unit. Un-powering a unit during a firmware or FPGA update may render the unit inoperable and require return to the manufacturer for repair.

7.1.3.2. Required Resources

1a) x55 HYPERION Tools.zip can be downloaded from:

<http://www.micronoptics.com/products/sensing-solutions/support/> under the si255 HYPERION subheading.

1b) Tools are distributed as LabVIEW 2014 executables and require that the 32 bit National Instruments 2014 LabVIEW Runtime engine be installed, which is available for free download at:

<http://www.ni.com/download/labview-run-time-engine-2014/4887/en/>

2) Upon extraction, open the HYPERION Tools folder and locate the Firmware and FPGA Updates directory.

Name	Type	Compressed size	Password ...	Size
firmware and fpga updates	File folder			
initialize presets	File folder			
network and configuration manag...	File folder			

3) Contained in that directory are si155 and si255 the Firmware/FPGA image files, as well as a directory that houses Firmware and FPGA Updater tool. Browse to the Micron Optics FW and FPGA Upgrade Tool folder.

Name	Date modified	Type	Size
fw and fpga upgrade tool	12/18/2015 8:24 AM	File folder	
R10.6.11.hyperion.si155.upg	12/18/2015 9:19 AM	UPG File	16,387 KB
R10.6.11.hyperion.si255.upg	12/18/2015 9:16 AM	UPG File	16,740 KB
REAMDE - IMPORTANT.txt	12/18/2015 9:24 AM	Text Document	1 KB

NOTE: there are two different firmware files for si155 and si255 instruments. The same firmware upgrade tool is used for both instruments, and it is important that the user select the correct corresponding *.upg file. Selecting the wrong file will render the instrument temporarily nonfunctional until such time that the FW upgrade process is re-run using the correct file.

4) In that folder is located a Micron optics x55 Upgrade.zip file. Extract the contents and be ready to run the HYPERIONUpgrade.exe.

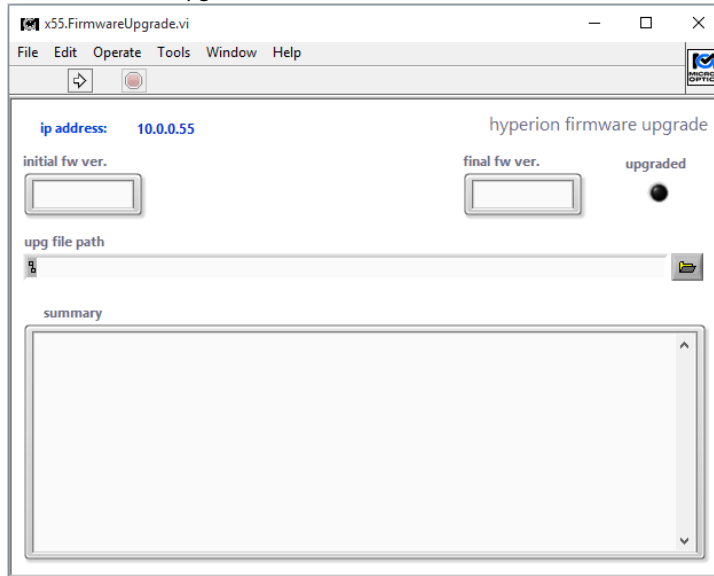


Name	Type	Compressed size	Password ...	Size
ipaddress.moi	MOI File	1 KB	No	
x55.FirmwareUpgrade.aliases	ALIASES File	1 KB	No	
x55.FirmwareUpgrade.exe	Application	1,012 KB	No	
x55.FirmwareUpgrade.ini	Configuration settings	1 KB	No	

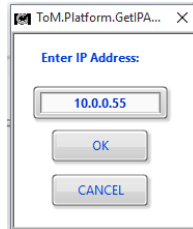


7.1.3.3. IP Address Management

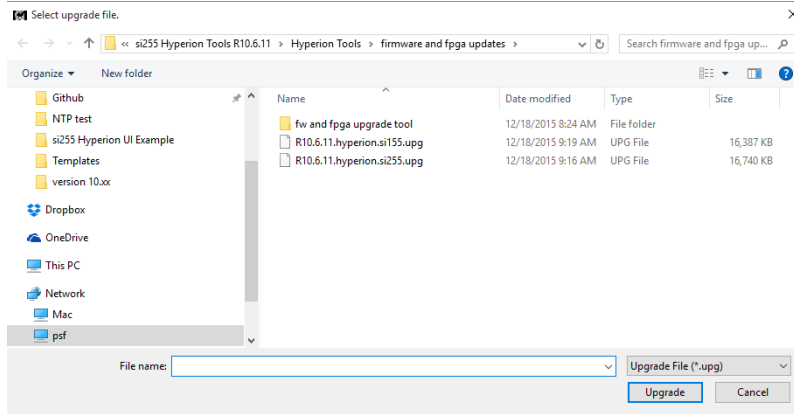
1) Run x55.FirmwareUpgrade.exe



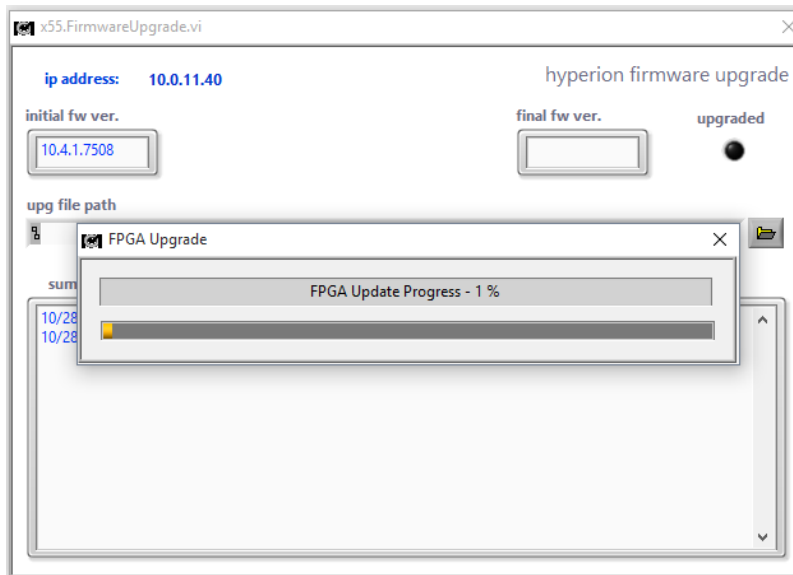
2) and Enter IP address when polled.



3) The program will then ask you to point to *.upg (upgrade) file. The default directory should house the *.upg file included with the upgrade tool. Click on the correct file and then click on the Upgrade button. Again, please use caution to choose the correct file, either si155 or si255.

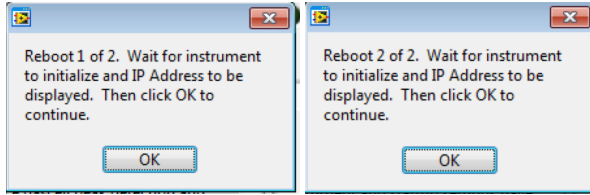


4) The utility will then transfer the contents of the (*.upg) file to the instrument, showing a progress bar like below to indicate the progress of the upgrade process. DO NOT power down the unit during this process.





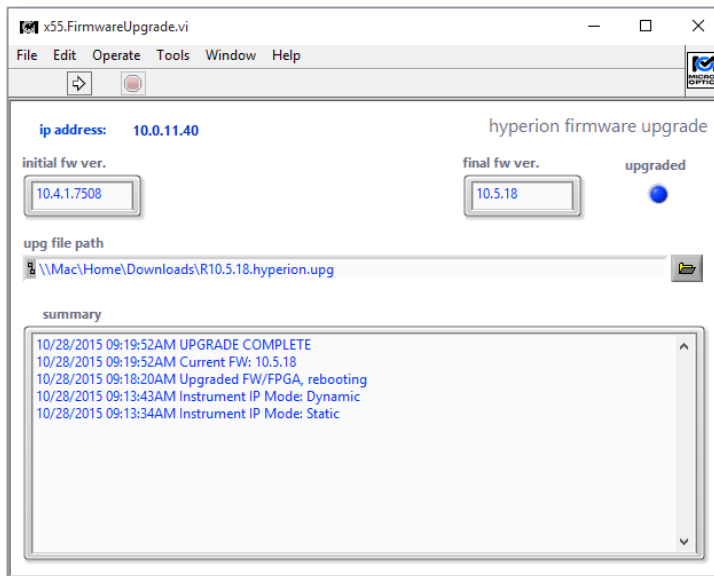
The unit will need to be rebooted once or twice during the process (depending upon the starting firmware version). The user will follow the onscreen instructions to complete the update



NOTE: if upgrading from a version 8 firmware to version 9 or 10, the module will go through two upgrade phases, each requiring a reboot. If the module should happen to be set for DHCP, please watch carefully after each reboot cycle to make sure that the IP address of the module and of the upgrade program are set to the same values. It has been observed in certain circumstances that the HYPERION unit may pull a different IP address after the first reboot, requiring that the user change the Upgrade Tool's settings accordingly.



Upon successful completion of the file transfers and setting on the si255, the Upgrade program will inform the user of its success.





7.1.4. Front Panel LCD Screen and Other Diagnostics

7.1.4.1. Front Panel Tools and Diagnostics

Initially, this front panel mounted LCD screen is used to display the IP address and Static/Dynamic status. In the future, this display, in conjunction with the panel mounted CTRL button will be used to display a variety of diagnostic and/or informational messages.

In combination with the CTRL button on the right of the front panel, the LCD screen can display the following.

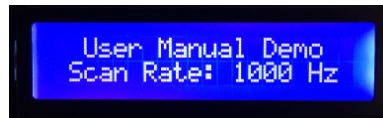
7.1.4.1.1 Information

A quick press of the CTRL button causes the LCD to toggle through multiple information screens, including such data as Firmware/FPGA versions, instrument serial number, instrument name, and laser scan rate. Two of these screens are shown below.

LCD screen showing the instrument serial number



LCD screen showing the laser scan rate



7.1.4.1.2 IP Settings

Once a button short press has landed the LCD on the IP address display, a 5 second long press (and release) will cause the instrument to switch from its assigned static IP address to DHCP mode, at which time it will attempt to pull an IP address from a connected DHCP server. A second 5 second long press (and release) will put the instrument back to its assigned static IP settings.



When on the IP settings screen at 15 second long press will cause the instrument to return to its factory default IP settings of address: 10.0.0.55, netmask 255.255.255.0.

8. FORM FACTOR SPECIFICS

NOTE: Please ensure that you reference the proper documentation for your particular module or instrument, as the required connections vary between models.

8.1. Field Modules, Small Form Factor

8.1.1. si155

8.1.1.1. Compliance

Testing for this product is underway. Successful completion of all testing will allow for the following claims:

This product complies with Part 15 of the FCC Rules. Operation is subject to the following two considerations: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This product complies with Emissions Standard EN61326-1, Subpart B Rules and Regulations Class A limits, EN61000-3-2 and EN61000-3-3.

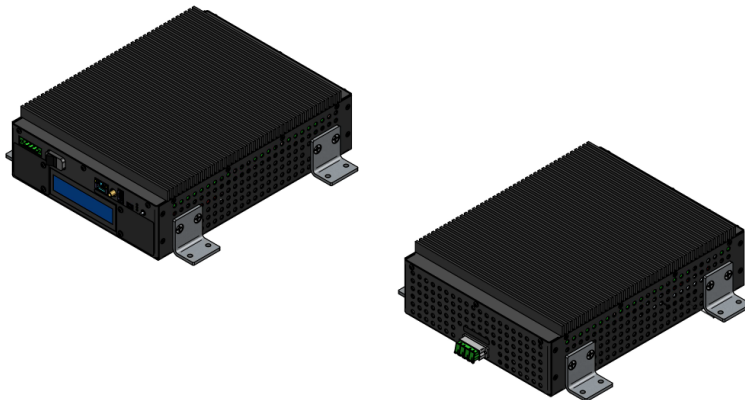
This product complies with Immunity Standard EN61326-1 (2000).

This product complies with Safety Standard EN61010-1.

This product complies with 21 CFR Subchapter J.

This product complies with applicable European Normatives and is CE compliant.

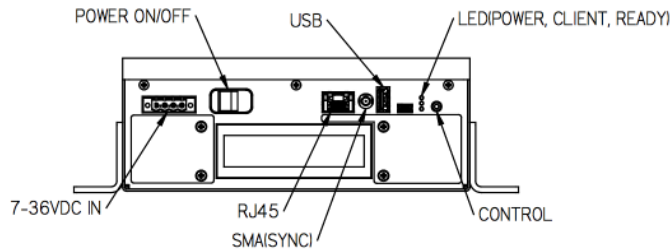
The laser safety WARNING labels are fixed on the module in the following locations (to be added).



8.1.1.2. Mechanicals

A. Ports and Indicators

i. Front View of the si155 Instrument



POWER ON/OFF SWITCH: Controls the supply of DC power to the si155 instrument.

(Not Labeled in diagram) **LCD:** Used to display the IP address and Static/Dynamic status and other status information.

CTRL: Used to toggle between LCD information screens and change IP settings.

USB: Manufacturer use only.

7 – 36 VDC IN: The front panel mounted connector is a four (4) terminal, removable terminal block that is intended for fixed wiring applications. Therefore, it is extremely important that the power be applied in the proper sequence:

With the switch in the OFF position and the power cord disconnected from power source...

1. Connect the power supply leads to the terminal block.
2. Plug the power connector block into the front panel of the si155.
3. Tighten the two (2) retaining screws.
4. Connect power cord to power source.
5. Move the power switch to the ON position.

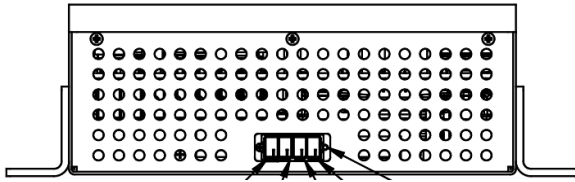
The si155 can operate from any DC power supply that outputs 7 - 36V and 80 W of power.

RJ45: This LAN connector is used for the Ethernet communication port.

SMA (SYNC): SMA connector used to output sync signal, useful for synchronization or triggering external data acquisition equipment. The square wave sync signal matches the laser scan speed setting of the instrument.

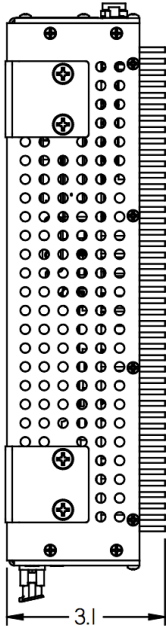
LED (POWER, CLIENT, READY): Currently, the red (POWER) LED indicates that power has been correctly supplied to the instrument. The amber (CLIENT) LED indicates that the firmware has successfully initialized. The green (READY) LED indicates that the internal laser is locked and under closed loop control.

ii. Rear View of the si155 Instrument

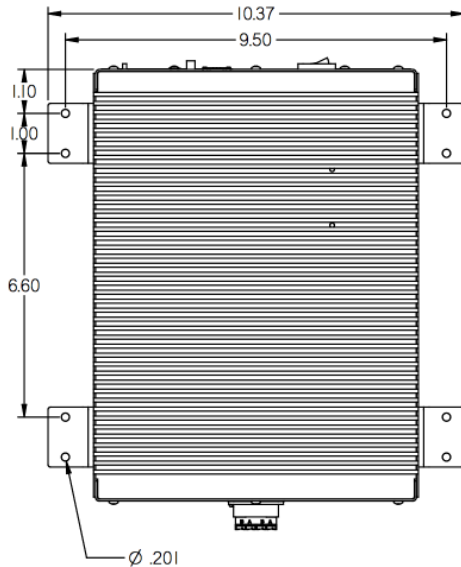


CH1 (-CH4) LC/APC: As many as sixteen optical connectors can be found on the rear panel of the si155 module. Use only LC/APC jumpers to connect to the front panel LC/APC connectors.

iii. Side View of the si255 Module



iv. Top view of the si155 Instrument





8.2. Field Modules, Large Form Factor

8.2.1. si255

8.2.1.1. Compliance

Testing for this product is underway. Successful completion of all testing will allow for the following claims:

This product complies with Part 15 of the FCC Rules. Operation is subject to the following two considerations: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This product complies with Emissions Standard EN61326-1, Subpart B Rules and Regulations Class A limits, EN61000-3-2 and EN61000-3-3.

This product complies with Immunity Standard EN61326-1 (2000).

This product complies with Safety Standard EN61010-1.

This product complies with 21 CFR Subchapter J.

This product complies with applicable European Normatives and is CE compliant.

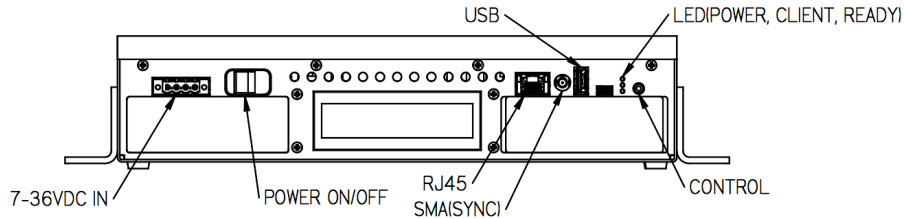
The laser safety WARNING labels are fixed on the module in the following locations.



8.2.1.2. Mechanical

8.2.1.2.1 Ports and Indicators

iv. Front View of si255 Instrument



POWER ON/OFF SWITCH: Controls the supply of DC power to the si255 instrument.

(Not Labeled in diagram) **LCD:** Used to display the IP address and Static/Dynamic status and other status information.

CTRL: Used to toggle between LCD information screens and change IP settings.

USB: Manufacturer use only.

7 – 36 VDC IN: The front panel mounted connector is a four (4) terminal, removable terminal block that is intended for fixed wiring applications. Therefore, it is extremely important that the power be applied in the proper sequence:

With the switch in the OFF position and the power cord disconnected from power source...

6. Connect the power supply leads to the terminal block.
7. Plug the power connector block into the front panel of the si255.
8. Tighten the two (2) retaining screws.
9. Connect power cord to power source.
10. Move the power switch to the ON position.

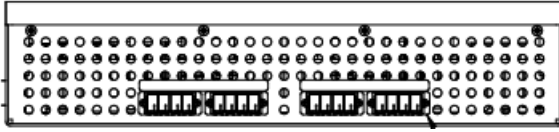
The si255 can operate from any DC power supply that outputs 7 - 36V and 80 W of power.

RJ45: This LAN connector is used for the Ethernet communication port.

SMA (SYNC): SMA connector used to output sync signal, useful for synchronization or triggering external data acquisition equipment. The square wave sync signal matches the laser scan speed setting of the instrument.

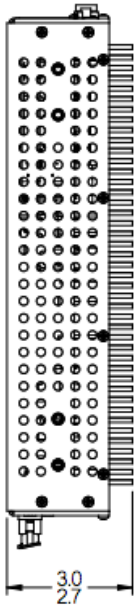
LED (POWER, CLIENT, READY): Currently, the red (POWER) LED indicates that power has been correctly supplied to the instrument. The amber (CLIENT) LED indicates that the firmware has successfully initialized. The green (READY) LED indicates that the internal laser is locked and under closed loop control.

v. Rear View of the si255 Instrument

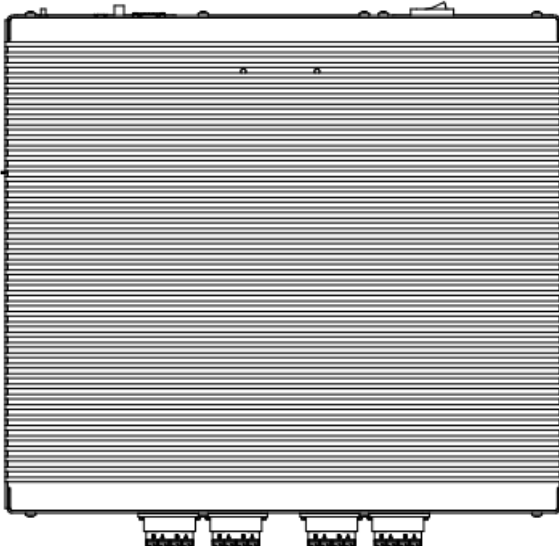


CH1 (-CH16) LC/APC: As many as sixteen optical connectors can be found on the rear panel of the si255 module. Use only LC/APC jumpers to connect to the front panel LC/APC connectors.

vi. Side View of the si255 Module



iv. Top view of the si255 Instrument





9. PLATFORM SPECIFICS

9.1. sm Sensing Modules

9.1.1. Module IP Address Management

On the module platform, the user communicates directly with the x25 or x30 optical core. In order to communicate, the user's host PC must be on the same logical network with the smXXX module.

9.1.1.1. Host PC and module are not on the same logical network.

If the host PC is not on the same logical network or the IP settings for the module are unknown, the module configuration utility can be used, as is explained in the following section. The user will need to have a VGA monitor and USB keyboard available.

9.1.1.2. Host PC and module are already on the same logical network.

If the host PC and the module are already on the same logical network, one of four methods can be used to set the IP address.

1. The configuration utility can still be used, provided that the user has a VGA monitor and USB keyboard available.
2. The web interface command console, as shown in section 5.1.3 of this document can be used to send commands directly to the x25 or x30 interrogator core. The appropriate IP settings commands are documented in sections 4.1.2.6 and 4.2.2.5 for the x25 and x30 cores, respectively.
3. The same commands referenced in option B can be sent to the module from any client developed command utility.
4. ENLIGHT can be used to connect to the module and set the IP address of that module by using the Edit IP Parameters pop-up window under the Hardware section on the Settings tab. See section 7.3.9 for details.

9.1.2. Module Keyboard/VGA Configuration Utilities

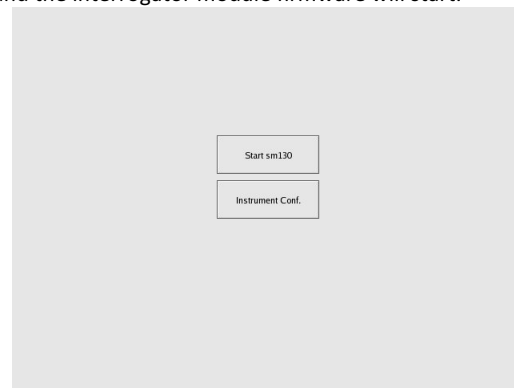
The smXXX modules are equipped with a front mounted VGA connection that offer a direct video output from the x25 or x30 core. Connection of a VGA monitor and external USB keyboard will allow a user to perform a small set of configuration tasks for the modules. The following three figures illustrate initialization menu, configuration window, and basic on-board GUI of the sm125 and sm130 modules.

NOTE: an external monitor and keyboard are not required for use of the sm125 and are suggested as a configuration convenience only.

Following a series of boot up messages and self-test indicators, an initializing interrogator module will display an image on screen as is seen in two images below. If a keyboard is not attached or a selection is not made, the default choice labeled "Start" or "Start sm130" will automatically be selected, and the interrogator module firmware will start.

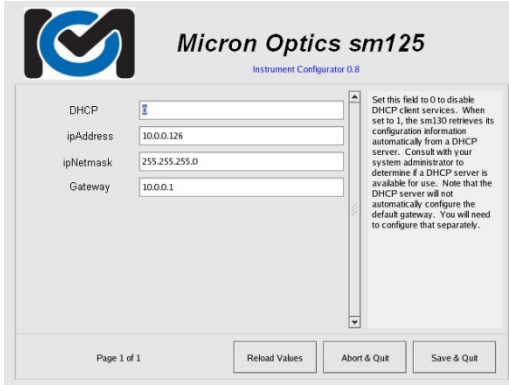


sm125 Initialization Window

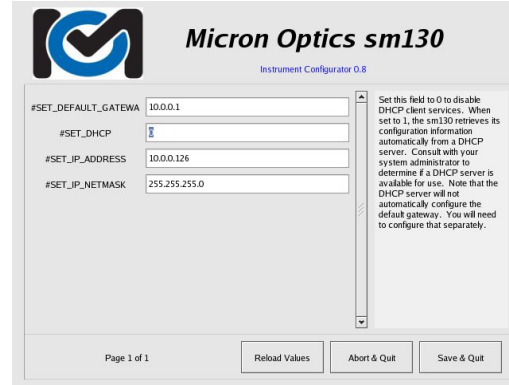


sm130 Initialization Window

If the "Instrument Conf." option is selected via the USB keyboard during initialization, the Instrument Configurator utility will launch, as shown in the following two figures. Navigating the arrow keys of the keyboard will allow the user to change the network configuration settings and save them.



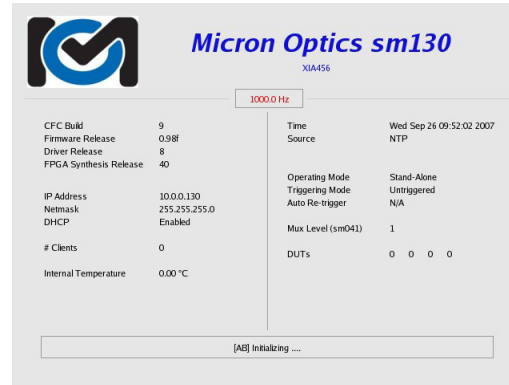
sm125 Configuration Utility



sm130 Configuration Utility



sm125 VGA output screen



sm130 VGA output screen

Once the main sm125 firmware is started, the basic screen seen above left will be displayed. The following information is displayed on this screen:

Module type:	e.g. sm125v2
Firmware Revision (F/W Rev):	e.g. 1.5
Compact Flash Build ([CFC...]):	e.g. 11
Module Serial Number (S/N):	e.g. SIA388
Ethernet status:	displays use of static IP or DHCP (e.g. eth0 [DHCP]; or eth0:)
IP Address:	e.g. 10.0.0.167
IP Netmask:	e.g. 255.255.255.0
IP Default Gateway:	e.g. 10.0.0.1

Once the sm130 firmware is started, the image above right will be displayed. The following information is displayed on the screen:

Module type:	e.g. sm130
Module serial number:	e.g. XIA456
CFC Build:	e.g. 9
Firmware Release:	e.g. 0.98f
Driver Release:	e.g. 8
FPGA Synthesis Release:	e.g. 40
IP Address:	e.g. 10.0.0.130



Netmask:	e.g. 10.0.0.130
DHCP:	displays whether enabled or disabled
# Clients:	displays number of current client socket connections (e.g. 0)
Internal Temperature:	e.g. 0.0 °C
Time:	displays the current internal system time (e.g. Wed Sept 26 09:52:02 2007)
Operating Mode:	displays current synchronization mode (e.g. Stand-Alone)
Triggering Mode:	displays current triggering mode (e.g. Untriggered)
Auto Re-trigger:	displays whether configured to automatically retrigger, if in a triggered mode (e.g. N/A)
Mux Level (sm041):	displays the present mux level of unit, as pertains to external switch multiplexing: 1 for 1-4 channels, 2 for 8 channels, 4 for 16 channels
DUTs:	displays the number of detected FBG DUTs on each channel, in the following sequence:
	1.1 2.1 3.1 4.1
	1.2 2.2 3.2 4.2
	1.3 2.3 3.3 4.3
	1.4 2.4 3.4 4.4

9.1.3. Module Web Interface Utilities

The smXXX interrogator modules are equipped with an internal web server that can be used to perform configuration, diagnostic, and firmware upgrade tasks. These internal web pages are best viewed with Internet Explorer, Mozilla Firefox, or the Opera browser.

All sm125, sm225, sm130, and sm230 modules are equipped with Web Interface utilities. The following images highlight the pages for the sm125 and sm130 modules, but are equally applicable to the sm225 and sm230 modules as well.

To access the internal sm125 web page, simply enter the IP address of any sm125 on your local network.

9.1.3.1. Home Page



sm125 Web Interface Home Page



sm130 Web Interface Home Page

The two images above show the main home pages of the sm125 and sm130 Web interfaces. These pages link the user to available Instrument Status, Command Console, and Instrument Upgrade pages.

9.1.3.2. Module Command Console Window

The first functional page on the sm125 and sm130 Web interfaces is the Command Console. From this console, the user can send any command to the connected sm125 or sm130 module and read the reply in the large text box at the bottom. A list of user commands can be pulled from the unit by issuing the #HELP command into the command line control and pressing the "Send Command" button.



sm125 Command Console Window



sm130 Command Console Window

9.1.3.3. Module Firmware Upgrade Page



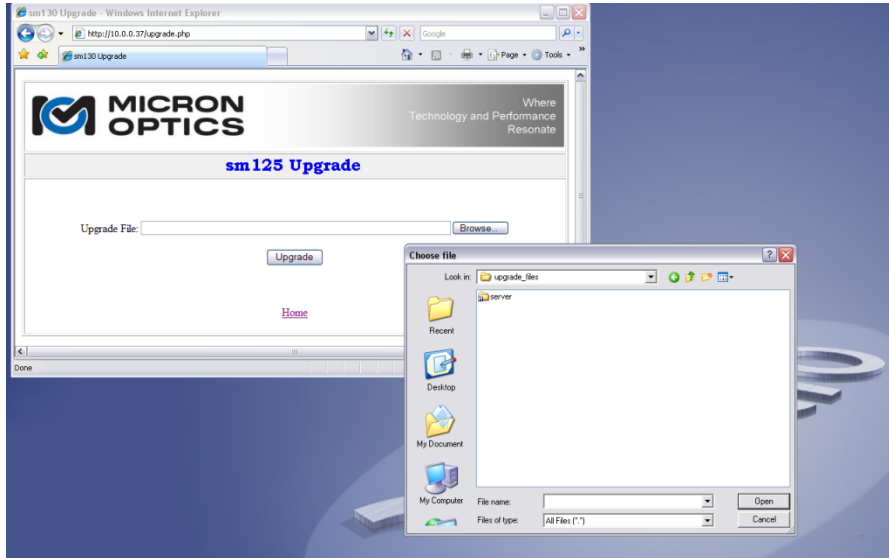
sm125 Firmware Upgrade Page



sm130 Firmware Upgrade Page

Lastly, the module web pages are used to upgrade the firmware to the connected sm125 or sm130 module. The newest upgrade files will be made available on the Micron Optics website, at www.micronoptics.com/support_downloads. The latest firmware will be packaged into a *.upg file, which contains all of the required firmware components, including CFC image, Firmware, Driver, FPGA synthesis or any other required components of the complete firmware build.

The following image shows a screenshot of the sm125 web interface in Upgrade mode. To upgrade the host sm125 module, the user will click on the Browse button to locate the latest *.upg file, and then press the Upgrade button. The web interface will advise the user to wait while the module upgrades and reboots itself. Upon rebooting, the unit will initialize and be ready for use when the green “ready” light is again illuminated.



Upgrading firmware to sm125 via Web Interface

9.2. si Sensing Instruments

This section covers the tools and actions required for management of firmware and connections to the Integrated ENLIGHT environment of the siXXX interrogation instrumentation line, which includes the si225, si230, si325, si725, and si730.

9.2.1. Instrument ENLIGHT Environment Upgrade

The latest version of ENLIGHT that is compatible with the Sensing Instrument Platform is version 1.0.38. It is recommended that all “si” sensing instruments operate this ENLIGHT version compatible with the Integrated ENLIGHT operating environment.

A zip file containing this latest “si” compatible release is available for download at http://micronoptics.com/sensing_software.php

To upgrade an siXXX instrument to this latest compatible release, download the zip file from the URL above and extract the contents to the root of a USB drive. Connect that drive to the USB port of an unpowered siXXX interrogator instrument. When power is applied to the instrument, it will recognize the ENLIGHT installation file on the root of the drive and ask the user if he wishes to upgrade. The user should confirm and the Integrated ENLIGHT environment will be upgraded. No reboot is required after upgrade. Should the ENLIGHT version include and require updated interrogator core firmware, the user will be so alerted. The process for updating interrogator core firmware through ENLIGHT is presented in section 7.3.9 of this document.

9.2.2. Instrument Integrated ENLIGHT Environment Connection Settings

9.2.2.1. Host IP Address Management

The network settings of the Integrated ENLIGHT instrument can be configured on the Settings tab of the ENLIGHT interface, as detailed in section 7.3.9 of this document. The instrument host IP address, Subnet Mask, and Default Gateway can all be manually controlled or automatically configured by DHCP.

9.2.2.2. Instrument Machine Naming Control

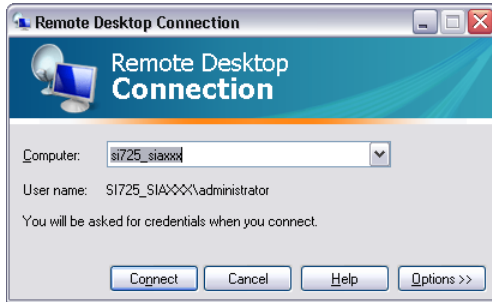
Additionally, the Integrated ENLIGHT instrument controls on the settings tab allow the user to assign a custom machine name for the interrogator instrument. The factory configured default machine name is based upon the model number and serial number of the interrogator instrument. For example, and si725 instrument with serial number SIAXXX will be named si725_SIAXXX by default.

In addition to addressing the instrument via the instrument host IP address, the unit can be addressed by machine name for Remote Desktop, Remote Data, and Ethernet Pass-through access purposes. The process for changing the instrument machine name through ENLIGHT is covered in section 7.3.9 of this document.

9.2.3. Instrument Remote Desktop Instrument Control

Remote configuration of the siXXX interrogation instrument can be facilitated by use of the Remote Desktop Protocol. Remote Desktop Protocol (RDP) is a multi-channel protocol that allows a user to connect to a networked computer . Microsoft refers to the official RDP server software as [Terminal Services](#) or Remote Desktop Services. The official client software is referred to as either [Remote Desktop Connection](#) (RDC) or Terminal Services Client (TSC). The following instructions show how to create a Remote Desktop Connection.

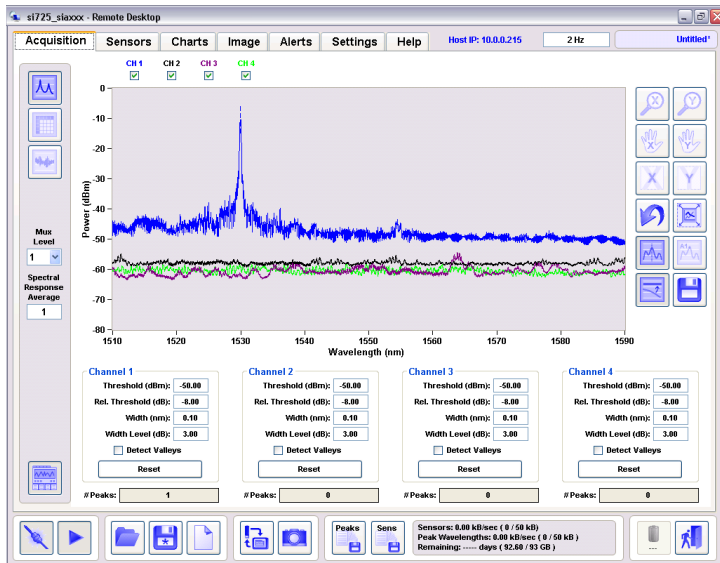
1. On the user’s computer, the user is to click Start, point to All Programs, and then point to Accessories.
2. In the Accessories menu, the user is to point to Communications, and then click Remote Desktop Connection.
3. In the Computer box shown in the following figure, the user is to type either the IP address or machine name of the target siXXX instrument.



1. The user will then be prompted for your credentials to log onto the siXXX interrogator. The correct credentials are as follows:

User name: Administrator
Password: micronoptics

Once connected, the user will see a complete image of the siXXX integrated ENLIGHT environment, as shown in the following image. All interrogator core setup, FBG and sensor creation, charting, alarming, and data saving functions can then be controlled remotely on the user's computer.



The Remote Desktop Connection can also be automated or integrated into a user's custom sensor monitoring application through the use of command line calls to Remote Desktop or the Remote Desktop Active X control.

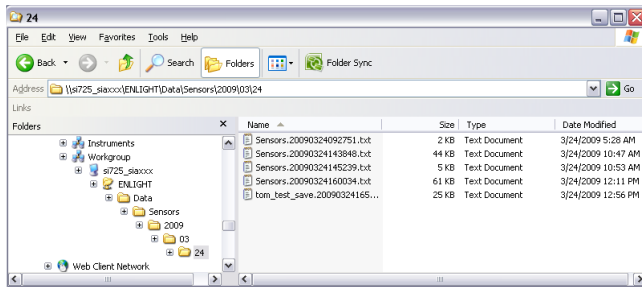
9.2.4. Instrument Local Data Storage and Remote Data Access

The Integrated ENLIGHT environment supports a host of local data storage features. These features can be accessed by direct touch screen, mouse, and/or keyboard controls or via Remote Desktop Connections, as highlighted in the previous section. The stored data can be accessed through two different methods.

The first method is to use the ENLIGHT Import/Export wizard to transfer files from the instrument internal hard drive to an external USB drive. This option will be covered in detail in section 7.2.2.2 of this manual.

In addition to direct, local access to the data, the stored files can also be remotely accessed over a network through the Shared Files feature of the Integrated ENLIGHT environment. To access the data, the connected user would simply need to browse the shared folders of the interrogator instrument.

For example, an si725 instrument with machine name si725_SIAXXX housed files that were of interest to the remote user, he could view, copy, and/or move those files using Windows Explorer or any other file browsing utility of his choice. The following image shows the listing of internally saved files for the si725 instrument as viewed by the remote networked user.



The organizational folder structure for saved data will be covered in section 7.3.7 of this document.

The shared data folders are secured and require the same login credentials used for Remote Desktop Connection:

Username: Administrator

Password: micronoptics

Remote Data Access allows users to access sensor and peak wavelength data logging files, screen capture files, full spectrum acquisition data files, image files, configuration files and firmware images. Access to these data stores can be accessed programmatically as well.

9.2.5. Ethernet Pass-through

The interrogator instrument configuration offers a suite of powerful onboard ENLIGHT tools for the user, as compared to the basic firmware of the smXXX optical interrogator module series. However, there may be circumstances where a user would prefer to communicate directly to the optical interrogator core by using the commands and protocols outlined in section 4 of this manual. An example of this type of use case could be the following.

Suppose that a user purchases an si730 Optical Sensor Interrogator Instrument for use in a development laboratory. Under most conditions, the user will prefer to employ the simple user interface and convenient data storage tools of the integrated ENLIGHT software. At some point, however, the user may find the need to collect unprocessed wavelength data from the optical interrogator core and integrate that data into his own custom data collection process, in much the same way that a user may traditionally interface an sm130 module. To facilitate this user scenario, the Integrated ENLIGHT environment includes an Ethernet Pass-through mode. When Ethernet Pass-through is enabled, the user may send commands and queries directly to the optical interrogator core of the instrument, as though it was a module configuration rather than an instrument. For example, when Pass-through is enabled, the x25 commands and protocols of sections 4.1.2.2 to 4.1.2.7 can be directly applied to collect data from the instrument's interrogator core.

In order to activate Pass-through mode, the Integrated ENLIGHT software needs to be disconnected from the interrogator core. The connect/disconnect feature is officially introduced in section 7.2.2.2 of this document, but is previewed in the following image.

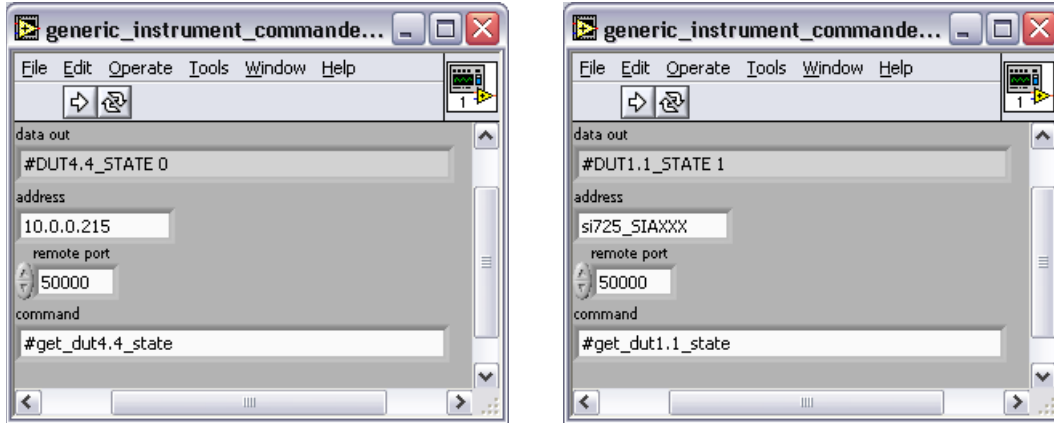
When the Connect/Disconnect button is disengaged, as shown below, Ethernet Pass-through is automatically enabled, allowing for direct Ethernet connections to the interrogator core.

Connect/Disconnect button (left) is disengaged, thereby enabling Ethernet Pass-through.





The x25 or x30 interrogator commands should be addressed to the IP address or machine name of the interrogator instrument, as defined in the Settings tab of the instrument’s ENLIGHT operating environment. The following images show a basic generic communication program passing commands directly to the interrogator core via the instrument’s host IP address (left) and machine name (right).



NOTE: Only port 1852 communications for the x30 and port 50000 communications for the x25 are supported via Ethernet Pass-through. For this reason, the web servers for the two interrogator cores are not available via Pass-through. Similarly, port 55000 internal peak detection commands are not supported for the x25 core via Pass-through, but port 50000 internal peak detection commands are.

NOTE: When an interrogator core is built into an instrument configuration it is said to be in “embedded mode”. Once in embedded mode the instrument host manages all IP settings for the core. As such, all commands to set IP address, Netmask, and default gateway of the interrogator core are ignored, replying with a placeholder reply of “0.0.0.0.” Additionally, the VGA output of the interrogator core is disabled, as all management responsibilities for that interrogator core have been relegated to the instrument host.

9.2.6. Safe Instrument Shutdown Procedures.

The Integrated ENLIGHT software should be shut down prior to turning off power to the instrument via hardware power switch. First, a soft shutdown sequence should be initiated by pressing the Exit button on the Integrated ENLIGHT Interface. Once a soft shutdown sequence is initiated, the green and amber LEDs will begin to cycle in phase with each other. As the shutdown sequence progresses, the frequency of this simultaneous flashing will increase. When the safe shutdown sequence is complete, the green and amber LEDs will both go dark and the si225 will yield a sequence of three audible beeps. At that time, it is safe to move the instrument power switch to the “off” position.

ENLIGHT exit button



CAUTION: Failure to follow safe instrument shutdown procedures may affect instrument stability or longevity. Please exit ENLIGHT and wait for the three beep sequence before setting the power switch to the off position.

9.3. sp Sensing Processors

9.3.1. Features and Options

The sp125 and sp130 Sensing Processors are industrial grade, high performance, processing module accessories for use with MOI sm125 and sm130 Optical Sensing Interrogators. The sp’s are built on fanless industrial PCs and facilitate communication and power control to the sm125 and sm130 module.



Custom software and application development are supported in both Linux and Windows XP operating systems. Maximum flexibility supports many applications.

The data processing environment enabled by the sp125 and sp130 allows users to easily bridge the gap between optical domain data and application specific engineering units like strain, temperature, pressure, and acceleration.

Data can be saved to the CompactFlash or 2.5" HDD. Raw or processed data can also be exported to other processors via Ethernet, RS-232/422/485, or USB.

Support for ACPI power management is provided. In addition, the sp units can remotely switch power on/off to the sm125 or sm130 interrogator via software.

Mechanical Details and port definitions can be found in section 6.1 of this document.

The sp125 and sp130 processors are also available as "internal processor options" for the sm225 and sm230 Rack Mount optical interrogator module configurations. Please visit www.micronoptics.com for details.

9.3.2. Support Materials

Included among the instrumentation support materials are two programmer's guides for the sp125 and sp130 Processor Modules. These guides highlight both Windows XP environment and Linux examples for programmatic control of the general purpose I/O (GPIO) ports of the Processor modules. Also included with the sp support materials are dll's and the associate documentation that support basic sm125 and sm130 communications and data retrieval. See www.micronoptics.com/support_downloads for details.

10. FORM FACTOR SPECIFICS

NOTE: Please ensure that you reference the proper documentation for your particular module or instrument, as the required connections vary between models.

10.1. Field Modules

10.1.1. sm125

10.1.1.1. Compliance

This product complies with Part 15 of the FCC Rules. Operation is subject to the following two considerations: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This product complies with Emissions Standard EN61326-1, Subpart B Rules and Regulations Class A limits, EN61000-3-2 and EN61000-3-3.

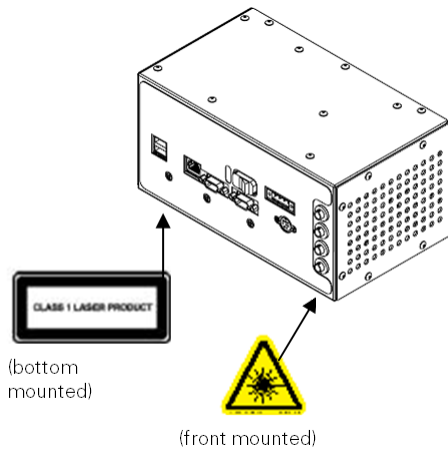
This product complies with Immunity Standard EN61326-1 (2000).

This product complies with Safety Standard EN61010-1.

This product complies with 21 CFR Subchapter J.

This product complies with applicable European Normatives and is CE compliant.

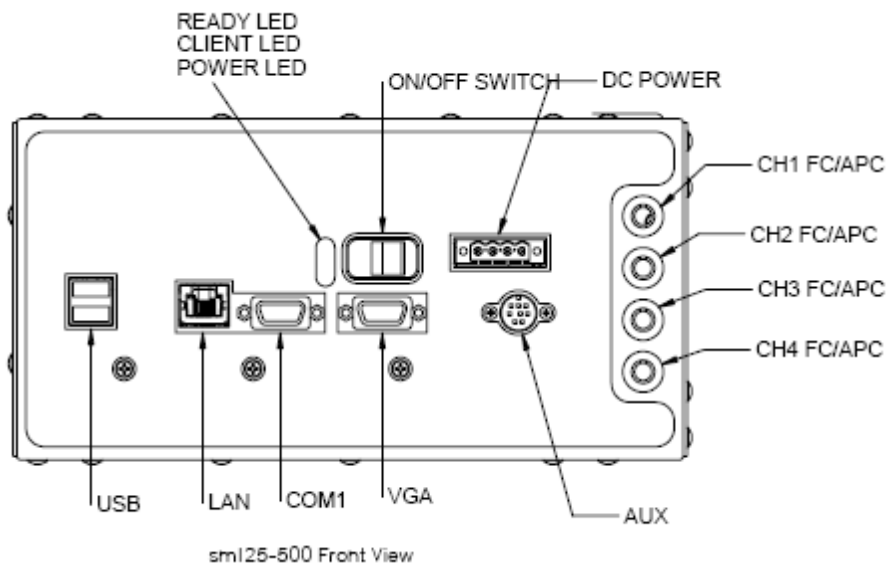
The laser safety warning labels are fixed on the module in the following locations.



10.1.1.2. Mechanical

A. Ports and Indicators

i. Front View of sm125 module



ON/OFF SWITCH: Controls the supply of DC power to the sm125 module.

VGA: This port can be used to attach an external monitor for diagnostic purposes.

COM1: Not Used

USB: May be used for an external USB keyboard for limited configuration tasks, including changes to IP settings.

DC POWER: The front panel mounted connector is a four (4) terminal, removable terminal block that is intended for fixed wiring applications. Therefore, it is extremely important that the power be applied in the proper sequence:

1. With the switch in the OFF position and the power cord disconnected from power source...
2. Connect the power supply leads to the terminal block.



3. Plug the power connector block into the front panel of the sm125.
4. Tighten the two (2) retaining screws.
5. Connect power cord to power source.
6. Move the power switch to the ON position.

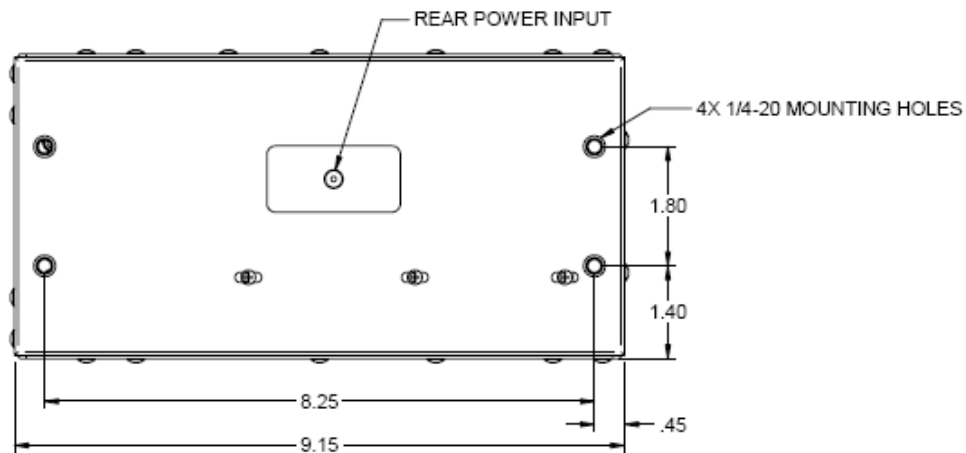
The sm125 can operate from any DC power supply that outputs 7 - 36V and 80 W of power.

LAN: This RJ-45 connector is used for the Ethernet communication port.

AUX: This port is an 8 pin DIN connector used to connect the sm125 to accessories, including the available sm041 Channel Expansion Module.

CH1 (-CH4) FC/APC: As many as four optical connectors can be found on the front panel of the sm125 module. Use only FC/APC jumpers to connect to the front panel FC/APC connectors.

ii. Rear View of the sm125 Module

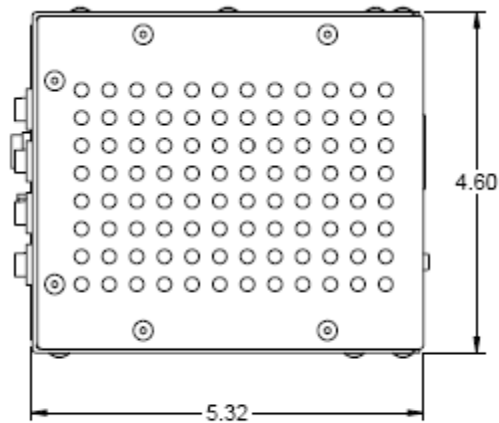


sm125-500 Rear View

REAR POWER INPUT: This rear panel mounted coaxial power connector is designed to handle power application and disconnect under all load conditions. Use this connector for all applications in which a fixed power connector is not required.

MOUNTING HOLES. These threaded holes on the rear of the sm125 allow for direct mounting of the module.

iii. Side View of the sm125 Module

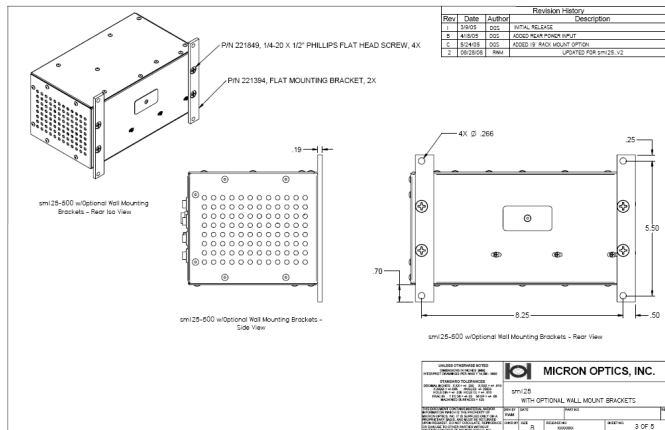


sm125-500 Side View

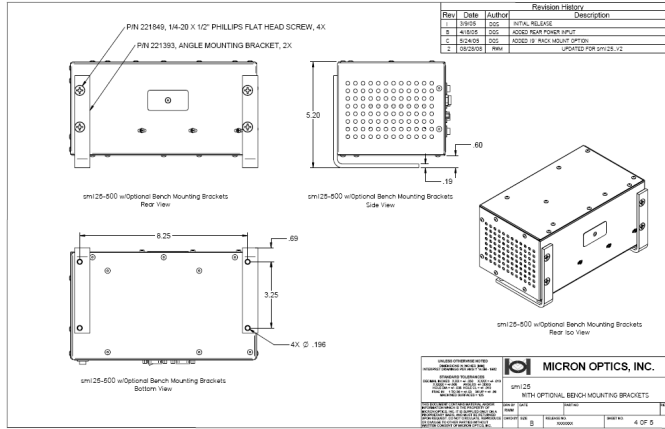
B. Other sm125 Mounting Options

There are several options for fixed mechanical mounting of the sm125 module. The following three images highlight these options.

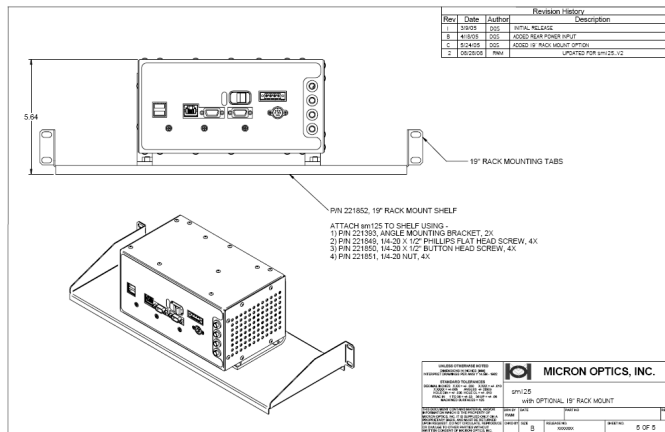
i. sm125 Mounting Option 1: Wall Mounting Bracket



ii. sm125 Mounting Option 2: Bench Mounting Bracket



iii. sm125 Mounting Option 3: Optional 19" Rack Mount



10.1.2. sm041

10.1.2.1. Compliance

This product complies with Part 15 of the FCC Rules. Operation is subject to the following two considerations: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This product complies with Emissions Standard EN61326-1, Subpart B Rules and Regulations Class A limits, EN61000-3-2 and EN61000-3-3.

This product complies with Immunity Standard EN61326-1 (2000).

This product complies with Safety Standard EN61010-1.

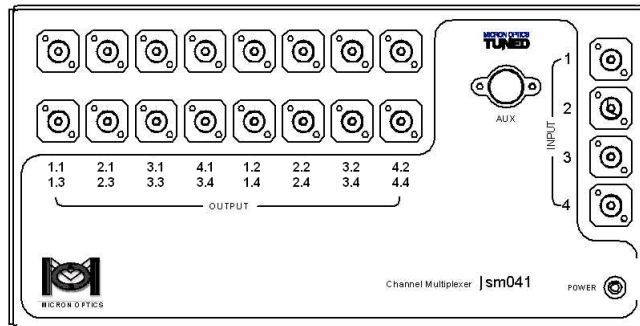
This product complies with 21 CFR Subchapter J.

This product complies with applicable European Normatives and is CE compliant.

10.1.2.2. Mechanical

A. Ports and Indicators

i. Front View of the sm041 Module



POWER LED: This red indicator confirms that power has been supplied to the module. A switch-type sm041 module (e.g. sm041-416) requires no separate power supply of its own, as all power and control for the multiplexer are supplied from the host interrogator module via the AUX port. A coupler-type sm041 module (e.g. sm041-016) is passive and requires only optical connectors. See INPUT and OUTPUT below.

AUX: This port is used to connect the sm041 module to the host interrogator module and supplies the electrical power and control signals to the multiplexer.

INPUT (1-4): These optical input FP/APC connectors are to be cabled to the four front panel optical connectors of the host module.

OUTPUT (1.1 – 4.4): These optical output FP/APC connectors are to be cabled directly to the optical sensors.

B. Optical Connections

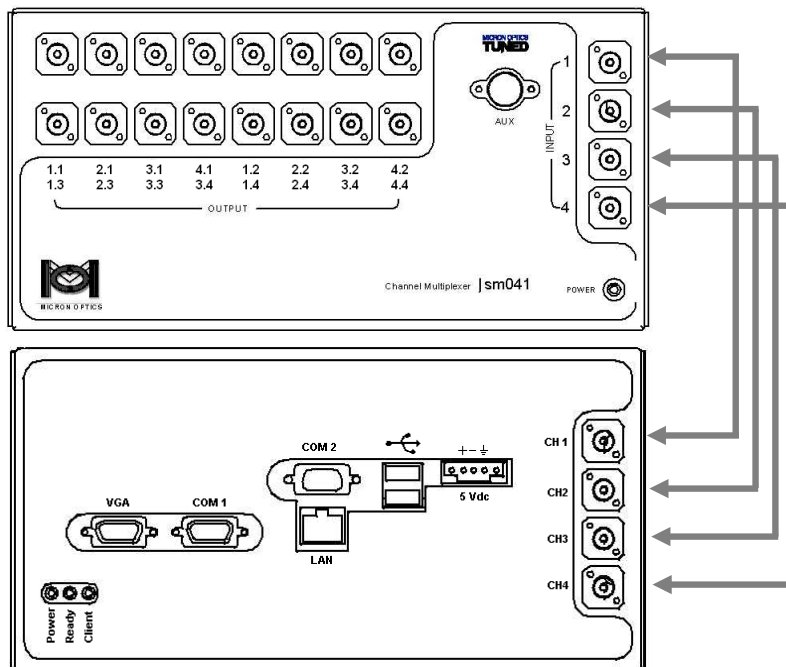


Diagram of sm041 optically connected to host instrument.

The input FC/APC optical connectors of the sm041 multiplexer module are to be connected directly to the front panel FC/APC optical connectors of the host interrogator module. The input channel number of the multiplexer module should exactly match the output channel number of the host interrogator.

10.1.3. sp125

10.1.3.1. Compliance

Testing for this product is underway. Successful completion of all testing will allow for the following claims:

“This product complies with Part 15 of the FCC Rules. Operation is subject to the following two considerations: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.”

This product complies with Emissions Standard EN61326-1, Subpart B Rules and Regulations Class A limits, EN61000-3-2 and EN61000-3-3.

This product complies with Immunity Standard EN61326-1 (2000).

This product complies with Safety Standard EN61010-1.

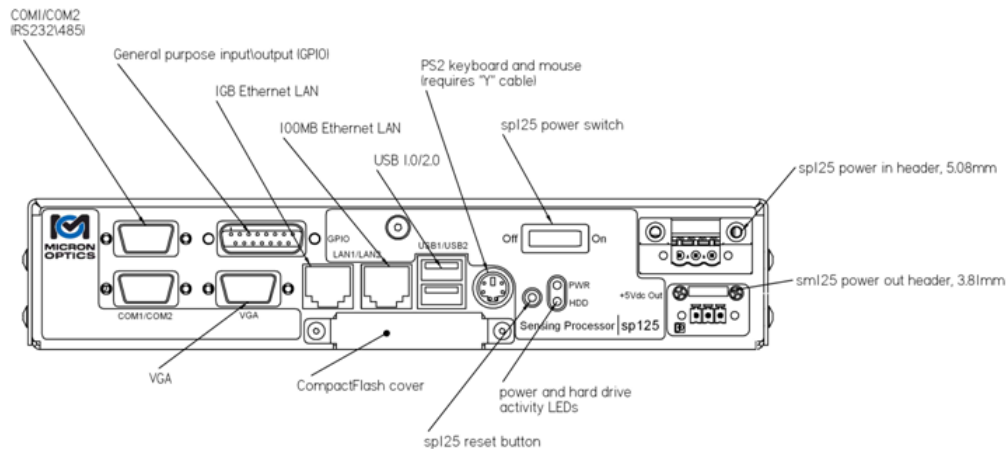
This product complies with 21 CFR Subchapter J.

This product complies with applicable European Normatives and is CE compliant.”

10.1.3.2. Mechanical

A. Ports and Indicators

i. Front Panel View of sp125 Module



VGA: Used for connection to an external VGA monitor.

COM1, RS-232: Available for other communications protocols on OEM or user developed basis.

COM2, RS-232/485: Available for other communications protocols on OEM basis.

USB1/USB2: Used for connection of any USB1.0/2.0 compatible device.

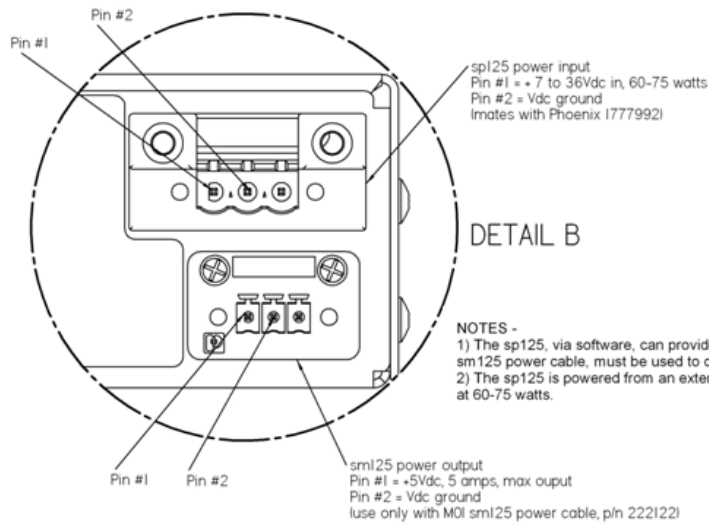
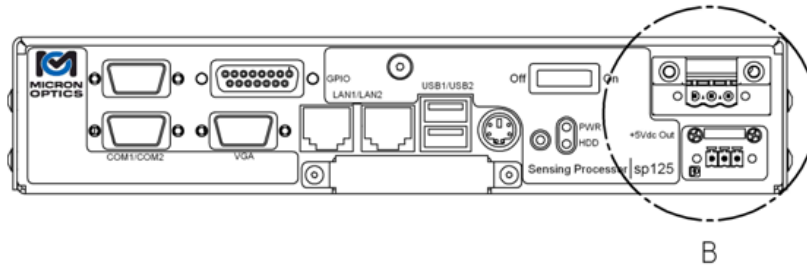
PS2: Used for connecting a PS2 keyboard and mouse. Requires a "Y" cable.

LAN1/LAN2: Used to connect sp125 to sm125 and/or external LAN. Connection to a sm125 requires a crossover cable.



GPIO: Female DB15 connector used to connect external, general purpose, input/output (GPIO) signals to sp125. Please refer to section on GPIO for more information.

ii. Power Connector Details



sp125 power input
Pin #1 = +7 to 36Vdc in, 60-75 watts
Pin #2 = Vdc ground
Imates with Phoenix I777992I

NOTES -
1) The sp125, via software, can provide power to a sm125. MOI p/n 222122. sm125 power cable, must be used to connect the sp125 to the sm125.
2) The sp125 is powered from an external source capable of supplying 7-36Vdc at 60-75 watts.

sm125 power output
Pin #1 = +5Vdc, 5 amps, max output
Pin #2 = Vdc ground
luse only with MOI sm125 power cable, p/n 222122I

iii. General Purpose Input/Output (GPIO) Wiring Diagram

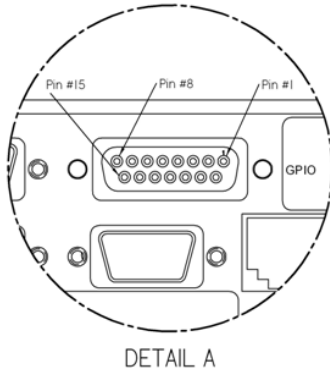
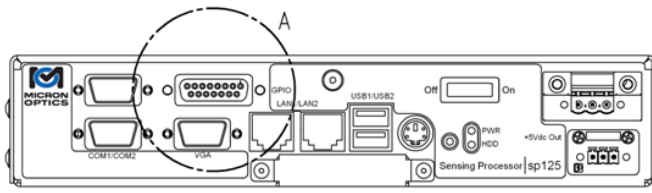
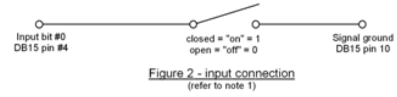
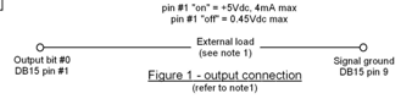


Table 1 - sp125 GPIO_DB15 female

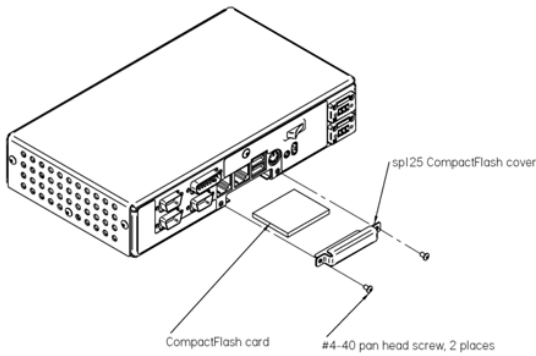
Pin	Description
1	Output Bit #0
2	Output Bit #1
3	Output Bit #2
4	Input Bit #0
5	Input Bit #1
6	Input Bit #2
7	not used
8	not used
9	signal ground
10	signal ground
11	signal ground
12	signal ground
13	signal ground
14	signal ground
15	signal ground



NOTES:

- GPIO line voltage and current ratings -
Input voltage low = -0.3 to +0.8Vdc
Input voltage high = +2.0 to +5.3Vdc
Output low(sinking) - voltage = 0.45Vdc max, current = 12mA
Output high(sourcing) - voltage = 5Vdc max, current = 4mA
(additional specifications available upon request)
- Refer to Table 1 for description of GPIO lines.
- Refer to figures 1 & 2 for examples of connecting external devices to the GPIO.
- Bit value of 0 = "off", bit value of 1 = "on".
- Any signal ground may be used with any GPIO input or output line.

iv. Compact Flash Access Port



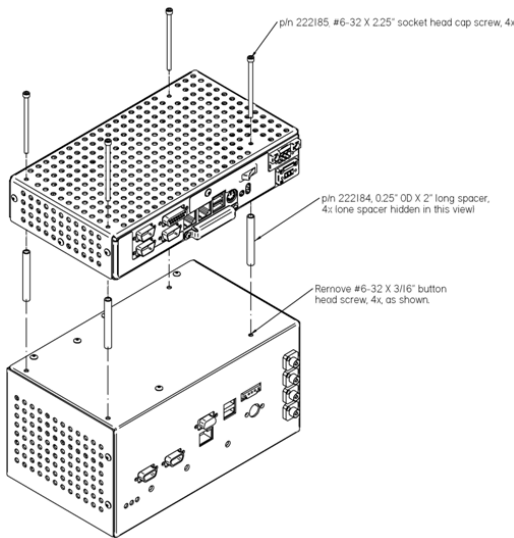
Instructions for installing CompactFlash in sp125

- 1) Remove #4-40 pan head screw, 2 places, as shown.
- 2) Remove sp125 CompactFlash cover.
- 3) Insert CompactFlash card into sp125. Please note that the CompactFlash card is keyed and will only align with the CompactFlash socket in one orientation.

CAUTION - DO NOT FORCE CompactFlash CARD. CARD SHOULD SEAT INTO SOCKET WITH GENTLE PRESSURE. IF RESISTANCE IS FELT THEN VERIFY THAT CompactFlash CARD IS NOT DAMAGED AND THAT THE CARD KEY PROPERLY ALIGNS WITH THE SOCKET KEY.

- 4) Install sp125 CompactFlash cover.

B. Attachment to sm125 Module



Instructions for attaching sp125 to sm125

1) Remove #6-32 X 3/16" button head screw from sm125 case, 4 places, as shown.

CAUTION - DO NOT REMOVE ANY OTHER SCREWS FROM sm125. THIS CAN RESULT IN DAMAGE TO THE OPTICAL COMPONENTS HOUSED WITHIN THE sm125.

2) Insert a #6-32 X 2.25" socket head cap screw through the sp125, 4 places, as shown. The screw should pass completely through the sp125 housing and protrude from the bottom of the sp125.

3) Insert the 0.25" OD X 2" long spacer into the sp125 from the bottom. The #6-32 X 2.25" screw installed in the previous step should protrude from the bottom of the sp125, and through the spacer.

4) Align the #6-32 X 2.25" screws with the holes in the sm125 and tighten, 4 places, as shown. MOI recommends applying a small amount of removable thread locking compound (Locktite 222MS) to each screw prior to installation. Screws should be torqued to a maximum of 9 inch-pounds.

CAUTION - FAILURE TO USE THE 2" SPACERS PROVIDED WITH THE sp125 MAY RESULT IN DAMAGE TO THE OPTICAL COMPONENTS HOUSED WITHIN THE sm125. FAILURE TO USE THE #6-32 X 2.25" SCREWS PROVIDED WITH THE sp125 MAY RESULT IN DAMAGE TO THE OPTICAL COMPONENTS HOUSED WITHIN THE sm125.

10.2. Rack Mount Units

Due to the possibility of multiple Ethernet, VGA and/or USB ports on the rear of the Rack Mount units, special attention should be paid to the Ports and Indicators section for these products.

CRITICAL NOTE: si225 and si230 Instruments with the Integrated ENLIGHT environment REQUIRE that a crossover cable be used in loopback configuration in order to function. Please ensure that the provided cable remains connected between the Laser Module Ethernet port and the LAN1 port of the si225 or si230 instrument.

10.2.1. sm225/si225

10.2.1.1. Compliance

Testing for this product is underway. Successful completion of all testing will allow for the following claims:

"This product complies with Part 15 of the FCC Rules. Operation is subject to the following two considerations: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This product complies with Emissions Standard EN61326-1, Subpart B Rules and Regulations Class A limits, EN61000-3-2 and EN61000-3-3.

This product complies with Immunity Standard EN61326-1 (2000).

This product complies with Safety Standard EN61010-1.

This product complies with 21 CFR Subchapter J.

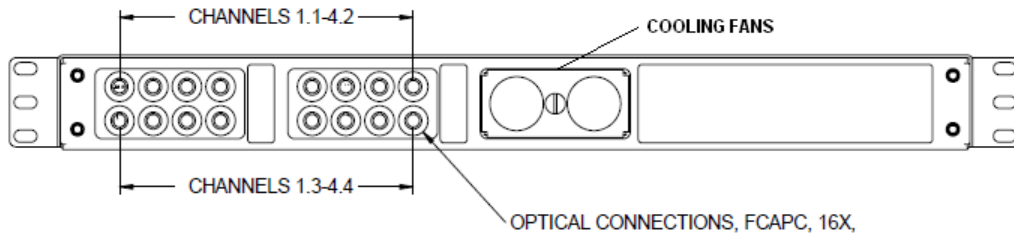
This product complies with applicable European Normatives and is CE compliant."

10.2.1.2. Mechanical



A. Ports and Indicators

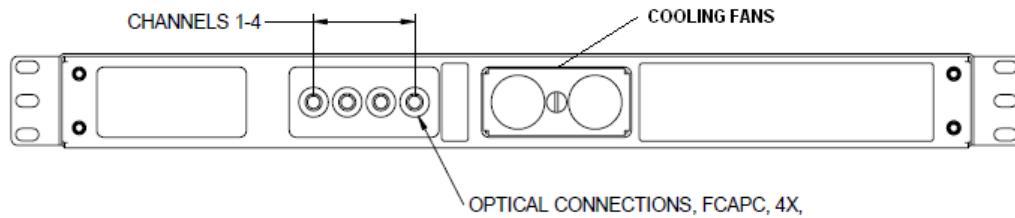
i. Front Panel View of sm225/si225, 16 channel version



CHANNELS 1.1 – 4.2, 1.3 – 4.4: FC/APC optical connectors.

COOLING FANS: Cooling fans feature user accessible dust screen for easy cleaning.

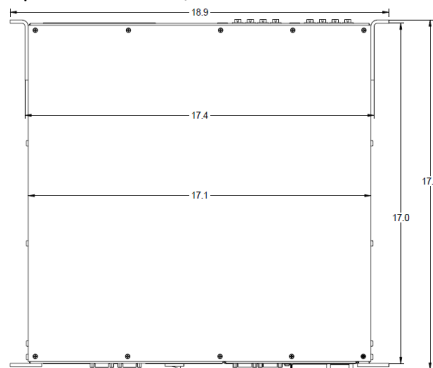
ii. Front Panel View of sm225/si225, 4 channel version



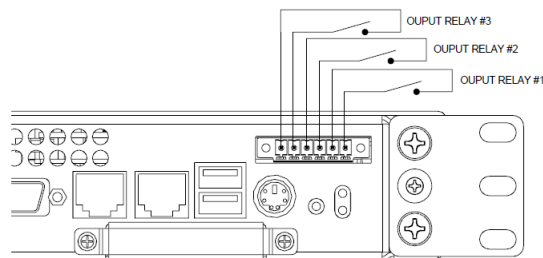
CHANNELS 1-4: FC/APC optical connectors.

COOLING FANS: Cooling fans feature user accessible dust screen for easy cleaning.

iii. Top View of sm225/si225 enclosure

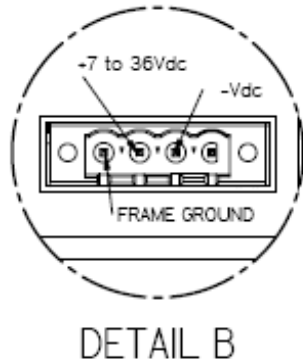


iv. sm225/si225 Relay Output

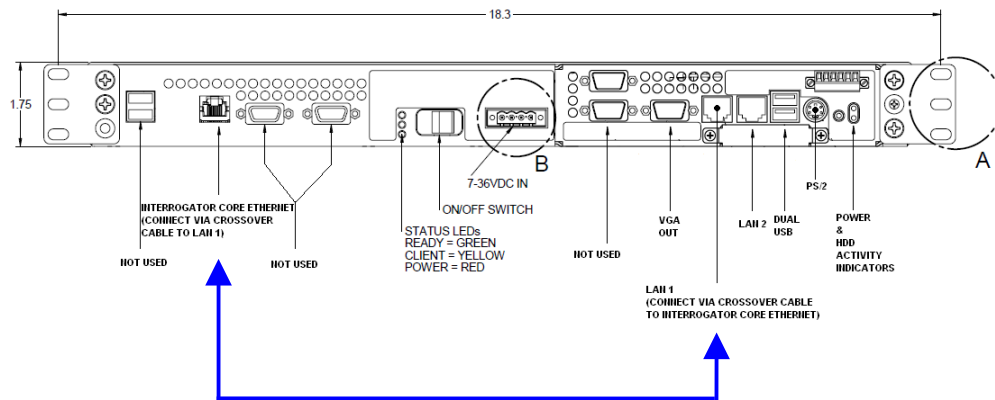




v. sm225/si225 Power Terminal Details.



vi. Rear View of si225 Instrument. Ethernet crossover loopback connector REQUIRED for operation.



CRITICAL NOTE: Ethernet crossover loopback connector REQUIRED for operation.

In the si225 instrument, there are several ports which are not used for normal operation. It is important to become familiar with the correct ports for use with the Integrated ENLIGHT environment.

INTERROGATOR CORE ETHERNET: This port to the x25 interrogator core must be connected to LAN1 of the si225 instrument via a crossover cable at all times.

POWER LED: This red indicator confirms that power has been supplied to the x25 interrogator core.

CLIENT LED: This amber indicator is illuminated when there is an open socket connection to the x25 interrogator core. This indicator will be active when either ENLIGHT is connected to the core, or a remote PC is connected directly to the interrogator core via the Ethernet Pass-through feature.

READY LED: This green indicator confirms that the initialization sequence of the interrogator core has completed and data connects to the core are possible.

NOTE: The amber client LED and the green ready LED are also used in combination to give a visible indication that the si225 instrument is in a safe shutdown sequence. Once a soft shutdown sequence is initiated via the Exit button on the Integrated ENLIGHT interface, the green and amber LEDs will begin to cycle in phase with each other. As the shutdown sequence progresses, the frequency of this simultaneous flashing will increase. When the safe shutdown sequence is complete, the green and amber LEDs will both go dark and the si225 will yield a sequence of three audible beeps. At that time, it is safe to move the instrument power switch to the “off” position.

ON/OFF SWITCH: Controls the supply of DC power to the si225 instrument.

7-36VDC IN: The front mounted connector is a four (4) terminal, removable terminal block that is intended for fixed wiring applications. Therefore, it is extremely important that the power be applied in the proper sequence:

1. With the power switch in the OFF position...
2. Connect the power supply leads to the terminal block.
3. Plug the power connector block into the front panel of the sm130.
4. Tighten the two (2) retaining screws.
5. Move the power switch to the ON position.

The si225 can operate from any DC power supply that outputs 7 - 36V and 80 W of power.

VGA OUT: This port can be used to attach an external monitor to the instrument. Used in combination with a USB keyboard and mouse, as user can locally operate to view the Integrated ENLIGHT graphical user interface.

NOTE: The si225 Instrument VGA port must be connected to a monitor device during boot up in order for the VGA drivers to be loaded by the processor. If no monitor is connected during boot up, no video signal will be available to the VGA port.

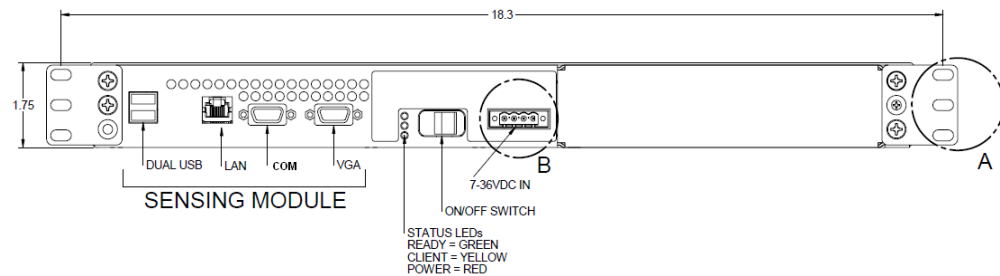
LAN 1: This port must be connected to the Interrogator Core Ethernet port via a crossover cable at all times.

LAN 2: This port is used to connect the si225 instrument to a user's network. This port is user configurable through the Integrated ENLIGHT interface and enables Remote Desktop, Remote Data, and Ethernet Pass-through functions.

PS2: A PS2 mouse or keyboard can be connected to this port.

POWER AND HDD ACTIVITY INDICATORS: The green power LED indicates that power has been supplied to the internal processor module of the si225 instrument. The amber LED indicates that there is active hard disk activity.

vii. Rear view of sm225 Module



POWER LED: This red indicator confirms that power has been supplied to the module.

READY LED: This green indicator confirms that the initialization sequence of the unit has completed and remote communications are possible.

CLIENT LED: This amber indicator is illuminated when there is an open remote connection to the module.

ON/OFF SWITCH: Controls the supply of DC power to the sm225 module.

VGA: This port can be used to attach an external monitor for diagnostic purposes.

COM: Not Used

DUAL USB: May be used for an external USB keyboard for limited configuration tasks, including changes to IP settings.

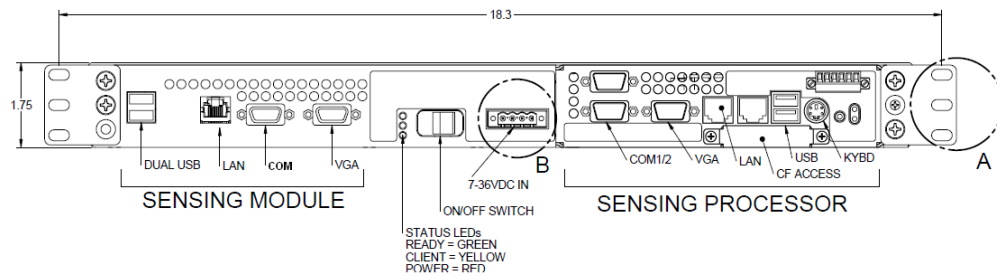
7-36VDC IN: The front panel mounted connector is a four (4) terminal, removable terminal block that is intended for fixed wiring applications. Therefore, it is extremely important that the power be applied in the proper sequence:

1. With the switch in the OFF position...
2. Connect the power supply leads to the terminal block.
3. Plug the power connector block into the front panel of the sm125.
4. Tighten the two (2) retaining screws.
5. Move the power switch to the ON position.

The sm225 can operate from any DC power supply that outputs 7 - 36V and 80 W of power.

LAN: This RJ-45 connector is used for the Ethernet communication port.

viii. Rear View of sm225 Module with Optional Sensing Processor



The ports for the sm225 Sensing Module are defined in the previous section. Addition of the Optional Internal Sensing Processor adds the following ports and connections.

VGA: Used for connection to an external VGA monitor.

NOTE: The Sensing Processor VGA port must be connected to a monitor device during boot up in order for the VGA drivers to be loaded by the processor. If no monitor is connected during boot up, no video signal will be available to the VGA port.

COM1, RS-232: Available for other communications protocols on OEM or user developed basis.

COM2, RS-232/485: Available for other communications protocols on OEM basis.

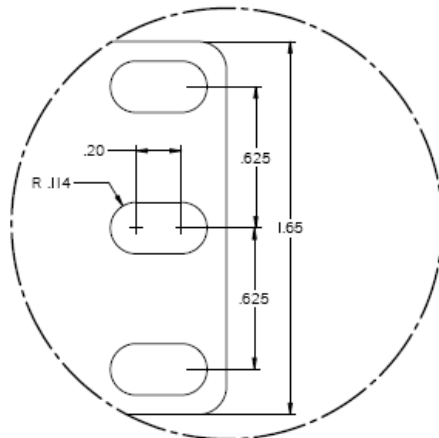
USB1/USB2: Used for connection of any USB1.0/2.0 compatible device.

PS2: Used for connecting a PS2 keyboard and mouse. Requires a "Y" cable.

LAN1/LAN2: Used to connect Sensing Processor to Sensing Module and/or external LAN. Connection to the Sensing Module requires a crossover cable.

B. Rack Mounting Tabs

The sm225/si225 modules and instruments can be rack mounted to any standardized ([EIA 310-D](#), [IEC 60297](#) and [DIN 41494 SC48D](#)) frame or enclosure of depth 18.5" or greater. Mechanical details for the included rack mount ears are seen in the image below.



10.3. Cabinet Installation Considerations

Should any Micron Optics Optical Interrogator module or instrument be installed into an equipment cabinet or rack, the following items should be considered:

- Elevated operating environment – If installed in a closed or multi-unit rack assembly, the operating ambient of the rack may be greater than the room ambient (40°C maximum).
- Reduced air flow – Installation of the equipment in a rack should be such that the amount of airflow required for safe operation is not compromised.
- Mechanical loading – Mounting of equipment in a rack should be such that a hazardous condition is not achieved due to uneven loading.
- Circuit overloading – Consideration should be given to the connection of the equipment to the supply circuit and the effect that overloading of circuits could have on over-current protection and supply wiring. Appropriate consideration of equipment nameplate rankings should be used when addressing this concern.
- Reliable Earth ground – Reliable Earth grounding of rack-mounted equipment should be maintained. Particular attention should be given to supply connections other than direct connection to the Branch (use of power strips).



11. ENLIGHT SENSING ANALYSIS SOFTWARE

11.1. ENLIGHT Introduction

11.1.1. Background and Motivation

Optical sensors and interrogators offer many unique advantages to conventional electrical sensors. However, for years, integration of an optical measurement system involved the selection of an appropriate sensor and interrogator combination, followed by an extensive software development effort by the end user to pull the components functionally together. MOI developed the ENLIGHT software package to ease the integration of the optical sensor system and facilitate many of the common features needed in such measurement systems.

Optical sensor systems share many operational characteristics with conventional electronic sensors systems. Concepts such as sensor limits and alarm conditions, for example, apply equally well to both types of systems and are thus well supported by software for either sensor class. Optical sensors, however, have special properties outside the scope of traditional sensor software that require attention, management, and control. Optical spectrum provisioning and spectral peak detection are two examples of such properties. The fact that FBG sensors can be serially multiplexed is a great architectural advantage, but introduces significant additional data processing challenges. MOI ENLIGHT endeavors to combine the traditionally useful features of conventional sensor software with the specific needs of the optical sensor system to create a single software tool for the optical sensor system integrator.

11.1.2. New and Different in ENLIGHT Version 1.5

ENLIGHT version 1.5 is a major new release that enhances existing features and adds new functionality, principally in the areas of data management, archiving, analysis, and access. Key among these additions is the new ENLIGHT Data Archive, which provides users with a reliable baseline of data storage functionality that is both complementary and completely independent from the existing user-selected data saving options located on the Save tab. The ENLIGHT Data Archive serves as a data safety net: an ever-present archival data store that is active from the moment the first FBGs and sensors are defined. The new Archive tab presents the user with a suite of indicators and controls used to plot and view sensor and FBG data that has been stored in the ENLIGHT Data Archive.

Some of these changes are minor and will not materially affect the user or the measurement system. However, some of the changes are fundamental in nature and will need to be thoroughly understood by the user prior to use. The user is encouraged to read through this section in detail to become comfortable with the changes before using this latest version. Throughout this section of the user guide, new features will be highlighted with a “**New in v1.5!**” indicator in the margin.

11.1.2.1. Features

New in v1.5!

1. The new ENLIGHT Data Archive and the Archive Tab
 - a. ENLIGHT v1.5 automatically stores FBG and sensor data into a Data Archive that is active from the moment that the first FBGs and sensors are defined. The Data Archive provides users with a reliable baseline of data archiving functionality that is both complementary and completely independent from the existing user-defined saving options location on the Save tab.
 - b. Data stored into the ENLIGHT Data Archive is governed by adjustable data archiving update interval and maximum file size controls.
 - c. The new Archive tab enables the user to retrieve FBG and sensor data from the ENLIGHT Data Archive and view that data on an archive chart.
 - d. Retrieved archive data can be copied or saved to file for easy export.
 - e. A new MOI designed Microsoft Silverlight™ Web Application example enables users to view and plot FBG and sensor data stored in the ENLIGHT Data Archive quickly and conveniently with a web browser.
2. Save Tab
 - a. In addition to local storage and email distribution, data files generated by the Save tab can now be transferred via FTP.
3. Settings Tab



- a. Application Settings
 - b. New to the Application Settings section are controls to enable and specify the port for the ENLIGHT Remote Command Interface or the new Web Interface.
4. Data Archiving Controls
- a. A new collection of controls has been added to the Settings tab to govern the location, update interval, and maximum size of the ENLIGHT Data Archive.

11.1.1.2.2. Considerations

Large sections of the ENLIGHT code base have been re-worked to provide for many of the tools outlined above. Micron Optics recommends reviewing the documentation contained herein to become familiar with the additional functions, features, plug-in requirements and interrogator model compatibility.

11.1.3. Scope and Support

11.1.3.1. Supported Instruments

MOI ENLIGHT supports a full suite of MOI interrogators based upon the x25 and x30 swept laser interrogator cores.

Throughout this manual, the reader will find references to x25 and x30 interrogator cores. The distinctions between the types of interrogator cores are important in many aspects of the software. A basic explanation of their differences is introduced in section 2.1 of this document. The implications of those differences as it relates the software features are presented here.

The x25 optical interrogator core is based upon swept wavelength laser based measurement technology that is enabled by full spectral scanning and data acquisition. x25 technology is focused on providing measurements with high absolute accuracy, software post-processing flexibility, and loss budget/dynamic range measurement performance. x25 technology based interrogators are typically characterized by lower acquisition rates and higher dynamic ranges, and also support continuous on-board NIST traceable wavelength reference components. x25 based interrogators are also ideally suited to measure many different optical sensor types, including FBGs, long period gratings, extrinsic Fabry-Pérot sensors, and many others. Interrogators based upon x25 technology are typically best suited for “static” measurement applications, requiring data acquisition rates in the low singles or tens of Hz.

The x30 optical interrogator core is based upon swept wavelength laser based measurement technology that is enabled by hardware detection, optimized for rapid data acquisition of many FBG sensors. x30 technology is focused on providing measurements with higher acquisition rates, moderate dynamic range, and continuous on-board referencing. Interrogators based upon x30 technology are typically best suited for “dynamic” measurement applications, requiring data acquisition rates in the hundreds to low thousands of Hz.

These two distinctions of scan rate and peak detection methodology will have some impact on the user interface for several of the ENLIGHT features. It is advisable that the user become comfortable with the differences between the two swept laser interrogator core platforms in order to best understand the documentation and subsequent use of the ENLIGHT software.

Micron Optics ENLIGHT is the default software interface for all MOI Sensor Interrogator systems.

NOTE: ENLIGHT v1.5 is compatible with all “sm” sensing modules and can be run on all “sp” sensing processors or other external

Windows computers with Microsoft .NET Framework 4. ENLIGHT v1.5 is NOT compatible with “si” sensing instruments with the Integrated ENLIGHT operating environment.

ENLIGHT v1.5 is compatible with the following sensing modules and sensing processors:

- sm125 (v.1 and 2)
- sm225
- sm130 (all versions)
- sm230 (all versions)

- sp125
- sp130

ENLIGHT v1.5 is NOT compatible with the following sensing instruments:

- si225
- si325
- si725
- si230
- si730

The latest version of ENLIGHT that is compatible with the Sensing Instrument Platform is v1.0.38. It is recommended that all “si” sensing instruments operate this ENLIGHT version compatible with the Integrated ENLIGHT operating environment.

11.1.3.2. Supported Sensors

MOI ENLIGHT has been designed to be compatible with all “MOI Tuned” designated Fiber Bragg Grating (FBG) based optical sensors, as well as a host of other FBG sensor designs.

Some advanced features, including MOI Sensor Templates, are available exclusively for FBG sensors bearing the “MOI Tuned” designation.

11.1.3.3. ENLIGHT Deployment Versions

ENLIGHT is deployed in two versions, desktop or integrated. Desktop ENLIGHT can be run as an application on any Windows XP, Vista, or Windows 7 computer with .NET 4.0 support.

Integrated ENLIGHT is built into a hardware specific Windows Embedded image and serves as the module firmware for certain MOI Sensor Interrogator products. The next section will outline the various deployment platforms for both integrated and desktop versions of ENLIGHT. The latest version of ENLIGHT compatible with the integrated ENLIGHT operating environment is v1.0.38.

11.1.3.4. Deployment Configurations

MOI ENLIGHT has been developed so as to be deployable on a number of MOI interrogation system platforms. Before identifying specific platforms, it will be instructive to become familiar with the components subsystems that comprise the MOI interrogation hardware product line. As introduced in section 2 of this document, the MOI Interrogator product offering is comprised of combinations of technology, platform and form factor components. Micron Optics ENLIGHT software is optimized to bring maximum benefit to each of the product configurations. The physical interface and configuration tools to the ENLIGHT software will be in part dependent on the specifics of the deployment platform. Some examples of those differences follow here.

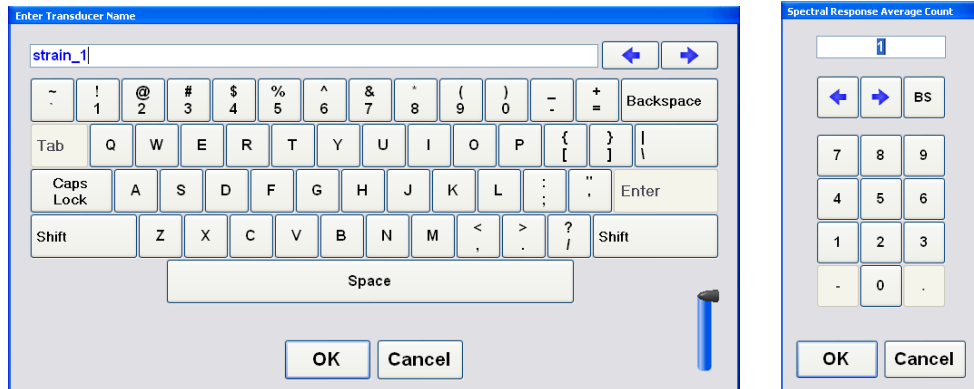
1. Integrated ENLIGHT on a portable interrogator.

Example: si325 Sensing Instrument with integrated ENLIGHT

The portable interrogator instrument is an implementation of an MOI measurement system which includes an x25 interrogator core, a processor module, batteries, support electronics, and a touch sensitive LCD screen. The primary data input device for an integrated ENLIGHT powered portable interrogator is a touch sensitive LCD screen. Alphanumeric inputs are captured through an on-screen pop-up virtual keyboard, which becomes visible when needed by the software. The two figures below show a virtual alpha-numeric keyboard and a virtual numeric keypad, which are used by the application for data entry.

Virtual Alpha-numeric Keyboard

Virtual numeric keypad



2. Desktop ENLIGHT on a stand-alone PC.

Example: User provided laptop or desktop Windows PC with desktop ENLIGHT connected to sm125 or sm130 Sensing Interrogator module.

MOI ENLIGHT can be installed and run on a stand-alone Windows PC. In this configuration, the ENLIGHT software would be user-configured to connect to a Sensing Interrogator module via an external Ethernet connection. In such an installation, the PC keyboard, mouse, and monitor would be the primary devices of inputting and reading data from the ENLIGHT software.

3. Integrated ENLIGHT on a rack mount Interrogator Instrument.

Example: si225 or si230 Sensing Instrument with integrated ENLIGHT

MOI offers complete, integrated sensor measurement systems in a 1U 19" rack mount configuration, which integrate the interrogator core, an optical multiplexer, and an internal processor unit preloaded with Integrated ENLIGHT software. In this implementation, the ENLIGHT software runs as integrated firmware on the interrogator module, and can be accessed through local keyboard/mouse/monitor devices, or remotely via a Windows Remote Desktop Protocol (RDP) connection.

4. Discrete Modules

Example: sp130 Sensing Processor with desktop ENLIGHT and sm130 Sensing Module

This architecture allows for the integration of an optical interrogator module, multiplexer module, and industrial grade sensor processor module. This approach may be used to facilitate a potentially large physical separation between sensor interrogator and processor modules. This may be the most convenient method of adding the features of MOI ENLIGHT to an existing sm125 or sm130 module installation.

5. Laboratory Instrument

Example: si725 or si730 Sensing Instrument with integrated ENLIGHT

Another implementation is a Laboratory optimized MOI interrogator which includes an interrogator core, a processor module, and a touch sensitive LCD screen. The primary data input device for an integrated ENLIGHT powered Laboratory interrogator is the touch screen, and is facilitated by virtual keyboards like those shown in the portable interrogator section. Alternatively, a USB keyboard/mouse unit can be used, if preferred.

11.1.4. Help and References

There are several sources of reference material available to ENLIGHT users. This section will identify some of those resources.

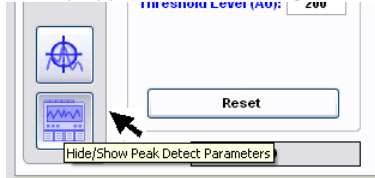
11.1.4.1. User Manual and White Papers

All documentation needs for MOI interrogator products are now served by this unified user manual. Micron Optics also supplies white papers explaining the theory of operation of the SL interrogator cores to address any additional detailed questions that may arise while using the MOI SL interrogation products. Current support materials are available for download at www.micronoptics.com/support_downloads.

11.1.4.2. Tool Tips

On deployments where an external mouse is used, floating Tool Tips are used to help define the function of certain features in the software. The following figure shows one of the hover labels on the ENLIGHT application.

Tool Tip appears over hovering mouse pointer



11.1.5. Supplementary Programs and Plug-ins

1. Microsoft .NET Framework 4

Micron Optics ENLIGHT requires that the Microsoft .NET Framework 4 be installed on a target computer before ENLIGHT will run. After the installation of ENLIGHT is complete, it will prompt the user to download the latest version of Microsoft .NET Framework 4.

Users can download and install Microsoft .NET Framework 4 for free from via the link at:

<http://www.microsoft.com/download/en/details.aspx?id=17851>

2. Microsoft Silverlight™

Microsoft Silverlight™ is a free plug-in, powered by Microsoft .NET Framework 4. This tool allows for ENLIGHT users to retrieve FBG and sensor data stored in the Data Archive to a Micron Optics designed Microsoft Silverlight™ Web Application example. The browser will prompt the user to install Microsoft Silverlight™ if the plug-in is not detected.

Additional Microsoft Silverlight™ resources can be found at <http://www.microsoft.com/silverlight/resources/>

11.2. Conceptual Overview

This section will outline of the basic concepts required to make full use of the ENLIGHT software. Additionally, this section will introduce the user to the structure of ENLIGHT and offer a context for subsequent details.

11.2.1. Concepts

11.2.1.1. x25 and x30 Interrogator Cores

As mentioned in a previous section, several features of the ENLIGHT user interface will change depending on the type of target interrogator core target. These differences will be explained in context throughout the remainder of the manual. For a more in-depth understanding of the differences between x25 and x30 optical interrogator cores, please refer to the section 4 of this document.

11.2.1.2. Defining Measured Elements

In the field of FBG-based sensing, the terms FBG, sensor, and transducer are often used interchangeably. In order to bring about a higher degree of sensor analysis precision, MOI has defined a number of terms, as they apply in the context of the MOI ENLIGHT software. Each term has a specific meaning and set of associated attributes. It is hoped that this additional degree of specificity will lead to clarity in definition and application of optical sensing elements.

- A. **Spectral Peaks** – As defined in the context of MOI ENLIGHT, “peaks” or “spectral peaks” refer to the direct wavelength output of the optical spectrum peak detection. In the case of an x25 interrogator core, peaks are defined by software post-processing of a calibrated full spectrum signal. In the case of an x30 interrogator core, peaks are the direct output from the internal hardware peak detection circuitry. In either case, the only attributes that a peak possesses are its current wavelength and amplitude. No additional attributes or properties are associated with peak elements.
- B. **FBGs** – Next in the hierarchy of measured elements is the concept of “FBGs”. FBGs are similar to spectral peaks in that they possess attributes of optical wavelength and amplitude, but have been enhanced with several other operational parameters. In MOI ENLIGHT software, FBGs carry with them the following additional concepts: 1) a maximum and minimum possible measurement wavelength, or wavelength range; the functionality of “wavelength binning”, where FBG index values are maintained, through events that might otherwise cause a “shift”; 2) averaging on a per-FBG basis; and 3) physical or optical path length location relative to the interrogator. Each of these features will be explained with more detail in subsequent sections of this document. For now, it is sufficient to recognize that these properties are attributed at the FBG level and not to spectral peaks.
- C. **Sensors** – The next higher-level measured element is the sensor. A sensor can be defined by a mathematical combination of one or more FBGs, yielding a calibrated output value of the desired measurement quantities, such as temperature, pressure, or strain. As a function of FBGs, the sensor carries with it the benefits of averaging and binning that have been defined for its constituent FBGs. Additionally, at the sensor level, the following concepts are introduced: calculated sensor range min and max values, historical min and max values, sensors type with associated units, sensor information and descriptions, averages, warning thresholds, and alarm thresholds. Sensors can also be flagged as “active/inactive”, “hidden/visible”, and can be used in rate-of-change calculations as well.

Introduced to the sensor element includes the concept and implementation of sub-expressions. Sub-expressions are mathematical definitions of quantities that are used by the main sensor expression to calculate the engineering quantity desired. Sensor expressions can be calculated from functions of FBGs, sub-expressions, constants, and other sensor values. The IDs and expressions defined in sub-expressions are specific to the sensor for which they are defined. As such, sub-expression IDs need not have unique names for each sensor as they can only be referenced by the specific sensor for which they have been defined.

It is now the sub-expression that bears the concept and implementation of “correlation of compensation” to the calling sensor. Correlation is the notion that a known positive or negative relationship exists between the operating spectral migration of a sub-expression and its calling sensor’s expression. The most common example of this is that a sensor’s thermal component of strain (expressed as a sub-expression) will track the direction of the total strain (expressed as a term of the sensor expression). By specifying a “correlation of compensation” between a sub-expression and a sensor expression, the optimum minimal spectral range required for the sensor measurement and thermal operating range can be achieved.

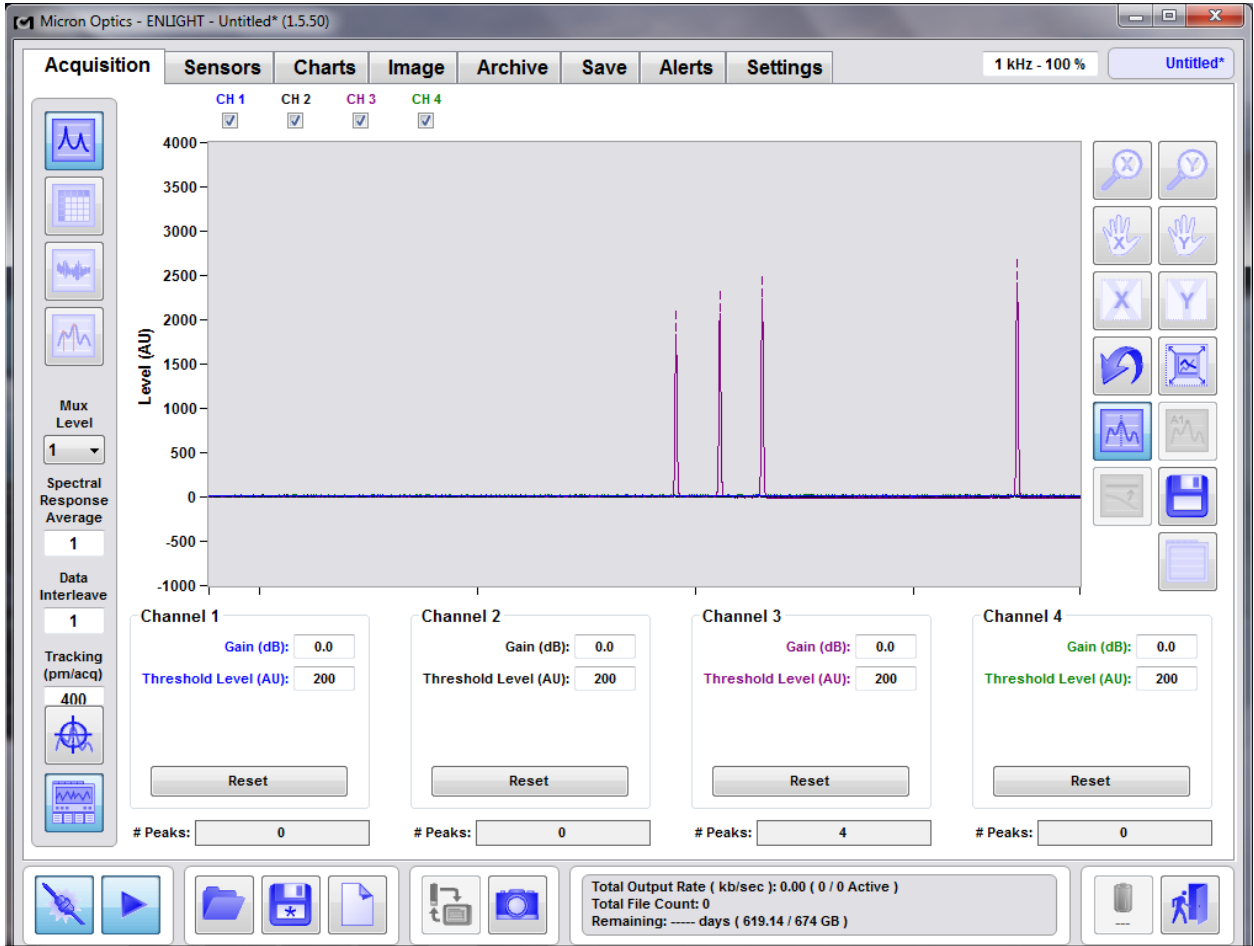
A key, predefined sub-expression is the DeltaT sub-expression, which defines the operating temperature range of each sensor element. DeltaT will be covered in greater detail in the description of the Sensors tab, sections 7.3.3.3 and 7.3.3.4.

- D. **Macro-sensors** – Last among measured elements is the macro-sensor. The macro-sensor is created as a mathematical combination of sensors, where a previously configured sensor can serve as a variable in the computation of another sensor (e.g. using the calibrated output of a temperature sensor in a thermal compensation calculation for a strain sensor). Support of macro-sensor functionality allows ENLIGHT to easily facilitate the calculations on large scale optical sensor deployments. Macro-sensors are also now facilitated in ENLIGHT through the use of the “Sub-Expressions” controls of the Edit Sensor Window.

11.2.2. Structure

This section will outline the design structure of the ENLIGHT software, introducing the main tools to control and make use of the software. The two main items of interest are the collection of global controls and the tab structure, which together define the operation of the high level operation of software.

ENLIGHT Top Level View



The preceding figure shows a full screen shot of MOI ENLIGHT with the Acquisition tab selected. Note that a large portion of the screenshot is filled with a tab control structure. This tab control dictates what features of the software are displayed based upon the tab selected. Note also that several controls and indicators are located outside of that tab structure. These features are visible on the GUI independent of the tab control state. The following section will detail those features.

11.2.2.1. Acquisition Rate



The acquisition rate indicator (top, right) displays the acquisition rate of the interrogator core from which ENLIGHT is collecting data. In the case of an x30 interrogator core, ENLIGHT also displays the percentage of remaining available buffer in this field.

11.2.2.2. Global Controls



Across the bottom of the ENLIGHT GUI is a series of global controls. These are as follows:

- A. Connections and Acquisition Cluster (bottom, left)



At the left side of the global controls bar is a cluster of two connection and Acquisition tools.

i. Connect to Module



Connects or disconnects ENLIGHT completely from the target interrogator.

NOTE: For an Integrated ENLIGHT configuration, the module connection must be disconnected in order to use Ethernet Pass-through.

ii. Acquire Data



Starts and stops data acquisition from the connected interrogator.

B. Configurations Management Cluster (bottom, left)



As part of the global controls bar is a cluster of configuration management tools. In MOI ENLIGHT, a configuration file stores all of the settings from all tabs of the software, including instrument settings, FBG definitions, sensor definitions, chart and image settings, archive settings, notification entries, and more. The configuration management cluster houses the three main tools required for managing configurations.

iii. Open Configuration



This control allows the user to open and load a previously saved configuration.

NOTE: When ENLIGHT is launched, the most recent saved configuration will load automatically.

iv. Close Configuration



This control allows the user to save the current configuration. Note that an asterisk (*) indicates that the current configuration has been modified since the last save.

v. New Configuration



This control allows the user to create a new default configuration.

vi. Current Configuration indicator (top, right)



Functionally associated with the configuration management cluster is the Current Configuration indicator, located on the far upper right of the GUI. This text indicator displays the name of the current configuration. Note that an asterisk (*) indicates that the current configuration has been modified since the last save.

C. Import/Export Tool (Integrated ENLIGHT deployments only) (bottom, center)



Located also among the global controls is the Import/Export tool. On an Integrated ENLIGHT deployment, this control launches a window that manages the transfer of data, images, configurations, and interrogator core firmware files to and from an external USB drive.

Import/Export Utility



The figure above shows the Import/Export utility that manages the transfer of data from ENLIGHT to and from an external USB drive. Radio buttons on the top allow the user to choose among data, images, configuration files, and firmware file. Data transfer is facilitated by simply highlighting the desired files and clicking on the arrow control that indicates the desired direction of file transfer.

For desktop deployments of ENLIGHT, file transfer is facilitated by standard Windows file management dialogs.

D. Screenshots (bottom, center)



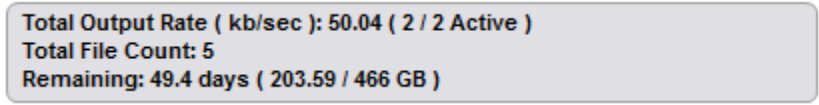
Next among the global controls is a universal screen capture tool. By pressing the camera icon, ENLIGHT will capture a bitmap image of the entire GUI and make that file available for transfer via the Import/Export or other file management tools. Screenshot files are saved into the ...\\Micron Optics\\ENLIGHT\\Data folder.



E. Data Saving

The concept of data storage will be covered in more detail in following sections of this manual. This section will serve to introduce the locations of data saving controls throughout the GUI.

i. Streaming Save Indicator



Central to the global controls cluster is a “streaming data” save indicator. As will be explained on the Save tab, data can be captured for spectral peaks, full optical spectrum, calibrated sensors, and events data. Controls to determine the rate and destination files for all types of streaming and triggered data are located on the Save tab.

The text indicator window above displays information about current data saving activity. Included among the display items are:

The rate at which data is being saved for all file types:	Total Output Rate (kB/sec):	42.20 kB/sec
The number of data saving profiles enabled and active:		2 / 2 Active
The number of files created by the current saving profiles:	Total File Count:	5
The remaining time capacity of local storage at the current rate:		49.4 days
	Remaining:	(203.59 / 466 GB)
The remaining disk space for storage, relative to the size of disk:		

ii. Other Save features



Across the various functional tabs, icons bearing some variation of a floppy disk indicate a local save function. These tab-specific save functions will be outlined in the appropriate tab summary.

F. Power and Exit Buttons (bottom, right)



At the end of the global controls cluster is a collection two power and exit buttons.

i. Battery Status (Portable Instrument deployments only)



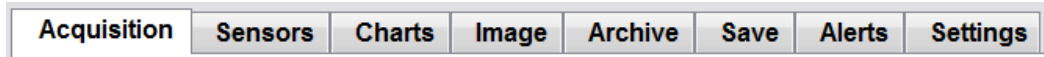
In an integrated application of ENLIGHT on a portable instrument, the Battery Status icon indicates if the unit is currently operating on AC power (charging) or the internal battery (discharging) and it displays the current level of charge present on the battery. When operating on AC power, pressing the Battery Status button will launch a pop up window that display the number of minutes until the battery is fully charged. When operating on the internal battery, the pop window will display the time remaining, in minutes. When executed from a desktop or stand-alone PC environment, the Battery Status button is disabled.

ii. Quit



This control ends the ENLIGHT application. In an integrated application of ENLIGHT on an interrogator instrument, the Quit button ends the ENLIGHT application and powers down the instrument. When the application is quit, the user will be prompted to save any changes to the current configuration file.

G. Tab Control



The majority of controls and indicators of the ENLIGHT software are contained within the main tab control. This tab control governs the flow of the software from data acquisition, to conditioning, display, and subsequent action. Details of each tab will be covered in section 7.3 of this document.

11.2.2.3. Control States

Understanding the three states for controls and indicators is important to the successful navigation MOI ENLIGHT. The three states are as follows:

i. On



The indicator above is in the “on” state. The edges of the icon show the button as concave, as if it was pressed, and the blue control is at its highest contrast. These two visual cues work together to inform the user that this button is on.

ii. Off



The indicator above is in the “off” state. The edges of the icon show the button as convex, as if it was not pressed, and the blue control is in a low contrast state. These two visual cues work together to inform the user that the button is off.

iii. Inactive



The indicator above is inactive, and not applicable to the utility as displayed. The icon appears as neither concave nor convex, and all of the blue hue has been replaced by a dull gray. A control or indicator in this state is not accessible and cannot be used.

11.2.2.4. Error Indicators

Throughout the ENLIGHT user interface, the color red has been reserved to indicate alarms, warnings or errors. Any background, icon, label, or indicator that is highlighted with either a dark or light red color should be given immediate attention.

11.3. Operational Overview

This section will outline the operation of ENLIGHT software. The first section will cover initial connection to the interrogator core. The middle sections will detail the operation of the functional tabs of the MOI ENLIGHT software. The final section will outline the details of the various data storage formats.

11.3.1. Module Selection and IP Settings

Upon first launching MOI ENLIGHT, the software will need to establish an Ethernet socket connection to an MOI interrogator core. In an Integrated installation of ENLIGHT, the software will come preconfigured to connect to the internal interrogator core, and no user configuration will be required.

11.3.1.1. Choosing Module and Configuring IP Settings

In a desktop or stand-alone installation, the user will be asked to choose the type of target module and configure ENLIGHT with the correct TCP/IP settings. Once the parameters are selected, ENLIGHT allows the user to check the settings with the “Test” button, pictured above as the icon with the “A+”. ENLIGHT will then attempt to make a connection to the interrogator as configured by the user and will inform the user whether or not the connection was successful via a pop-up message.

NOTE: Once a connection has been successfully established and the configuration is subsequently saved, ENLIGHT will make note of the last successful configuration. As such, the next time that ENLIGHT is launched on that processor, it will attempt to connect to the same interrogator target, thereby eliminating the need to re-configure IP settings on each launch of the software.

Advanced connection settings include the ability to specify a port and timeout for the module connection. In most cases, the default values should be used for the socket connection.

11.3.1.2. Connection Splash Screen



Connecting 

Once the configuration settings have been established, ENLIGHT will attempt to establish a constant connection with the interrogator core. During that time, a splash screen will appear, as shown in the figure above. Attempts to connect can be aborted by clicking on the red X control.

11.3.2. Acquisition tab

Once a connection is established the ENLIGHT user interface will default to the Acquisition tab. This first operational tab is the portion of the ENLIGHT software where all interrogator core hardware level configurations are established. The principal purpose of the Acquisition tab is to confirm that the optical FBG components have been effectively connected to the interrogator, and that the spectral information is successfully collected by the measurement system. The ultimate goal of the Acquisition tab is to yield a stable and repeatable output of spectral peaks that accurately reflect the optical sensor network to be measured. Once the peaks have been successfully polled from the module, subsequent tabs will add meaning and value to those raw measurements. In the Acquisition tab, there are many tools to aid in the optimization of FBG component peak data acquisition.

11.3.2.1. Data Acquisition

On the left side of the Acquisition tab, there are several vertically oriented controls that govern the connection state to the optical interrogator. These three controls are described in this section.

A. MUX Level



The MUX level control configures the ENLIGHT user interface and all of the corresponding module settings for use with an optical multiplexer unit. See section 7.3.2.7 of this document for additional details.

B. Spectral Response Average



This control sets the number of averages to be applied to the spectral amplitude traces. In an x25 interrogator, peak detection is performed on the full spectrum trace. By first averaging the full spectrum data, better repeatability of the peak detected values can be achieved. In an x30 interrogator, spectral averaging can improve the noise characteristics of the Spectral Diagnostic View trace, though it does so at the expense of optimal time response and without effect on peak data. Most users choose to leave this setting at the default '1' and instead use averaging for sensors.

C. Data Interleave (x30 core only)





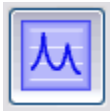
The data interleave control allows the user to set an interleave level, which can be used to reduce the data rate from the interrogator. For example, if a data interleave of 10 is applied to a 1 kHz x30 interrogator core, then data will stream to ENLIGHT at a rate of 100Hz.

D. Tracking Window



The Tracking Window control allows the user to specify how far a tracked peak can move between successive scans and still be detected and considered as a valid peak. This control is a required element of the ENLIGHT Peak Tracking algorithm that prevents sensor indexing issues in the event of a missing or extraneous peak. "Peak Tracking" is outlined in more detail in a following section.

11.3.2.2. Spectrum View

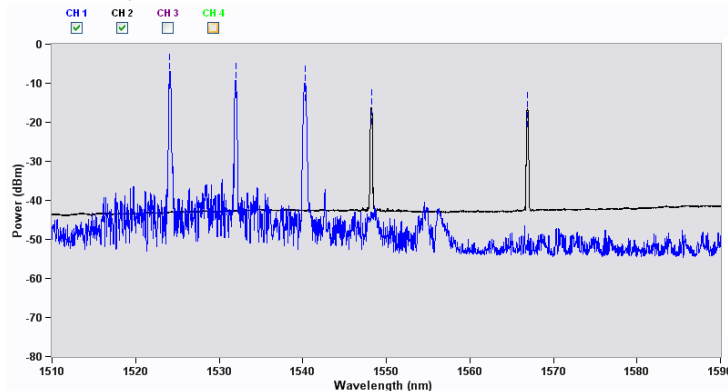


On the Acquisition tab, there are four primary modes from which the user can choose in order to view collected data from the interrogator or interrogator core. The first of these views is Amplitude View. In Amplitude View, data is collected from the interrogator and displayed in an X-Y plot.

The details of Amplitude View are somewhat dependent on whether the target interrogator core is an x25 or x30 core. The following two sub-sections introduce the Amplitude View and the differences between interrogator core types.

A. x25 interrogator core

x25 Core Amplitude Measurement

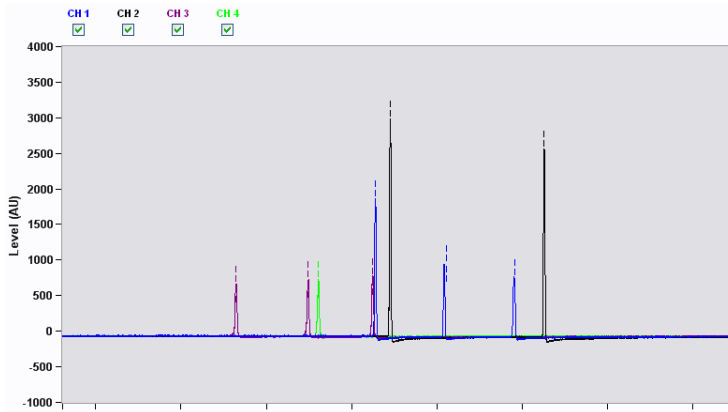


For an x25 interrogator core:

- Full spectrum is the primary measurement mode of the interrogator.
- Full spectrum is fully calibrated and accurate in wavelength
- Amplitude reflects received power in units of dBm (except in dB, where normalized)
- Receiver bandwidth of interrogator core is matched so as to accurately reflect full spectrum of spectral peaks

B. x30 interrogator core

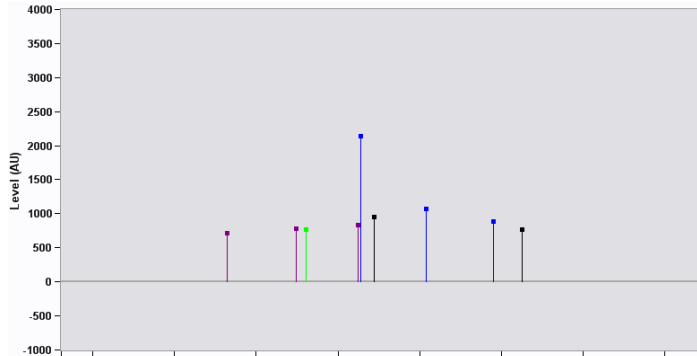
- i. x30 Core Amplitude Measurement – Spectral Diagnostic View equipped



For an x30 interrogator core, equipped with Spectral Diagnostic View:

- Full spectrum is NOT the primary measurement mode of the x30 core
- Intended only as a visual cue that the system is properly configured, in terms of gain, FBG component count, gross FBG shape, relative position, etc.
- X axis displayed terms of fractional counts...no absolute wavelength information implied
- Amplitude levels are relative and used to ensure proper signal-to-noise ratio and prevent signal saturation
- Receivers are intentionally band-limited to reduce noise; spectral peak shapes are not indicative of actual FBG components.
- Offsets between sensor shape maxima in the Spectral Diagnostic View and the peak locations are expected and a function of sensor bandwidth.

ii. Peak locations and amplitudes only



For x30 interrogator cores without the Spectral Diagnostic View feature, Amplitude View shows the count location and peak amplitude for each of the spectral peaks.

C. Zoom and Pan

i. Zoom X and Y



Allow for independent or simultaneous zooming of the X and Y axes of the active plot.

Left mouse button double clicking will Zoom In on the Amplitude Chart. Right mouse button double clicking will Zoom out.

ii. Pan X and Y



Allow for independent or simultaneous panning of the X and Y axes of the active plot.

iii. Autoscale X, Y. Autoscale X and Y to Full Scale.



- Autoscale X automatically adjusts the X scale to the full range of active X data.
- Autoscale Y automatically adjusts the Y scale to the full range of active Y data.
- Autoscale X and Y to Full Scale simultaneously sets the X and Y scales to the full range of the measurement unit, independent of the range of the active data.

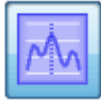
iv. Undo Zoom



Sets the X and Y scales back to the values one step prior to the previous scaling action. (disable Autoscale X,Y for Undo Zoom to persist)

D. Peaks and FBGs

i. Show Peaks



Superimposes indicators on the amplitude plot that identify the spectral locations at which spectral peaks have been found. Peaks are plotted on the amplitude graphs with short, dashed vertical line segments.

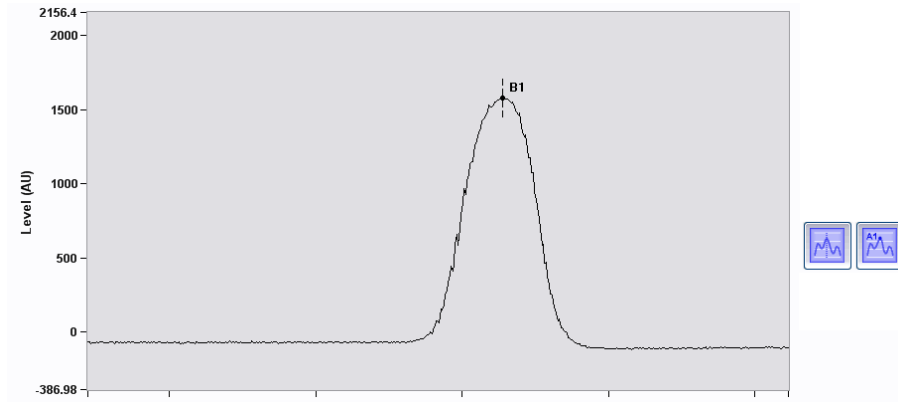
ii. Show FBGs



Superimposes indicators on the amplitude plot that identify the spectral locations at which defined FBGs have been found. The "Show FBGs" indicator helps the user to visually associate defined FBG elements with detected spectral peaks and the surrounding optical spectrum, for diagnostic purposes.

The concept of FBGs will be officially introduced and more thoroughly explained in section 7.3.3.1 of this document.

iii. Resulting Peak Plot



The above plot shows a spectral section (solid black line) that results in a spectral peak (dashed vertical black line) and a defined FBG (black dot with black text label).

11.3.2.3. Peak Detection

A. Hide/Show Peak Detection Parameters (x25 interrogator core only)



Alternately hides or shows the array of peak detection parameter controls.

B. x25 interrogator core peak detection parameters

Channel 1

Threshold (dBm):	-50.00
Rel. Threshold (dB):	-8.00
Width (nm):	0.10
Width Level (dB):	3.00

Detect Valleys

Reset

Peaks: 1

This cluster of controls is used to define the peak detection parameters for an x25 core, where peak detection is performed as a post processing operation on the full optical spectrum. The resulting number of detected peaks for the channel is displayed in the # Peaks indicator below the channel controls group.

i. Parameters:

- a. Threshold (dBm)
- b. Rel. Threshold (dB)
- c. Width (nm)
- d. Width Level (dB)

A complete explanation of the x25 peak detection parameters and techniques for optimization can be found in section 4.1.1.2 of this document.

ii. Detect Valleys



Detect Valleys

This boolean checkbox control causes the peak detection algorithm for the specified channels to target valleys (spectral features with upward concavity) rather than peaks (spectral features with downward concavity).

iii. Reset Parameters



This control resets the four controls above to their default values. These defaults are set to work well with standard MOI FBGs and sensors, as well as most common sensor FBGs from other sources.

iv. # Peaks Indicator



Shows the total number of peaks that meet the criteria set forth by the four peak detection parameters above.

C. x30 interrogator core peak detection parameters



This group of controls is used to define the peak detection parameters for an x30 interrogator core, where peak detection is performed in the analog electronic circuitry of the interrogator. The resulting number of detected peaks for the channel is reported in the # Peaks indicator below the cluster.

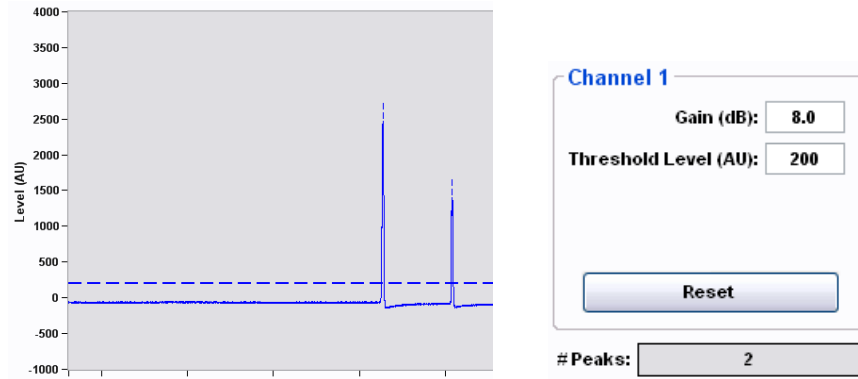
i. Parameters:

a. Gain (dB):

This parameter sets the receiver gain in calibrated 0.1 dB steps, from a minimum of 0 dB to a maximum of 20 dB. Receiver gain can also be set to un-calibrated negative values as low as -30 dB, allowing the user to minimize gain settings for additional system flexibility.

b. Threshold Level:

This parameter sets an internal hardware threshold below which all signals are ignored for purposes of hardware peak detection. The value of the threshold level is normalized to the amplitude of the signal as reported in the amplitude plot. As such, the amplitude view can directly assist the user in choosing effective threshold level settings.



Note the transient dashed line that appears for 1 second to indicate that the Noise Threshold value is changed. This indicator also appears briefly when the Gain control is changed. Dashed lines also indicate threshold settings for x25 interrogator cores.

D. Peak Tracking

Peak detection in an x30 core is not performed in post processing of a sampled full optical spectrum. Rather, peak detection in the x30 core is performed at the hardware level, yielding an output data set from the module that is principally comprised of peak wavelength information. Advance internal x30 data processing features, including wavelength averaging and sensor distance compensation, are conducted on a per-sensor basis and are applied to the sensors as a function of its index on the serialized sensor fiber. For example, consider a case where an sm130 module has detected three spectral peaks, at wavelengths of 1510, 1520, and 1530 nm. Now imagine that 1000 averages have been applied to the 1520 nm peak. As long as all three peaks are detected by the module, then the data is reported as (measurement noise neglected):

1510.000 1520.000 1530.000.

Imagine, though, that the second sensor drops below the detection threshold, and suddenly, only two spectral peaks are detected by the sm130 module. Then, the data should be reported as:

1510.000 1530.000,

with the 1530 nm wavelength, formerly the third peak, now occupying the second index of the wavelength array.

However, since averaging has been applied to second detected spectral peak, there will be a transition in the averaging buffer from 1520 nm to 1530nm. The first 1000 datasets returned from the module following the shift of the 1530nm sensor would look like this.

1510.000 1520.010
 1510.000 1520.020
 ...
 1510.000 1529.980
 1510.000 1529.990

Restated, the first 1000 acquisitions following the disappearance of the 1520nm peak would be erroneous. Furthermore, the position in which the peak wavelength is reported to the remainder of the ENLIGHT application would shift, further affecting FBG and sensor calculations downstream. To prevent these types of peak shift errors, the concept of peak wavelength tracking has been introduced to the x30 interrogator cores. With peak tracking enabled, the previous scenario would result in datasets as follows:

1510.000 1520.000 1530.000

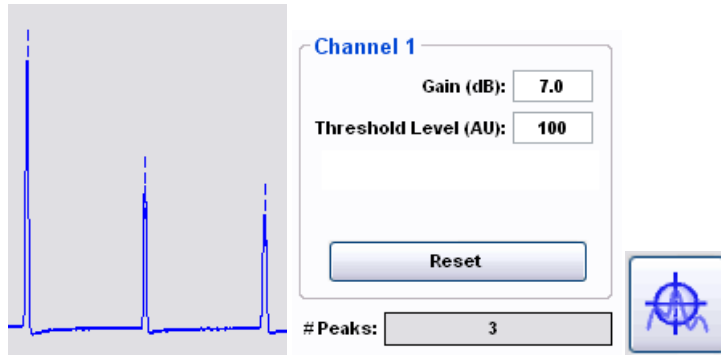


1510.000	-----	1530.000
1510.000	-----	1530.000
...
1510.000	-----	1530.000
1510.000	-----	1530.000
1510.000	-----	1530.000

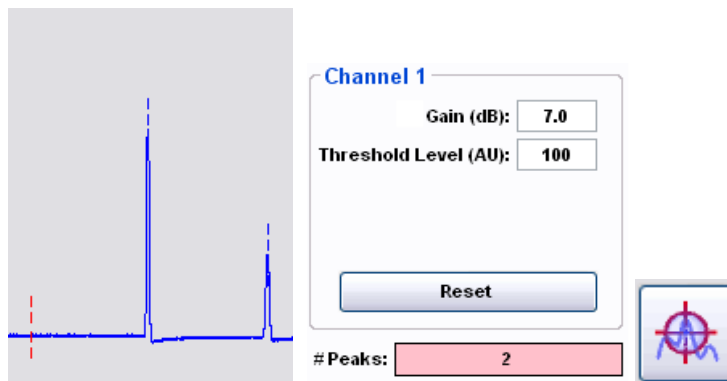
With peak tracking enabled, the peak wavelengths continue to report in their specified position, ensuring that all sensor specific operations (averaging, etc.) continue to be applied to the intended peak wavelength. The ability to serially connect FBG sensors is one of the key advantages of the technology. The peak tracking feature of ENLIGHT brings that benefit of serial sensor connections to the application while eliminating the otherwise problematic side effects.

The following examples highlight how peak tracking is used.

The graphic below depicts a starting condition. Three peak wavelengths are being tracked and reported by the x30 interrogator core.



Case I. One of the tracked peaks is not detected. Three visual indications are used to alert the user as to the nature of the error. First, the background color for the “#Peaks” indicator changes from gray to red. In this way, the user is immediately alerted that the number of detected peaks (2 in this case) is not equal to the number of tracked peaks. Second, a dashed vertical line is shown in the spectral position where the x30 core expects a tracked peak, but does not find a detected peak. Third, the “Reset Wavelength Tracking” button is highlighted with a red “target” to indicate that Wavelength tracking may need to be reset.

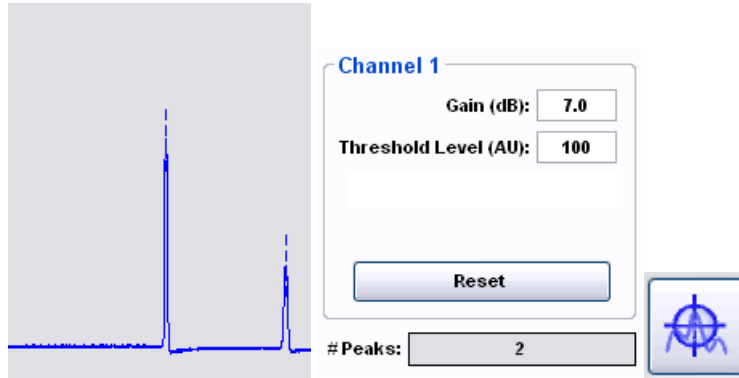


There are several ways to rectify the alarm condition caused by a mismatch of detected and tracked peaks.

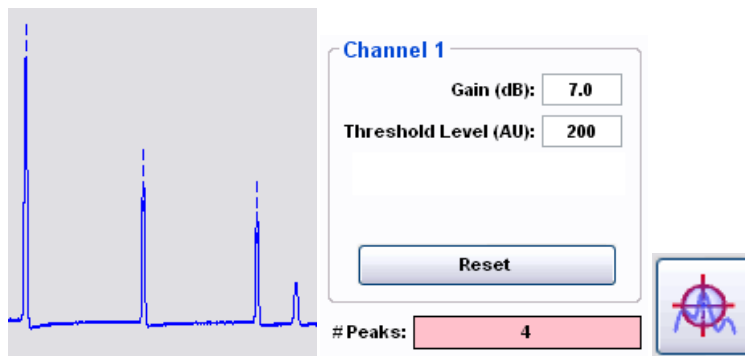
Resolution A: Identify and address the root cause at the optical level. In this case of this example, the spectral peak was attenuated with a sharp bend in the fiber. The first option would be to correct that bend, restoring the spectral peak to its original amplitude. When the module again detects the third spectral peak at the expected spectral location, the alarm state indicators will be removed.



Resolution B: Choose to remove the entry from collection of tracked peaks, by pressing the Reset Wavelength Tracking button. If the missing spectral peak has been intentionally removed from the system or otherwise cannot be recovered, the “Reset Wavelength Tracking” can be used to reset the tracking, causing the x30 core to reduce the number and position of tracked peaks to meet the number and position of detected peaks.



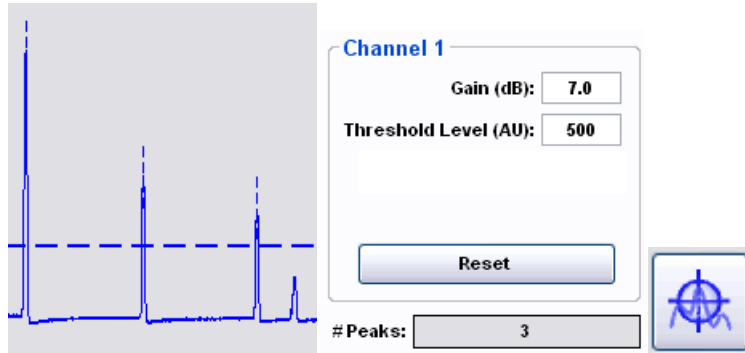
Case II. An additional peak is introduced to the system, yielding a greater number of detected peaks than tracked peaks.



The preceding graphic shows a condition where a spurious peak has been introduced to the sensor system, and has been detected by the x30 interrogator. In this example, the three tracked and detected peaks are indicated by the blue vertical lines. The red-highlighted “# Peaks” indicator shows that a total of four peaks have been detected, though only three have been tracked. This fourth peak will not be included in any subsequent FBG or sensor definitions unless included among the tracked peaks.

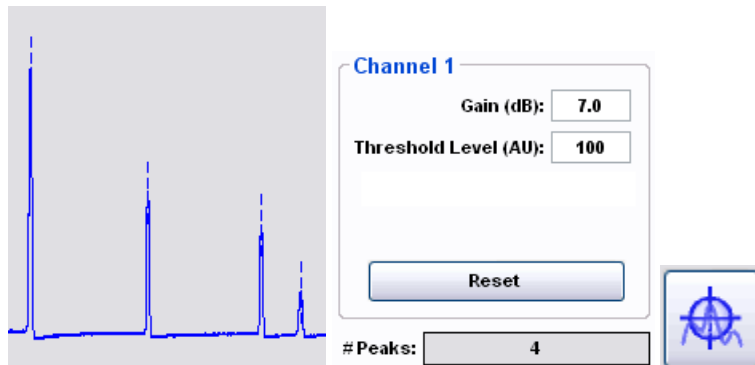
To rectify the condition where the x30 interrogator core detects more peaks than it is tracking, one of two actions can be taken.

Resolution A. Change gain and/or threshold value to avoid spurious peak.



If the gain and/or Threshold level is changed such that the spurious peak falls below the threshold, then the peak will no longer be detected, yielding an equal number of detected and tracked peaks. This action would be appropriate if the peak was indeed extraneous and not a part of the measurement system. This action is show in the images above.

Resolution B. Reset Wavelength Tracking to include new peak.



If the new peak is an intended component of the measurement system, then the “Reset Wavelength Tracking” control can be used to create a corresponding entry among the tracked peaks collection. Peak tracking has also been implemented for x25 cores.

11.3.2.4. Table View



This control changes the Acquisition tab view to Table View.

Spectral Peaks table

Channel 1		Channel 2		Channel 3		Channel 4	
WL	Level	WL	Level	WL	Level	WL	Level
1524.1112	-7.9	1548.2193	-13.6				
1531.9663	-10.1	1566.8943	-14.1				
1540.2709	-10.9						



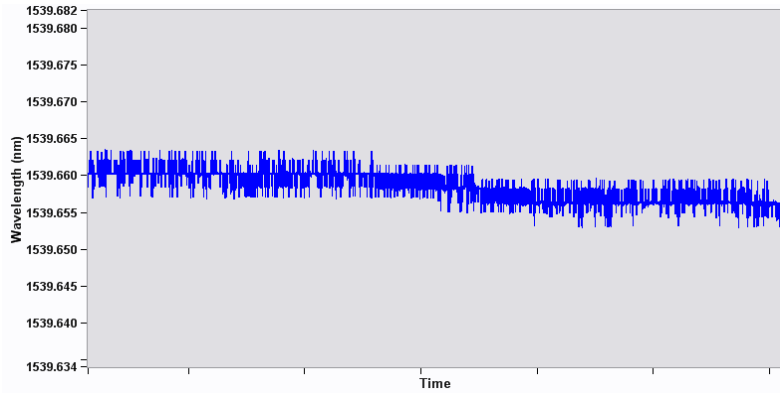
The Spectral Peaks table displays the wavelength and amplitude level information for all of the spectral peaks detected on a per-channel basis. x25 interrogator cores report level in dBm and x30 interrogator cores report level in arbitrary units.

11.3.2.5. Chart View



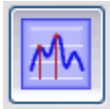
This control changes the Acquisition tab to Chart View.

Spectral Peak Chart View



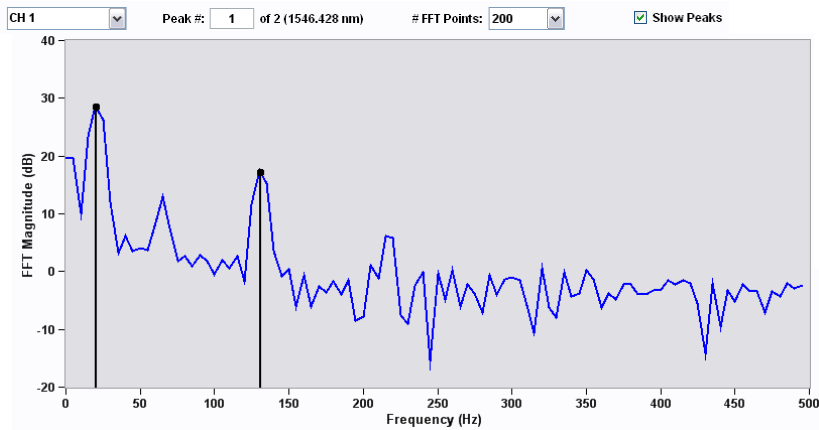
The Spectral Peak Chart View plots the wavelength value of one spectral peak as a function of time. A “Channel” control and “Peak #” control select the spectral peak to be plotted. A “# Points” control dictates the plot depth of the chart.

11.3.2.6. FFT View (x30 interrogator cores only)



This control changes the Acquisition tab to FFT (Fast Fourier Transform) View.

Spectral Peak FFT view



The Spectral Peak FFT View plots the frequency response of one spectral peak as measured by an x30 interrogator core. A “Channel” control and “Peak #” control select the spectral peak whose frequency response will be plotted. The “# FFT points” control selects how many sequential data points are used to calculate the FFT. The “Show Peaks” control overlays indicators of detected FFT feature in the Fourier spectrum.

11.3.2.7. Multiplexing

The base measurement capacity of a four channel x25 or x30 interrogator core can be expanded by use of an optical multiplexer module. Whether assembled as a set of discrete modules (i.e. an sm125 optical sensing interrogator with an sm041 channel multiplexer) or as an integrated module (i.e. si225 instrument with integrated ENLIGHT), multiplexing can be used to expand the measurement capacity of the optical sensing system.

Certain deployments of integrated ENLIGHT (e.g. portables, like si325) will not show multiplexing controls, as they do not apply.

A. Enabling



The “Mux Level” control sets the multiplexer level for the interrogator core, causing the unit to send control signals to the multiplexer and begin measurements on the multiplexed system. The Mux Level setting can be changed at any point of the measurement using MOI ENLIGHT. The Mux Level acts as a multiplier to the number of received channels. For example, a four channel unit with a Mux Level of 4 will be configured to measure 16 sub-channels. Several features of the Acquisition tab will change in response to the Mux Setting. The next few items will summarize those changes.

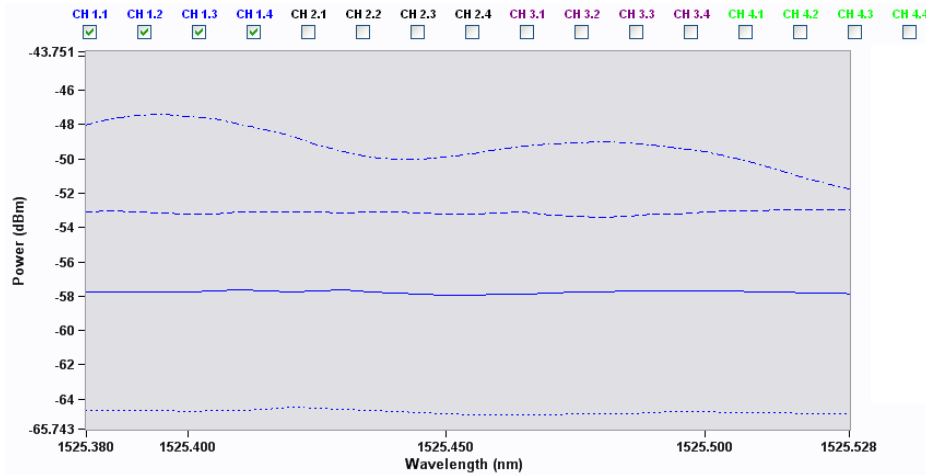
B. Channel/Sub-Channel controls and definitions

NOTE: Legacy users may be familiar with the previous “all integer” standard for multiplexer notation. Please see the appendix entitled “Using Legacy Notation for MOI Multiplexers” in the appendices section of this document.

The maximum number of interrogator core measurement fibers is four. These fiber connections are referred to as “channels” and are designated by the integers 1, 2, 3, and 4. These channels can be connected to the input fibers of a multiplexer module, which are then optically switched sequentially into either two or four output fibers for each input fiber. The multiplexed output fibers are referred to as “sub-channels” and are designated with a decimal fractional number, ranging from 1.1 to 4.4. The integer part of the number (the digit on the left of the decimal point) refers to the interrogator core channel to which the sub-channel is multiplexed. The fractional part of the number (the digit on the right of the decimal point) refers to the multiplexer position of that sub-channel. The full decimal fraction notation uniquely identifies the sub-channel.

With this decimal fraction nomenclature, the use of an integer label (e.g. 2) always implies an interrogator core “channel”. The use of a decimal fraction (e.g. 4.4) always implies a multiplexed “sub-channel”. For MOI optical sensing interrogators, certain functions and features are implemented on a channel basis and will always use an integer channel number as a parameter. Other features and functions are implemented on a “sub-channel” basis and will always use a decimal fraction sub-channel number as a parameter. Details for these functions and their associated commands are outlined in section 4 of this document for each particular interrogator core type: x25 or x30. The controls and indicators used in ENLIGHT also respect this convention.

C. Graph identification



When the Mux Level is changed from 1 to 4, an interrogator core with a total number of four channels is multiplexed into 16 sub-channels. As is seen in the preceding figure, on the amplitude view of the Acquisition tab, the component sub-channels of each group share a common color. sub-channels are differentiated by line style. For each channel group (X = 1–4) => X.1: solid, X.2: long dash, X.3: short dash, X.4 dot-dash.

D. Peak detection (PD) parameters

i. Multiplexed x25 core channels: Unique PD parameters for each sub-channel

Channel 1

Threshold (dBm):

Rel. Threshold (dB):

Width (nm):

Width Level (dB):

Detect Valleys

CH 1.1

CH 1.1

CH 1.2

CH 1.3

CH 1.4

For x25 interrogator cores, peak detection is performed as a post processing step on acquired spectrum. As such, the peak detection parameters can be individually specified for sub-channels, as shown in the figure above.

ii. Multiplexed x30 cores channels: common PD parameters for each group of sub-channels

Channel 1

Gain (dB):

Noise Threshold:

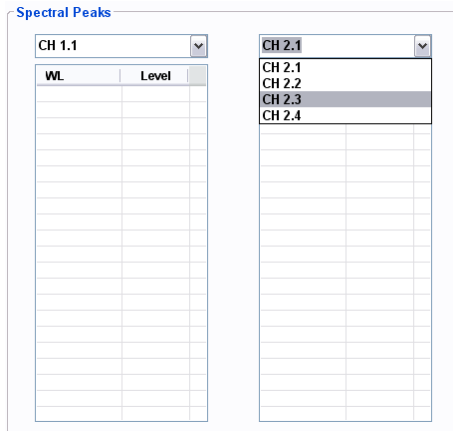
Peaks

For x30 interrogator cores, peak detection is performed in channel hardware and during the actual laser scan. As such, the peak detection parameters are common among all of the sub-channels that make up a channel group, as



shown in the figure above. Note that the resulting “# Peaks” indicator for the channel now reflects the two or four multiplexed sub-channels for that channel group, corresponding to the selected MUX level.

iii. Table indicators



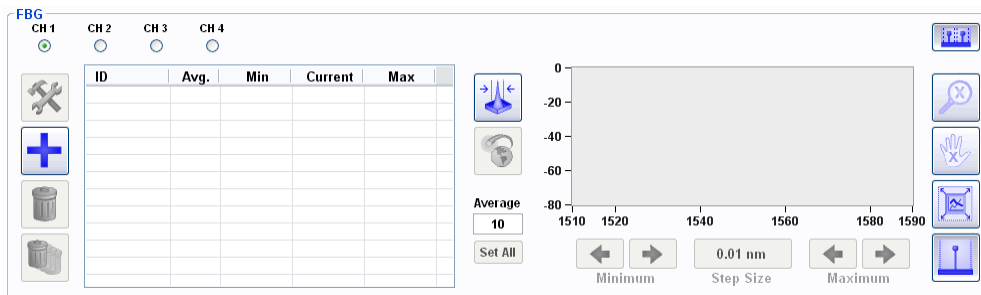
For a multiplexed system, the channel label atop each table converts from a plain text label to a drop down list control, allowing the user to select any sub-channel for display from among each channel group.

11.3.3. Sensors tab

Once optical spectrum has been collected and peaks detected on the Acquisition tab, the resulting spectral peak features are ready for use in the next layer of data processing: the Sensors tab.

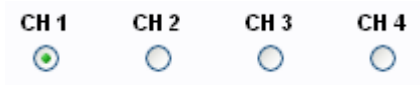
The Sensors tab is the section of the ENLIGHT software where spectral peaks are assigned certain attributes and properties and thus defined as FBGs. Those resulting FBG elements can then serve as the basis for calculations of higher order, calibrated sensor elements. The Sensors tab is subdivided into two sections. The top half of the tab deals with the concept of FBGs, while the bottom half deals with Sensors.

11.3.3.1. FBGs



The top half of the Sensors tab is equipped with the tools to create and manage FBG elements.

A. Channel Controls



FBGs are defined and managed on a per-channel basis (or on the basis of sub-channel, for a multiplexed system). The Channel Controls allow the user to select a single channel for which FBGs will be viewed and managed at a time.

B. FBG Table

ID	Avg.	Min	Current	Max
FBG_A1	1	1538.830	1539.831	1540.830
FBG_A2	1	1550.706	1551.706	1552.263
FBG_A3	1	1553.263	1553.818	1554.820

The FBG table summarizes the attributes of the existing FBGs for the specified channel. The parameters that the table reports for each FBG are as follows:

- i. **FBG ID (ID):**
This variable ID is associated with an FBG at the time of its creation so that subsequent sensor expressions can use the variable in mathematical expressions. IDs are automatically assigned to created FBG elements, but can be customized. FBG variable ID names can be any unique alphanumeric, and ENLIGHT will automatically replace any space in a name with an underscore. Strictly numeric names may not be used, as they confuse the expression parser.
- ii. **Averages (Avg.):**
The number of averages to be applied to that FBG.
- iii. **Minimum:**
The user-specified lowest wavelength that an FBG should be able to reach over the defined operating conditions. This wavelength represents the lower bound of the FBG's wavelength "bin". The minimum value of the wavelength bin is an important design parameter for the sensor system, as it is used to calculate the measurement and thermal operating ranges of its associated sensor quantities. If an FBG wavelength moves past the defined bin, the FBG value is ignored and omitted from the FBG table and all subsequent sensor calculations.
- iv. **Current:**
The currently measured wavelength of the FBG.
- v. **Maximum:**
The user-specified highest wavelength that an FBG should be able to reach over the defined operating conditions of the sensor into which it has been assembled. This wavelength represents the upper bound of the FBG's wavelength "bin". The maximum value of the wavelength bin is an important design parameter for the sensor system, as it is used to calculate the measurement and thermal operating ranges of its associated sensor quantities. If an FBG wavelength moves past the defined bin, the FBG value is ignored and omitted from the FBG table and all subsequent sensor calculations.

NOTE: With the version 1.0 release of ENLIGHT, the concepts of "Absolute" or "Referenced" FBG modes have been replaced by FBG calculation shorthand notations. This change eliminates the ambiguity of FBG references as they appear in sensor expressions. For example, an FBG_A1 in "absolute" mode would yield a value of 1550.001 in an expression while an FBG_A2 in "referenced" mode might yield a value of 0.011. With this former method, there was not immediately obvious means of knowing what value an FBG in a calibration expression might yield, leading to user confusion. The elimination of the FBG mode and the addition of FBG calculation shorthand notations remove this potential for confusion. Details for these shorthand expressions are presented in the section 7.3.3.2 of this document.

Additionally, global FBG zeroing has been replaced by independent sensor zeroing. Details for this process are outlined in section 7.3.3.5 of this document.

i. Table Buttons

On the Sensors tab, there are several buttons located near the FBG table. The following entries offer a brief explanation for each of those buttons.

i. Edit FBG



This control launches a data entry form to manage the attributes of the selected FBG.

ii. Add FBG



Creates a new FBG element and launches a data entry form for the user to input the attributes of that new FBG.

iii. Delete FBG



Deletes the selected FBG.

iv. Delete All FBGs on Channel



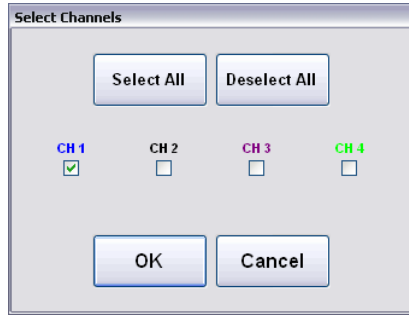
Deletes all of the FBGs on the selected channels or sub-channels.

v. Create FBGs



This control automatically creates a collection of FBG elements on the selected channel. The number of FBGs created will be equal to the number of tracked spectral peaks found by the Acquisition tab for the corresponding channel. The upper and lower wavelength limits will be automatically defined as the FBG's spectral "bin" according to the parameters defined by the following graphic. FBG IDs are automatically generated but may be later changed by the user in the Edit FBG window.

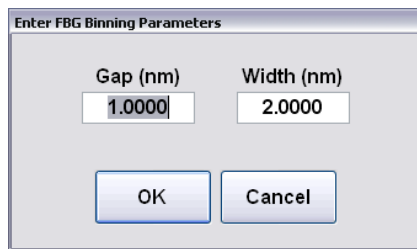
Enter Binning Parameters



When a Create FBGs operation is initialized, a Select Channels popup window launches. A series of Boolean checkboxes offer the user control of which channels to select for automatic FBG bin creation, with the currently active channel pre-selected. “Select All” and “Deselect All” buttons offer a convenient means of mass channel selection. In the case that a multiplexer is employed in a configuration, the Select Channels window scales appropriately.

A similar multiple-channel selection window is employed to enable multi-channel delete and average FBG actions.

Once the desired channels are selected, the Binning Parameters popup window launches.



The user enters data to define the desired maximum “Width (nm)” of the spectral bins and the minimum acceptable “Gap (nm)” between bins. MOI ENLIGHT then automatically defines wavelength bins for each FBG according to the requirements.

When new bins that are created alongside or in between existing bins, ENLIGHT will attempt to create the new bins according to the specified parameters, but will first show deference to the existing FBG range settings, so as to not disrupt previously configured FBGs and corresponding sensors. Priority is given to bin gap over bin width, in the case of a conflict.

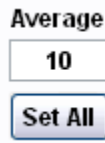
vi. Enable/Disable Distance Compensation



This control causes ENLIGHT to perform distance compensation calculations for the channel selected on the FBG section of the Sensor tab. In order for distance compensation calculations to be performed, all FBG elements on the selected must be assigned a non-zero distance. Distance compensation calculations are applied to the FBGs of the selected channel, as well as the user-defined FBG bin limits.

Global module support for distance compensation can be enabled or disabled in the Application Settings of the Settings tab. Please see the appendix entitle “Speed of Light Effects on SL-based Wavelength Measurements” in the Appendices section of this document.

vii. Average



These controls set the number of averages to be applied to the selected FBG. Use of the “Set All” button applies the specified number of averages to all FBGs on the selected channels or sub-channels.

ii. Editing or adding FBGs

The preceding data entry form is used to set/modify the attributes for the selected FBG. This form is launched by both the “Add FBG” and “Edit FBG” buttons, as well as a double click event on one of the FBG entries on the table, in the case of desktop deployments.

In addition to the aforementioned ID, Minimum, Maximum and Average parameters, the Edit sensor window presents the following controls.

a. Auto Button

This control will automatically assign the FBG ID to the next available based upon the following alpha-numeric designation: sub-channels 1.1 – 4.4 are associated with letters A-P and FBGs on each sub-channel are associated numerically with the order in which they were created. For example, the fourth FBG created on sub-channel 1.2 would be assigned the ID “FBG_E4”.

b. Distance (m)

Sets the distance to the FBG for use in distance compensation calculations.

Global module support for distance compensation can be enabled or disabled in the Application Settings of the Settings tab. Please see the appendix entitled “Speed of Light Effects on SL-based Wavelength Measurements” in the Appendices section of this document.

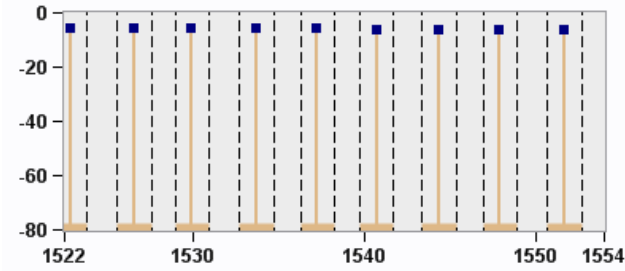
c. Alpha (sec)

Alpha is a customized version of a discrete low pass filter implementation where the alpha factor is scaled inversely to the value of the signals derivative. Use of the Alpha parameter allows the user to create a corrected macro-sensor where the low pass filtered (alpha) computation of the sensor can be subtracted from the sensor, yielding an effectively high pass filtered version of that sensor which is not sensitive to slow variations or offsets.



A larger value of Alpha yields a slower response time. A key example of this process would be to eliminate slow-moving thermal offset in a fiber optic accelerometer measurements.

C. FBG Spectral graph – shows spectrum and FBG range for selected channel

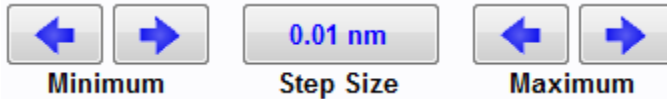


On the right side of the FBGs section is an FBG spectral graph display. This display is used to provide a quick visual assessment of the wavelength provisioning of the optical FBG elements on the selected measurement channel. Each FBG is represented by a vertical line with a square dot on the top. Its X axis position indicates the wavelength of the FBG, and its Y axis height reflects the intensity (x25 core) or relative intensity (x30 core) of the reflected, detected signal. The thick horizontal bases of the FBG indicators on the Spectral Chart display the wavelength bin for each FBG. Optional vertical dashed line indicators can also be enabled to help visualize the FBG position relative to their respective wavelength bins. FBG spectral ranges or “bin widths” can be directly modified by dragging appropriate dashed vertical limit indicator.

i. FBG Spectral Graph Control Buttons

The following control buttons are collocated with the FBG Spectral graph:

a. Bin Limit Controls



This collection of controls can be used to incrementally move the minimum and maximum FBG bin limits either up or down by a selectable step size. These controls are most useful on a touch screen deployment of ENLIGHT, or for fine tuning of FBG bins for setting exact sensor ranges (details on this process follow in a later section of this document.)

b. Zoom X



Allows for zooming of the X axis on the FBG Spectral graph.

Left mouse button double clicking will Zoom In on the FBG Spectral graph. Right mouse button double clicking will Zoom out.

c. Pan X



Allows for panning of the X axis on the FBG Spectral graph.



d. Autoscale X



Sets the X scale to the full range of the measurement unit.

e. Show FBG Limits



Shows/hides the dashed, vertical line indicators that highlight the upper and lower wavelength limits of each FBG's spectral bin.

f. Hide/Show FBGs



Hides/shows the entire FBG section of the Sensors tab. When hidden, the Sensor management tools from the bottom of the Sensors tab expand to fill the entire tab. FBG controls can again be shown by either clicking the Hide/Show FBG control or any of the unselected Boolean channel controls.

11.3.3.2. Sensors

The bottom half of the Sensors tab is equipped with the tools to create and manage sensor elements.

Sensors

ID	Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.	
✗ tower_temp_os4100	C	-----	-59.948	-59.460	2.702	99.818	100.331
✓ tower_strain_os3100	με	-59.9/2.7/100.4	-3755.350	-2554.555	-69.475	2539.949	2545.891
✓ pile_temp_os4200	C	-----	-40.103	-20.000	25.725	100.000	120.087
✓ deck_strain_os3600	με	-30.0/-3.1/30.0	-1250.260	-1000.000	-207.308	1000.000	1250.179

Filter: All Group 1 Refresh .moi .txt

A. Table

ID	Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.	
✗ tower_temp_os4100	C	-----	-59.948	-59.460	2.702	99.818	100.331
✓ tower_strain_os3100	με	-59.9/2.7/100.4	-3755.350	-2554.555	-69.475	2539.949	2545.891
✓ pile_temp_os4200	C	-----	-40.103	-20.000	25.725	100.000	120.087
✓ deck_strain_os3600	με	-30.0/-3.1/30.0	-1250.260	-1000.000	-207.308	1000.000	1250.179

The Sensors table summarizes the attributes of the existing sensors for the measurement system. An individual sensor can be selected by clicking anywhere on its corresponding row. Traditional shift-click and control-click methods can be employed to select multiple sensors for further action.



The parameters that the table reports for each sensor are as follows:

- i. ID:
The ID of the sensor will be used for all charting, image displays, event logs, and data files in later tabs of the software. IDs can be any unique alphanumeric, and ENLIGHT will automatically replace any space in that ID with an underscore. Strictly numeric IDs may not be used, as they confuse the expression parser. It is not recommended to use Sensor IDs that may be duplicated by the automatic FBG variable ID naming convention (e.g. "FBG_A1" or "FBG_F22".)

NOTE: Sensor IDs with shorter lengths display better in the limited space of sensor indicators on the Image tab.

- ii. Units Column (unlabeled):
Displays the units that correspond to the measurement type for each of the defined sensors (strain, temperature, pressure, acceleration, displacement, wavelength, and custom)
- iii. Relative T:
Displays the relative temperature range over which the sensor can operate, based upon the range afforded the DeltaT sub-expression by its corresponding FBG's defined wavelength bin. Details outlining the relationships between FBG bins and sensor thermal/measurement ranges follow in section 7.3.3.3 of this manual.
- iv. Range Min
Displays the minimum value of the measurement range limit defined for the sensor. This value is calculated from the available wavelength spectrum for each of the sensor's components FBGs, sub-expressions, and other constituent sensors via the sensor calibration expression.

NOTE: The minimum and maximum sensor ranges are calculated across all possible combinations of all component optical sensor expressions and sub-expressions, with respect to each component FBG's full wavelength bin, taking compensation correlation relationships into account.

NOTE: If a sensor range is calculated before one of its component FBGs has been measured, a range min/max of "Infinity" may result. In this event, the user can zero the sensor after the FBG is present and measured and allow ENLIGHT to recalculate the range.

NOTE: If a component FBG's bin limits are not sufficiently wide for a compensated macro-sensor to cover both the required temperature range and any measurement range (strain, displacement, etc.), the range indicators will show an entry of three dashes (- - -). A real, achievable numerical range can be realized by either increasing the wavelength bin width of the measurement FBG or by decreasing the required relative temperature range for the temperature reference sensor/FBG.

- v. Alarm Min:
Displays the minimum alarm limit for the sensor, as defined by the user. A current value less than this limit will constitute an alarm condition.
- vi. Current:
Displays the current measured/calculated value for the sensor.
- vii. Alarm Max:
Displays the maximum alarm limit for the sensor, as defined by the user. A current value greater than this limit will constitute an alarm condition.
- viii. Range Max:
Displays the maximum value of the measurement range limit defined for the sensor. This value is calculated by ENLIGHT from the available wavelength spectrum for each of the sensor's components FBGs via the sensor calibration expression.



B. Sensor Function Buttons

a. Create and Delete Sensor Controls

i. Edit Sensor



Launches a data entry form to manage the attributes of the selected sensor.

ii. Add Sensor



Creates a new sensor element and launches a data entry form to for the user to input the attributes of that new sensor.

iii. "Create From Definition"



Creates a new sensor element using pre-defined calibration and thermal compensation templates for MOI manufactured optical sensors. Please see section "7.3.3.4. Creating Sensors from MOI Definitions (*.mst) Files" for details.

iv. "Create From Existing Sensor"



Creates a new sensor element based upon an existing and highlighted sensor. Please see section 7.3.3.6 for an example of this feature in use.

v. Delete Sensor



Deletes the selected sensor.

vi. Delete All Sensors



Deletes all of the sensors from the current configuration.

b. Manage Limits and Alarms



i. Activate/Deactivate Sensor



Flags the selected sensor as “active”, thereby individually qualifying the sensor for participation in alerting events. An active sensor is also indicated by a blue check mark on the table, just to the left of the sensor name. An inactive sensor is indicated with a red “X”.

ii. Reset Sensor Alarm



Restores the table entry for the selected sensor to its pre-warning and pre-alarm state. Re-qualifies the selected sensor for participation in alerting events.

iii. Reset All Sensor Alarms



Restores the table entries for all sensors to the pre-warning and pre-alarm states. Re-qualifies all sensors for participation in alerting events.

iv. Zero Sensor



Zeroes the selected sensor(s) sensors by zeroing all of the FBG values in the sensor expression. See section 7.3.3.5 of this document for more detail.

v. Zero All Sensors



Zeroes all sensors by zeroing all of the component FBG values in all of the sensor expressions. See section 7.3.3.5 of this document for more detail.

vi. Set Sensor Limits to Range



Sets the alarm limits equal to the minimum and maximum calculated sensor ranges for the selected sensor.

c. Manage Sensors table

i. Import Sensor from MOI Config File



Imports the Sensors table and definitions from an existing MOI configuration file. This function can be very useful when migrating from one interrogator platform to another on a given measurement system (e.g. from sm125 to sm130).

ii. Save Sensors to Text File



Captures the current sensor configurations and makes it available as a file via the import/export tool.

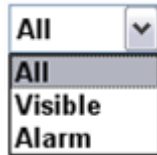
Exported Sensor configuration file

	A	B	C	D	E	F	G	H
1	Name	Active	Type	Warn	Min	Current	Max	Expression
2	saddl_temp	TRUE	Temperature	0.8	20.44909707	21.0486	21.2225	0.0399651332531352*B1^3-181.791772316077*B1^2+
3								

iii. Filter Sensors table

Allows the user to reduce the number of sensors displayed on the table based upon several criteria in order to more easily view certain segments of the full sensor collection. Reducing the number of displayed sensors also lowers CPU usage and improves system performance.

1. Filter All/ /Visible/Alarm



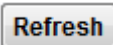
Allows the user to choose among displaying “All” sensors, only sensors which have been declared as “Visible”, or only sensors that are presently or have recently been in an alarm state without a reset action applied.

2. Filter Groups



Allows the user to display a configurable subset of the full sensor collection by selecting one of the preconfigured groups. Sensors can be assigned to a group using the Edit Sensor window. Ten independent groups are provided for organizing and filtering sensor collections.

3. Refresh





Allows the user to view an updated list of sensors in an alarm state when the sensor Filter displays Alarm. All sensors that have gone in and out of an Alarm state will be removed from the list when the Refresh button is clicked.

C. Editing and Adding Sensors

Range
 Min: -65.091 Max: 110.552 Current: 19.849

Parameters
 Active Visible Rate Of Change
 ID: TEMP_1
 Group:
 Type: Temperature
 # Averages: 100
 Minimum: -10.000
 Maximum: 60.000
 Warn Thresh.: 0.80 (0.0 - 1.0)
 Expression: C3*(T8_1+Lambda0S)^3+C2*(T8_1+Lambda0S)^2+C1*(T8_1+Lambda0S)+C0
 Expression is valid

Name	Value
Lambda0S	0
C3	3.687872179219...
C2	-17362.199485356
C1	27246689.18031...
C0	-14252887035.1...

Sub-Expressions

ID	Expression	Compensation
		None
		None
		None

OK Cancel Apply

This Edit Sensor dialog is a data entry form used to set/modify the attributes for the selected sensor element. This form is launched by use of the “Add Sensor” and “Edit Sensor” buttons, as well as a double click event on one of the sensor entries on the table, in the case of a desktop deployment.

a. Setup

i. Range

This section of the data entry form displays the calculated range minima and maxima of the sensor elements.

1. Minima and Maxima

Min: Displays the calculated minimum value of the range for the selected sensor.

Max: Displays the calculated maximum value of the range for the selected sensor.

2. Set Limits to Range



Sets the alarm limits equal to the minimum and maximum of the calculated sensor range for the selected sensor.

3. Current

Shows the current value of the selected sensor.

ii. Parameters - Defining Additional Parameters on Main Tab

Many of the basic sensor parameters (ID, Relative T, Range Min/Max, Alarm Min/Max) that are set with the sensor data entry form were defined in the explanation of the Sensors table. Additional sensor attributes managed by the data entry form include:

1. Active

 Active

A sensor is declared as “Active” for purposes of Sensors table display and alerts management. A sensor that is made Active is marked with a ‘check’ whereas an inactive sensor is marked with an ‘X’ in the Sensors table display. Only Active sensors will produce alerts in the Events log in the Alerts tab.

2. Visible

 Visible

A sensor is declared as “Visible” for purposes of Sensors table display. Sensors which are not specified as visible are seen only when the display filter is set to All and are seen in lighter colors of gray and light blue, as shown by sensor “tower_temp_os4100” in the following image.

ID	Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.	
<input checked="" type="checkbox"/> tower_temp_os4100	C	-----	-59.948	-59.460	2.769	99.818	100.331
<input checked="" type="checkbox"/> tower_strain_os3100	µε	-59.9/2.8/100.4	-3755.350	-2554.555	-71.203	2539.949	2545.891
<input checked="" type="checkbox"/> pile_temp_os4200	C	-----	40.103	-20.000	25.750	100.000	120.087
<input checked="" type="checkbox"/> deck_strain_os3600	µε	-30.0/-3.1/30.0	-1250.260	-1000.000	-208.325	1000.000	1250.179

3. Rate of Change

 Rate Of Change

When the Rate of Change box is selected, a 5th order Savitsky-Golay filter is used to compute the first derivative for that sensor. When a sensor has Rate of Change selected, all display and alarming functions for that sensor will reference the derivative of that sensor and not its actual value. Sensors will be displayed in purple on the Sensors table when in Rate of Change mode.

NOTE: Any macro-sensor that references a sensor in Rate of Change mode in its expression will operate on the actual value of that sensor and not its derivative.

4. Averages

Averages:

Sets the number of averages to be applied to that particular sensor. With sensor averaging, multiple sensors can reference a single un-averaged FBG and not be uniformly subject to averaging implemented at the FBG level. Averaging is supported at both the FBG and Sensor levels and can be used in combination to achieve the desired degree of noise reduction and time response that is optimal for each application.

5. Group

Group: ▼

Allows the user to assign the selected sensor to one of ten independent pre-defined sensor groups, for purposes of display filtering on the Sensors table. In addition to organizing and simplifying Sensors table viewing, use of Groups and the associated table filter will reduce the number of displayed Sensors, which lowers CPU usage and improves system performance. Individual sensors may be assigned to maximum of one group.



6. Type

Type:

Assigns the type of the defined sensor (strain, temperature, pressure, acceleration, displacement, wavelength, and custom) and assigns the associated default units for that sensor type.

7. Constants

Constants

Name	Value
Lambda0S	-23.7413
C3	4.13980947477103
C2	-18843.3096330642
C1	28589996.1036361
C0	-14459427071.5871

The Constants table allows the user to define Names and Values for constants that are used in sensor expressions and sub-expressions, such that standard sensor calibration equations can be conveniently employed in the sensor expression field. Constants defined for a sensor are unique to that sensor and cannot/will not be referenced by other sensors or their expressions.

To add a constant, the user will enter a Constant name in the data input box under the Name column of the Constants table and enter a value in the data input box under the Value column of the Constants table, then click on the Update button. To edit or delete a constant, the user will first click on the table to highlight a line, and then use the Update or Delete button to complete the action.

8. Expression:

The Sensor Expression is used to calculate a new quantity (most likely a physical quantity such as temperature, strain, etc.) from a combination of one or more FBG's, sub-expressions, and/or other Sensors.

FBG IDs and other Sensor IDs can be used as variables in the expression. Portions of the expression may be grouped to indicate how the calculation should be performed using parentheses (), brackets [], or braces {}.

Expressions can also reference other sensors to form macro-sensors. There is no explicit limit on the depth of recursion that can exist in macro-sensors.

(1) Functions supported by the Sensor expression parser:

The following arithmetic operators are supported: '+', '-', 'x', '/', '^'.

The following Boolean operators are supported: '<', '>', '=', '&', '|', '!', '<>', '>=', '<='.

Strings concatenation is supported with the following operators: '+', '&'.

The following functions are also available:

ABS:

Calculate the absolute value of a number [usage: ABS(X)].

SIGN:

Calculate the sign of a number [usage: SIGN(X) returns -1 if X < 0, +1



if $X > 0$, 0 if $X = 0$]

TRUNC:
Discards the fractional portion of a number [usage: TRUNC(-3.2) returns -3, TRUNC(3.2) returns 3].

CEIL:
Return the smallest integer value that is greater than a specified value [usage: CEIL(-3.2) returns 3, CEIL(3.2) returns 4].

FLOOR:
Return the largest integer value that is less than a specified value [usage: FLOOR(-3.2) returns -4, FLOOR(3.2) returns 3].

SQR:
Calculate the square of a number [usage: SQR(X)].

SQRT:
Calculate the square root of a number [usage: SQRT(X)].

INTPOW:
Raise Base to an integral power. [usage: INTPOW(2, 3.2) returns 8].

POW:
Raise Base to any power [usage: POW(2, 3.2) returns 9.19].

EXP:
Raise e to a specified value [usage: EXP(X)].

LN:
Calculate the natural log of a number [usage: LN(X)].

LOG:
Calculate the base 10 log [usage: LOG(X)].

LOGN:
Calculate the log base N of X [usage: LOGN(10, 100) = 2].

SIN:
Calculate the sine of the angle X (radians) [usage: SIN(X)].

COS:
Calculate the cosine of the angle X (radians) [usage: COS(X)].

TAN:
Calculate the tangent of the angle X (radians) [usage: TAN(X)].

ASIN:
Calculate the arcsine (radians) of X [usage: ASIN(X)].

ACOS:
Calculate the arccosine (radians) of X [usage: ACOS(X)].

ATAN:
Calculate the arctangent (radians) of X [usage: ATAN(X)].

SINH:
Calculate the hyperbolic sine of the angle X (radians) [usage: SINH(X)].

COSH:
Calculate the hyperbolic cosine of the angle X (radians) [usage: COSH(X)].

COTAN:
Calculate the cotangent of the angle X (radians) [usage: COTAN(X)].

VAL:
Returns the floating point numeric value of the string argument [usage: VAL("3.1") returns 3.1].

MIN:
Returns the minimum of two values [usage: MIN(2,3) returns 2].

MAX:
Returns the maximum of two values [usage: MAX(2,3) returns 3].

SUM:
Returns the sum of its arguments [usage: SUM(2,3,5,) returns 10.
There is no preset limit on the number of parameters.

IF:
The IF(b, case1, case2) function provides branching capability. If b is not 0, then it returns case1, else it returns case2 [usage: IF(HEIGHT, 3/HEIGHT, 3) will make sure 3/HEIGHT does not cause division by zero].



RND:

Returns a random floating point number between 0 and 1 [usage: RND()].

STR: Returns the string representation of the passed value [usage: STR(123) returns "123"].

SUBSTR:

Returns a substring. The first parameter is the string, the second parameter is the starting index (0 based), and the last parameter is the number of characters to copy [usage: SUBSTR("Hello", 1,3) returns "ell"].

STRLEN:

Returns the length of the specified string. [usage: STRLEN("abc") returns 3].

CONCAT:

Returns the concatenated strings [usage CONCAT("abc","def",...) returns "abcdef". There is no preset limit on the number of parameters].

TRIM:

Removes all leading/trailing spaces from a specified string [usage: TRIM(" abc ") returns "abc"].

(2) FBG calculation shorthand notations

Many sensor expressions are calculated from referenced or normalized values of FBG wavelengths. Take for example a basic constant temperature strain sensor:

$$\text{Strain } (\mu\epsilon) = 1e6 * ((\lambda - \lambda_0) / \lambda_0),$$

where λ is the current value of the FBG wavelength λ_0 and is the FBG reference wavelength.

This expression can be greatly simplified through the use of the following shorthand notations. Assuming X is the FBG name:

X_0: The reference value when the FBG was zeroed (for a given Sensor)

X_D: X - X_0, the current value minus the reference value.

X_N: X_D / X_0 which expands to (X - X_0) / X_0 (useful for strain expressions)

X_F: The "filtered" version of the FBG value (must set Alpha on the FBG setup dialog...larger alpha = slower response)

Thus, the above expression can be simplified to the following:

$$\text{Strain } (\mu\epsilon) = 1e6 * \lambda_N.$$

These FBG calculation shorthand notations are used extensively in the MOI sensor configuration templates described later in this section of this document.

NOTE: A sensor is automatically zeroed and its component FBG reference values recorded when that sensor is created.

iii. Sub-Expressions

Sub-Expressions		
ID	Expression	Compensation
<input type="text"/>	<input type="text"/>	None
<input type="text"/>	<input type="text"/>	None
<input type="text"/>	<input type="text"/>	None

Sub-expressions are mathematical definitions of quantities that are used by the main sensor expression to calculate the engineering quantity desired.

1. ID

The ID of the sub-expression will be used for reference in the sensor expression that operates on it. IDs can be any unique alphanumeric, and ENLIGHT will automatically replace any space in that ID with an underscore. Strictly numeric IDs may not be used, as they confuse the expression parser. It is not recommended to use IDs that may be duplicated by the automatic FBG variable ID naming convention (e.g. "FBG_A1" or "FBG_F22".)

2. Expression

The expression is used to calculate a quantity based upon the values of FBGs or other sensor elements.

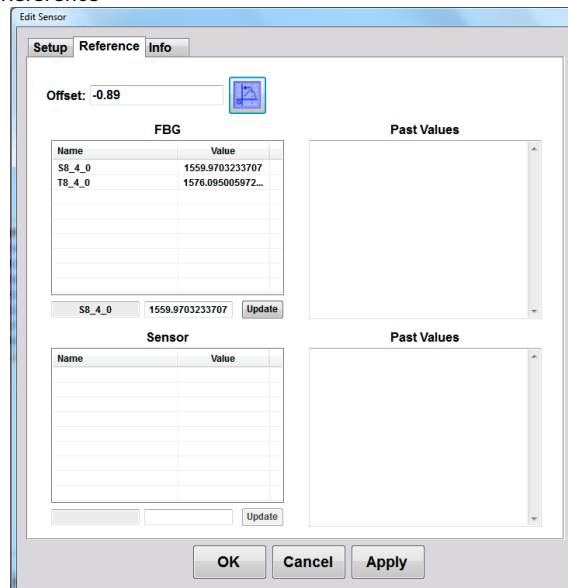
3. Compensation (Correlation)

Compensation correlation is the notion that a known positive or negative relationship exists between the operating spectral migration of a sub-expression and its calling sensor's expression. The compensation control sets the correlation between the sub-expression and its calling expression. The effect of this compensation correlation relationship is defined in the examples of 7.3.3.3 and 7.3.3.4.

iv. Warn Thresh.:

The fractional span of the total alarm range which is defined as a warning threshold (as a prelude to the alarm threshold). For example, if the Alarm Min is set to -1000, the Alarm Max is set to 1000, and the warning threshold is set to the default value of 0.8, then warning alerts would be asserted when the sensor value reached min/max warning limits of -800 and 800, respectively.

b. Reference



The Reference tab on the Edit Sensor window displays the composition of a sensor referencing multiple FBG or sensor elements.

i. Offset

Displays the compensation value added to or subtracted from the net sensor value when the user clicks the Zero The Sensor button.

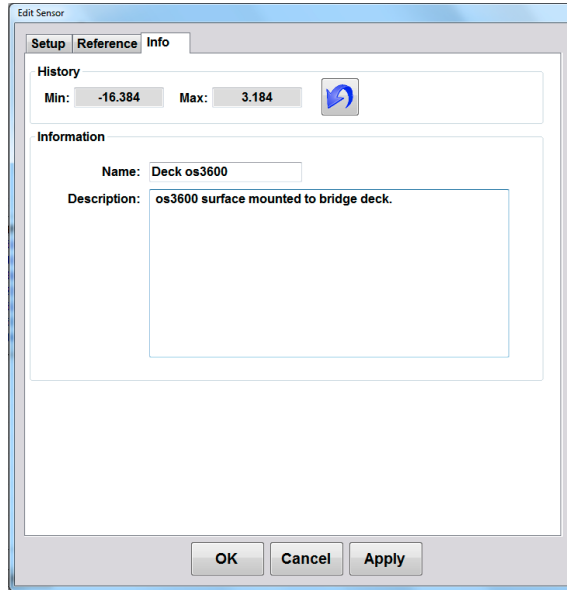
ii. FBG

Displays the FBG names and current wavelength values used to compose the sensor.

iii. Sensor

Displays the Sensor names and current engineering unit values used to compose the sensor.

c. Info



i. History



Displays some simple statistics on recent sensor values

1. Min: displays this lowest recorded sensor value since the last reset.
2. Max: displays the highest recorded sensor value since the last reset
3. Reset:



Resets the recorded statistics on the selected sensor.

ii. Information

1. Name:
Allows assignment of a Sensor Name (independent from the Sensor) ID to the sensor element that will be saved with the sensor data in any logged data files.
2. Description:
Provides a data entry field for a descriptive summary of that sensor element that will be saved with the sensor name in any logged data files.

D. Moving Sensors on the Sensors table

Once defined, the order in which sensors appear in the Sensors table can be changed to optimize sensor and macro-sensor organization.

To move a sensor entry on the tab, click and select that entry with the mouse or touch screen. Once selected, the entry will be highlighted with a gray background, as is seen in the following figure and the Move Sensor arrow buttons on the bottom right of the chart will be enabled.



ID		Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
✓ tower_temp_os4100	C	-----	-59.948	-59.460	2.726	99.818	100.331
✓ tower_strain_os3100	μϵ	-60/2.7/100	-3755.350	-2554.555	-69.396	2539.949	2545.891
✓ pile_temp_os4200	C	-----	-40.103	-20.000	25.723	100.000	120.087
✓ deck_strain_os3600	μϵ	-30/3.1/30	-1250.260	-1000.000	-208.922	1000.000	1250.179



Note that the Move Sensor buttons require that all sensors be shown before the buttons can be enabled. This measure ensures that hidden sensors are not overlooked when the list is organized. The following figure shows the same list as above, with the sensor “deck_strain_os3600” reordered to the top of the table.

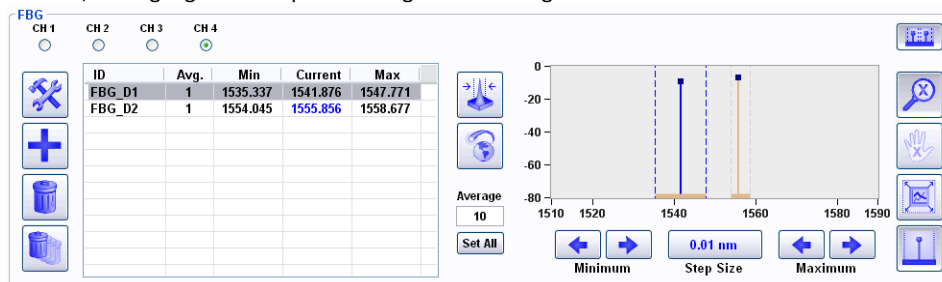
ID		Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
✓ deck_strain_os3600	μϵ	-30/3.1/30	-1250.260	-1000.000	-208.922	1000.000	1250.179
✓ tower_temp_os4100	C	-----	-59.948	-59.460	2.726	99.818	100.331
✓ tower_strain_os3100	μϵ	-60/2.7/100	-3755.350	-2554.555	-69.396	2539.949	2545.891
✓ pile_temp_os4200	C	-----	-40.103	-20.000	25.723	100.000	120.087



E. Navigation tools and color-coded visual cues.

a. Identifying relationships between Sensors, FBGs, and Channels.

- i. Selecting an FBG from the table highlights the corresponding FBG on the FBG spectral graph. Likewise, right clicking near a sensor or dragging a sensor’s limits on the graph highlights that FBG on the table. Note that if an FBG is selected, the highlighted FBG plot is changed from orange to blue.



In the image above, the FBG “FBG_D1” has been selected, and its image has been highlighted on the graph in blue to aid with visual association. Also note that the FBG bin limits for all other FBG entries have been grayed out, further highlighting the selected FBG and its bin controls.

- ii. Selecting a sensor from the Sensors table highlights the channel or channels on which that sensor has component FBGs. For example, if a temperature compensated strain sensor has its strain measurement FBG on channel 1 and



its temperature compensation FBG on channel 4, then when the sensor is selected on the Sensors table, the channel controls above the FBG chart will highlight channels 1 and 4 with a blue label.



ID	Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
✓ tower_temp_os4100	C	-----	-59.948	-59.460	2.652	99.818
✓ tower_strain_os3100	µε	-60/2.7/100	-3755.350	-2554.555	68.465	2539.949
✓ pile_temp_os4200	C	-----	-40.103	-20.000	25.778	100.000
✓ deck_strain_os3600	µε	-30/3.1/30	-1250.260	-1000.000	-210.321	1000.000
✓ deck_strain_os3100	µε	-66/0.0/94	-3630.469	-----	0.035	2672.485

In the image above, the sensor “deck_strain_os3100” has been selected, and the selector controls for channels 1 and 4 have been highlighted in blue to indicate that those two channels house component FBGs of the selected sensor.

When a channel with a component FBG of the selected sensor is shown, the component FBGs of that selected sensor are highlighted in both the FBG table and graph with blue accents.

ID	Avg.	Min	Current	Max
FBG_D1	1	1535.337	1541.876	1547.771
FBG_D2	1	1554.045	1555.854	1558.677

ID	Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
✓ tower_temp_os4100	C	-----	-59.948	-59.460	2.642	99.818
✓ tower_strain_os3100	µε	-60/2.7/100	-3755.350	-2554.555	-68.226	2539.949
✓ pile_temp_os4200	C	-----	-40.103	-20.000	25.697	100.000
✓ deck_strain_os3600	µε	-30/3.1/30	-1250.260	-1000.000	-210.872	1000.000

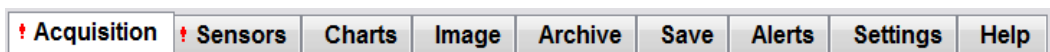
In the images above, the sensor “tower_strain_os3100” has been selected, and channel 4 has been chosen for display in the FBG table and chart. The FBGs “FBG_D1” and “FBG_D2” have been highlighted in the table with blue labels to indicate that they are computational components of the sensor “tower_strain_os3100”. Also, the FBG indicators on the graph which corresponds to “FBG_D1” and “FBG_D2” have been highlighted in blue and gray respectively (as opposed to non-highlighted orange) to indicate their association with the selected sensor “tower_strain_os3100.” The blue color highlights a fundamental FBG of the sensor expression, while the gray color highlights an FBG used in the thermal sub-expression.

b. Use of red color cues for diagnostics and troubleshooting.

The color red is used throughout the ENLIGHT user interface to indicate warning or alarm conditions which require user attention. The following sections will outline some of the indicators and present them in a natural order that might be used to diagnose a sensor installation or configuration problem.

i. Tab Error Indicators

Visible from all operational tabs are the tab error indicators. These red exclamation point icons are visible whenever an error condition exists on the corresponding tab. In the case of the image below, there are error conditions on both the Acquisition and Sensors tab which require attention.



ii. Sensor Warning/Alarms

ID		Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
✓ tower_temp_os4100	C	-----	-59.948	-59.460	1.740	99.818	100.331
✗ tower_strain_os3100	µε	-60/2.6/100	-3755.350	-2554.555	-----	2539.949	2545.891
✓ pile_temp_os4200	C	-----	-40.103	-20.000	25.680	100.000	120.087
✓ deck_strain_os3600	µε	-30/-3.2/30	-1250.260	-1000.000	-212.297	1000.000	1250.179

On the Sensors tab, the red exclamation point to the left of the sensor informs the user that there has been some sort of warning or alarm event for that sensor. A following section will detail what the various colors and positions of the indicators mean within the Sensors table.

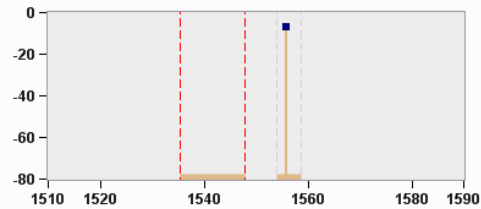
iii. Channel Alarm Indicators



Regardless of which channel is currently selected for viewing in the FBG table and graph, any channel with an alarm condition will be highlighted among the channel selector controls. In this way, the user is able to rapidly navigate to the source of the reported problem.

iv. FBG Alarm Indicators

ID	Avg.	Min	Current	Max
FBG_D1	1	1535.337	-----	1547.771
FBG_D2	1	1554.045	1555.828	1558.677

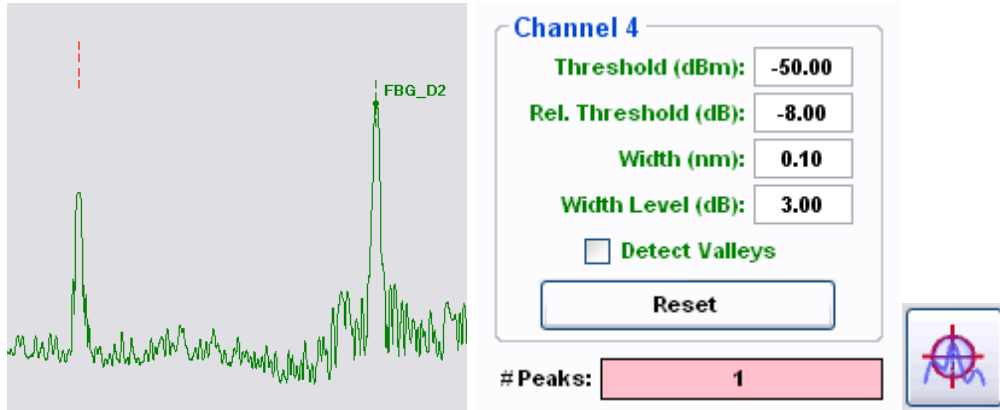


When a channel with a red label is selected for viewing on the FBG table and graph, two additional uses of red indicators help to guide the user to a diagnosis. First, the Current Wavelength entry for the offending FBG will be highlighted to indicate if the FBG is out of range. Secondly, once that FBG is selected for highlighting in the graph, the vertical range indicator lines that correspond to that FBG will be highlighted in red.

If an FBG’s current wavelength value either exceeds its min or max wavelength limit or cannot be read, then no value for current wavelength will be reported. Note, however, that the FBG is still represented in both the table and spectral graph above. ENLIGHT maintains the concept of the created FBG, even if the peak that was used to create the FBG is no longer present. In that way, the removal of a single FBG from the measured data will not have a “shift” or “ripple effect” on other FBG values that surround it. The required averaging and referencing operations will continue to run correctly on the remaining FBGs. The sensors that are calculated based upon the indexed FBGs will not be subject to a wholesale shifting of FBG indices. This simple concept and its implementation constitute one of the subtle, but key strengths of the MOI ENLIGHT software for FBG sensor systems.

v. Peak Wavelength Tracking errors

As mentioned in the Acquisition tab section 7.3.2, peak wavelengths are tracked by index within the x30 interrogator cores and by ENLIGHT for x25 cores. When the number of tracked peaks is not equal to the number of detected peaks, a red background is used to indicate a problem. Likewise, the “target” color on the “Reset Wavelength Tracking” button turns from blue to red, and a red dashed vertical line is used to indicate the relative spectral position of the expected tracked peak.



Following the example further, it is seen that the first spectral peak on channel 4 (which corresponds to FBG “FBG_D1”) has fallen below the Rel. Threshold Level of -8 dB. With this information, the user can either seek to correct the root cause of the attenuation or change the peak detection parameters in order to re-enable detection of the missing spectral peak.

The colors yellow and red are also used to convey conditions of warning and alarm, respectively, on the indicators of the Charts and Image tabs, which will be introduced in a later section.

c. Sensor warning and limit indicators on Table (with persistence)

The Sensors table also uses background colors to indicate a current or past warning or alarm state for a sensor. The next four examples show how sensor warning and alarm events are indicated on the Sensors tables.

i. Sensor currently in warning (negative)

ID		Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
✓tower_temp_os4100	C	-----	-59.948	-59.460	0.030	99.818	100.331
✓tower_strain_os3100	µε	-60/0.1/100	-3755.350	-2554.555	-20.945	2539.949	2545.891
✓pile_temp_os4200	C	-----	-40.103	-20.000	25.805	100.000	120.087
!deck_strain_os3600	µε	-30/-3.1/30	-1250.260	-1000.000	-812.512	1000.000	1250.179

In the figure above, the macro-sensor “deck_strain_os3600” shows that it has been in a warning state on the low end of its range, as is indicated by the yellow background on the “Alarm Min” entry, and is still in the warning state, as is indicated by the yellow background on the “Current” value entry.

ii. Sensor currently in alarm (greater than maximum) with warning history (less than minimum)

ID		Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
✓tower_temp_os4100	C	-----	-59.948	-59.460	0.030	99.818	100.331
✓tower_strain_os3100	µε	-60/0.1/100	-3755.350	-2554.555	-20.945	2539.949	2545.891
✓pile_temp_os4200	C	-----	-40.103	-20.000	25.805	100.000	120.087
!deck_strain_os3600	µε	-30/-3.1/30	-1250.260	-1000.000	1169.139	1000.000	1250.179

In the figure above, the macro-sensor “deck_strain_os3600” shows that it has been in a warning state on the low end of its range, as is indicated by the yellow background on the “Alarm Min” entry, and has been in an alarm state on the high end of its range, as is indicated by the light red background on the “Alarm Max” entry. The sensor is still in an alarm state, as is indicated by the light red background on the “Current” value entry.

iii. Alarm Reset

ID		Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
✓tower_temp_os4100	C	-----	-59.948	-59.460	0.294	99.818	100.331
✓tower_strain_os3100	µε	-60/0.3/100	-3755.350	-2554.555	-15.756	2539.949	2545.891
✓pile_temp_os4200	C	-----	-40.103	-20.000	25.741	100.000	120.087
✓deck_strain_os3600	µε	-30/-3.2/30	-1250.260	-1000.000	-207.690	1000.000	1250.179

Targeted use of the “Reset Sensor Alarm” button or blanket use of the “Reset All Sensor Alarms” button will restore the Sensors table to its pre-warning and pre-alarm state, and re-qualify all sensors that were originally configured to participate in alerting events (active sensors).

11.3.3.3. Sensor Operating Temperature Range: A Manually Created Macro-Sensor Example

Fiber Bragg Gratings elements are intrinsically responsive to both temperature and strain stimuli. In order to extract exclusively mechanical strain (or other transducer derived) information from an FBG based sensor, some form of temperature compensation must be performed. There are a number of different methods that can be used for different applications. In this macro-sensor example, a separate temperature sensor is used to measure the local temperature in the area of the FBG strain sensor. That temperature data is used to estimate the thermally induced strain on the FBG strain sensor, defined as the thermal component of strain. Finally, the thermal component of strain is subtracted from the total measured strain, and the result is mechanical strain. This example also demonstrates how to choose the correct compensation correlation setting for the sensor’s sub-expressions, as well as how to choose the FBG bin widths to facilitate the correct measurement and operating temperature ranges for the macro-sensor. More information on temperature compensation can be found at <http://www.micronoptics.com/support/downloads>.

ID	Avg.	Min	Current	Max
FBG_D1	1	1540.877	1541.876	1542.877
tower_1	1	1554.764	1555.763	1556.764

First, a temperature sensor is identified as source of the temperature compensation information for the macro-sensor. This could take the form of either a relative or an absolute temperature sensor. For this example, FBG_D1 is identified as the component FBG of that temperature sensor.

Note that the default wavelength bin width of 2nm has been applied to FBG_D1.

The second defined FBG on channel 4 represents the strain measurement FBG. Suppose for this example that this FBG has been built into strain sensor housing and is responsive to both strain and temperature. It has been designated by the custom FBG ID “tower_1”.

The strain sensor element created to reference the FBG “tower_1” requires both the relative wavelength change of the FBG and its nominal wavelength as part of the calibration. The FBG calculation shorthand notation for accessing these quantities will be used in and demonstrated by this example.

The first sensor created is the relative temperature sensor. In this example, the FBG based temperature sensor is called “tower_temp_os4100” to reflect the relative temperature change of the tower of a target bridge. The sensor is designated as a type “Temperature” and a calibration expression is applied that incorporates the delta wavelength of the FBG as referenced by the shorthand notation “FBG_D1_D”. This component-specific calibration expression would be supplied to the user as part of a qualified FBG-based relative temperature sensor.

Range

Min: -64.505 Max: 136.626 Current: ---

Parameters

Active Visible Rate Of Change

ID: tower_temp_os4100

Group: [dropdown]

Type: Temperature [dropdown]

Averages: 100

Minimum: -10.000

Maximum: 60.000

Warn Thresh.: 0.80 (0.0 - 1.0)

Expression: $C3*(FBG_D1_D)^3+C2*(FBG_D1_D)^2+C1*(FBG_D1_D)+C0$

Expression is valid

Name	Value
Lambda0S	4.2336
C3	3.471656322026...
C2	-16427.2422292...
C1	25910320.26410...
C0	-13622633762.3...

Update Delete

ID	Expression	Compensation
		None [dropdown]
		None [dropdown]
		None [dropdown]

OK Cancel Apply

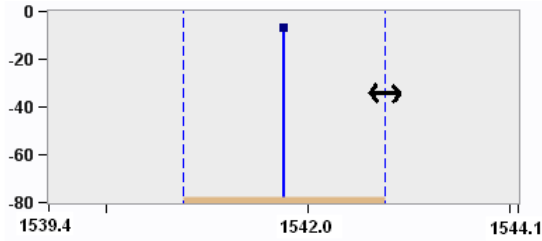
Note that this sensor was created using both the Constants table and the FBG calculations shorthand notation X_D .

When the expression is applied to the variable, the range for the resulting sensor is calculated in engineering units. The range is calculated over the full possible wavelength range or ranges of the components FBGs for the sensor and any of its component sub-sensors or sub-expressions. In the case of this simple relative temperature sensor, the sensor calculation is dependent on only one FBG, and the resulting sensor range represents the span of the expression as evaluated over the full range of the defined FBG: FBG_D1.

Note that the default width of 2nm assigned to FBG D1 has been used to calculate the range of the sensor "tower_temp_os4100", yielding a relative temperature range of -34.618 degrees C to 34.585 degrees C. The user would likely note that the calculated range of the sensor does not yet match the specifications for the sensor from the datasheet or the desired operating temperature range for the macro-sensor. This means that the default 2nm FBG range must be modified until the resulting sensor range matches the expected and desired measurement range of that sensor.

To modify the sensor range, the user will close the Edit Sensor window and move up to the FBG section of the sensor tab. In order to affect the range of the computed sensor, the user will change the range of the constituent FBG used in the sensor calculation. In this example, the wavelength limits for FBG_D1 will be modified.

These limits can be changed in one of two ways. First, the Edit FBG window can be used to directly key in new wavelength limits. Alternately, and perhaps more intuitively, the bin limits for the FBG sensor can be modified directly on the FBG spectral graph by dragging the corresponding limit indicator, as shown below. When the limits are changed, the related sensor range values are instantaneously calculated to reflect that change. In this way, the user can monitor the desired measurement parameter range of temperature while editing the FBG wavelength range graphically.



The image above shows the user dragging the wavelength range maximum for FBG_D1 until the maximum bin limit of the temperature sensor “tower_temp_os4100” reads desired maximum operating delta T over ambient. Assume for this example that the current temperature is 25 degrees and the user wants to provision sufficient wavelength to operate from -20 to 100 degrees C. In this case, the user will drag the lower and upper wavelength limits of the FBG bin until the sensor “tower_temp_os4100” shows a relative temperature range of -45 to 75 degrees C.

ID	Avg.	Min	Current	Max
FBG_D1	1	1540.574	1541.875	1544.045
tower_1	1	1551.226	1555.761	1561.013

ID	Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
tower_temp_os4100	C	-45.027	-34.618	-0.005	34.586	75.091

This method of modifying sensor ranges has two principal benefits. First, it allows the user to set up the sensor measurement system in terms of the desired measurement quantities of strain, temperature, acceleration, etc. Secondly, by using the FBG wavelength bin limits as the direct controls for sensor range configurations, the user becomes aware of the FBG spacing, as it pertains to optical spectral provisioning and management.

With these steps, the first of the sensors is created.

In the next steps, the temperature data from the os4100 relative temperature sensor is used to estimate the thermally induced strain component of the nearby FBG strain sensor. The functional form for this expression is highlighted in the appendix entitled “Optical Strain Gage Temperature Compensation Techniques” in the Appendices section of this document.

Range

Min: --- Max: --- Current: ---

Parameters

Active Visible Rate Of Change

ID: tower_strain_os3100

Group: ---

Type: Strain

#Averages: 1

Minimum: ---

Maximum: ---

Warn Thresh.: 0.80 (0.0 - 1.0)

Expression: $(1e6 * tower_1_N) / Fg - EpsT0$

Expression is valid

Constants

Name	Value
Fg	0.89
CTEs	11
C1	6.156
C2	0.7

Update Delete

Sub-Expressions

ID	Expression	Compensation
EpsT0	$(DeltaT) * (C1 / Fg + CTEs - C2)$	Positive
DeltaT	tower_temp_os4100	None

OK Cancel Apply

The strain calculation for the “tower_strain_os3100” takes the form of “strain = total strain – thermal component of strain”.

In the above expression, “total strain” is calculated as

$$(1e6 * tower_1_N) / Fg$$

Note that as Fg has been defined in the Constants Table, its name can be used in the calculation expression. Tower_1_N represents the normalized wavelength $X_N = (X - X_0) / X_0$ in the FBG calculation shorthand notation.

In the above expression “thermal component of strain” is represented by the ID EpsT0 and is defined as a sub expression. EpsT0 is a function of DeltaT, which is defined as the previously created relative temperature sensor “tower_temp_os4100”. The sub-expression ID of “DeltaT” is a special sub-expression designation used exclusively for representing the relative temperature range over which the defined sensor will operate. In this example, the strain sensor “tower_strain_os3100” will be configured to operate over a temperature range determined by DeltaT, which in this example is defined as “tower_temp_os4100”.

As DeltaT defines the operating relative temperature range of the sensor, the thermal component of strain, EpsT0, will be a function of Delta T. As seen in the above example, EpsT0 is calculated as:

$$(DeltaT) * (C1 / Fg + CTEs - C2)$$

C1, Fg, CTEs, and C2 have all been defined in the Constants Table and can be referenced by name in the Expressions and Sub-expressions for this sensor. Also note that this sub-expression EpsT0 references the ID from a previously defined sub-expression: “DeltaT”.

NOTE: Find Fg and other sensor constants on the Sensor Information Sheets that accompany each MOI Tuned sensor.

With the declaration of DeltaT and EpsT0, all of the constants, sub-expressions, and expressions have been sufficiently defined such that the full expression for “tower_strain_os3100” can be calculated. However, one additional setting must

be configured in order to correctly calculate the operating temperature and strain measurement ranges for this thermally compensated macro-sensor and the required wavelength ranges for its component FBGs.

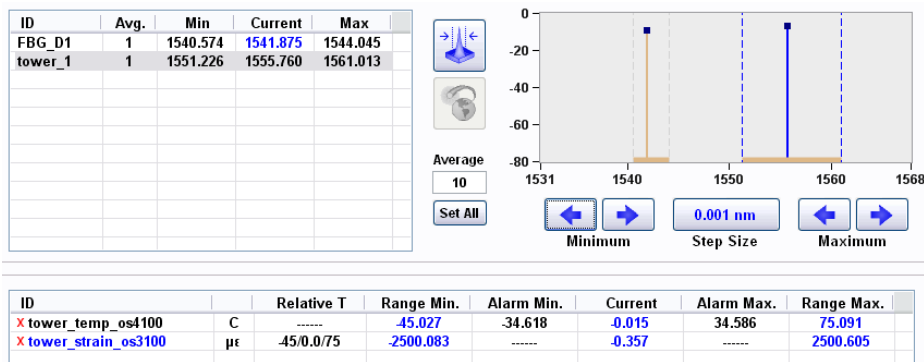
Note in the Sub-Expression section the configuration setting “Compensation”, which is used to determine how the range of a particular sub-expression will affect the resulting range of the sensor that references it. In the general case, the correlation would be set to “none”. The “none” setting is used when there is no known correlation between the changes in value of sensor’s expression and any of its sub-expressions. An example of this “lack of correlation” would be a differential temperature measurement between two disjoint parts of a system. The temperature at point A may increase while the temperature at point B may decrease, or vice versa. Both temperatures may increase, or both may decrease. The system designer imparts no expectation on the two sensors that comprise the differential temperature macro-sensor. In this case, a correlation selection of “none” would be made. The full wavelength range of both optical temperature sensors would be used to calculate the potential range of the differential macro-sensor, resulting in the widest possible range of differential temperature ranges.

If there is a known relationship between a sensor and one or more of its sub-expressions, a correlation relationship of “positive” or “negative” may be chosen. The present example of a temperature compensated strain sensor benefits from the use of a “positive” correlation relationship to best calculate the complete sensor measurement range.

In the case of this temperature compensated strain sensor, it is known that for most materials the thermally induced strain on the sensor increases with increasing temperature. Thus, it can be concluded that the estimated “EpsT0” sub-expression will change with positive correlation to the temperature compensated strain sensor “tower_strain_os3100”. Selecting the “positive” compensation parameter for sub-expression “EpsT0” will cause the range calculation to factor in the thermal requirements of the complete macro-sensor into the wavelength ranges of the strain measurement FBG (tower_1), thereby provisioning sufficient wavelength for both the strain measurements and the change in wavelength due to thermal effects. The provisioned thermal range will be governed by the temperature range of the defined relative temperature (DeltaT): “tower_temp_os4100”.

In an approach similar to that of the “tower_temp_of4100” temperature sensor previously introduced, the wavelength limits of the strain sensor “tower_strain_os3100” will be modified by the user to set the sensor measurement range to the designed limits, thereby setting both the proper range for the strain sensing element and provisioning the proper portion of optical spectrum for the operation of the component FBGs over the designed temperature range. Note that the color association guides show the strain FBG in blue while the thermal compensation FBG is shown in gray:

The sub-expression “DeltaT” is a special and unique sub-expression ID that identifies its component FBG as the thermal compensation FBG for the calling sensor, enabling the gray color association on the FBG table and defining the relative temperature range for the calling sensor, as shown on the Sensors table.



In the images above the limits for the “tower_1” FBG have been dragged to set the range of the “tower_strain_os3100” sensor to a designed +/-2500 microstrain limit, over a -45 to 75 degrees C relative operating temperature range. The selected temperature range, as well as the current temperature, relative to when the sensor was created, is displayed on the table in the Relative T column.

11.3.3.4. Creating Sensors from MOI Definitions (*.mst) Files – Multiple Examples

The example of the previous section demonstrated the sensor concepts of sub-expressions, relative operating temperature range (DeltaT), and sub-expression compensation (correlation). Additionally, the example demonstrated the use of FBG calculations shorthand notation (X_D, X_N) and the use of calibration expressions.

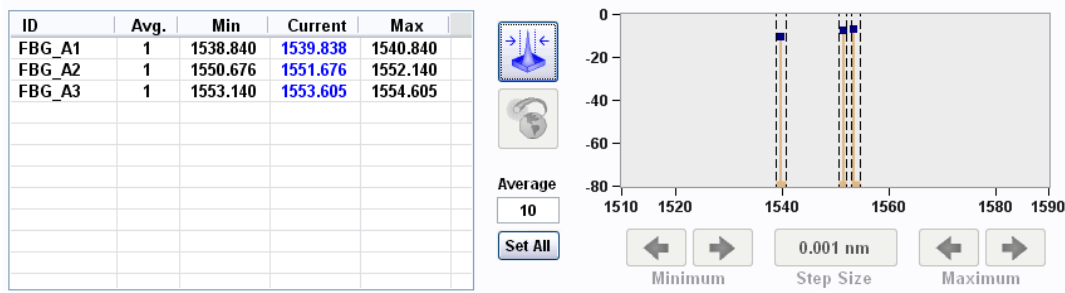
This section will show how creation of sensors can be greatly simplified through the use of MOI Definition Files. MOI definition files are distributed with ENLIGHT for MOI manufactured optical sensors. These definition files include the required calibration and compensation expressions for all MOI engineered optical sensors and the associated recommended thermal compensation approaches.

NOTE: Sensor specific calibration coefficients are distributed with the Sensor Information Sheets that accompany MOI Tuned sensors. Examples of Sensor Information sheets are available for download at www.micronoptics.com/support_downloads.

A. Example: Self-Compensated os3600 Strain Sensor

For this example, assume that two FBGs on channel 1 are known to be elements of an os3600 thermally compensated strain sensor. The following steps demonstrate how to simply create and provision wavelength for this sensor over the expected operating temperature range.

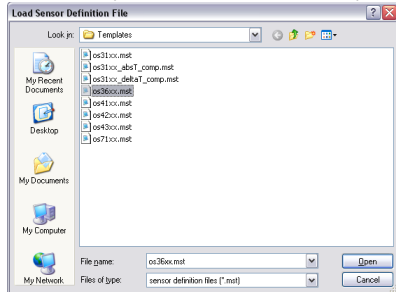
First, FBGs are automatically detected by ENLIGHT and default FBG bins are created on channel 1.



Next, on the Sensors tab, the “Create From Definition” button is pressed.



The user is presented with a list of options for *.mst definition files. For this example, the os36xx.mst file is selected.



A window with title “Create os36xx Sensor” opens.



Create os36xx Sensor

ID: Model: (Strain)

Description:

Expression:

EpsT0: (Positive)

DeltaT: (None)

Int. Expr. 3:

Constants

Fg:

CTEs:

C1:

C2:

St:

C6:

C7:

C8:

C9:

C10:

FBGs

Channel	FBG	
P2: <input type="text" value="-- All --"/>	<input type="text" value="-- Select FBG --"/>	Temp. Gage
P1: <input type="text" value="-- All --"/>	<input type="text" value="-- Select FBG --"/>	Strain Gage
FBG 3: <input type="text"/>	<input type="text" value="-- Select FBG --"/>	
FBG 4: <input type="text"/>	<input type="text" value="-- Select FBG --"/>	

Sensors

Sensor 1:

Sensor 2:

Sensor 3:

Sensor 4:

NOTE: The OK button is enabled when all user-defined fields (white background) have been adequately populated.

The MOI sensor definition helps to streamline the creation of this macro-sensor by automatically defining the following elements:

1. Model: automatically populated as "os36xx" and identified as a strain type sensor.
2. Description: automatically populated as "Internally temperature compensated long gage strain sensor".
3. Expression: automatically populated, incorporating both total and thermal component of strain terms.
4. EpsT0: the thermal component of strain is automatically defined and declared as positively correlated to the complete strain expression.
5. DeltaT: the relative operating temperature range is automatically defined as a function of one of the component FBGs.
6. Sensor model-consistent constants: C1, C2, and St are automatically defined and populated, further reducing data input requirements for the user.
7. Variable IDs P1 and P2: defines the two FBG quantities to be used in the expression and/or sub-expressions and links each to a filtered selection drop-down list from which the temperature and strain FBGs can be selected.

User defined quantities are clearly identified with a white background, in contrast to the gray background of the automatically populated controls. In order to finish creating the temperature compensated strain sensor, the user simply selects the FBGs to be used, enters the sensor-/substrate-specific calibration constants (Fg and CTEs) and names the sensor. In this example, the following entries complete the creation of the sensor:

P2 (temp) = FBG_A2 (from sensor information sheet)
 P1 (strain) = FBG_A3 (from sensor information sheet)
 Fg = 0.83 (from sensor information sheet)
 CTEs = 11 (user supplied)
 ID = deck_strain_os3600 (user supplied)

Once all editable data is entered, the "Create os36xx" window is closed and the following sensor is created:



Edit Sensor

Setup Reference Info

Range
 Min: -832.826 Max: 832.826 Current: ---

Parameters
 Active Visible Rate Of Change
 ID: deck_strain_os3600
 Group:
 Type: Strain
 # Averages: 1
 Minimum:
 Maximum:
 Warn Thresh.: 0.80 (0.0 - 1.0)
 Expression: (1e6 * S8_3_N) / Fg . EpsT0
 Expression is valid

Constants

Name	Value
Fg	0.83
CTEs	11
C1	0.796
C2	10.1
St	23.8

Update Delete

Sub-Expressions

ID	Expression	Compensation
EpsT0	(1e6 * (FBG_A2_N / C1)) + (DeltaT * (CTEs - C2))	Positive
DeltaT	1E3 * FBG_A2_D / St	None

OK Cancel Apply

Note that all constants, expressions, sub-expressions (IDs, expressions, and compensation) have been automatically ported to the "Edit Sensor" window from the "Create os36xx Sensor" window. The special DeltaT sub-expression ID has been assigned and will be used to identify the operating relative temperature range for the sensor.

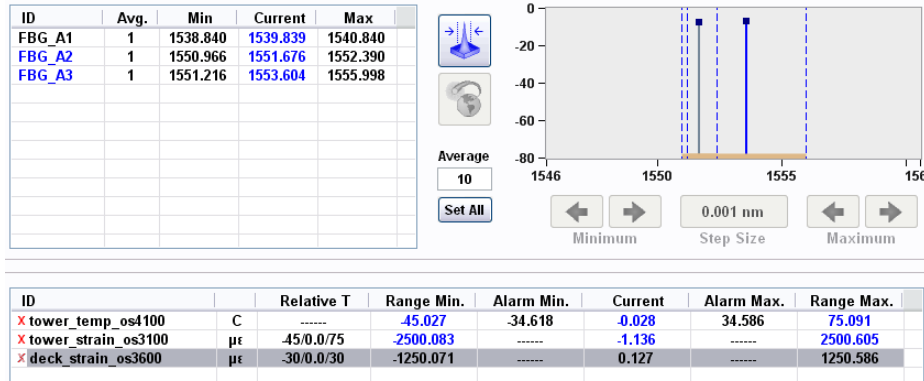
ID	Avg.	Min	Current	Max
FBG_A1	1	1538.840	1539.840	1540.840
FBG_A2	1	1550.676	1551.677	1552.140
FBG_A3	1	1553.140	1553.605	1554.605

Average: 10
Set All

Minimum Step Size Maximum

ID	Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
X tower_temp_os4100	C	-----	-45.027	-34.618	-0.033	34.586
X tower_strain_os3100	µε	-45/0.0/75	-2500.083	-----	-0.637	2500.605
X deck_strain_os3600	µε	-42/0.1/19	-----	-----	1.642	-----

The final step for the os3600 example is to modify its component FBG wavelength bins to reflect the relative operating temperature and strain measurement ranges.



The image above shows the os3600 sensor with the temperature FBG (gray) bin modified to yield an operating relative temperature range of +/- 30 degrees C and the strain FBG (blue) bin modified to yield a strain measurement range of +/- 1250 microstrain. Because a positive compensation correlation was established between the total strain and the thermal component of strain, the apportioned spectrum for the two FBGs are exactly enough to contend with the mechanically and thermally induced wavelength changes needed to measure +/-1250 microstrain over a +/- 30 degree C operating range.

NOTE: In certain circumstances (e.g. os3600 sensors) overlapping FBG wavelength bins are required in order to support the full measurement range over the full operating temperature range of a collection of sensors using a minimum optical spectrum for each sensor. While overlapping bins are functional in and permitted by ENLIGHT, caution should be exercised in their usage.

B. Example: Absolute Temperature Probe os4200.

The associated FBG for the sensor is identified as FBG_A1. The “Create From Definition” button is pressed, and the os42xx.mst file is selected. FBG_A1 is selected from the pull down list, constants from the sensor-specific information sheet are populated in the corresponding fields, and the user supplies a sensor ID, resulting in the following image:



Create os42xx Sensor

ID: Model: (Temperature)

Description:

Expression:

Int. Expr. 1:

Int. Expr. 2:

Int. Expr. 3:

Constants

Lambda0:

C3:

C2:

C1:

C0:

C6:

C7:

C8:

C9:

C10:

FBGs

Channel	FBG	
P1: -- All --	FBG_A1	Temp. Probe
FBG 2:		
FBG 3:		
FBG 4:		

Sensors

Sensor 1:

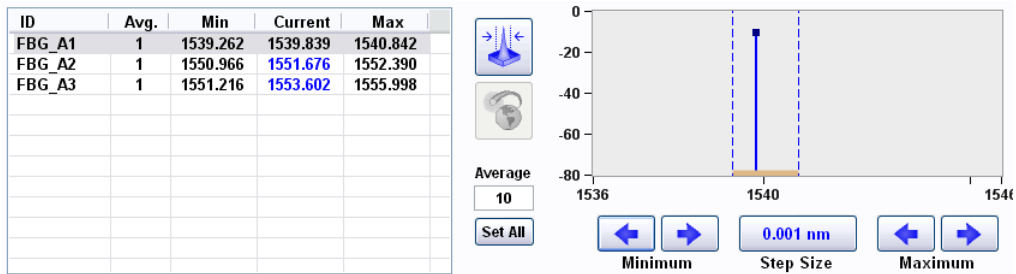
Sensor 2:

Sensor 3:

Sensor 4:

NOTE: os42xx and os43xx sensors require a large number of significant digits for the calibration expressions in order to yield the high absolute temperature accuracy that these sensors offer. Copy (control-c) and paste (control-v) operations are supported to ease data entry from other sensor data records.

After clicking OK, the sensor is created. The user then drags the FBG bin limits for FBG_A1 to yield the desired temperature measurement range of -40 to 120 degrees C.



ID	Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.	
x tower_temp_os4100	C	-----	45.027	34.618	0.075	34.586	75.091
x tower_strain_os3100	µε	-45/0.1/75	-2500.083	-----	0.425	-----	2500.605
x deck_strain_os3600	µε	-30/0.0/30	-1250.071	-----	-1.413	-----	1250.586
x pile_temp_os4200	C	-----	40.103	-----	25.587	-----	120.037

C. Example: os3100 strain sensor, compensated by os4200 absolute temperature sensor.

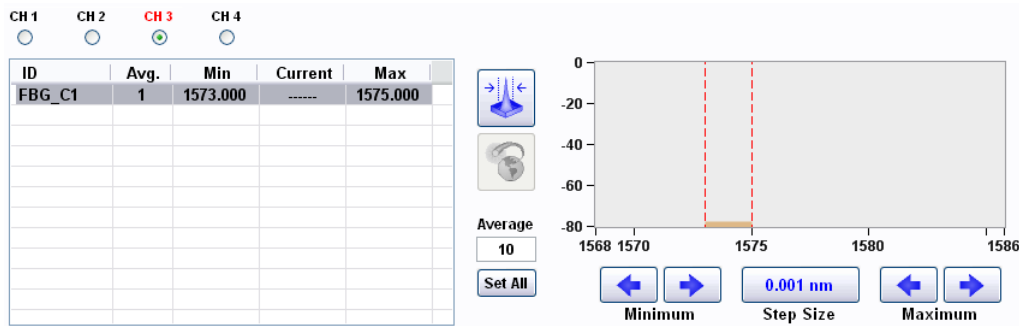
This example will demonstrate how to reference an existing absolute temperature sensor for thermal compensation of an os3100 strain sensor, using the "Create From Definition" feature of the Sensor tab.



Additionally, this example will demonstrate how a sensor element can be created without having one or more of the component FBGs optically connected to the system.

Consider the possibility that an os3100 sensor will be added to the measurement system, but is not yet available for optical connection to the interrogator. If the nominal information about the sensor is already known, the sensor element can still be created and wavelength provisioned in ENLIGHT.

Assume that the os3100 nominal wavelength is 1574 nm and will be connected to channel 3 of the interrogator. The FBG element can be manually created as shown below.



The absence of the grating is indicated by the red bin limits and channels 3 label, but sensors can still be created from the FBG_C1 reference.

The sensor definition “os31xx_absT_comp.mst” is selected for the “Create From Definition” action.

The 'Create os31xx Sensor' dialog box contains the following fields and sections:

- ID:** pile_strain_os3100
- Model:** os31xx (Strain)
- Description:** Externally absolute temperature compensated strain sensor.
- Expression:** (1e6 * FBG_C1_N) / Fg - EpsT0
- EpsT0:** DeltaT * (C1 / Fg + CTEs - C2) (Positive)
- DeltaT:** AbsoluteTemp_D (None)
- Int. Expr. 3:** ()
- Constants:**
 - Fg: 0.89
 - C1: 6.156
 - C2: 0.7
 - CTEs: 11
 - C5: (empty)
 - C6: (empty)
 - C7: (empty)
 - C8: (empty)
 - C9: (empty)
 - C10: (empty)
- FBGs:**

Channel	FBG	
P1:	FBG_C1	Strain Gage
FBG 2:		
FBG 3:		
FBG 4:		
- Sensors:**
 - AbsoluteTemp: (dropdown menu showing 'pile_temp_os4200' selected) (Temperature)
 - Sensor 2: tower_temp_os4100 ()
 - Sensor 3: pile_temp_os4200 ()
 - Sensor 4: (dropdown menu) ()

Buttons: OK, Cancel

As with the previous two examples, most fields are automatically populated by the *.mst file. For this sensor, the user chooses an ID, a substrate CTEs, and associates the strain measurement FBG (here FBG_C1). Additionally, the user is required to select the temperature element that will be used for temperature compensation for the sensor. ENLIGHT presents the user with an automatically filtered list of temperature sensors from which the user is to choose. In this example, the absolute temperature sensor “pile_temp_os4200” is used, resulting in a sensor as in the following image:



Edit Sensor

Setup Reference Info

Range
 Min: --- Max: --- Current: 1.078

Parameters
 Active Visible Rate Of Change

ID: pile_strain_os3100
 Group:
 Type: Strain
 # Averages: 1
 Minimum:
 Maximum:
 Warn Thresh.: 0.80 (0.0 - 1.0)
 Expression: (1e6*FBG_C1_N)/Fg.EpsT0
 Expression is valid

Constants

Name	Value
Fg	0.89
CTEs	11
C1	6.156
C2	0.7

Update Delete

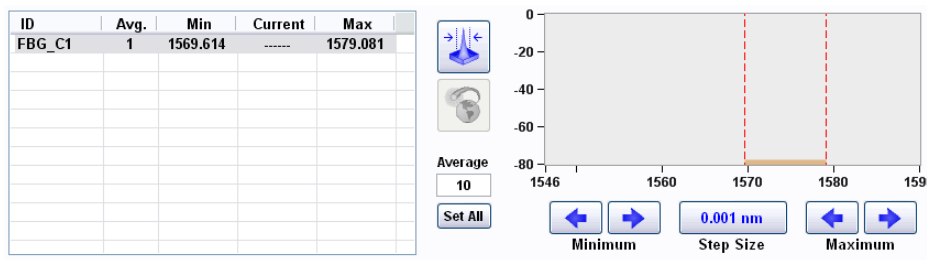
Sub-Expressions

ID	Expression	Compensation
EpsT0	DeltaT*(C1/Fg+CTEs-C2)	Positive
DeltaT	pile_temp_os4200_D	None
		None

OK Cancel Apply

Note that the special DeltaT sub-expression references the absolute temperature sensor “pile_temp_os4200” with the “_D” delta calculation shortcut, yielding a relative temperature quantity, as is needed for all DeltaT calculations. As with the os3600 example before it, EpsT0 uses DeltaT in a positively correlated compensation equation to normalize the thermal component of strain on the os3100 strain sensor.

Dragging the FBG limits for FBG_C1 (despite the fact that it has not yet been optically connected) defines the strain range for the compensated os3100 macro-sensor over the operating temperature range dictate by the compensating os4200 sensor.



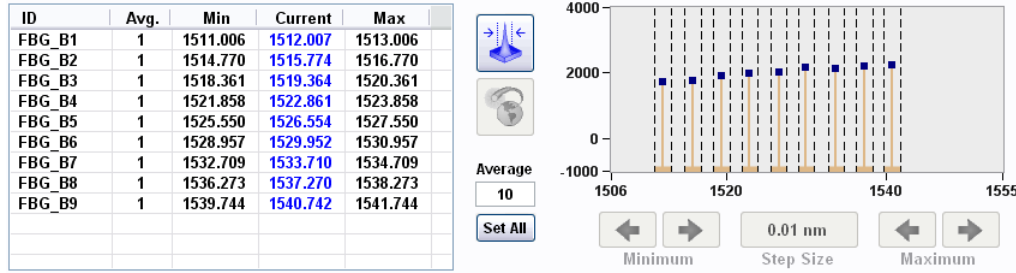
ID	Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.	
X tower_temp_os4100	C	-----	-45.027	-34.618	-0.059	34.586	75.091
X tower_strain_os3100	µε	-45/0.1/75	-2500.083	-----	-0.454	-----	2500.605
X deck_strain_os3600	µε	-30/0.1/30	-1250.071	-----	-0.368	-----	1250.586
X pile_temp_os4200	C	-----	-40.103	-----	25.583	-----	120.037
X pile_strain_os3100	µε	-66/0.0/94	-2000.001	-----	-----	-----	2000.600

D. Example: an array of distributed os4100 sensors, used to detect distributed temperature changes.



This example shows how new sensors can be created using the Create From Sensor duplication button. Additionally, the use of Rate of Change and single sided Alarm Limits will be demonstrated.

First, the array of FBGs is identified on channel 2 and bins are automatically created by ENLIGHT.



Next, the “Create From Definition” button is used to create a sensor configuration from the os41xx.mst file. After the sensor is created, its properties can be changed using the Edit sensor button. In the image below, the Rate of Change control is selected.

Edit Sensor

Setup Reference Info

Range
 Min: -31.824 Max: 37.380 Current: 0.514

Parameters
 Active Visible Rate Of Change

ID: os4100_a
 Group: [Dropdown]
 Type: Temperature [Dropdown]
 # Averages: 1
 Minimum: [Field]
 Maximum: [Field]
 Warn Thresh.: 0.80 (0.0 - 1.0)
 Expression: 1e3 * (FBG_B1_D / St)
 Expression is valid

Constants

Name	Value
St	28.9

 Update Delete

Sub-Expressions

ID	Expression	Compensation
		None [Dropdown]
		None [Dropdown]
		None [Dropdown]

OK Cancel Apply

This sensor will now report its derivative, calculated by a five point Savitsky-Golay convolution, for plotting and alarm purposes.

With os4100_a completed and reporting the rate of temperature change, a new sensor can be created by highlighting os4100_a on the Sensors table and clicking the “Create From Existing Sensor” button.





Parameters

Active
 Visible
 Rate Of Change

ID:

Group:

Type: **Temperature**

Averages:

Minimum:

Maximum:

Warn Thresh.: (0.0 - 1.0)

Expression:

Constants

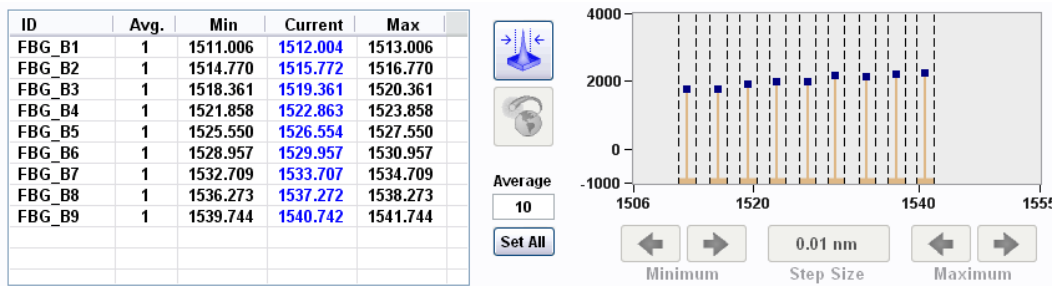
Name	Value
St	28.9

Update Delete

A new sensor is created that replicates the expression, type, constants, averages, alarm ranges, warning threshold, and rate of change settings.

The user would populate the ID field for this new sensor and edit the FBG variable in the expression to point to a different FBG and click OK to complete the sensor creation.

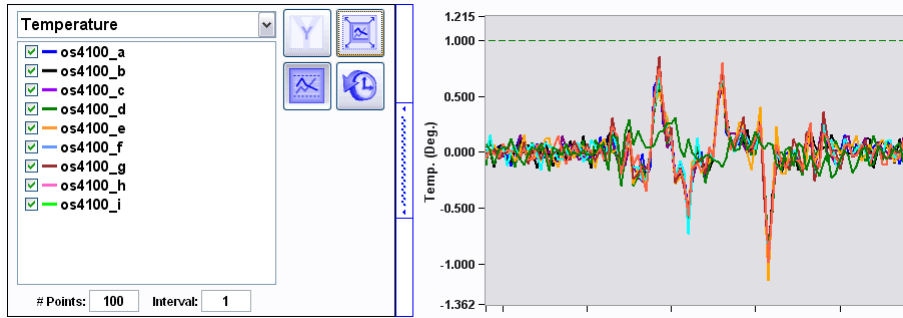
This process of replication is repeated until all of the FBGs on channel 2 are represented by os4100 expressions, resulting in the following screen image.



ID	Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
✓ os4100_a	C	-----	-----	0.010	0.500	-----
✓ os4100_b	C	-----	-----	-0.115	1.000	-----
✓ os4100_c	C	-----	-----	0.170	1.000	-----
✓ os4100_d	C	-----	-----	0.105	1.000	-----
✓ os4100_e	C	-----	-----	-0.063	1.000	-----
✓ os4100_f	C	-----	-----	-0.033	1.000	-----
✓ os4100_g	C	-----	-----	-0.103	1.000	-----
✓ os4100_h	C	-----	-----	0.040	1.000	-----
✓ os4100_i	C	-----	-----	0.073	1.000	-----

Note that the Current sensor values for each of the os4100's _a through _i are displayed in purple, signifying a rate of change quantity.

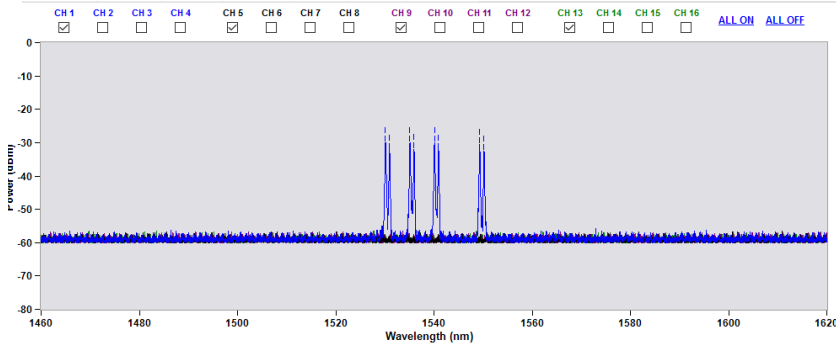
Note: The Rate of Change is calculated for the sensor for purposes of plotting and alarming only. Any other sensor that references a "Rate of Change" sensor in its expression will operate on the sensor's actual value, not on the calculated Rate of Change.



In the image above, ENLIGHT plots the Rate of Change for each of the nine os4100 relative temperature sensors and applies a positive limit of 1. An alarm will be triggered if any one of the nine sensors changes more than one degree C in a single acquisition.

11.3.3.5. Referencing Sensors (independent FBG zeroing)

Connected to 8 FBGs



Automatically detect. Have live values and create bin regions automatically.

ID	Avg.	Min	Current	Max
FBG_A1	1	1529.127	1530.123	1530.452
FBG_A2	1	1530.614	1530.941	1531.909
FBG_A3	1	1534.187	1535.165	1535.433
FBG_A4	1	1535.596	1535.924	1536.922
FBG_A5	1	1539.097	1540.095	1540.423
FBG_A6	1	1540.586	1540.911	1541.911
FBG_A7	1	1540.291	1540.289	1540.648
FBG_A8	1	1540.827	1540.187	1551.184

ID	Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
test_referencing_with_j...	pt	12 / 0.1 / 36	-495.915	---	-4.051	495.399

Edit the sensors, see reference values stored as current value of FBG when the sensors were created (expected behavior)

Edit Sensor

Setup Reference Info

Offset: 0.000 -- NOT SAME AS ZEROING --

FBG	
Name	Value
FBG_A1_0	1530.1263591214
FBG_A2_0	1530.938143578...

FBG_A1_0 1530.1263591214 Update

Sensor	
Name	Value

Update

OK Cancel Apply

FBG

ID	Avg.	Min	Current	Max
FBG_A1	1	1529.500	1530.126	1530.500
FBG_A2	1	1530.500	1530.939	1531.500
FBG_A3	1	1534.107	1535.110	1535.433
FBG_A4	1	1535.596	1535.924	1536.922
FBG_A5	1	1539.097	1540.097	1540.423
FBG_A6	1	1540.586	1540.913	1541.911
FBG_A7	1	1548.291	1549.291	1549.648
FBG_A8	1	1549.827	1550.186	1551.184

Average: 10

Set All

Minimum 0.001 nm Step Size Maximum

ID	Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
x test referencing_with J...	μ s	-16 / 0.1 / 20	-140.444	4.333	-	-140.056



Edit Sensor

Setup Reference Info

Offset: 0.000 -- NOT SAME AS ZEROING --

FBG		Past Values
Name	Value	
FBG_A1_0	1530.126936334...	6/22/2017 8:49:58 AM - 1530.12693633406
FBG_A2_0	1530.935863969...	

FBG_A1_0 1530.12693633406 Update

Sensor		Past Values
Name	Value	

Update

OK Cancel Apply

FBG

Name	Value
FBG_A1_0	1530.128278820...
FBG_A2_0	1530.938920847...
FBG_A3_0	1535.108251952...
FBG_A4_0	1535.921499662...

FBG_A1_0 1530.12827882066 Update

FBG

Name	Value
FBG_A1_0	1530.000131149...
FBG_A2_0	1531.000087239...
FBG_A3_0	1534.999929320...
FBG_A4_0	1536.0000263868

FBG_A1_0 1530.00013114991 Update

ID		Relative T	Range Min.	Alarm Min.	Current	Alarm Max.	Range Max.
X test_1	µε	-16 / 0.4 / 20	-140.444	-----	3.637	-----	-140.056
X test_2	µε	-15 / 0.1 / 20	-133.141	-----	0.700	-----	-132.751
X difference	µε	-----	-721.420	-----	-0.145	-----	744.606
X difference_2	µε	-----	-718.198	-----	-0.706	-----	747.830
X difference_3	µε	-----	-718.622	-----	-1.827	-----	747.404



Edit Sensor

Setup Reference Info

Offset: 0.000 -- NOT SAME AS ZEROING --

FBG	
Name	Value
FBG_A1_0	1530.1263591214
FBG_A2_0	1530.938143578...

FBG_A1_0 1530.1263591214 Update

Sensor	
Name	Value

Update

OK Cancel Apply

Edit Sensor

Setup Reference Info

Offset: 0.000 -- NOT SAME AS ZEROING --

FBG	
Name	Value
FBG_A1_0	1530.1263591214
FBG_A2_0	1530.938143578...

FBG_A1_0 1530.1263591214 Update

Sensor	
Name	Value

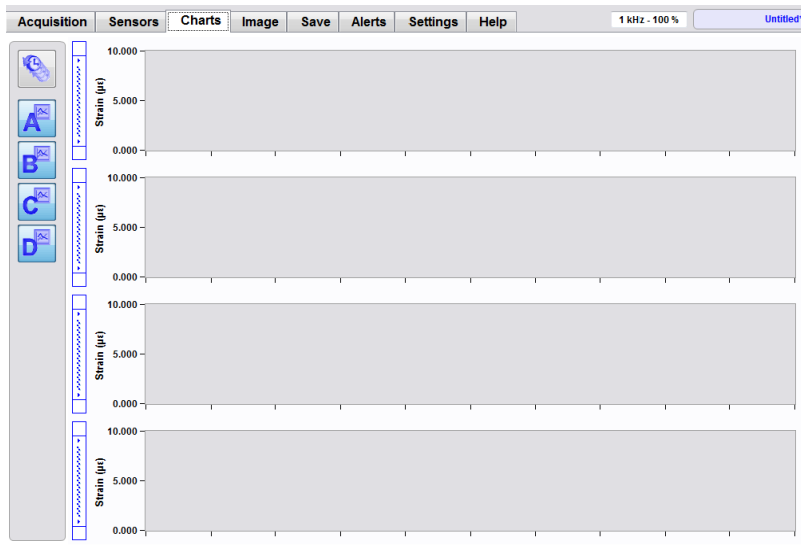
Update

OK Cancel Apply

11.3.4. Charts Tab

Once sensors and their warning/alarm levels have been defined, the first of the data visualization tabs can be used. The Charts tab allows the user to independently configure as many as four simultaneous strip chart indicators for display, with each strip chart plotting data for multiple sensors of the same type.

Top level view of Charts tab



11.3.4.1. Show/Hide Chart



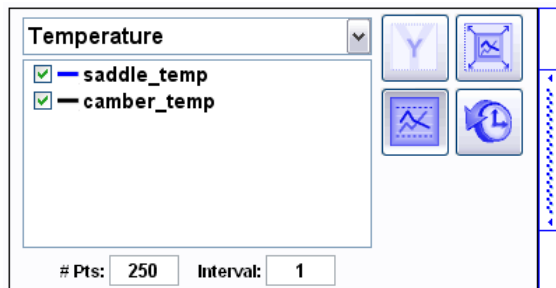
To the left side of the chart displays, there are four buttons labeled with letters from A to D. These controls independently show or hide each of the four available strip charts. The charts automatically rescale in the vertical direction to fill the available space on the tab.

11.3.4.2. Reset All



Above the chart display controls, the “Reset All” button can be used to reset data on all of the four strip charts, ensuring synchronization of all four chart entries.

11.3.4.3. Show/Hide Chart Setup



To the left side of each visible strip chart is an expanding control dock. When the dock is closed, the controls are hidden and the chart expands in the horizontal to fill the available space. When the user clicks on the blue vertical dock expansion tab, the controls again become visible and the chart constricts accordingly.



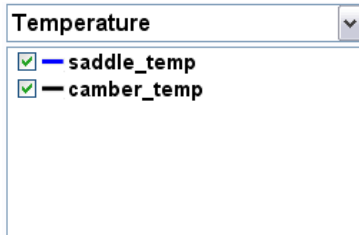
The following entries explain the details of the chart controls.

A. Sensor Type



Each chart is defined by a specific sensor type, and only sensors of that defined type will be made available for plotting on that chart. In this way, a common set of units can be applied to the Y axis of the plot. Additionally, there is a better probability that the magnitudes of the sensor values on a single plot will be similar, reducing the spread of data values that could render the scales impractically large. For example, by grouping temperature sensors on one plot and strain sensors on another, the software avoids the complication of trying to plot 23 degrees C and 2500 microstrain on the same Y axis.

B. Sensor List



Once a sensor type is defined for the chart, then a list of sensors for that type is presented to the user for addition to the chart. In the example above, the two strain sensors that were made available were selected for plotting on chart A. Note that each sensor in the list is identified with a color key that matches the color of the sensor plot on the chart.

C. Scale Buttons

i. Auto Scale Y-Axis



Automatically adjust the Y scale according to the displayed sensor data.

ii. Zoom All



Automatically adjust the Y scale to all present data, including the sensor data and the alarm limits.

iii. Hide/Show Limits



Alternately hides/shows the sensor alarm limits on the chart.

iv. Reset Chart





Erases all current data and restarts the chart.

D. # Points and Interval

i. # Points

Pts:

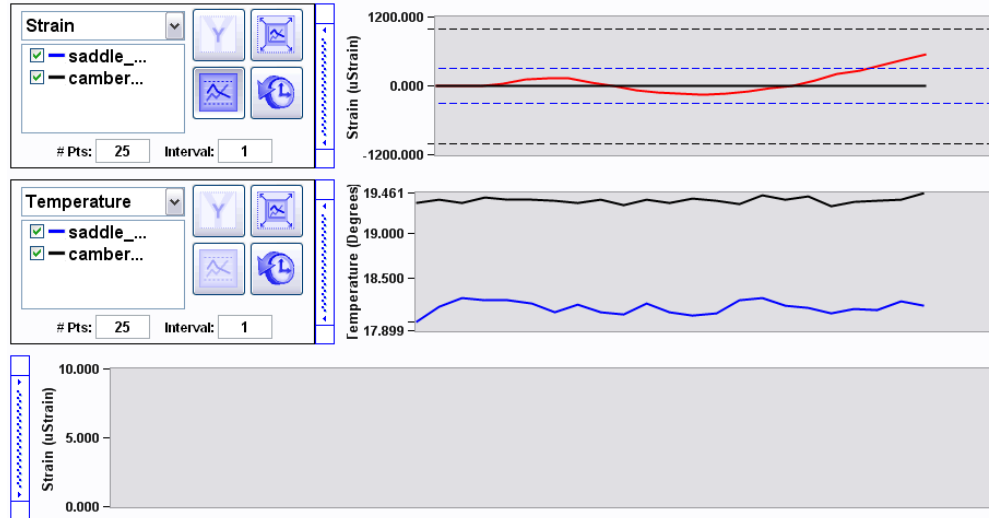
Defines the displayed span of the chart X scale.

ii. Interval

Interval:

Determines the interval at which the chart is updated. For example, an interval value of 20 on a 1 kHz x30 interrogator core would cause the chart to update a rate of 50 times per second.

E. Warning and alarm indications



When a sensor goes into warning or alarm condition, the color of that sensor plot on the chart turns from its predefined color to either yellow or red, respectively. In the above figure, it can be seen that the “saddle_strain” sensor has exceeded its maximum limit value and is thus in an alarm state. As such, the color of its trace has turned from blue to red.

11.3.5. Image tab

The second tab used for sensor data visualization is the Image tab. The following section will outline how to use the Image tab to create custom sensor system monitoring configurations.

11.3.5.1. Select Image



On the bottom, right of the Image tab is the “Select Image” control, as seen above. After clicking the button, the user will be prompted with a file dialog that allows him to choose an image file that has been loaded into the ENLIGHT Image folder via the Import/Export tool.

11.3.5.2. Hide/Show Image Setup



Like the chart controls from the previous section, the image setup is controlled by an expanding dock control on the left side of the display. By clicking on the blue vertical dock expansion tab, the user will reveal the list of available sensors for the Image tab.

Image tab with image selected and one sensor placed



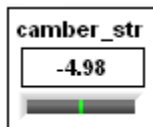
The purpose of the Image tab is to place indicator elements that represent the current measurements of sensors onto logical positions on a display image. For example, the image above shows a picture of a monitored bridge. The two macro-sensors representing the temperature compensated strain measurements of the bridge saddle and camber could be placed in a position of the image that shows these two bridge parts. Such visual associations between the sensor indicators and an image or diagram of the measurement subject help the user to more intuitively perceive and interpret measurement data.

In order to place a sensor indicator on the image, the user selects that sensor from the list presented in the expanded dock control. When a sensor is selected, ENLIGHT adds an indicator in the geometric center of the image. If multiple sensors are added at the same time, ENLIGHT will stagger stack the indicators so that all are still visible. In order to place the indicator in a desired location, the user clicks on the full indicator object (via mouse or finger control, depending on the deployment platform) and drags the indicator to that specific position.

11.3.5.3. Indicator types

All indicators are placed onto the image as a default “Full” style indicator. The following entries will present the three different indicator types and how to interpret them.

A. Full Indicator



The default Full indicator style is shown above and is comprised of three elements. The top line of the indicator displays the first ten characters of the sensor name, followed by the units, space permitting. The center numeric portion of the indicator displays the current value of the sensor. The bottom line shows a tank indicator of the sensor value. The length of the tank reflects the full specified limits range of the sensor, and the current value is indicated by the position of the



wiper. On this indicator, a green tank indicator implies that the sensor is in range. A yellow indicator implies that the sensor is in a warning state. A red indicator implies that the sensor is presently in an alarm state, and the direction of the indicator of the alarm infraction, either max or min.

NOTE: Of the three indicator types, only the FULL type allows the user to reposition the indicator.

B. Gage Indicator



By tapping the mouse pointer onto a full indicator, the user can switch that indicator to Gage style. As is seen on the image above, the Gage style indicator shows the full specified limits range of the sensor as the circumference of the round gage. The green arc section indicates the portion of the range neither in warning nor alarm state. The yellow arc sections indicate a warning state. When the sensor exceeds its specified alarm limits, then the background of the Gage indicator turns red to indicate its alarm condition.

C. LED Indicator



The LED indicator type simply shows the current state of the sensor as in range, in warning state, or in alarm state, by showing colors green, yellow, or red, respectively.

D. Global Indicator settings



Indicator types can be individually changed by tapping on the indicator, or changed en masse by use of one of the above "Show All" controls. The control on the left sets all indicators to Full style. The control in the middle sets all indicators to Gage style. The control on the right sets all indicators to LED style.

Zoom of image view with mixed indicator types



The image above shows the Image View with a number of mixed indicator styles.

11.3.6. Archive Tab

The ENLIGHT Data Archive provides users with a reliable baseline of data storage functionality that is both complementary and completely independent from the existing user-selected data saving options located on the Save tab. The ENLIGHT Data Archive serves as a data safety net: an ever-present archival data store that is active from the moment the first FBGs and sensors are defined. The Archive tab presents the user with a suite of indicators and controls used to plot and view FBG and sensor elements that have been stored in the ENLIGHT Data Archive.

11.3.6.1. Data Archive Management Controls

The Data Archiving controls govern the location, update interval, and maximum size of the ENLIGHT Data Archive and are located on the Settings tab. Details for use of these controls are presented in section 7.3.9.6 of this user guide.

11.3.6.2. ENLIGHT Data Archive Display Parameters

The controls and indicators of the Archive tab provide simple access to sensor and FBG elements stored in the ENLIGHT Data Archive and a means to plot that content over date ranges of interest.

The Archive tab allows for easy access to data stored in the Data Archive. The following controls are the key tools for modifying chart content.

A. Display Parameters

The screenshot shows a control panel titled "Display Parameters". It contains two date pickers: "Start" with the value "1/ 1/2012" and "Stop" with the value "1/31/2012". Below these is a "Delta (minutes):" input field with the value "10.00". At the bottom right of the panel is a blue "Update Chart" button.

The Display Parameters controls enable the user to select the date ranges and the time granularity with which ENLIGHT retrieves sensor and FBG data from the ENLIGHT Data Archive for plotting on the archive chart. Since ENLIGHT stores sensor and FBG data to the ENLIGHT Data Archive continually from the moment a sensor is configured, selection of these date ranges via the Display Parameter controls does not limit future data from being stored to the Data Archive. Rather, it simply selects the data that will be retrieved from the Data Archive for plotting and viewing on the archive chart.

i. Start & Stop Dates

These calendar controls enable the user to select the date ranges over which ENLIGHT retrieves sensor and FBG data from the ENLIGHT Data Archive. The default Start date is the date the original Data Archive is created, and the default Stop date is 30 days later.

ii. Delta

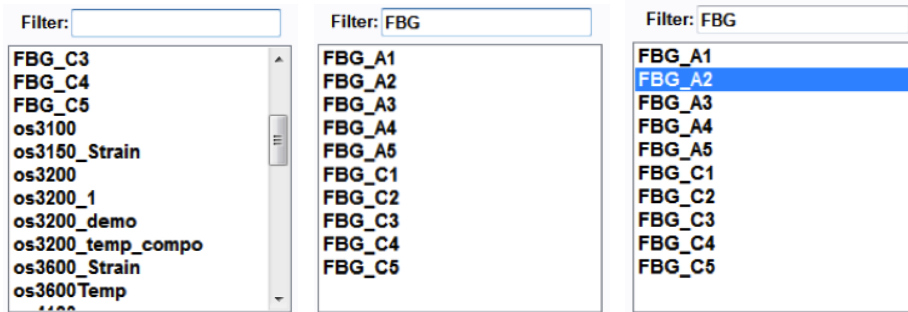
The Delta value control selects the time differential between successive FBG or sensor elements to be retrieved from the ENLIGHT Data Archive and displayed on the archive chart. If the default settings remain in place, the Delta value is set to 10 minutes and the default archive update interval is set to 1 minute. With these default setting, the archive chart will display only 1 out of every 10 recorded FBG or sensor values stored in the Data Archive.

iii. Update Chart

This button updates the archive chart to reflect any changes to the Display Parameter settings on the Archive tab. Note that if modification of the Start or Stop date does not update the chart Update Chart is clicked, the user should make sure the Auto-Zoom X button is selected to display the entire date range.



B. Filter



The Filter control helps users to quickly find and access target FBG or sensor elements among the complete collection of Data Archive elements. All defined FBG and sensor names stored in the ENLIGHT Data Archive are displayed in an archive elements list indicator on the Archive tab. The user can quickly locate specific subsets of FBG or sensor elements by typing a name in the Filter.

In the series of screenshots above, the functionality of the Filter control is demonstrated. In the leftmost screenshot, the available archive elements indicator shows all of the FBG and sensor elements stored in the ENLIGHT Data Archive. In the middle screenshot, the user entered the letters “FBG” in the Filter control, thereby reducing the list of FBGs and sensors to elements with names containing the letters “FBG”. In the rightmost screenshot, the user then double clicks on “FBG_A2” to add data from that FBG element to the Archive chart. See section 7.3.6.3 for additional examples.

i. Add Selected Plots To Chart



This control will add a highlighted FBG or sensor from the archive elements list indicator to the archive chart. Note, double-clicking a single FBG or sensor name in the archive elements list indicator will also add it to the chart.

ii. Remove Selected Plots From Chart



This control will remove all highlighted FBG or sensor elements in the archive elements list indicator from the archive chart. Note that right-clicking on a single FBG or sensor name in the legend on the rightmost side of the Archive tab also presents an option to remove its archive data from the chart.

iii. Remove All Plots From The Chart



This control will remove all archive data from the chart.

C. Summary



Summary
Name: A os3600
Points: 416593
Min/Max: -1362.98 - 1558.54
Start/Stop: 4/16/2011 - 2/27/2012
[Display Selected Element](#)

The Summary indicator will display the total number of data points for a selected FBG or sensor element that is stored in the ENLIGHT Data Archive, that element’s minimum and maximum values, and the full range of dates over which the ENLIGHT Data Archive holds entries for that element.

In order to populate the Summary for a particular FBG or sensor element, the user will highlight that element in the archive elements list indicator, then then click the Display Selected Element button to load this information in the window.

D. Zoom and Pan

i. Zoom X and Y



Allow for independent or simultaneous zooming of the X and Y axes of the active plot.

Left mouse button double clicking will Zoom In on the active plot.

Right mouse button double clicking will Zoom out.

Note, the min/max date range or value can be manually adjusted by double-clicking and entering an earlier or later date or number.

ii. Pan X and Y



Allow for independent or simultaneous panning of the X and Y axes of the active plot.

iii. Autoscale X, Y. Autoscale X and Y to Full Scale.



- Autoscale X automatically adjusts the X scale to the full range of active X data.

- Autoscale Y automatically adjusts the Y scale to the full range of active Y data.

- Autoscale X and Y to Full Scale simultaneously sets the X and Y scales to the full range of the measurement unit, independent of the range of the active data.

iv. Undo Zoom



Sets the X and Y scales back to the values prior to the previous scaling action.

E. Clipboard & Grid Buttons

i. Copy Chart Data To Clipboard



This control takes the FBG and/or sensor data that has been retrieved from the ENLIGHT Data Archive and plotted on the Archive tab chart and copies those data points to the system clipboard when the button is clicked.

ii. Save Chart Data To Disk



This control takes the FBG and/or sensor data that has been retrieved from the ENLIGHT Data Archive and plotted on the Archive tab and saves those data points to a data file when the button is clicked. The default path to this stored data file for Windows XP is C:\Documents and Settings\My Documents\Micron Optics\ENLIGHT\Data or for Windows 7 is C:\Libraries\Documents\Micron Optics\ENLIGHT\Data.

iii. Copy Chart Image To Clipboard



This control copies an image of the current archive chart to the clipboard when the button is clicked.

iv. Hide/Show Grid



This control shows or hides a default grid to the archive chart.

11.3.6.3. ENLIGHT Data Archive Retrieval and Archive Chart Display Examples

The following usage examples will help to demonstrate some of the features of the Archive tab. To facilitate these demonstrations, consider the following measurement scenario. Multiple temperature and strain sensors have been stored by ENLIGHT into the ENLIGHT Data Archive according to the following Data Archiving control settings, from the Settings tab:

Data Archiving	
Name (0.000 GB)	Update Interval
ENLIGHT	1 min
Import Export/Backup Delete	Maximum Size
	1.0 GB

As soon as each of the FBG and sensor elements are defined, the wavelength and converted measurement data are stored into the ENLIGHT Data Archive at the default Update Interval of 1 minute. After 30 days of data have been stored into the ENLIGHT Data Archive, the user wishes to plot and view data for certain FBGs and sensor over a one week subset of that time period. The following examples will demonstrate methods for doing so.

1. Adding FBG or sensor elements to the archive chart



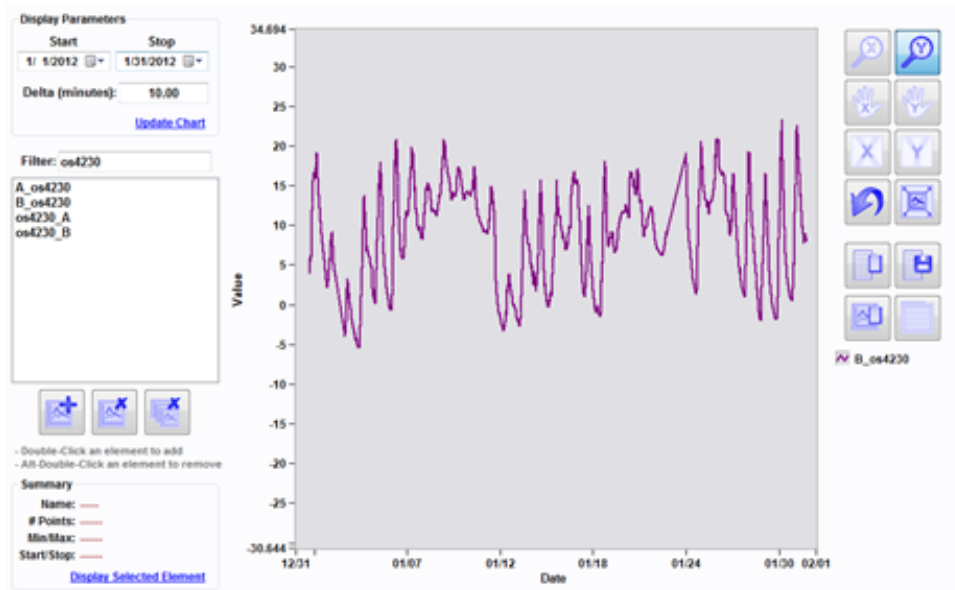
Example 1 a

Goal:

To add a single sensor element to the archive chart using the archive elements list and Filter control.

Implementation:

In the screenshot below, the user first limits the FBG and sensor elements displayed in the archive elements list by typing “os4230” in the Filter control. Within this filtered list, the user can then choose a specific temperature sensor and add it to the archive chart by double-clicking the selected sensor name. Alternatively, the customer could highlight the sensor, in the archive elements list and click the “add selected plots to chart button.”



Example 1 b

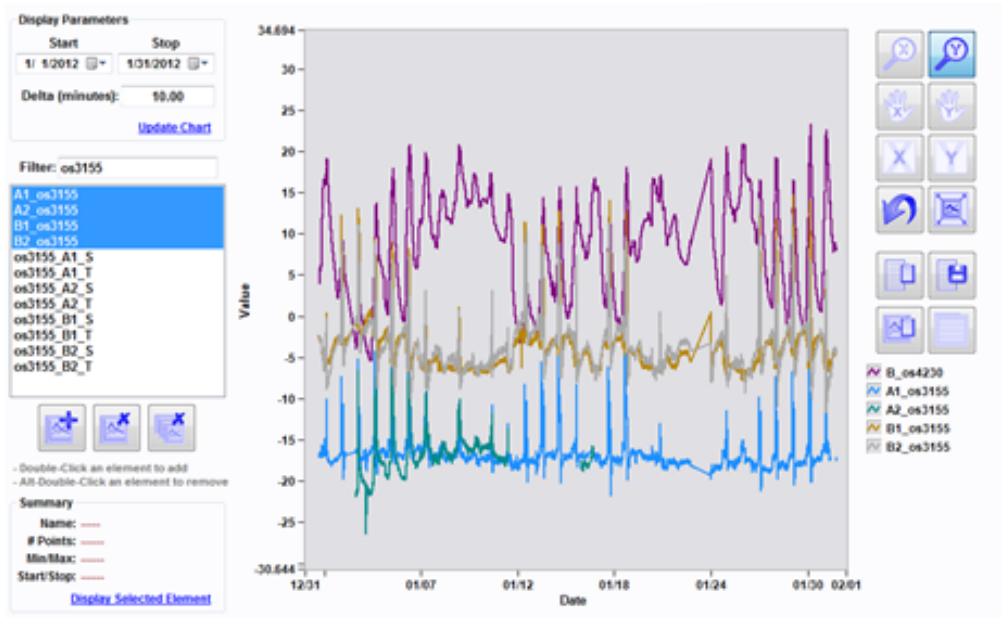


Goal:

To add multiple sensing points to the archive chart using the archive elements list and Filter control.

Implementation:

In the screenshot below, the user first limits the FBG and sensor elements displayed in the archive elements list by typing “os3155” in the filter control. Within this filtered list, the user can then choose to add multiple sensors at one time to the archive chart by holding down the “control” key on the keyboard, highlighting multiple sensors, and then clicking the ‘Add Selected Plots To Chart’ control.



Example 1 c

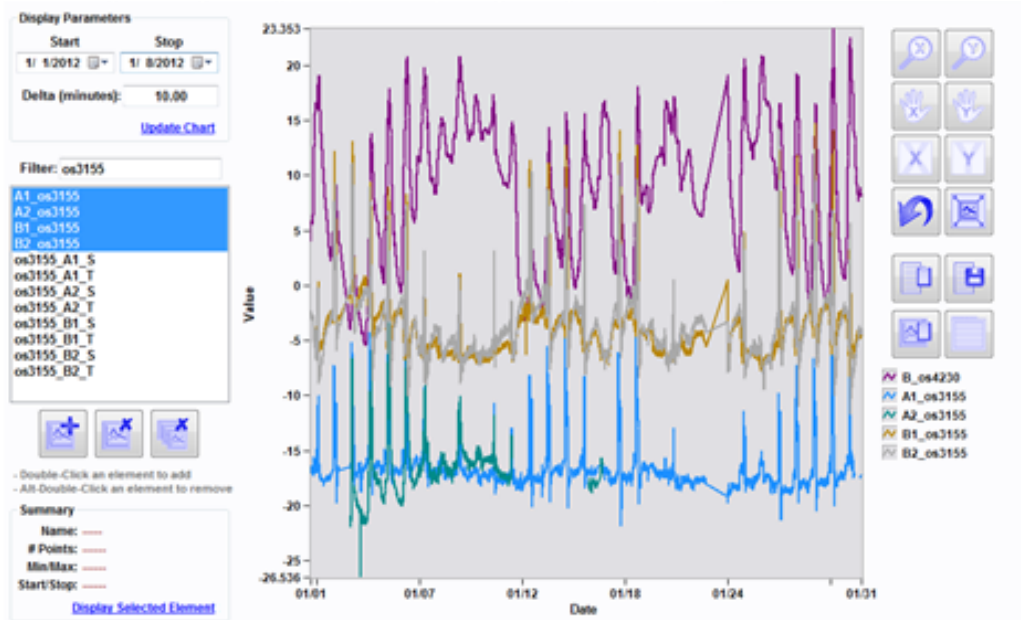


Goal:

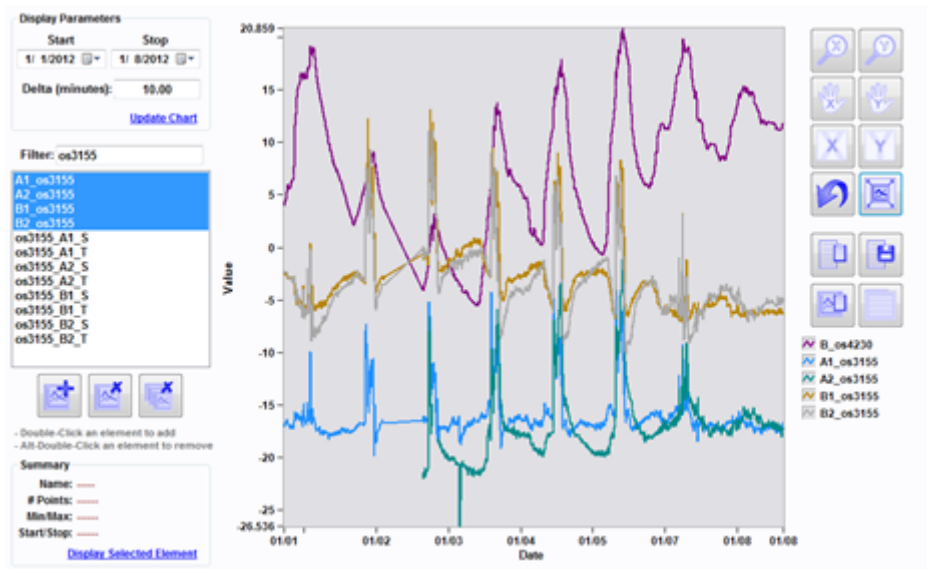
To select the date range over which the ENLIGHT Data Archive elements are plotted.

Implementation:

Now that all desired sensors have been plotted to the archive chart, the user can adjust the archive chart display date range to reduce the plotted data from a 30-day range to a one-week range. The following screenshot shows the user double-clicks on the 1/31/2012 archive display stop date, and then changes the day from 31 to 8.



As soon as the user clicks “Enter” on the keyboard or clicks off the editable box, the archive chart displays a date range from 1/01 to 1/08.



2. Removing selected plots from the Archive chart

Example 2 a

Goal:

To remove a single selected plot from the Archive chart using the archive elements list and Filter control.

Implementation:

This archive elements list allows users to filter the FBG and sensor elements listed and remove sensors with the same simple methods used to add those elements to the archive chart. To remove an element from the archive chart, the user can simply highlight that FBG or sensor element and click the Remove Selected Plots From Chart button. Note that holding down the Alt button on the keyboard and double-clicking will remove the sensor as well.

Example 2 b

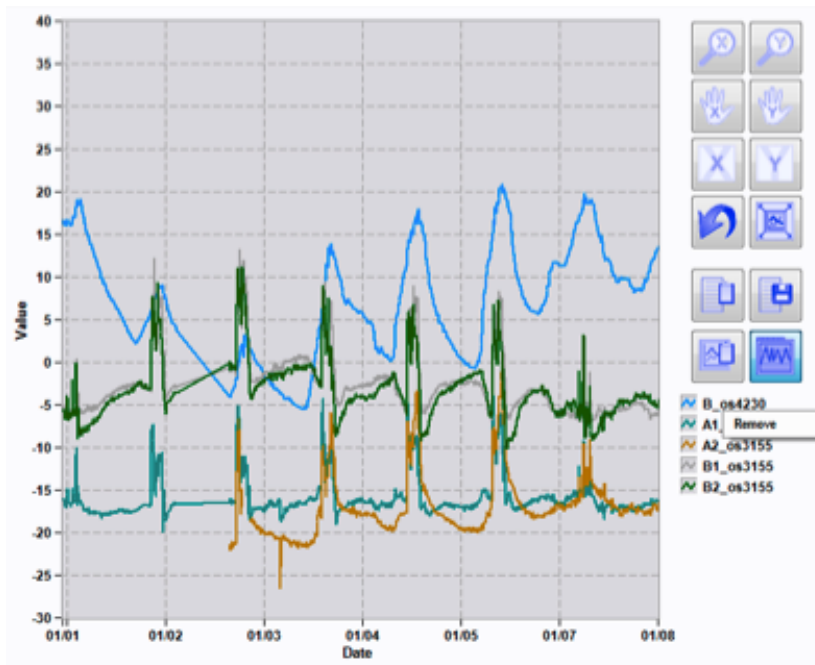


Goal:

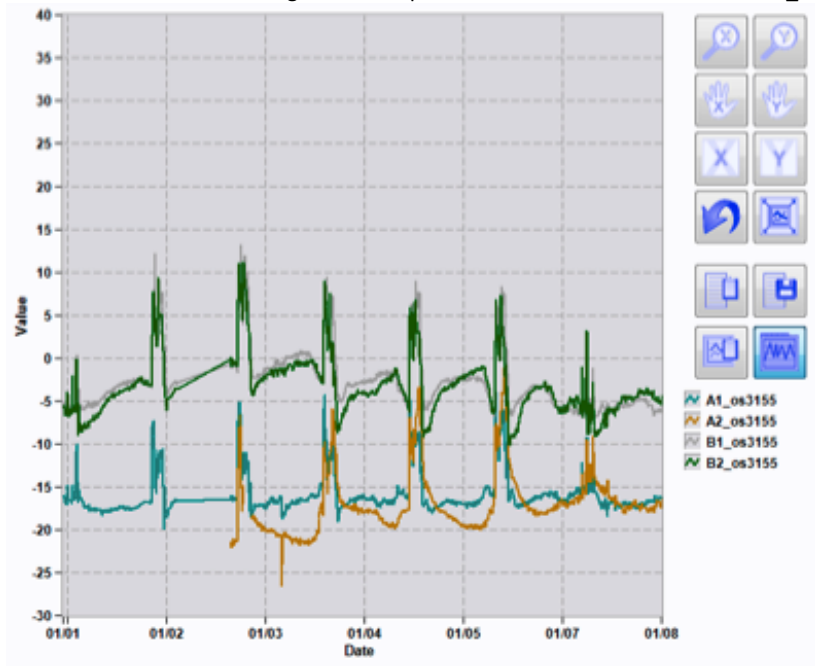
To remove a single selected plot from the Archive chart via the archive chart legend

Implementation:

This feature enables users to remove a single FBG or sensor element from the archive chart by selecting that element from the archive chart legend. The screenshot below shows the user removing a specific sensor element from the archive chart by right-clicking on “B_os4230” in the legend and selecting “Remove”.



The screenshot below no longer includes plotted data for the sensor element “B_os4230.”



11.3.7. Save Tab

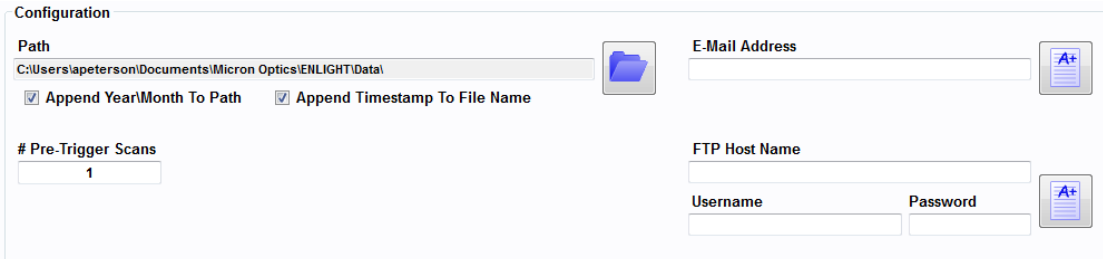


In addition to the new ENLIGHT Data Archive functionality detailed in section 7.3.6, the existing user-defined data saving options remain.

New in v1.5!

In addition to local file storage and email distribution options, a new FTP transfer function has been implemented in ENLIGHT v1.5.

11.3.7.1. Configuration



The following controls govern some of the common attributes for all of the supported data saving profiles:

A. Path

This control allows the user to choose the location in which all ENLIGHT data files are stored. By default, ENLIGHT creates data files and folders under the user’s My Documents folder, under Micron Optics\ENLIGHT\Data folders.

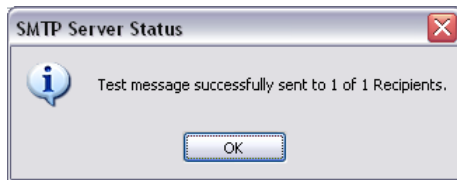
When logging data files, ENLIGHT will automatically create subfolders for the year and the month in which the data files were created, making it easy to locate and manage the resulting files. The Append Timestamp To File Name is also a default setting which will add the defined Timestamp Format.

B. # Pre-Trigger Scans

This control defines how many sensor and spectral peak acquisitions ENLIGHT maintains in an internal buffer as “Pre-Trigger Scans”. Should the user choose to include pre-trigger scans in a triggered sensor data saving profile, then ENLIGHT can also include the data that would precede the trigger event.

C. E-Mail Address

This control defines an email address to which data files can be sent, if so configured. The test button to the right of the data entry field creates and sends a test email from the selected return address to the defined recipients on the Save tab using the SMTP settings from the Settings tab, Notification section. A successful test email will be indicated by the following pop up message:



New in v1.5!

D. FTP Parameters



These controls define an FTP site and account credentials for transfer of data files, if so configured. The test button to the right of the data entry fields creates and sends a test file to the defined FTP site.

i. Host Name

This control defines the FTP server host name for the target FTP site to which data files will be transferred.

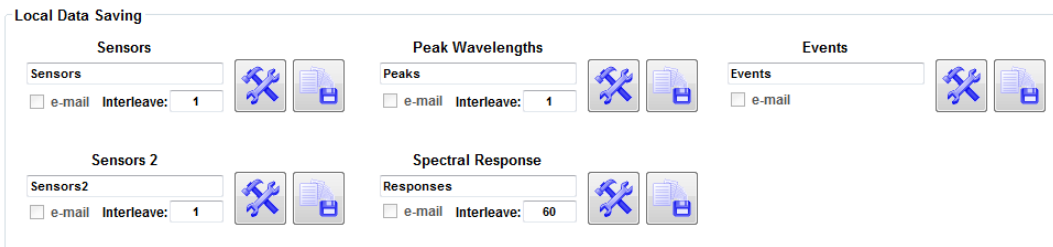
ii. Username

This control defines the user account name for the FTP site.

iii. Password

This control defines the password associated with the user account associated with the previously selected Username.

11.3.7.2. Local Data Saving



A. Common controls and indicators

ENLIGHT now supports multiple, simultaneous data saving profiles. The following controls are common to the five major file types for saving:

a. File name base



ENLIGHT will automatically append a time stamp and file extension to each of the data files that it generates to the file base name defined by this control. For example, a peaks data file created at 11:21:22 on January 31st, 2010 would be named Peaks.20100131112122.txt.

In the screenshot above, “Peak Wavelengths” is the profile type and “Peaks” is the base file name.

b. e-mail



Data files can be emailed to the defined recipient at the time that the files are closed. This control enables the email function for the associated data saving profile.

c. Interleave



Defines the rate divider at which data will be saved from a streaming ENLIGHT acquisition. This interleave is aggregated with the data acquisition interleave from the Acquisition tab.



For example, if an Acquisition tab interleave of 10 is applied to data from a 1kHz sm130-700 and a Save tab interleave of 20 is applied to a data save profile, data would be acquired from the module at $1000\text{Hz}/10 = 100\text{Hz}$ and data would be logged to disk at a rate of $100\text{Hz}/20 = 5\text{ Hz}$.

- d. Edit Data Logging Settings



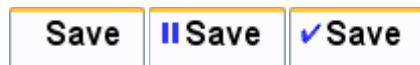
This button launches a configuration window for the associated data save profile.

- e. Log (file type) to disk



This button enables the associated data saving profile.

- f. Data Saving Disabled, Enabled, Active



The status of ENLIGHT data saving is displayed by the presence or absence of an indicator icon on the left of the Save tab. When no data saving profiles are active, the Save tab has no blue icon displayed. When data saving is for any profile is enabled but none are currently active, a blue “pause” icon is displayed. When data saving for any profile is currently active, a blue “check” icon is displayed.

B. Data Saving Profile Types

ENLIGHT now supports five independent data save profiles for four different data types.

- a. Sensors and Sensors 2

The sensor profile logs calibrated sensor data to a text file based upon profile configuration parameters. Two independent sensor data saving profiles are currently supported.

NOTE: When a sensors data saving profile is enabled, all FBG and Sensor controls are locked to prevent changes.

- b. Peak Wavelengths

The peak wavelengths profile logs raw spectral peaks as returned by the interrogator to a text file based upon profile configuration parameters. One Peak Wavelengths profile is currently supported.

- c. Spectral Response

The Spectral Response profile logs the full optical spectral response data as returned by the interrogator to a text file based upon profile configuration parameters. One Spectral Response profile is currently supported.

- d. Events

The Events profile logs the complete contents of the events table on the Alerts tab to a text file based upon profile configuration parameters. One Events profile is currently supported.

C. Data Saving Profile Controls

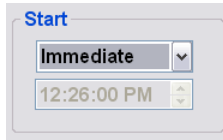
The following controls allow the user to craft when and how the data log files are generated, stored, and transmitted.

NOTE: Saving times, lengths, intervals, and size limits are all best effort and subject to finite timing resolution and



restrictions.

a. Start



This group of controls governs when the associated data saving profile begins.

i. Mode

- Immediate:

Data saving will initiate as soon as the Log button is pressed.

- Next Minute/Hour/Day:

Data saving will initiate when the PC system clock advances to the next specified unit of time. Use of these setting can offer a convenient means of starting multiple data saving profiles simultaneously.

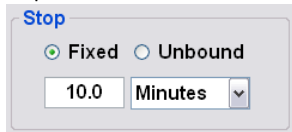
- Time:

Data saving will initiate when the PC system clock reaches time specified by the time control below.

ii. Time Control

- Specifies the time at which a saving profile with Start Mode->Time will initiate.

b. Stop



This group of controls governs when the associated data saving profile will end.

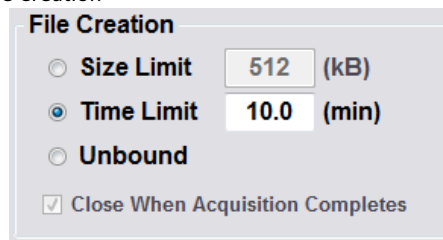
i. Fixed

Data saving will continue for the amount of time governed by the numeric time and units controls in the Stop controls group.

ii. Unbound

Data saving will continue until the profile is disabled by releasing the Log button.

c. File Creation



This group of controls governs how file sizes for the associated data saving profile will be limited.

i. Size

When a data file reaches the size determined by the (kB) control, the file will be closed and emailed (if so configured).

ii. Time

When a data file has been open for an amount of time greater than the (min) control, the file will be closed and emailed (if so configured).



iii. Unbound

No actions will be taken to limit the data file size. The file size will be determined exclusively by the data saving frequency, rate, and overall duration (from Start to Stop).

iv. Close When Acquisition Completes

Independent of the Limit By control setting, the data profile can be configured to close and email (if so configured) the data file when any repeated or triggered acquisition event completes. File Creation controls can be used in combination with this setting to further tailor how/when files are closed and/or emailed.

d. Acquisition

Acquisition

Continual Length: 10.0 Seconds

Repeated Interval: 1.0 Minutes

Triggered On Warning Auto Reset

On Alarm Extend

On Missing FBG

This collection of controls governs how data is collected during the time between the Start and Stop conditions.

i. Continual

In Continual mode, data is logged to file at the maximum rate determined by the acquisition rate and any interleave settings.

ii. Repeated

In repeated mode, data can be periodically logged for user-defined length of time, repeating at a user-defined interval.

1. Length – the amount of time that a single acquisition lasts.
2. Interval – the time period between the start of repeated acquisitions. By definition, the Interval setting must be longer than the Length setting.

iii. Triggered (Sensor profiles only)

Sensor data saving profiles can be configured to log data when one or more of the following trigger conditions are met.

1. On
-Warning:
Data logging will begin if any selected sensor goes into a warning state.
-Alarm:
Data logging will begin if any selected sensor goes into an alarm state.
-Missing FBG:
Data logging will begin if an FBG is identified as missing from its defined bin.
2. Auto Reset
Once a trigger event has occurred and the acquisition is complete based on its Length setting, an active Auto Reset setting of True will reset the triggered profile such that a subsequent trigger event initiates another data logging acquisition.
3. Extend
If a trigger event that initiates a data saving acquisition repeats during the data logging acquisition Length, an active Extend setting of True will cause the data acquisition Length to extend from that additional trigger event by an additional amount of time defined by Length.



e. File Contents

File Contents
Timestamp Format: Full
 Include Header
 Details Module Config.
 Initial File Only

i. Timestamp format

All time stamp formats originate from the interrogator core (x25 or x30). Like any PC clock, it will drift. In your case you'll use the NTP server to keep this drift under control.

None:

ENLIGHT will not append any form of a timestamp for each consecutive data acquisition.

Delta:

ENLIGHT stores the original UTC time from the sm130 and subtracts successive time stamps to report a delta time in 0.000000 format. We report to the microsecond to reduce rounding errors in the millisecond. The original UTC time is included in the header (if save header is selected).

Full:

ENLIGHT uses your PC clock to create the initial timestamp and then adds the Delta for successive timestamps. The PC clock is used so that local time zone settings are reflected in the timestamps. Timestamps are presented in mm/dd/yyyy/hh:mm:s.000000.

Native:

ENLIGHT reports the interrogator's internal clock reading for each dataset in microseconds since UTC 00:00:00 on January 1, 1970.

ii. Include Header

When checked this control appends a header to the top of all logged data files that incorporates information about the ENLIGHT configuration as in the following x30 example:

Date: 2/9/2010 21:33:10.44458

Data Save Interval: 1
Timestamp Format: Delta

Name:
Description:



Module Type: SM130
Mux Level: No Switch
HW Acquisition Rate: 1000 Hz
Wavelength Tracking: 400 pm / acquisition Data Interleave: 1

IP Address: 10.0.0.58
Port: 1852

IDN: sm130
Image ID: 2.0h
S/N: SIA6HF

When a header is included in the file, the first line of the file is populated with a number that defines the number of lines that of header information before logged data begins.

iii. Initial file only

For data save profiles that generate more than one data file, this setting allows the user to restrict the header to the first file only.

iv. Module Configuration

Includes module configuration information in the file header as in the following x30 example:

CH 1 Configuration:
Distance Compensation Enabled: False
Spectral Average Count: 1
Gain (dB): 0.0 dB
Noise Thresh.: 200

CH 2 Configuration:
Distance Compensation Enabled: False
Spectral Average Count: 1
Gain (dB): 0.0 dB
Noise Thresh.: 200

CH 3 Configuration:
Distance Compensation Enabled: False
Spectral Average Count: 1
Gain (dB): 0.0 dB
Noise Thresh.: 200

CH 4 Configuration:
Distance Compensation Enabled: False
Spectral Average Count: 1
Gain (dB): 0.0 dB
Noise Thresh.: 200

f. Included Data (Sensor profiles only)



This collection of controls governs additional included data for Sensor data saving profiles.

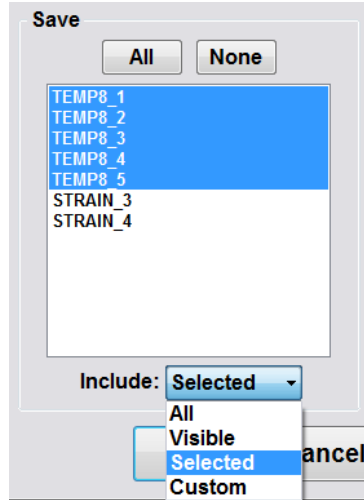
i. FBG

If selected, each sensor log entry will also include the raw wavelength values for the sensors' component FBGs.



- ii. Pre-Trigger (Triggered acquisition only)
If selected, the logged data file will include a number datasets from before the trigger event, determined by the “# Pre-Trigger Scans” control.

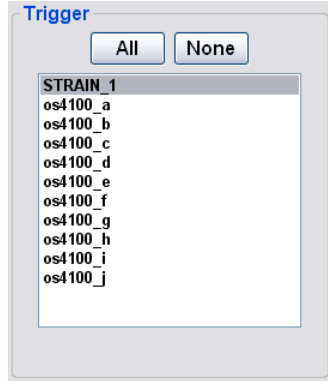
g. Save (Sensors profiles only)



This collection of controls governs which sensors are saved to the logged data file.

- i. Selection Mode
 - 1. All – All sensors from the Sensors table will be saved to the data file, regardless of when the sensors were created relative to when the data save profile was created.
 - 2. Visible – All visible sensors from the Sensors table will be saved to the data file, regardless of when the sensors were created relative to when the data save profile was created.
 - 3. Selected
Allows the user to choose specific sensors for inclusion in the logged data file.
 - a. Individual selection (control-click to select)
 - b. All – All of the sensors present at the time the data save profile was created will be included. Any sensors created after the data save profile was created will not be included unless the profile is specifically modified to include them.
 - c. None

h. Triggered Sensors (Triggered Acquisition mode only)



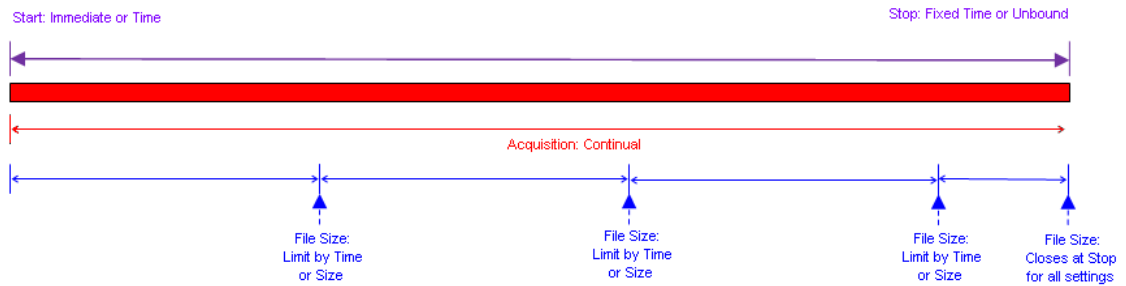
This collection of controls governs which sensors are monitored for trigger events to initiate a triggered data saving acquisition.

- Individual selection (control-click to select)
- All – All of the sensors present at the time the data save profile was created will be used as trigger sources. Any sensors created after the data save profile was created will not be used unless the profile is specifically modified to include them.
- None



D. Data saving examples:

1. Save **continuously without acquisition limit**, close (and email) files at file size or time limit from defined Start until Stop.



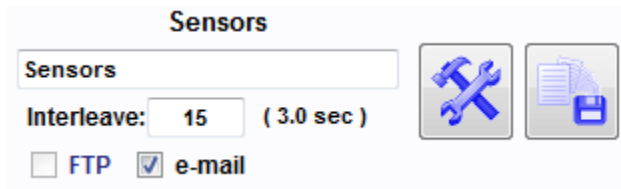
Example 1a

Goal:

Save data from a 1Hz sm125 every 15 seconds Hz for 7 days, closing and emailing file every night at midnight.

Implementation:

Interleave: 15. E-mail: True. Start: Time: 12:00:00am. Stop: Fixed = 7 Days. File Size: Time Limit =1440 min. Acquisition: Continual.





Example 1b

Goal:

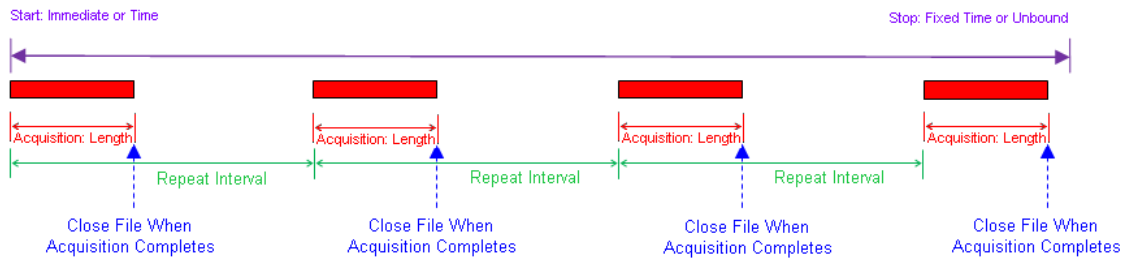
Save data from a 1 kHz sm130 at a rate of 500 Hz for 2 hours, closing file (without emailing) when file sizes reach 1 MB.

Implementation:

Interleave: 2. Email: False. Start: Immediate. Stop: Fixed = 2 Hours. File Size: Size Limit = 1024 kB. Acquisition: Continual.



- 2. Save for limited acquisition lengths (time), repeat acquisitions on a pre-defined time interval. Close (and email) files when each acquisition completes from the defined Start until Stop.



Example 2a

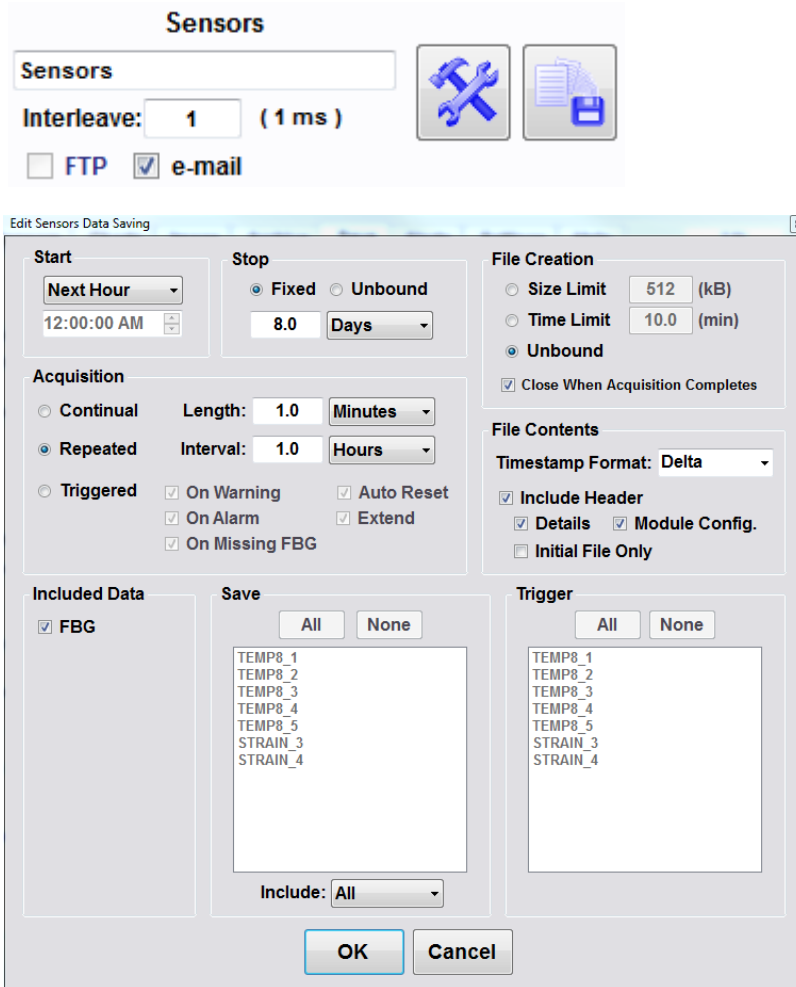
Goal:



Save and email 1 minute bursts of full speed data from a 1kHz sm130 data every hour for 8 hours starting at the top of the next hour.

Implementation:

Interleave: 1. E-mail: True. Start: Next Hour. Stop: Fixed = 8 Hours. File Size: Unbound, Close When Acquisition Completes: True. Acquisition: Repeated, Length: 1 Minute, Interval: 1 Hour.



Example 2b

Goal:

Save 5 minutes of 1 Hz data from an sm125 system every 2 hours. Close and email file every 12 hours. Continue this pattern for one year.

Implementation:

Interleave: 1 E-mail: True. Start: Next Hour. Stop: Fixed = 365 Days. File Size: Time Limit = 720 (min). Acquisition: Repeated, Length: 5 Minutes, Interval: 2 Hours.



Sensors

Sensors

Interleave: (1 ms)

FTP e-mail

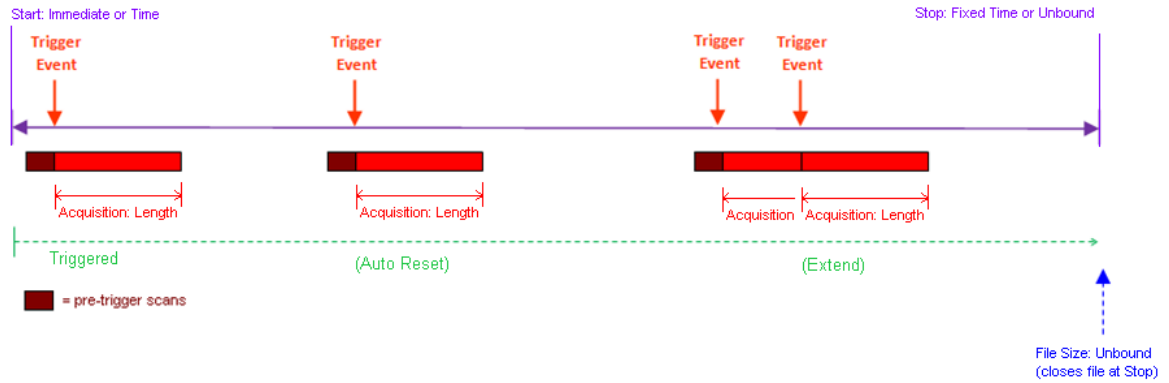
 

Edit Sensors Data Saving

Start Next Hour 12:00:00 AM	Stop <input checked="" type="radio"/> Fixed <input type="radio"/> Unbound 365.0 Days	File Creation <input type="radio"/> Size Limit 512 (kB) <input checked="" type="radio"/> Time Limit 720.0 (min) <input type="radio"/> Unbound <input checked="" type="checkbox"/> Close When Acquisition Completes
Acquisition <input type="radio"/> Continual Length: 5.0 Minutes <input checked="" type="radio"/> Repeated Interval: 2.0 Hours <input type="radio"/> Triggered <input checked="" type="checkbox"/> On Warning <input checked="" type="checkbox"/> Auto Reset <input checked="" type="checkbox"/> On Alarm <input checked="" type="checkbox"/> Extend <input checked="" type="checkbox"/> On Missing FBG		File Contents Timestamp Format: Delta <input checked="" type="checkbox"/> Include Header <input checked="" type="checkbox"/> Details <input checked="" type="checkbox"/> Module Config. <input type="checkbox"/> Initial File Only
Included Data <input checked="" type="checkbox"/> FBG	Save All None TEMP8_1 TEMP8_2 TEMP8_3 TEMP8_4 TEMP8_5 STRAIN_3 STRAIN_4 Include: All	Trigger All None TEMP8_1 TEMP8_2 TEMP8_3 TEMP8_4 TEMP8_5 STRAIN_3 STRAIN_4

3. Save for limited acquisition lengths (time), trigger acquisitions (with Auto Reset for additional triggers and Extend acquisition time for repeated trigger events within the acquisition time). Save all Pre-Trigger Scans on each acquisition. Close (and email) files When Acquisition Completes complete from the defined Start until Stop.

Example 3a



Goal:

It is expected that trains crossing a bridge will trigger a warning event on sensor STRAIN_1 and will clear the bridge in less than two minutes. The goal is to save 1 kHz data for 10 seconds before and 2 minutes after each train triggers a STRAIN_1 warning. Should a train be longer or moving more slowly than expected, 1 kHz data should continue to be acquired until the train has completely passed. Continue this program for 24 hours, emailing an update at the end.

Implementation:

Pre-Trigger Scans: 10,000. Interleave: 1. E-mail: True. Start: Immediate. Stop: Fixed = 24 Hours. File Size: Unbound. Close When Acquisition Completes: False. Acquisition: Triggered, On Warning: True, On Alarm: False, On Missing FBG: False, Auto Reset: True (multiple trains), Extend: True (slow or long trains). Length: 2 Minutes. Include Pre-Trigger Data: True. Triggered Sensors: STRAIN_1. Saved Sensors: All.

Pre-Trigger Scans:





Sensors

Sensors

Interleave: (1 ms)

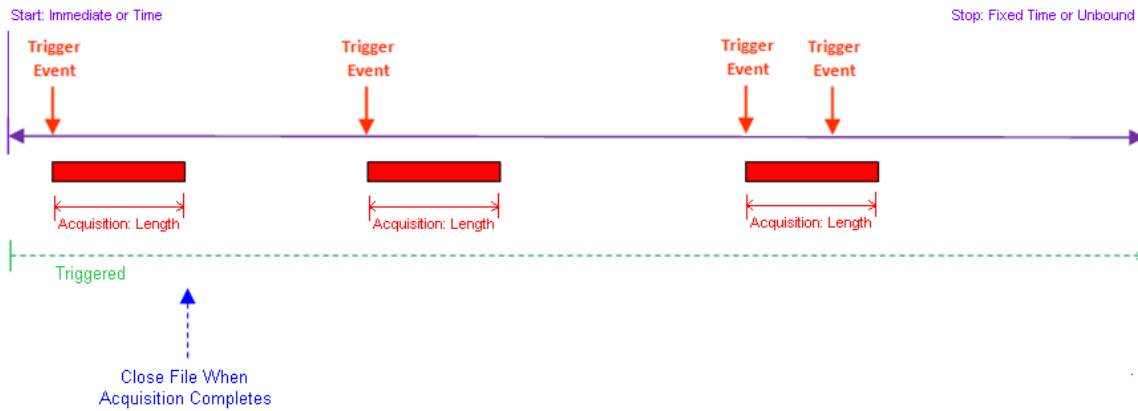
FTP e-mail



Edit Sensor1 Data Saving

Start Immediate 12:00:00 AM	Stop <input checked="" type="radio"/> Fixed <input type="radio"/> Unbound 24.0 Hours	File Creation <input type="radio"/> Size Limit 256 (kB) <input type="radio"/> Time Limit 10.0 (min) <input checked="" type="radio"/> Unbound <input type="checkbox"/> Close When Acquisition Completes
Acquisition <input type="radio"/> Continual Length: 10.0 Seconds <input type="radio"/> Repeated Interval: 1.0 Hours <input checked="" type="radio"/> Triggered <input checked="" type="checkbox"/> On Warning <input type="checkbox"/> Auto Reset <input type="checkbox"/> On Alarm <input type="checkbox"/> Extend <input type="checkbox"/> On Missing FBG		File Contents Timestamp Format: Native <input checked="" type="checkbox"/> Include Header <input checked="" type="checkbox"/> Details <input checked="" type="checkbox"/> Module Config. <input type="checkbox"/> Initial File Only
Included Data <input checked="" type="checkbox"/> FBG <input checked="" type="checkbox"/> Pre-Trigger	Save All None STRAIN_1 Include: Selected	Trigger All None STRAIN_1

Example 3B



Goal:

Save all 1kHz sm130 sensor for 6 seconds and send e-mail immediately the first time any perimeter sensor reaches a warning, alarm or sensor missing state. Start at 8 pm and maintain watch program for 1 ½ hours.

Implementation:

Interleave: 1. e-mail: True. Start: Time->8:00:00 pm. Stop: Fixed->90 Minutes. File Size: Unbound. Close When Acquisition Completes: True. Acquisition: Triggered, On Warning: True, On Alarm: True, On Missing FBG: True, Auto Reset: False, Extend: False, Length: 6 seconds. Include Pre-Trigger Data: False. Triggered Sensors: All. Saved Sensors: All.

Sensors

Sensors

Interleave: (1 ms)

FTP e-mail

Edit Sensor1 Data Saving

Start Time 8:00:00 PM	Stop <input checked="" type="radio"/> Fixed <input type="radio"/> Unbound 90.0 Minutes	File Creation <input type="radio"/> Size Limit 256 (kB) <input type="radio"/> Time Limit 10.0 (min) <input checked="" type="radio"/> Unbound <input checked="" type="checkbox"/> Close When Acquisition Completes
Acquisition <input type="radio"/> Continual Length: 10.0 Seconds <input type="radio"/> Repeated Interval: 1.0 Hours <input checked="" type="radio"/> Triggered <input checked="" type="checkbox"/> On Warning <input type="checkbox"/> Auto Reset <input checked="" type="checkbox"/> On Alarm <input type="checkbox"/> Extend <input checked="" type="checkbox"/> On Missing FBG	Included Data <input checked="" type="checkbox"/> FBG <input type="checkbox"/> Pre-Trigger	File Contents Timestamp Format: Delta <input checked="" type="checkbox"/> Include Header <input type="checkbox"/> Details <input checked="" type="checkbox"/> Module Config. <input type="checkbox"/> Initial File Only
Save <input type="button" value="All"/> <input type="button" value="None"/> STRAIN_1 Include: Selected	Trigger <input type="button" value="All"/> <input type="button" value="None"/> STRAIN_1	<input type="button" value="OK"/> <input type="button" value="Cancel"/>

Note: Files will always close (and email, if so configured) at the end of the stop condition, independent of any other File Size settings.

11.3.8. Alerts Tab

The Alerts tab is the section of the ENLIGHT software where instrument, application, and FBG/sensor events are logged and communicated. The Alerts tab is divided into two sections. The top half of the tab deals with the transmission of notification messages, while the bottom serves to log events.

11.3.8.1. Message Notifications

The top section of the Alarms tab serves to manage the distribution of notifications via email.

A. Message Notifications table

Message Notification	
Name	E-Mail Address
✓ ENLIGHT_user	enlight_user@isp.com
✗ ENLIGHT_user_2	ENLIGHT_user_2@isp.com

Notification Delay (sec): 10

The notifications table shows the list of defined recipients for notification events. The parameters that the table reports for each recipient are as follows:

- i. Name:
The name of the recipient who is to receive notifications.
- ii. Email address:
The email address at which the recipient is to receive notifications.

B. Buttons

The following buttons are collocated with the notifications table

- i. Edit Recipient



Launches a data entry form to manage the attributes of a selected recipient.

- ii. Add Recipient



Creates a new recipient entry and launches a data entry form to for the user to input the attributes of that new recipient.

- iii. Delete Recipient



Deletes the selected recipient.

iv. Delete All Recipients



Deletes all recipients from the current configuration

v. Activate/Deactivate Recipient

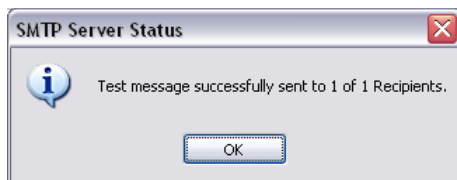


Qualifies the selected recipient to participate in notification events

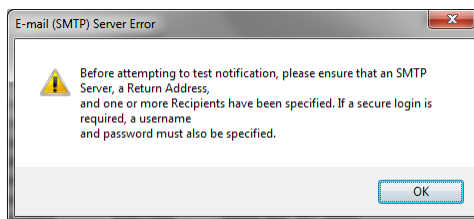
vi. Test E-mail



Creates and sends a test email from the selected return address to any active recipients on the Alerts tab using the SMTP settings from the Settings tab, Notification section. A successful test email will be indicated by the following pop up message:



An unsuccessful attempt will be indicated by a different message, outlining details of the failure as follows.



vii. Enable/Disable Alerts

Alerts Enabled



Any active recipient will receive the subscribed alerts when available. Active recipients will be identified by a blue check and a black font on the recipients table. Inactive recipients will be identified by a red X and gray font on the notifications table.

Alerts Disabled.



All recipients, whether active or inactive, will be identified with a gray font on the notifications table, as alerts are disabled and no messages will be sent.

viii. Notification delay

Notification Delay (sec):

Sets the delay time in seconds over which all alert notifications will be aggregated into a single message.

C. Configure notifications

	Sensor	Instrument	Application
Alarm	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Warning	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Information	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>


OK Cancel

The preceding figure shows the data entry form used to configure the attributes of a selected notification recipient. In addition to the Information parameters already discussed in the previous section, there are controls to filter the type and severity of information that will be considered for triggering a notification event.

In the figure above, there is a 3 by 3 entry matrix of check box controls. The column labels show the three possible sources of notification information: sensor, instrument, and application events. The row labels show the three levels of severity of the events from those sources: Information, Warning, and Alarm. Use of this control matrix allows for customization with regard to the source and severity of the events that will trigger notification events for each individual recipient.

D. Sound notifications

Sound Notification



- Play On Alarm**
- Latch On Alarm**
- Play On Warning**
- Latch On Warning**

Interval (sec)

The Sound Notification feature alerts the user with an alarm when defined warning and alarm parameters have been met.

- i. **Play:**
Will play a sound for the entire duration any defined settings have reached an Alarm or Warning state.
- ii. **Latch:**
Will play a sound when any Alarm or Warning is set-off until the specific alert been acknowledged and cleared by the user.
- iii. **Interval**
Defines the interval at which the sound will not play.

11.3.8.2. Event Log

The event log shows a listing of occurrences produced by the measurement system. The complete list can be filtered for viewing based upon several criteria controls.

A. Table

Event Log

Date	Severity	Type	Description
6/26/2008 3:08:43 PM	Information	Sensor	'cs_4' expression status changed to 'True'
6/26/2008 3:08:43 PM	Information	Sensor	'cs_4' expression changed from " to '0'
6/26/2008 3:08:43 PM	Information	Application	Transducer 'cs_4' added
6/26/2008 3:08:35 PM	Information	Sensor	'cs_3' expression status changed to 'True'
6/26/2008 3:08:35 PM	Information	Sensor	'cs_3' expression changed from " to '0'
6/26/2008 3:08:35 PM	Information	Application	Transducer 'cs_3' added
6/26/2008 3:08:27 PM	Information	Sensor	'cs_2' expression status changed to 'True'
6/26/2008 3:08:27 PM	Information	Sensor	'cs_2' expression changed from " to '0'
6/26/2008 3:08:27 PM	Information	Application	Transducer 'cs_2' added
6/26/2008 3:07:40 PM	Information	Application	Configuration saved to 'Monitoring Station 2.cfg' on 6/26/2008 3:07 PM
6/26/2008 3:07:40 PM	Information	Application	Configuration successfully saved to 'Monitoring Station 2.cfg'
6/26/2008 3:06:55 PM	Alarm	Sensor	FBG 'A3' could not be located in wavelength range (1538.29 nm, 1541...
6/26/2008 3:01:15 PM	Alarm	Sensor	Transducer 'saddle_strain' value of 360.36 has exceeded its alarm lim...
6/26/2008 3:00:50 PM	Warning	Sensor	Transducer 'saddle_strain' value of 218.61 has exceeded its warning L...
6/26/2008 2:58:44 PM	Information	Instrument	Scanning enabled
6/26/2008 2:58:43 PM	Information	Instrument	Port 16 disabled

The image above shows the event log with multiple entries. To the left side of the chart are a series of filter controls that limit which events are shown in the table. To the right side are two controls used to delete or save the event log.

B. Severity filters

These controls are used to filter the events shown in the table based on severity.

i. Hide/Show Alarm Events



Hide/Show events classified with a severity of "Alarm". A severity of "Alarm" indicates that something potentially dangerous or fatal has occurred.

ii. Hide/ Show Warning Events



Hide/Show events classified with a severity of "Warning". A severity of "Warning" indicates that something important has occurred in the system.

iii. Hide/Show Information Events



Hide/Show events classified with a severity of "Information". A severity of "Information" indicates that a non-threatening event has occurred or that an aspect of the programs operations has been modified.



C. Source Filters

i. Hide/Show Sensor Events



Hide/Show events generated by Sensors.

ii. Hide/Show Instrument Events



Hide/Show events generated by the Interrogator Instrument or module interrogator core.

iii. Hide/Show Application Events



Hide/Show events generated by the MOI ENLIGHT Application.

D. Delete



Deletes all events from the Event Log.

E. Save



Captures the event log and makes it available as a file via the import/export tool.

The event log is automatically cleared upon exit of an ENLIGHT session.


11.3.9. Settings Tab


The settings tab is used to manage global settings for the application. The tab is subdivided into the sections outlined below.

11.3.9.1. Hardware

Hardware

ENLIGHT Rev: 1.5.25

IP Address:	<input type="text" value="10.0.0.79"/>		Serial #:	<input type="text" value="SIA8Q6"/>
Port:	<input type="text" value="50000"/>		FW Rev.:	<input type="text" value="2.109"/>
Timeout (sec):	<input type="text" value="5"/>		SL Core:	<input type="text" value="SM125v2"/>



This section allows the user to configure the settings for and view information about the target instrument or SL interrogator core to which ENLIGHT will connect.

A. IP Address

Displays the IP address of the target interrogator module or core.

B. Port

Displays the port over which TCP/IP communication will take place. A compatible default port number is automatically assigned when the target module type is selected by the user.

C. Timeout (sec)

Displays the TCP/IP communication timeout. ENLIGHT will timeout if no client responds to the configured IP address. The default x25 and x35 interrogator module default timeout duration is 5 seconds. These settings are typically not changed by the user.

D. S/N

Displays the serial number for the presently connected SL interrogator core.

E. FW Rev:

Displays the firmware revision of the target SL interrogator core.

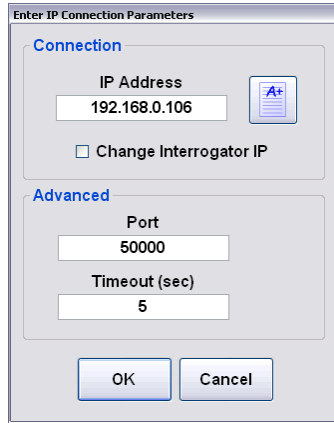
F. SL Core:

Displays the interrogator core type, by model number.

G. Edit Connection



Launches a data entry form for configuring the connection parameters of IP address, port, and timeout of the target interrogator core, as seen in the following image.



When already connected to an Interrogator core, the IP address of that core can be changed by updating the IP Address field and checking the “Update Interrogator” checkbox. After clicking “OK”, the x25 or x30 core will reboot in order to facilitate the change of IP address, and a socket connection will be re-established with the interrogator core on the new, updated IP address.

H. Timeout

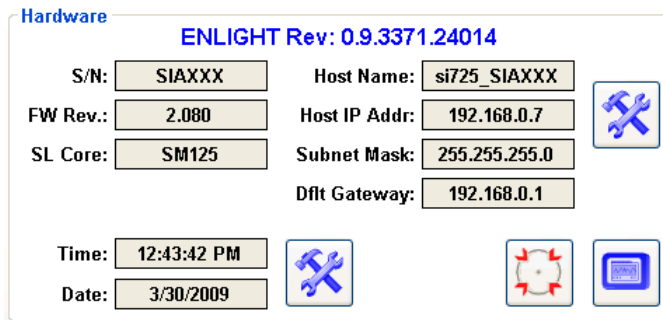
Displays the TCP/IP communication timeout.

I. Command Console



Launches a communication window that allows for direct commands and queries to the optical interrogator core. All commands outlined in sections 4.1.2.6 and 4.2.2.6 of this document are supported by the command console, except for commands ignored by “embedded” interrogator cores, as noted in section 5.2.5 of this document.

The remainder of this section shows the controls that pertain to Integrated ENLIGHT as embedded in Interrogator Instruments.



J. Host Name

Displays the host name of the interrogator instrument, which can be used for Remote Desktop connections and Remote Data Access.

K. Host IP Addr., Subnet Mask, Dflt Gateway



Displays the host IP address, Subnet Mask, and Default Gateway of the interrogator instrument, which can also be used for Remote Desktop connections and Remote Data Access.

Calibrate Touch screen (si3XX and si7XX Instruments only)



Initiates a touch screen calibration routine that can be used to restore alignment accuracy of the data input device for touch screen equipped interrogator instruments.

L. Time

Shows the current time setting for the interrogator instrument

M. Date

Shows the current date setting for the interrogator instrument.


N. Edit Time and Date



Launches a settings form for configuring the current time and data of the interrogator instrument, as shown in the following image.



Notification

Pending Messages
 
 (15 minute retry interval)

Return E-mail Address

IP Address or Name **Port** **SSL**

Username **Password** **Login**

A. Pending Messages

ENLIGHT will queue any email alerts that failed to send and continue to retry send every 15 minutes for a period of 24 hours.

B. Delete Pending Messages



Deletes any email alerts in the retry queue.

C. Return Email Address

Set the return email address used for all notification messages.

D. IP address or Name

Sets the IP address or name of the SMTP server used to send out email notification messages.

E. Port

Sets the port for communication to the SMTP server that is used to send out email notification messages.

F. SSL

Configures the outgoing email notifications to use secure socket layer communication to the SMTP server

G. Login

Indicates that a login is required for use with the specified SMTP Server.

H. Username

Sets the username for use in authentication by the specified SMTP server.

I. Password

Sets the password associated with the aforementioned Username for use in authentication by the specified SMTP Server

11.3.9.3. Configuration Notes



Notes entered into the Name and Description fields will be included in all email alerts generated by the present ENLIGHT configuration.

Configuration Notes

Name:

Description:

A. Name

The name of the configuration to be included with the email alerts

B. Description

A text description of the measurement configuration to be included with the email alerts

11.3.9.4. Application Settings

Application Settings

Virtual Keyboard & Keypad Remote Command Interface

Lock Configuration Port: #Clients:

Distance Compensation Web Interface (<http://andrewp>)

Display Full Screen

A. Virtual Keyboard & Keypad Enabled

Enables or disables the on-screen virtual keyboard and keypad, intended for use with touch screen instruments. The touch screen can be disabled for use with an external USB keyboard and mouse, if desired.

B. Lock Configuration

Disables access to most controls of the ENLIGHT user interface, to prevent inadvertent control value changes.

C. Distance Compensation

Globally enables the distance compensation feature. See the appendix entitled “Speed of Light Effects on SL-based Wavelength measurements” in the Appendices section of this document for details

D. Remote Command Interface

Enables the ENLIGHT Remote Command interface, which allows for programmatic data extraction from the ENLIGHT program. Please see section 7.4 of this document for details.

New in v1.5!

E. Enable Web Interface

Enables the ENLIGHT Web Interface, which allows for http access to data stored in the ENLIGHT Data Archive. Please see section 7.5 of this document for details.



11.3.9.5. Module Diagnostics

Module Diagnostics

Diagnostic File Path
 C:\Documents and Settings\tchaber.DOMAIN1\My Documents\Micron Optics\ENLIGHT\Data\

Extracts a core diagnostics file that can be used by Micron Optics to diagnose field issues with instruments and modules. Will reboot the module and generate a file that can then be sent via email to an MOI contact.

A. Select Target Location



Selects the target directory in which the module diagnostics file will be stored.

B. Create Diagnostics File




Initiates the core diagnostics file generation process.

New in v1.5!

11.3.9.6. Data Archiving

Data Archiving

Name (0.016 GB)
 ENLIGHT

[Import](#) [Export/Backup](#) [Delete](#) 

Update Interval
 1 min

Maximum Size
 1.0 GB

The Data Archiving controls govern the location, update interval, and maximum size of the ENLIGHT Data Archive. FBG and sensor data that is stored in the Data Archive can be plotted and viewed with the controls and indicators of the Archive tab, detailed in section 7.3.6 of this manual.

A single archive entry includes all defined FBG’s, raw wavelength values, defined sensors, and sensor measurement values in converted engineering units. Data Archive entries are complementary and completely independent from data configured to save to file on the Save tab. Data Archive entries are made automatically by the ENLIGHT software and do not require any configuration by the user.

A. Data Archive Management

The following controls manage the specific target Data Archive.

i. Name

Name (0.016 GB)
 ENLIGHT

This indicator displays the name of the active ENLIGHT Data Archive. The current size of that particular Data Archive is displayed in between the parentheses.

ii. Select ENLIGHT Data Archive Control

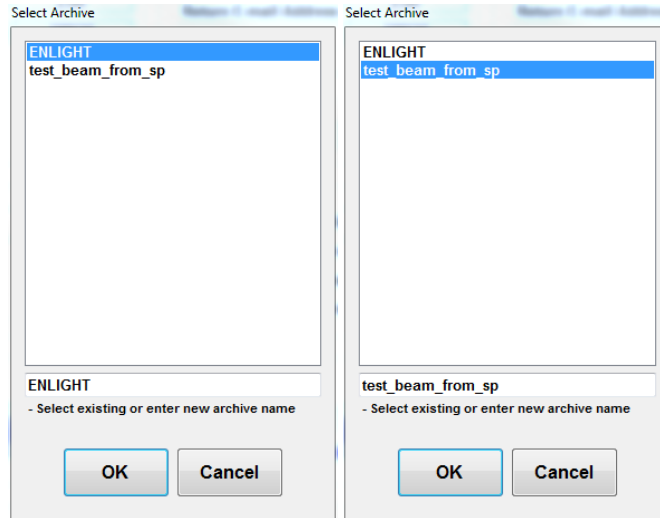


This control enables the user to select an existing or create a new ENLIGHT Data Archive.

1. Selecting an existing Data Archive

After clicking on the Select Data Archive control, the user will be presented with a list of existing Data Archives. To open one of these existing archives, the user can highlight the desired file and click "OK" to open that file.

The screenshots below show 'ENLIGHT' as the original active archive. The user then highlights a different archive, named 'test_beam_from_sp,' and then clicks OK to load that archive.



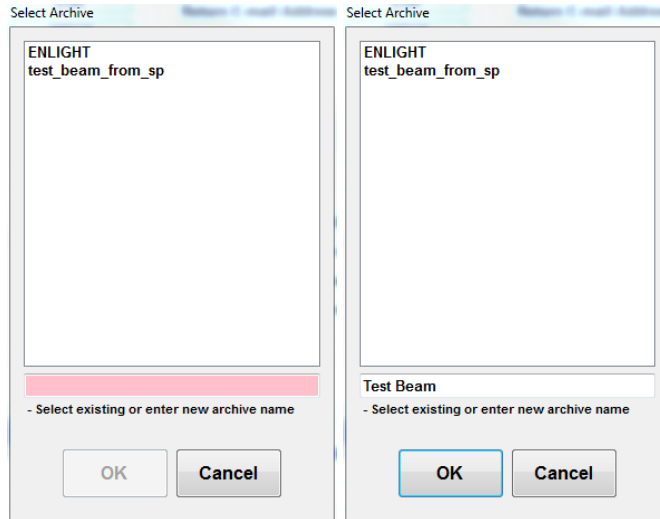
The screenshot below shows the selected archive successfully loads when the Name field displays the archive name as well as the total file size in between the parenthesis.





2. Creating a new Data Archive

The user can create a new Data Archive by typing the desired name in the blank field. The screenshots below show 'Test Beam' is entered in this field, which will be the new Data Archive name.



When the user hits the enter key or clicks the OK button, the new archive name will be displayed in the name box. Note, 0.000GB of data confirms no data has yet been stored in this new file.





iii. Export/Backup

Export/Backup

This control exports a copy of the active archive to a default target location of ...\Documents\Micron Optics\ENLIGHT\Data.

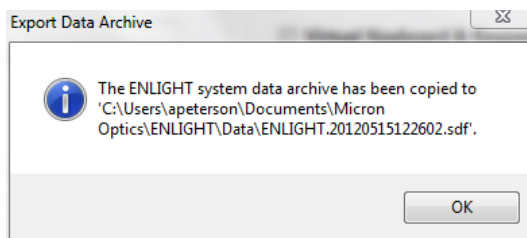
The Export/Backup feature serves as a means to create redundant backups of critical Data Archives, as well as facilitating migration of an ENLIGHT Data Archive to another installation of ENLIGHT.

Data Archives that approach the set Maximum Size will automatically truncate the oldest element data to prevent the file from exceeding the set file size limit. As such, it is recommended to consistently backup the archive when it approaches the maximum file size.

When the user clicks the Export/Backup button, he will be prompted to select a single Data Archive.



Once the user clicks OK, a pop-up window will open displaying the target file location upon the successful export of the file.

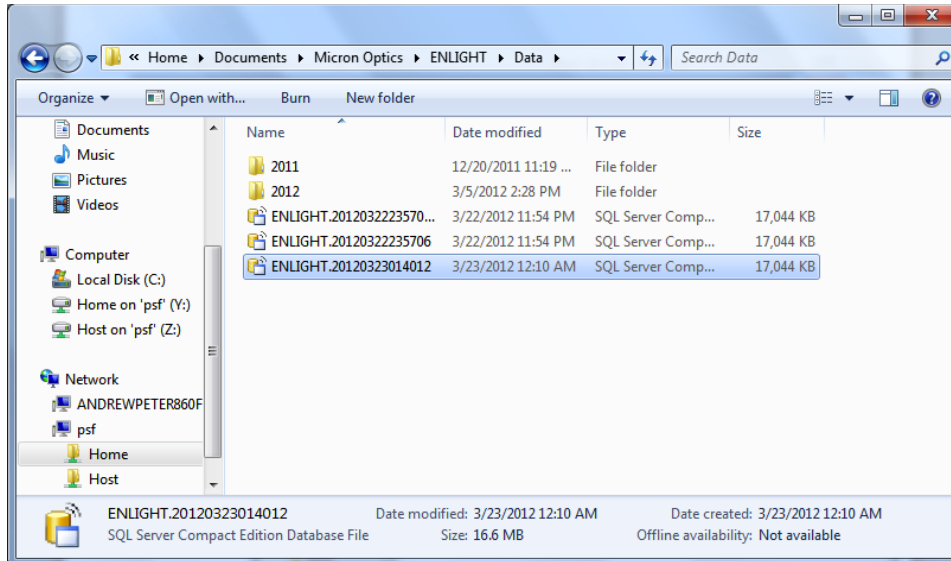


iv. Import

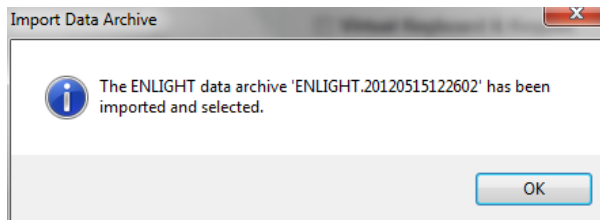
Import

The Import button allows the user to load exported archive files for viewing in the Archive tab.

When the Import button is clicked, a pop-up window will open containing archive files available for import into ENLIGHT. The screenshot below shows this window with a target file selected to be imported.



When the user double-clicks or hits Open, the file will load in ENLIGHT.

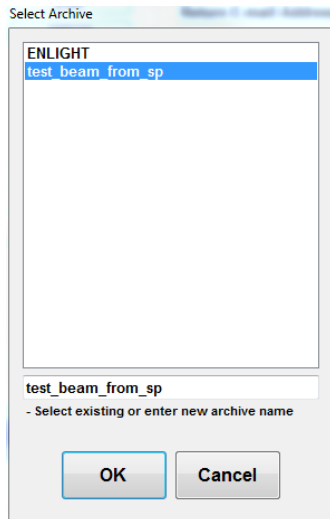




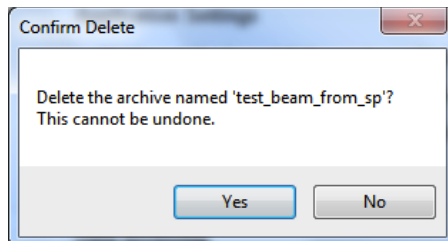
v. Delete

Delete

This function deletes the selected archive. In the event a Data Archive is no longer needed, this function will permanently delete the file.



If the user highlights a file name, and presses OK, a pop-up confirmation window will open.



Confirming the delete by clicking yes will permanently delete the file.

Note that the user cannot delete the active Data Archive running in ENLIGHT.



B. Update Interval

Update Interval

This control allows the user to set the time interval between subsequent ENLIGHT Data Archive entries. The default update interval is one minute.

C. Maximum Size

Maximum Size

This control allows the user to manage the maximum size of the archive. The default maximum Data Archive size is 1.0 GB.

A significant amount of information can be stored in the Data Archive, but there are two parameters that govern the rate at which the archive approaches the maximum size. The first is the total number of defined FBG and sensor elements, and the second is the time between update intervals. Although 1 GB of data is a relatively small file size relative to modern hard-disk drive capacities, this file size is large enough to store one year’s worth of data of 250 FBG and sensor elements, stored at the one minute default update interval. Users expecting to record more FBG or sensor elements over a longer period of time should adjust the data archiving controls to the maximum file size and a conservative update interval.

Once the maximum Data Archive size has been reached, consecutive archive entries will be truncated from the backend per each update interval so that the archive does not exceed the maximum size. Micron Optics recommends that a new Data Archive is created before the file reaches the maximum size in order to preserve all archived data.

11.3.9.7. Firmware Upgrade

Firmware Upgrade

File Location

Compatibility with the latest release of ENLIGHT software may require that the firmware for the target MOI interrogator core be updated. Each release of ENLIGHT will include the latest firmware for the target interrogator cores. Once connected to the interrogator, the user uses the button with the folder icon to browse to the target firmware image, and then use the button with the arrow button to initiate the firmware upgrade. On-screen instructions will guide the user through the installation process.

11.3.9.8. News & Updates

News & Updates

[Follow On Facebook](#) [Follow On Twitter](#)

Get up to date information on optical sensing technology and other news by following us on Facebook and Twitter.

11.4. ENLIGHT Remote Command Interface**11.4.1. TCP/IP Remote Control Interface**

ENLIGHT supports a collection of commands with which calibrated sensor and other types of data can be programmatically retrieved from the ENLIGHT program for use in other software applications. The following sections outline how connections are made to ENLIGHT, how commands/replies are sent/received, and what commands are supported by the interface.

11.4.1.1. Overview

The remote interface is accessed by establishing a TCP/IP socket between the client software and ENLIGHT. A maximum of 5 clients can be connected simultaneously. To establish a connection, the client software must have access to the same Ethernet subnet that is hosting ENLIGHT. By default, ENLIGHT accepts incoming connection requests on port 1853. If required (i.e. running multiple instances of ENLIGHT), the port can be changed in the Command Interface section of the Setting tab. Additionally, the firewall on both the client and ENLIGHT machine must be configured to allow TCP/IP traffic through the configured port.

11.4.1.2. Communication Protocol

ENLIGHT remote interface commands are case-insensitive ASCII strings that begin with a # character and are terminated with a linefeed (ASCII char 10 [0x0a in hex, '\n' in C, vLf in Visual Basic]). The maximum command length is 2048 characters. If more than 2048 characters are received before the command termination, the command receive is aborted and no response is sent. If the termination character is received before the maximum command length is reached, the received text will be parsed and processed. The string immediately following the initiation character represents the command to be executed. The command and its arguments are all separated by one or more spaces. Multiple spaces anywhere in the received string are ignored. All successfully received commands will produce a standard 6-byte response header in addition to any returned data. The response header contains a response payload length, type, and status. The format of the response is shown in the table below:

4-byte unsigned integer (Little Endian): Response length (number of bytes following the entire header)
 1-byte unsigned integer: Response Type
 1-byte unsigned integer: Response Status

The current Response Type values are:

CommandResponse = 0
 StreamingData = 1
 ArchiveData = 2
 Event = 3

The current Response Status values are:

Success = 0
 Unexpected Failure = 1
 Processing Error = 2
 Access Denied = 3
 Invalid Command = 4
 Invalid Number Of Arguments = 5
 Invalid Argument = 6



11.4.1.3. Data Retrieval (polling vs. streaming)

ENLIGHT supports two methods for retrieving data (i.e. peaks, amplitudes, sensors). The simplest method (data polling) is to asynchronously request the data through repeated execution of a command such as #GET_SENSOR_VALUES. Data polling is only appropriate for low-speed acquisition where the application does not require consistent and accurate acquisition timing. The second, more robust method (data streaming) allows high-speed continuous data to be delivered to the client application. The data is buffered in ENLIGHT on a per-client basis and thus ensures that all of the collected data is transmitted even when the network throughput temporarily drops below the rate required for the requested data. In both cases, the client can specify exactly which data to send (i.e. which sensors or FBGs), reducing the network bandwidth required between the host and client. Simultaneously various forms of data (sensors, FBGs, Events) can all be streamed over a single client connection in addition to non-streaming control commands.

11.4.1.4. User Command Set

As of this writing, the following commands are supported by the ENLIGHT Remote Command Interface:

#HELP

Syntax: #HELP

Description: The help command returns a complete listing of the ENLIGHT remote interface commands.

Example: #HELP

Response: #CLEAR_REFERENCE
#GET_AMP_CH
#GET_CAPABILITIES

#GET_ENLIGHT_VISIBLE

Syntax: #GET_ENLIGHT_VISIBLE

Description: Determine if the ENLIGHT User Interface is currently being displayed on the host machine.

Returns: 1 if visible; 0 if hidden

Example: #GET_ENLIGHT_VISIBLE

Response: 1



#SET_ENLIGHT_VISIBLE

Syntax: #SET_ENLIGHT_VISIBLE [visible]

Description: Hide/Show the ENLIGHT User Interface.

Parameters:

visible = 1 to show the interface, 0 to hide the interface

Example: #SET_ENLIGHT_VISIBLE 0

Response: The ENLIGHT interface is now hidden.

#GET_MODULE_CONNECTED

Syntax: #GET_MODULE_CONNECTED

Description: Determine if ENLIGHT is currently connected to the module.

Returns: 1 if connected; 0 if disconnected

Example: #GET_MODULE_CONNECTED

Response: 1

#SET_MODULE_CONNECTED

Syntax: #SET_MODULE_CONNECTED [connected]

Description: Connect/Disconnect ENLIGHT from the Module. If requested state of the connection is equal to the requested state, no action is taken. The command does not return until the requests completes so several seconds may be required. Ensure that the socket timeout is sufficiently large to account for the delay.

Parameters:

connected = 1 to connect to the module, 0 to disconnect from the module

Example: #SET_MODULE_CONNECTED 0

Response: The module is now disconnected.

#GET_ENLIGHT_VERSION

Syntax: #GET_ENLIGHT_VERSION

Description: Determine the ENLIGHT version.

Returns: The version of ENLIGHT currently installed.

Example: #GET_ENLIGHT_VERSION

Response: 1.0.0



#GET_ENLIGHT_CONFIGURATION

Syntax: # GET_ENLIGHT_CONFIGURATION

Description: Retrieve the name of the loaded ENLIGHT configuration file.

Returns: The configuration file name.

Example: #GET_ENLIGHT_CONFIGURATION

Response: 122011.moi

#SET_ENLIGHT_CONFIGURATION

Syntax: #SET_ENLIGHT_CONFIGURATION

Description: Load an ENLIGHT Configuration file from the default Configuration directory. To work properly, the filename cannot contain spaces.

Unless specified otherwise, the operating ENLIGHT Configuration file will save before the new Configuration file loads.

Also, regardless of the setting in the loaded configuration file, the Remote Command Interface will be enabled to ensure remote access is not removed.

Returns: A response confirming the specified ENLIGHT Configuration loaded successfully.

Example: #SET_ENLIGHT_CONFIGURATION 120211

Response: The specified configuration file was loaded.

#GET_MODULE_DIAGNOSTICS

Syntax: #GET_MODULE_DIAGNOSTICS

Description: Retrieve the module internal diagnostic data. The retrieved data can be used by Micron Optics to remotely investigate hardware and software issues related to the operation of the module. THIS COMMAND WILL CAUSE THE MODULE TO REBOOT UPON COMPLETION!!!

Returns: A binary array of data that corresponds to the contents of a password protected zipped file containing the module internal diagnostic data.

Example: #GET_MODULE_DIAGNOSTICS

Response: [binary array]

#GET_FBG_PROPERTIES

Syntax: #GET_FBG_PROPERTIES

Description: Get the setup information for all of the FBGs.

Returns: A chart listing defined FBG parameters and instantaneous wavelength data.

Example: #GET_FBG_PROPERTIES

ID	Channel	Current	Averages	Wavelength Min ...
FBG_C1	CH 3	1551.86661	1	1550.87221...
FBG_C2	CH 3	1555.960658	1	1554.968774...



#GET_SENSOR_PROPERTIES

Syntax: #GET_SENSOR_PROPERTIES

Description: Get the setup information for all of the Sensors.

Returns: A chart listing defined Sensor parameters, measurement threshold settings, and instantaneous measurement data.

Example: #GET_SENSOR_PROPERTIES

Response:	Name	Current	Warning Threshold	Warning Minimum...
	Strain_1	1.49226	0.8	-3371.0177605409...

#GET_SENSOR_IDS

Syntax: #GET_SENSOR_IDS

Description: Retrieve the list of defined Sensor IDs.

Returns: A space delimited list of Sensor IDs.

Example: #GET_SENSOR_IDS

Response: OS3100_L1 OS3100_R1 OS4100_Main

#GET_FULLSPECTRUM

Syntax: #GET_FULLSPECTRUM

Description: Gets the full optical spectral response (reflected amplitude vs value).

If no arguments are supplied, all enabled sub-channels (based on the Mux Level and type of connected optical switch) are returned. Each sub-channel is returned as a 20-byte header followed by the data. The header consists of a double (8-byte) value start + double (8-byte) value delta + integer (4-byte) number of response bytes. Each requested sub-channel is returns sequentially in the response bytes.

8 - byte double	8 - byte double	4 - byte integer	(n * 8) - byte double	Channel 1
WL Start	WL Step	# Points (n)	Amplitudes	

8 - byte double	8 - byte double	4 - byte integer	(n * 8) - byte double	Channel 2
WL Start	WL Step	# Points (n)	Amplitudes	

8 - byte double	8 - byte double	4 - byte integer	(n * 8) - byte double	Channel 3
WL Start	WL Step	# Points (n)	Amplitudes	

8 - byte double	8 - byte double	4 - byte integer	(n * 8) - byte double	Channel 4
WL Start	WL Step	# Points (n)	Amplitudes	

Total Length = (Number of Channels * (20 + (n * 8))) bytes

Example: #GET FULLSPECTRUM

Response: An ASCII character set. Please see the description above for parsing instructions.



#GET_SENSOR_VALUES

Syntax: #GET_SENSOR_VALUES [ids]

Description: Get the current values of a list of specified Sensor IDs. If no arguments are supplied, the values of all defined Sensors are returned. The number of IDs passed as arguments is limited by the maximum command length of 2048 bytes.

Parameters:

If specified, a space delimited list of Sensor IDs. If an ID does not exist, NaN is returned. If no ids are specified, the values for ALL Sensors are returned.

Returns: A space delimited list of Sensor values.

Example: #GET_SENSOR_IDS OS3100_L1 OS4100_MAIN

Response: -100.325 2.647

#GET_STREAMING_ENABLED

Syntax: #GET_STREAMING_ENABLED

Description: Determine if data streaming is enabled/disabled.

Returns: 1 if enabled; 0 if disabled

Example: #GET_STREAMING_ENABLED

Response: 1

#SET_STREAMING_ENABLED

Syntax: #SET_STREAMING_ENABLED [enabled]

Description: Enable/Disable data streaming. This is a global setting that turns on/off all forms of data streaming.

Parameters:

enabled = 1 to start streaming data to the client, 0 to stop streaming data to the client.

Example: #SET_STREAMING_ENABLED 1

Response: Data streaming disabled.



#GET_STREAMING_SENSOR_ENABLED

Syntax: #GET_STREAMING_SENSOR_ENABLED

Description: Determine if sensor data streaming is enabled/disabled.

Returns: 1 if enabled; 0 if disabled

Example: #GET_STREAMING_SENSOR_ENABLED

Response: 1

#SET_STREAMING_SENSOR_ENABLED

Syntax: #SET_STREAMING_SENSOR_ENABLED [enabled]

Description: Enable/Disable sensor data streaming.

Parameters:

enabled = 1 to start streaming sensor data to the client, 0 to stop streaming sensor data to the client.

Example: #SET_STREAMING_SENSOR_ENABLED 1

Response: Sensor streaming enabled.

#GET_STREAMING_SENSOR_IDS

Syntax: #GET_STREAMING_SENSOR_IDS

Description: Get the IDs of the Sensors to include in the streaming sensor data.

Returns: a space delimited list of sensor IDs.

Example: #GET_STREAMING_SENSOR_IDS

Response: OS3100_L1 OS4100_Main

#SET_STREAMING_SENSOR_IDS

Syntax: #SET_STREAMING_SENSOR_IDS [id_1] [id_2] ... [id_x]

Description: Set the IDs of the Sensors to include in the streaming data. At least one ID must be specified.

Parameters:

id_1 to id_x: A space delimited list of Sensor IDs whose values are to included in the sensor data stream to the client.

Example: #SET_STREAMING_SENSOR_IDS OS3100_L1 OS4100_Main

Response: 2 IDs added. 2 of the specified IDs currently exist.



#SET_STREAMING_SENSOR_ALL

Syntax: #SET_STREAMING_SENSOR_ALL

Description: Set ENLIGHT to stream all available Sensors. All does not mean all currently defined (i.e. it is not a snapshot), but all regardless of Sensors being added/removed. It is ALL!

Example: #SET_STREAMING_SENSOR_ALL

Response: All sensors set to stream.

#GET_STREAMING_SENSOR_DIVIDER

Syntax: #GET_STREAMING_SENSOR_DIVIDER

Description: Get the Sensor streaming data divider.

Returns: The current divider applied to sensor streaming data.

Example: #GET_STREAMING_SENSOR_DIVIDER

Response: 5

#SET_STREAMING_SENSOR_DIVIDER

Syntax: #SET_STREAMING_SENSOR_DIVIDER [divider]

Description: Set the Sensor streaming data divider (for example, a value of 5 will reduce the data rate such that every fifth data point is transmitted to the remote client.

Parameters:

Divider is an integer greater than 0

Example: #SET_STREAMING_SENSOR_DIVIDER 5

Response: The streaming sensor divider set to 5

#GET_STREAMING_EVENT_ENABLED

Syntax: #GET_STREAMING_EVENT_ENABLED

Description: Determine if ENLIGHT event data streaming is enabled/disabled.

Returns: 1 if enabled; 0 if disabled

Example: #GET_STREAMING_EVENT_ENABLED

Response: 1

#SET_STREAMING_EVENT_ENABLED

Syntax: #SET_STREAMING_EVENT_ENABLED [enabled]

Description: Enable/Disable ENLIGHT event data streaming.

Parameters:

enabled = 1 to start streaming event data to the client, 0 to stop streaming event data to the client.

Example: #SET_STREAMING_EVENT_ENABLED 1

Response: ENLIGHT event data streaming enabled.



#GET_DATA_ARCHIVE_DIRECTORIES

Syntax: #GET_DATA_ARCHIVE_DIRECTORIES [sub-path]

Description: Get the directories (separated by tab) for a specified sub-path (from the Local Data Storage Path defined in ENLIGHT on the Save Tab).

Parameters:

sub-path is the path below the primary data path defined on the ENLIGHT Save tab.

Returns: A **TAB** delimited list of sub-paths.

Example: #GET_DATA_ARCHIVE_DIRECTORIES \2010

Response: 2010\01 2010\02

#GET_DATA_ARCHIVE_FILE_NAMES

Syntax: #GET_DATA_ARCHIVE_FILE_NAMES [sub-path | date] [pattern]

Description: Get the file names (separated by tab) of the Data Archives for a specified date or sub-path (from the Local Data Storage Path defined in ENLIGHT on the Save Tab). An optional search pattern (i.e. peaks*) can be specified as the second argument.

Parameters:

sub-path is the path below the primary data path defined on the ENLIGHT Save tab.

date is the date on which the file was created.

pattern is an optional string used to filter the returned list of file paths.

Returns: A **TAB** delimited list of file names (with sub-path).

Example: #GET_DATA_ARCHIVE_FILE_NAMES \2010\01 sensor*

Response: 2010\01\Sensors2.20100130083000.txt

#GET_DATA_ARCHIVE_FILE

Syntax: #GET_DATA_ARCHIVE_FILE [file path]

Description: Get a specified archived data file. The file to retrieve is specified by its sub-path from the Local Data Storage Path (defined in ENLIGHT on the Save Tab) in addition to the file name (i.e. \2010\01\Sensors.20100108155600.txt).

Parameters:

file path is the combination of the file sub-path and name.

Returns: The contents of the Data Archive file.

Example: #GET_DATA_ARCHIVE_FILE 2010\01\Sensors2.20100130083000.txt

Response: byte array of file contents



#DELETE_DATA_ARCHIVE_NAME

Syntax: #DELETE_DATA_ARCHIVE_NAME

Description: Delete all existing data from the archive for a specified name.

Example: #DELETE_DATA_ARCHIVE_NAME ENLIGHT

Response: All data for 'ENLIGHT' has been deleted.

#DELETE_DATA_ARCHIVE_VALUES

Syntax: #DELETE_DATA_ARCHIVE_VALUES

Description: Delete the data from the archive between the start and stop date for a specified name.

Example: #DELETE_DATA_ARCHIVE_NAME Sensor1 1/1/2012_1:00_pm 2/1/2012_1:00_pm

Response: All data for 'Sensor1 1/1/2012_1:00_pm 2/1/2012_1:00_pm' has been deleted.

#SENSOR_ZERO_ALL

Syntax: #SENSOR_ZERO_ALL

Description: Zero all of the Sensors present on the system.

Example: #SENSOR_ZERO_ALL

Response: All Sensors have been zeroed.

#SENSOR_ZERO_GROUP

Syntax: #SENSOR_ZERO_GROUP [group]

Description: Zero all of the Sensors present in a specified group.

Parameters:

group is the name of the group whose Sensors will be zeroed.

Example: #SENSOR_ZERO_GROUP Group1

Response: 5 Sensor(s) have been zeroed.

#SENSOR_ZERO_IDS

Syntax: #SENSOR_ZERO_IDS [id_1] [id_2] ... [id_x]

Description: Zero all of the specified Sensor IDs.

Parameters:

id_1 to id_x: A space delimited list of Sensor IDs whose values are to be included in the sensor data stream to the client.

Example: #SENSOR_ZERO_IDS OS3100_L1 OS4100_Main

Response: 2 Sensor(s) have been zeroed.



#REBOOT

Syntax: #REBOOT

Description: Reboot the host pc on which ENLIGHT is executing.

Example: #REBOOT

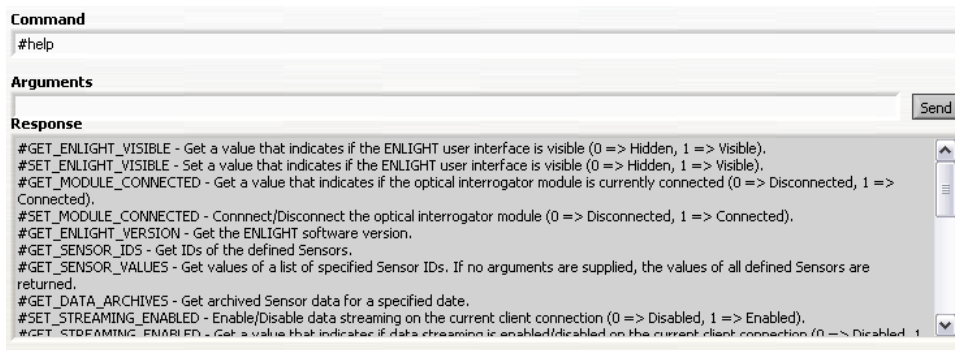
Response: Host rebooting.

11.4.2. ENLIGHT Remote Command Interface LabVIEW Example

11.4.2.1. Introduction

Included with the ENLIGHT distribution is a sample LabVIEW program that demonstrates some of the functionality of the ENLIGHT Remote Command Interface (RCI). This example features five functional tabs that introduce various features and commands of the RCI and is distributed as full source code for user review and/or modification. Please find this LabVIEW example installed with ENLIGHT in the following directory: Program Files/Micron Optics/ENLIGHT/Examples/LabVIEW

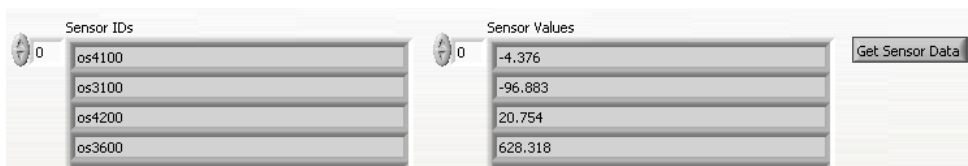
11.4.2.2. Command



The first functional tab demonstrates a basic command utility for the ENLIGHT RCI. The “Command” and “Parameter” controls are used to send a parameterized command to the RCI and display the response below.

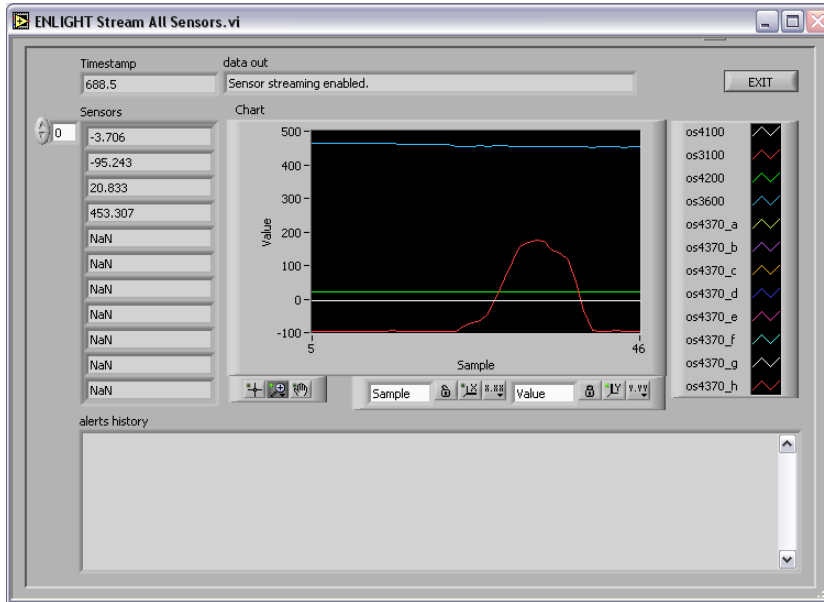
By default, the command window starts with the #HELP command, which returns a complete, updated list of supported commands to the user.

11.4.2.3. Get Sensors



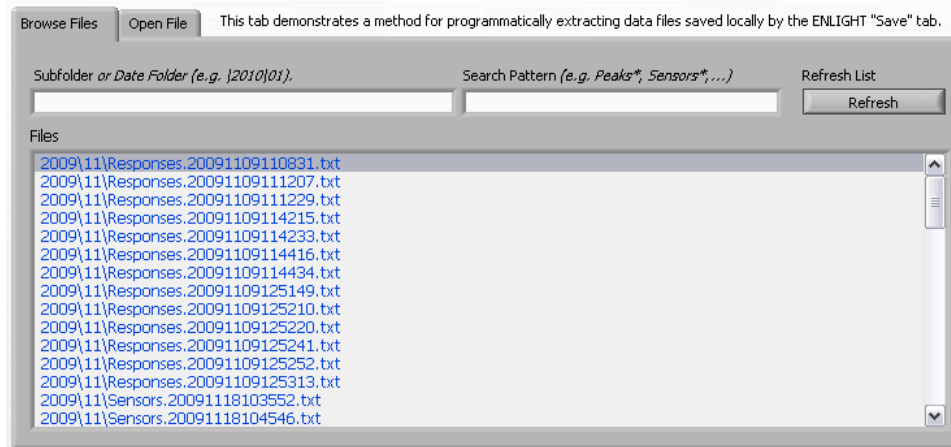
The next functional tab demonstrates how to extract a single, un-buffered collection of sensor IDs and sensor values. This tab exercises the #GET_SENSOR_ID and #GET_SENSOR_VALUES commands to return the latest data sets when the Get Sensor Data button is pressed.

11.4.2.4. Stream Sensors



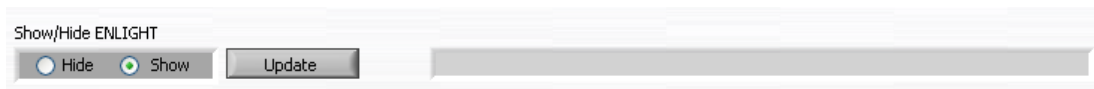
The third functional tab launches a sub-routine that exercises some of the streaming data commands to facilitate a continuous, buffered stream of calibrated sensor data from the ENLIGHT RCI to the LabVIEW client.

11.4.2.5. Get Data Archives



The fourth functional tab demonstrates a method for programmatically extracting data files saved locally by the ENLIGHT save tab. The `#GET_DATA_ARCHIVE_FILES` command is used to list the files present on the user-specified subfolder and the `#GET_DATA_ARCHIVE_FILE` command is used to extract and display the contents of the specified file.

11.4.2.6. Advanced



The final tab of the LabVIEW RCI example demonstrates how the front panel image of the ENLIGHT program can be programmatically shown or hidden using the `#SET_ENLIGHT_VISIBLE` command.

New in v1.5!**11.1. ENLIGHT Web Interface**

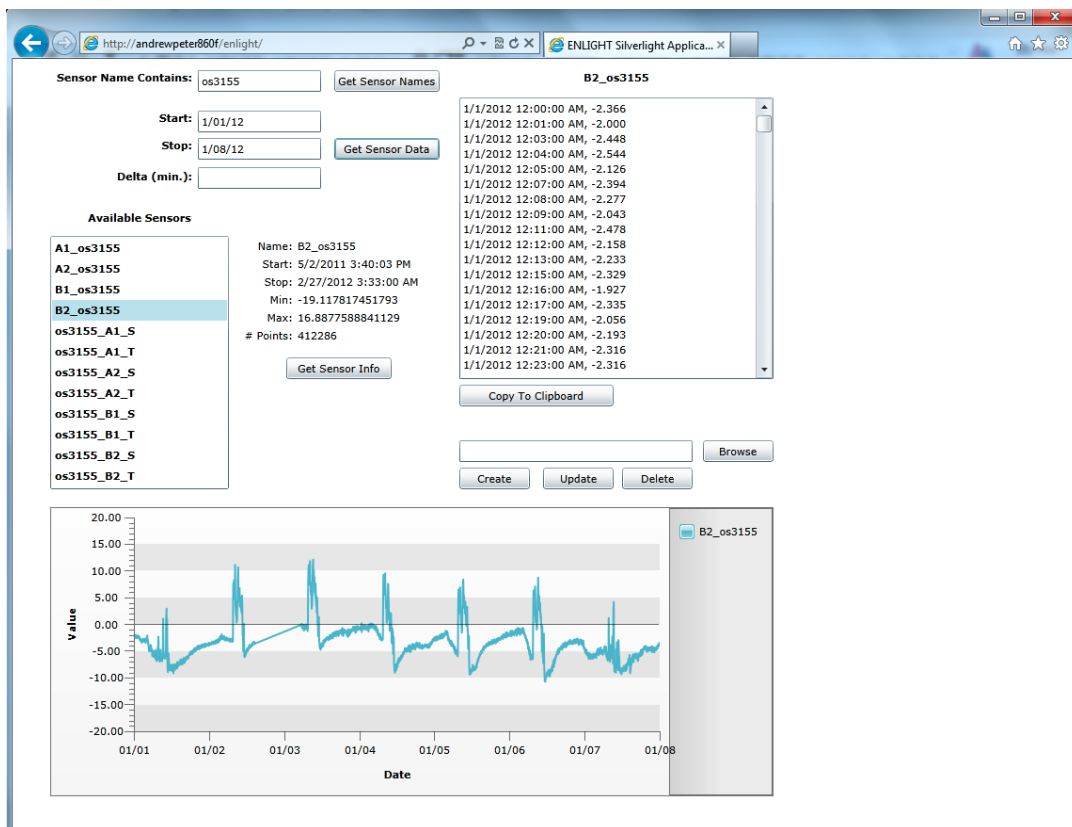
The ENLIGHT Web Interface provides a convenient means to retrieve and view FBG and sensor data that has been stored into the ENLIGHT Data Archive. The Web Interface enables configurable consumption and display of archived data through a custom user developed web application based upon a RESTful HTTP interface.

Micron Optics has implemented a simple technology demonstration of such an HTTP interface using Microsoft Silverlight™. The features and utility of this example are defined below. Developers interested in designing their own web and/or mobile applications to interface the ENLIGHT Data Archive through the ENLIGHT Web Interface are encouraged to contact Micron Optics for an SDK and development support.

ENLIGHT Web Interface Microsoft Silverlight™ Application

The ENLIGHT Web Interface Microsoft Silverlight™ Application enables the user to retrieve data stored in the ENLIGHT Data Archive using a web browser. Checking the Web Interface box under Applications Settings in the Settings Tab will activate the [http://\(processor name\)/ENLIGHT/ Web Interface](http://(processor name)/ENLIGHT/ Web Interface). The user can follow the hyperlink to access this Web application (Internet Explorer or Firefox recommended).

The screenshot below shows sensor information in the Web Interface, which is designed to operate in a similar method as the Archive Tab. Additional information on general operation of the Archive Tab can be found in section 7.3.6.2 of this manual.



12. APPENDICES

12.1. Using Legacy Notation for MOI Multiplexers

As mentioned earlier in this user guide, Micron Optics offers a suite of optical multiplexers which can be used to expand the number of measurement fibers for a sensor measurement system. In an effort to enhance clarity, Micron Optics has introduced a new primary naming and numbering convention for these multiplexed fibers. To maintain compatibility with legacy installation, MOI instrumentation continues to support the historical nomenclature and naming convention. This section will review the historical nomenclature, reference the new convention, and offer a means of correlating the two.

The maximum number of optical connects made to any MOI interrogator core is four. These parallel fiber connections are referred to as “channels”. Channels are designated by the integer labels “1”, “2”, “3”, and “4”. These channels can be connected to the input fibers of a multiplexer module. Each multiplexer input fiber can be optically switched sequentially into either two or four output fibers which would each, in turn be connected to sensor fibers.

Historically, MOI convention has been to refer to these multiplexed output fibers as “ports”, “multiplexer channels”, or just “channels”. The output fibers have been designated by the integer labels 1 – 8 or 1 – 16, with the following mapping to the SL interrogator core channels:

<u>Interrogator core Channel</u>	<u>Multiplexer output (mux channels/channels/ports)</u>
1	1, 5, 9, 13
2	2, 6, 10, 14
3	3, 7, 11, 15
4	4, 8, 12, 16

As some of the multiplexed channels share name labels with the interrogator core channels, there is a degree of ambiguity and some opportunity for confusion, particularly in circumstances where interrogator cores and multiplexer modules share a common enclosure, and the intermediate fiber connections are not visible to the user.

To clarify and simplify use of the combined interrogator core and multiplexer modules, MOI has introduced the sub-channel nomenclature based upon an “integer fraction” numbering convention, as outlined in an earlier section of this document.

The following table shows how to correlate the new “integer fraction” numbering convention with the legacy “all-integer” convention.

<u>New Convention (channel ->sub-channels)</u>	<u>Old Convention (channel ->ports)</u>
1 -> 1.1, 1.2, 1.3, 1.4	1 -> 1, 5, 9, 13
2 -> 2.1, 2.2, 2.3, 2.4	2 -> 2, 6, 10, 14
3 -> 3.1, 3.2, 3.3, 3.4	3 -> 3, 7, 11, 15
4 -> 4.1, 4.2, 4.3, 4.4	4 -> 4, 8, 12, 16

12.2. Configuring TCP-IP Settings for Interrogator Instrument Connections

12.2.1. Selecting Settings to Connect to the Module.

If you are not familiar with the various classes of IP addresses, seek help from your network administrator. They can assist in choosing a compatible IP address and netmask for both interrogator instrument and host computer. Though the following examples specifically reference configuration steps for the si255 module, the instruction apply equally well any Ethernet equipped HYPERION Instrument.

As was mentioned in Section 6.2 of this document, data transfer to and from the interrogator core can be facilitated through the Ethernet port of the instrument. The unit can be connected to an existing network through a hub or it can be connected directly to a host PC using a crossover Ethernet cable. Either arrangement requires that the instrument and the host be on the same logical network. In other words, the host PC and the instrument must be set for the same netmask and different, but compatible IP addresses. Using the default IP address and Netmask for the si255 instrument of 10.0.0.55 and 255.255.255.0, respectively, the following settings can be used to connect to the si255:



Case 1: Correct Settings

	Host PC	si255
Netmask	255.255.255.0	255.255.255.0
IP Address	10.0.0.121	10.0.0.55

Reason they work: The si255 and host PC share the same Netmask. The sm130 and host PC have different IP addresses of the same IP class.

Case 2: Incorrect Settings

	Host PC	si255
Netmask	255.255.255.0	255.255.255.0
IP Address	10.0.0.55	10.0.0.55

Reason they don't work: The si155 and host PC have the same IP address. The si155 needs a unique address on the network. If any other PC connected to the network is assigned to the same IP address, communication will fail.

Case 3: Incorrect Settings

	Host PC	si255
Netmask	255.255.255.0	255.255.255.0
IP Address	192.168.1.1	10.0.0.55

Reason they don't work: The si155 and host PC are not on the same subnet, as dictated by the IP address. See your network administrator to select compatible values.

Case 4: Incorrect Settings

	Host PC	si255
Netmask	255.0.0.0	255.255.255.0
IP Address	10.0.0.121	10.0.0.55

Reason they don't work: The si155 and host PC do not have identical Netmask settings.

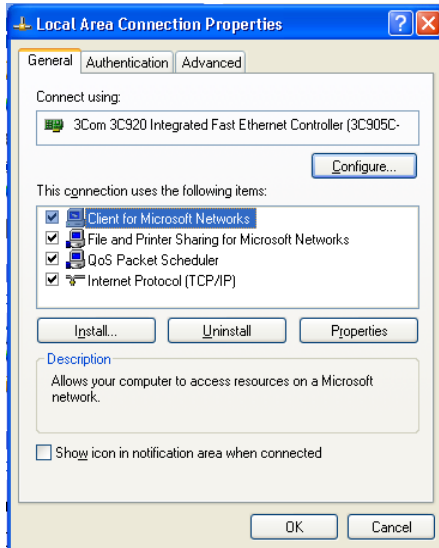
The NetworkConfigExample.exe program can be used to change the Netmask and IP address settings of the interrogator instrument. See details in Section 7.1.2 of this document.

Should there be any uncertainty regarding the settings of an interrogator instrument, the LCD screen on the front panel of to the instrument will display the current network settings.

12.2.1.1. Setting the Netmask and IP Address on the Host PC.

12.2.1.1.1 Windows XP

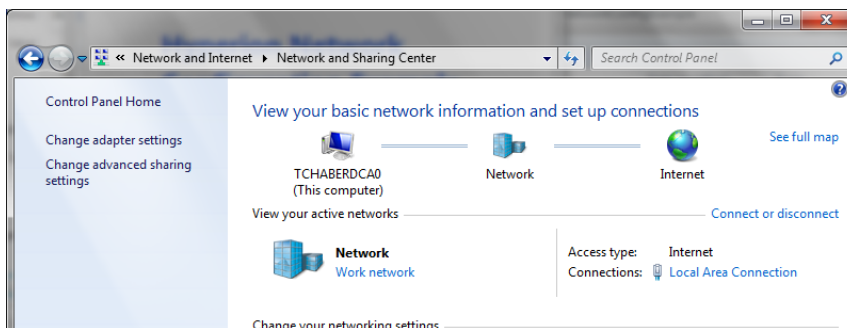
On a Windows XP machine, click and the Start Bar and select "Control Panel." Click on the icon entitled "Network Connections" to open. When open, highlight the "Local Area Network" selection, right click, and choose "Properties". The following window will open.



Highlight the “TCP/IP” entry and click “Properties”. You will then have access to change the IP address and subnet mask. Click “OK” to save settings.

12.2.1.1.2 Windows 7/8

On a Windows 7 machine, click on the Start Bar and select “Control Panel”, then choose “Network and Internet”, and then “Network and Sharing Center”.



Locate and click on the “Local Area Network” link in the window.