



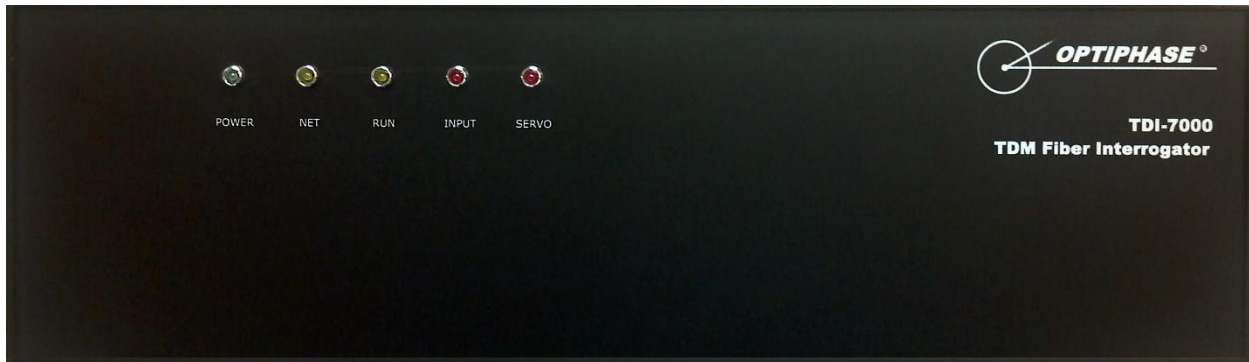
TDI-7000 TDM Fiber Interrogator

User Manual

Manual Part Number: 0035-3920-TDI_7000_User
Revision: E
Date: September 14, 2011

TDI-7000

TDM Fiber Interrogator



Notice

This publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of Optiphase, Inc. The information contained in this document is subject to change without notice.

Optiphase and the Phase Interlock Circle Device logo design is a registered trademark of Optiphase, Inc. All other trademarks, products and company names mentioned herein are the property of their respective owners.



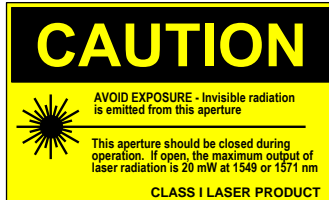
Optiphase, Inc.
7652 Haskell Avenue
Van Nuys, CA 91406
Phone: 818-782-0997
Fax: 818-782-0999

Copyright © 2011 Optiphase, Inc. All Rights Reserved.

SAFETY

The following general safety precautions must be observed during all phases of operation of this product. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture and intended use of the product. Optiphase, Inc. assumes no liability for the customer's failure to comply with these requirements.

DO NOT substitute parts or modify equipment: Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification to the product. Return the product to Optiphase for service and repair to ensure that safety features are maintained.



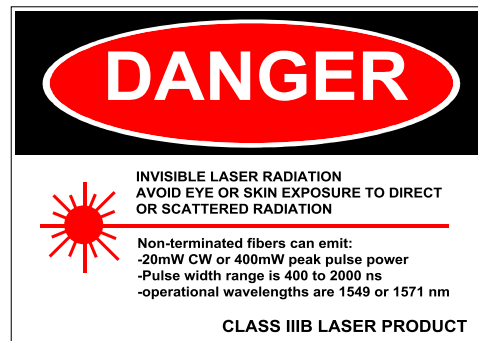
This product complies with 21CFR1010.2 as a CLASS I device.

SERVICE AND TECHNICAL SUPPORT

For questions regarding the use and operation of Optiphase products, or to report product or documentation issues contact technical support.

Optiphase must perform any adjustment, maintenance, or repair of this product. Only Authorized Service and Repair technicians may open the product chassis.

Contact Optiphase technical support via email: support@optiphase.com



USER MANUAL UPDATES

This user manual may contain descriptions and screen captures that are different than the control software that is delivered with a particular TDI-7000. They are representative of the functions of the instrument for a particular release. For current information, check the Optiphase web site at www.optiphase.com or contact support@optiphase.com.

USE AND EXPORT

User acknowledges and agrees to maintain, control and bear full responsibility for the use and distribution of the Product in accordance with all U.S. Export Administration Laws and Regulations and all exporting countries' export laws and regulations. Further, User also certifies that the goods are for commercial use and that the goods will not be sold or used with the intention of installation for military purposes. User agrees that the Product will not be sold or used with the intention of being used for purposes associated with chemical, biological or nuclear weapons or missiles capable of delivering such weapons, nor will they be re-sold if User knows or suspects that they are intended or likely to be used for such purposes.

REFERENCE DOCUMENTS

0035-3920-TDI_7000_User (Rev. E)

This manual

0035-3925-DsuUserInterface (Rev. D)

Gives an example of one implementation of an interface to the TDI-7000

0035-1269-PdrGain (Rev. A)

Describes the Optical Gain Level servo mechanism using the PDR

0035-1270-TdmMaskSwitch (Rev A)

Describes the Polarization Diversity Receiver mask switching approach

TDI7000.tconfig

Default filename for the configuration text file used by the Host software

U.S. Patent No 6778720

Dual Slope Fiber Optic Array Interrogator

08/17/2004

TABLE OF CONTENTS

INTRODUCTION.....	9
HARDWARE	11
Block Diagram - External Power Supply	12
Block Diagram - Internal Power Supply	13
Source	14
Pulser	14
Compensator.....	14
PZT Driver	14
EDFA.....	14
Mux Optics	14
PDR.....	15
DSU	17
Connections to the TDI-7000	19
Multiple DSU Systems	20
Front Panel LEDs.....	21
CONTROL SOFTWARE	23
Installing the software	23
Starting the application.....	24
Main Window.....	25
Acquisition Status.....	26
Operation Status Reporting.....	28
Status Bar.....	28
Status Log	28
System Setup	29
Configuration Files	30
Module Maintenance.....	31
Searching for modules on the network	32
Configuring a module	33
Laser Temperature.....	34
EDFA.....	34
System Parameters.....	35
Mask Parameters	37
Array Parameters	38
Pulse Parameters.....	40
Modulation Parameters	40
Sensor Sampling.....	43

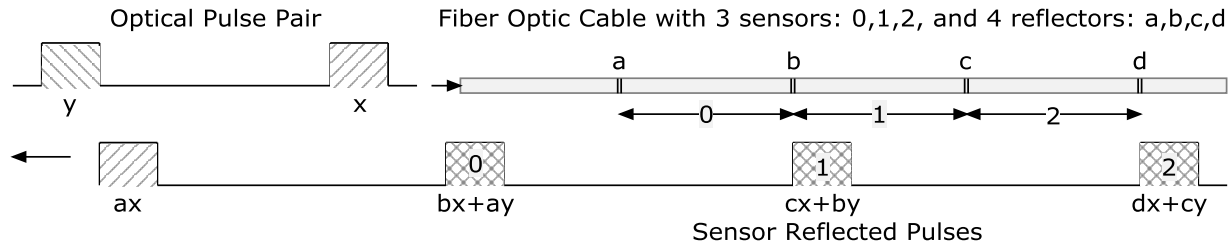
Input Parameters.....	44
Trigger Parameters	46
Analog Output	47
Switch Timing.....	48
ADC Data Viewing and Acquisition	49
Modulation Amplitude Servo Utility	52
Sine Table Phase Adjust Utility.....	53
Input Level Utility.....	55
Sine / Cosine Data Viewing and Acquisition	56
Phase Data Acquisition	57
Tools menu.....	58
Setup	58
Status File Viewer	58
Fixed Data Acquisition	59
Continuous Data Acquisition.....	60
Status Information	61
Continuous Triggered Data Acquisition	62
OPERATION	63
Requirements.....	63
Laser Wavelength	64
Modulation and Sample Frequencies	65
Optical Pulse Width.....	66
Receiver Saturation.....	67
ADC Samples.....	70
Modulation Amplitude.....	73
Modulation Phase	74
Sensors	75
Setup Procedure	76
Timing Summary	77
MAINTENANCE	78
Properties	78
Reset Module	79
Update Firmware.....	79
Modify Network Settings	80
Blink RUN LED.....	81
EDFA Console	81
Operations.....	82
Display FPGA Control Registers.....	82

Display FPGA Status Registers	83
Display Timing Table.....	83
Reset Instrument	84
DIGITAL DATA.....	85
Introduction.....	85
Optiphase.Settings File Format	86
Settings	86
Arrays	86
Groups.....	86
Comments.....	86
Example	86
Configuration File Specification.....	87
Groups.....	87
Binary Configuration File Layout	88
Text Configuration File Layout	88
Timing Table Files.....	89
Status File Specification.....	89
Groups.....	89
DSUAcquisitionInfo Group Elements	89
SystemInfo Group Elements	92
Block Status Group Elements	94
Binary Status File Layout	96
Text Status File Layout	96
Phase Sensor Data Formats and Calculations	97
Phase32: Normal Sensor Mode	97
Phase64: Extended Sensor Mode	97
PhaseFloat	97
Conversions	98
Rollovers	99
External Fringe Counter	100
Trigger Detection.....	101
Block Time.....	102
Frame Time.....	102
Trigger Time.....	102
Sensor Time.....	104
Sine/Cosine Data Format.....	106
ADC Data Format.....	106

ELECTRICAL/OPTICAL INTERFACE	107
SPECIFICATIONS	110

INTRODUCTION

The TDI-7000 TDM Fiber Interrogator instrument is designed to acquire readings from an array of sensors embedded in a fiber optic cable. A pair of optical pulses output from the TDI-7000 is launched into the cable and each sensor, defined as the fiber between two partial reflectors, returns an optical pulse in a unique time slot which carries the sensor reading.

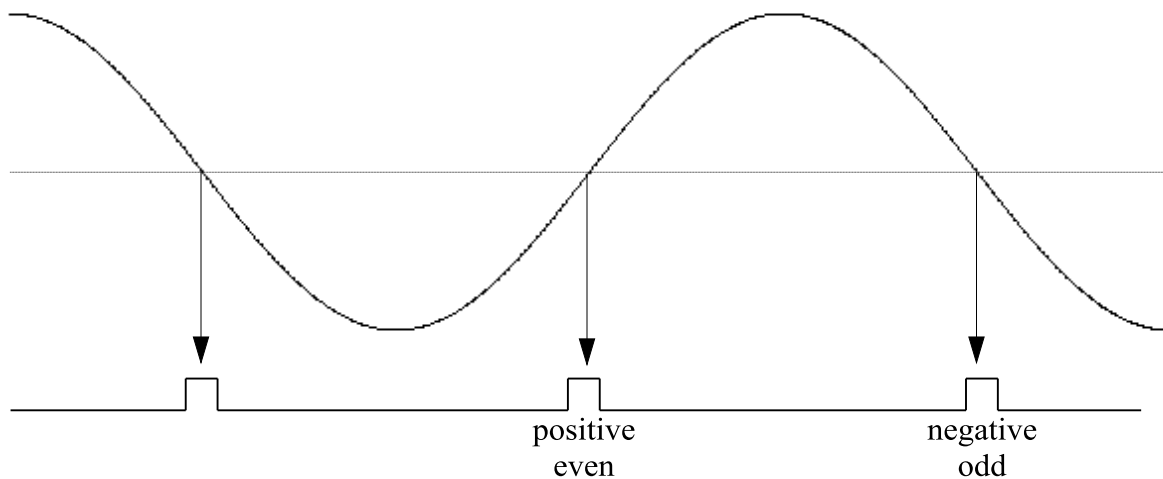


This is so because a sensor has a partial reflector at its start which reflects the trailing input pulse and a reflector at its termination that reflects the leading pulse so that it arrives back at the start to overlap the reflected trailing pulse. The superposition of the two pulses forms an interferometer, where only the leading pulse travels through the sensor and therefore picks up the sensor's optical path changes induced by the environment of the sensor. The trailing pulse becomes the reference against which the changes in the leading pulse are measured.

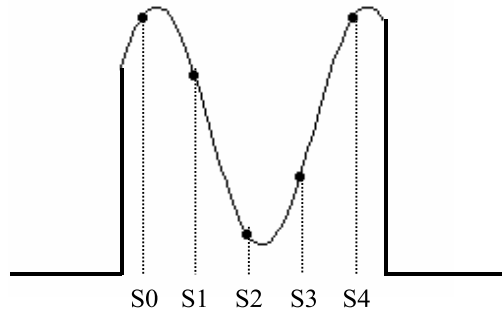
A reading of a sensor is expressed in units of the source wavelength which is about 1550 nanometers. This is converted by the TDI-7000 into an integer where 20 bits represent one wavelength of change.

The two input optical pulses are formed from a common source pulse which is split into a long path and a short path and then recombined. A known optical path change is applied to the split pulses so that the relative optical path change between the two is linearly increasing. During the time period of the pulses, this change covers a full wavelength and provides a known modulation which allows the determination of the sensor measurement. This modulation is induced by using a fiber stretcher in the split paths. A sine voltage is applied to the fiber stretcher and the linear portion of the sine at the zero crossings is used to induce the modulation tone.

This method of modulation results in one or two interrogation pulses, the first modulated by the positive slope and called the even pulse, and the second modulated by the negative slope and called the odd pulse, as shown in the following diagram.



The optical pulse returned by a sensor carries the sensor reading in its amplitude which is the result of the interference of the leading and trailing pulse that was input to the sensor cable. The linear modulation converts this amplitude into a sine variation over the time period of the pulse. The phase of this sine variation encodes the sensor reading. The phase is extracted by sampling the sensor return pulse five times where the time difference between each sample is the time it takes the modulation to change the relative phase difference between the leading and trailing pulse by one quarter wavelength.



The time period of the optical pulse must be greater than the time it takes the modulation to change the relative phase difference between the leading pulse and trailing pulse by one wavelength plus the pulse rise time.

Each sensor optical pulse return is the superposition of the leading and trailing input pulses to the sensor cable. Light has the property called polarization where the light field has direction in space. Since the polarization of the leading optical pulse can be different than of the trailing optical pulse, a condition called optical fading is possible when their directions of polarization are 90 degrees apart. This condition is eliminated by splitting the sensor return optical pulse into three equal pulses and passing each through its own polarizer oriented 120 degrees from the other two. Three optical to electrical detectors are used along with three analog to digital converters to take five samples of each of the three polarization states resulting in fifteen samples per sensor return. This polarization diversity receiver scheme insures that at least one set of five samples provides a measurement of the sensor signal that has not faded.

The sample condition where the time difference between each sample is the time it takes the modulation to change the phase between the leading and trailing pulse by one quarter wavelength is assured by adjusting the amplitude of the modulation sine such that the zero crossing slope is correct. The fifteen samples from a selected sensor are used to compute an error from the correct amplitude. This error is summed and used to correct the amplitude in an integral servo with an adjustable time constant.

A measure of the optical power available to the array sensors is found by summing a sample of the non-interfering pulse return ax from each of the three polarization masks and comparing the sum to a designated level. An integral servo maintains this optical power level with a specified time constant.

The TDI-7000 instrument and software provides the following functions to accomplish the sensor measurements:

- Synthesize a sine modulation tone at a programmable frequency
- Trigger an optical pulse at the positive and negative slope of the modulation zero crossing
- Sample each sensor return pulse five times on each of three detectors
- Compute a modulation amplitude error and servo the amplitude to the correct value
- Compute the sensor measurement from the 15 sensor samples
- Transmit the sensor measurements to a Host PC
- Accept configuration data from a Host PC
- Provide status information to a Host PC
- Provide setup and test modes under PC control
- Be able to reconfigure firmware for the FPGA and DSP through the Host PC interface

HARDWARE

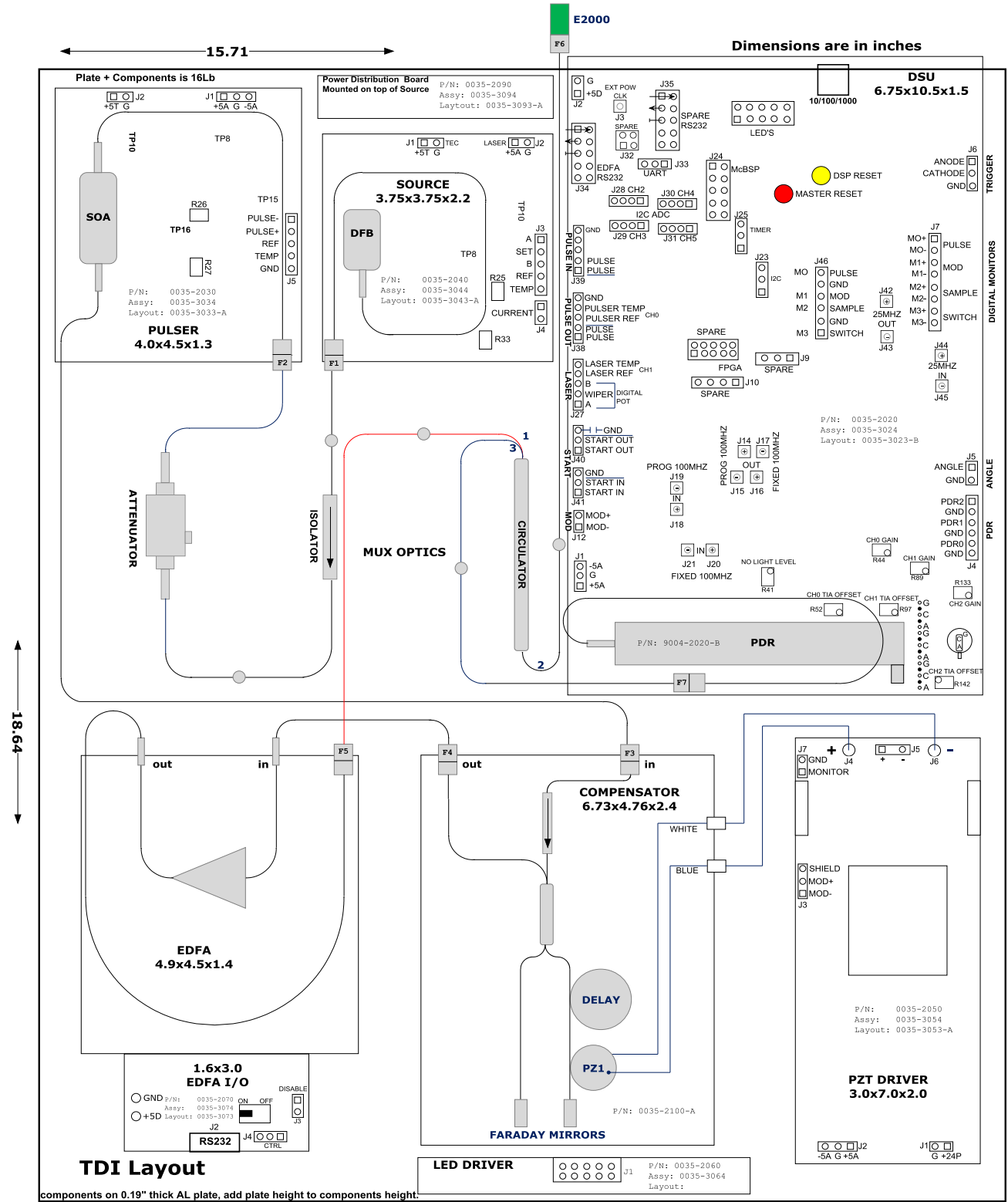
The physical layout of the TDI-7000 is shown in the block diagram on the next page.

The following table lists the major sections:

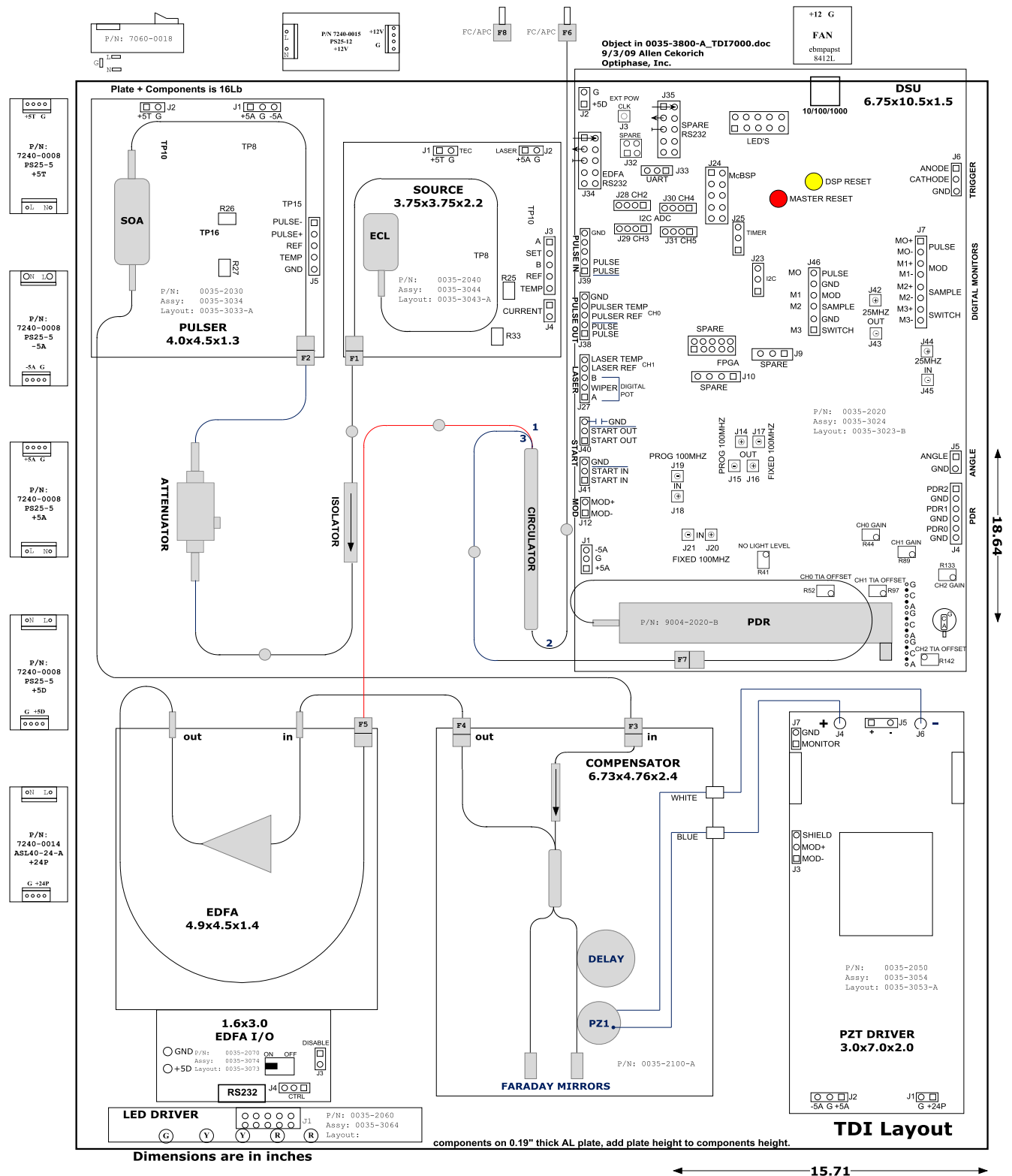
Section	Description
SOURCE	Laser light source
PULSER	Optical switch to form pulse
COMPENSATOR	Generates a delayed pulse with modulation
PZT DRIVER	Voltage amplifier for driving modulator in compensator
EDFA	Optical amplifier
MUX OPTICS	Interconnection optics
PDR	Polarization diversity receiver
DSU	Digital signal processing and control unit

These sections are described on the following pages.

Block Diagram - External Power Supply



Block Diagram - Internal Power Supply



Source

The light for the TDI-7000 is generated in the source section which is shown as an External Feedback Laser (ECL) although other options are possible including an external customer supplied light source. The ECL mounts on PCB 0035-2040 which provides the current and temperature control for the ECL. The temperature of the ECL is set using connector J3 which connects to a programmable digital pot J27 on the DSU. This allows turn on hysteresis and operating temperature of the ECL to be set. The light from the optical source exits the optical connector F1.

Pulser

The light from the optical source enters an isolator to prevent reflections from returning to the source and then passes through a manually adjusted attenuator to match the light level required by the pulser assembly at the optical connector F2. The pulser PCB 0035-2030 is a fast optical shutter that uses a Semiconductor Optical Amplifier that turns the continuous source light into the On/Off pulse required to interrogate the sensor array. The pulse duration is set by the electrical signal on connector J5 which comes from the DSU at connector J38 and is programmed by the user to work with the sensor length used in the array.

Compensator

The optical pulse generated by the Pulser enters the compensator on the optical connector F3. The compensator input is protected by an optical isolator to prevent reflections from going back to the pulser and then the light is split into a short path and a long path by a 50/50 optical coupler. The short path is terminated with a Faraday Mirror to return the light to the coupler. The long path includes a delay section and the Piezo-Electric modulator (PZ1) which induces a linear phase modulation during the duration of the optical pulse by linearly stretching the optical path as the pulse passes through the compensator.

The short and long paths of the compensator are constructed so that their length difference causes a time delay between the pulse reflected in the short arm and the pulse reflected in the long arm. This time delay is set to be equal to the time it takes one pulse to travel twice the length of a sensor. This means that the compensator must be matched to the length of a sensor. **Changing the sensor length requires that the compensator in the TDI-7000 be changed as well.** The two pulses from the compensator exit from the optical connector F4.

PZT Driver

The linear phase modulation induced on the optical pulse traversing the long path of the compensator is generated by the PZ1 transducer which changes a voltage into a stretching of the optical fiber wound on it. The voltage source comes from the DSU on connector J12 and goes to the PZT Driver connector J3. The PZT Driver PCB 0035-2050 is a differential amplifier that boosts the voltage level and supplies the current required to drive the PZT which is essentially a capacitor.

EDFA

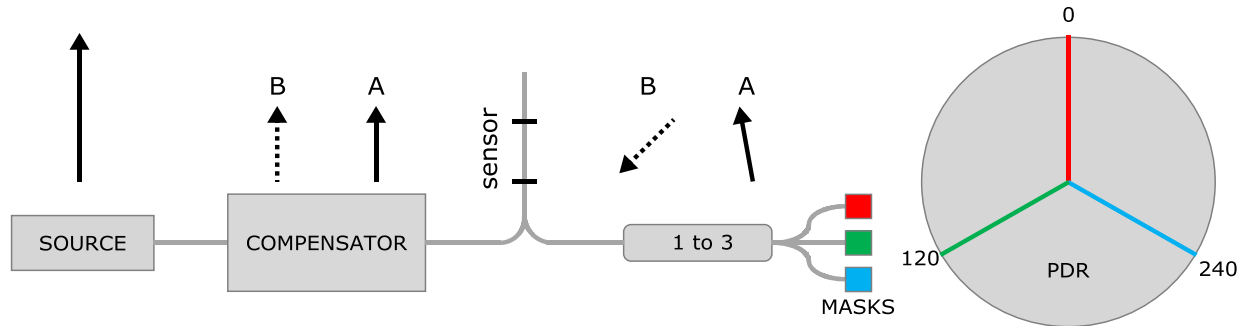
The input of the Erbium Doped Fiber Amplifier is connected to the compensator optical output at F4. The purpose of the EDFA is to boost the light level of the two compensator pulses. The amplification level of the EDFA is set through the RS232 interface on connector J2 which is connected to J34 on the DSU allowing operator programming of the EDFA gain. The EDFA output goes to the optical connector F5.

Mux Optics

The multiplexer optics contains an isolator, a manual optical attenuator and a circulator. The isolator protects the optical source from back reflections and connects to the manual attenuator that sets the light level at the pulse input. The circulator routes the two optical pulses from the EDFA output F5 to the sensor array connector F6 as the light passes from input port 1 to the output port 2 of the circulator. The combined optical sensor pulses returned from the sensor array at F6 are now routed to the Polarization Diversity Receiver on the DSU from port 2 to port 3 of the circulator.

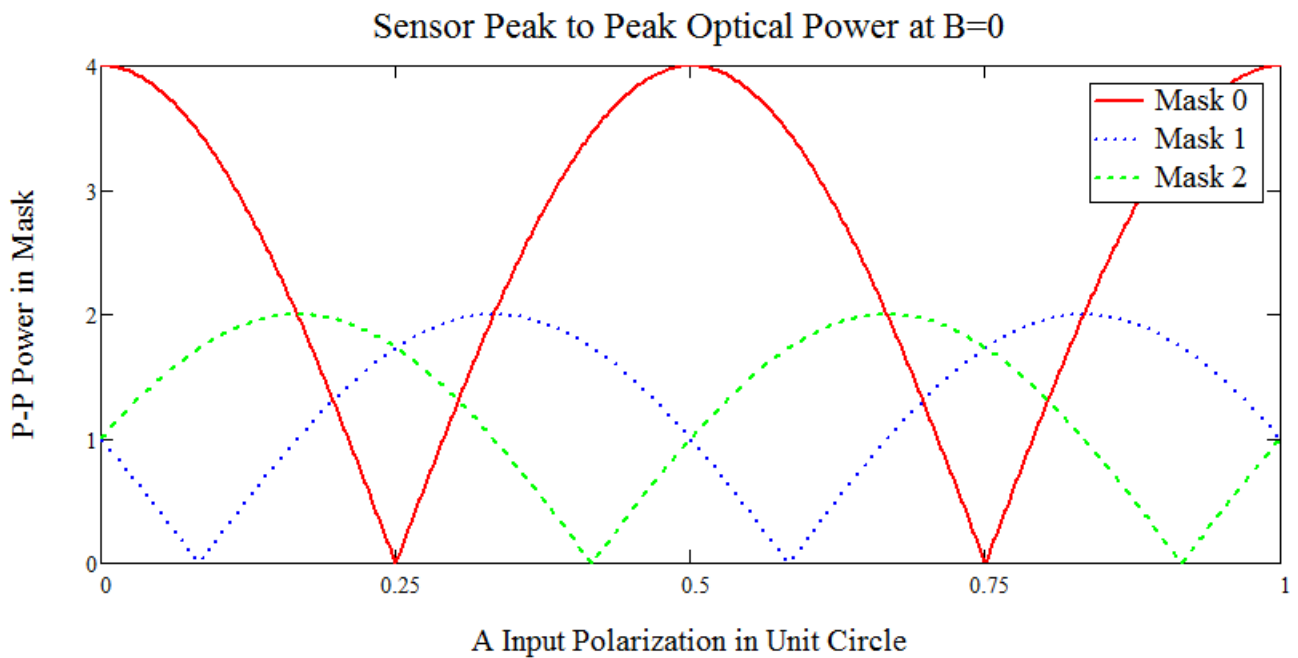
PDR

The light in the dual slope system starts out polarized from the source. It is then directed to the compensator where a Michelson is used to split the light and delay one optical pulse from the initial pulse resulting in two optical pulses A and B with the same polarization. However, when the two pulses are sent to a sensor, the initial pulse that passes through the sensor and returns to meet the second pulse at the input to the sensor has random polarization relative to the light in the second pulse. This situation is diagrammed below.



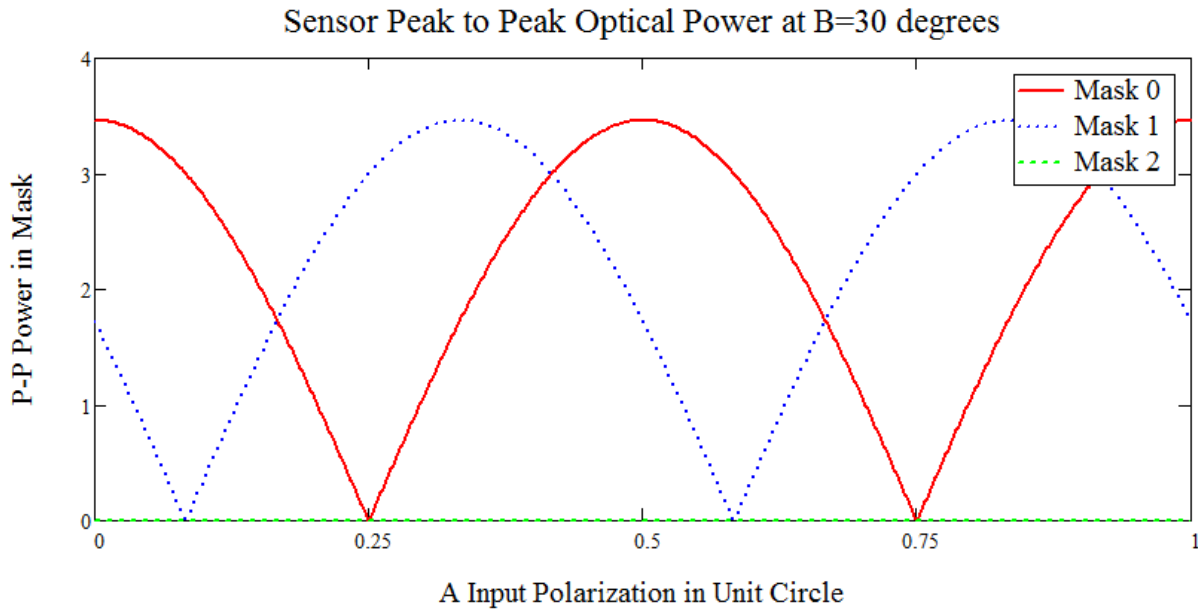
The two pulses A and B are shown at the input to the polarization masks (red, green and blue) that are oriented at the 120 degree divisions on the unit circle. The PDR is designed so that the angle between the A polarization and the B polarization is the same at each mask. Therefore, the polarization component of A that is aligned with the red mask and the polarization component of B that is aligned with the red mask pass through the red mask and interfere at the red mask optical receiver. The same is true for the green and blue receivers.

The polarization of the A optical pulse can have any angle on the unit circle as can the B optical pulse. The example below shows the linear polarization of A going from 0 to 360 degrees while the linear B polarization is fixed parallel to the red mask.



The optical power in each pulse is the same with a magnitude of one. The interference quadruples the peak to peak output power but the other two masks peak at half the maximum power.

For linear polarization, the worst case is when the B pulse is 30 degrees away from the red mask giving a maximum peak to peak power of 3.5 while the green mask is always zero since the B pulse is orthogonal to it and produces only the constant offset light from the A pulse.



The polarization states of A and B can each occur at any location on the Poincare sphere. The PDR design still insures that sufficient visibility remains.

DSU

The Dual Slope Unit PCB 0035-2020 is the control center of the TDI-7000. A functional diagram of the DSU is shown on the next page. The Polarization Diversity Receiver mounts on the DSU and converts the light from the sensors into voltage levels that are sampled by the 100MHz analog to digital converters. The receivers also have dynamic gain control through variable gain amplifiers.

Each sensor return pulse is sampled five times for each of the three receivers in the PDR. The fifteen samples are processed by the Spartan gate array to produce three sine/cosine pairs, one pair for each receiver. This data is then bundled into three 32-bit words and transferred to the Digital Signal Processor. This forms the raw data for the sensor signal computation which is done in the DSP.

The gate array also controls the modulation. The sine modulation table resides in the gate array and each entry is clocked out to the digital to analog converter in sequence once every 40ns, returning to the start of the table when a modulation cycle is complete. The sensor sample data is also used to measure the amplitude of the modulation and check whether each sensor sample is separated by $\pi/2$ radians which respect to the phase of the sensor. A correction error is generated which is used to update a modulation amplitude servo multiplier M.

The modulation sine table is part of a larger timing table that is sequenced at a 25MHz rate. The timing table initiates an optical pulse through the control bit fields: pulse enable and pulse phase. These allow the start of pulse to be controlled to 10ns. A one in the pulse enable bit field starts logic that looks at the pulse phase bit field and initiates the pulse after zero, one, two or three 10ns clocks have passed depending on whether the pulse phase field contains a value of 0, 1, 2 or 3. The sample enable and sample phase bit fields in the timing table work the same way to control the latching of ADC samples. The gain field is sent to a 10-bit digital to analog converter which sets the level of the variable gain amplifier providing dynamic gain control.

The timing table also generates timing information with respect to a modulation cycle including the first positive slope sample and the even or odd slope pulse. There is also a general purpose switch bit that can be used to synchronize an external device to the operation of the timing table.

The gate array also provides an ADC Data read function that passes the raw samples of one receiver at a 50 MHz rate to the DSP. This ADC Data function is used to setup the TDI-7000.

There are two types of DSUs, MAIN and AUX. An AUX is synced to a MAIN DSU, and typically exists in multiple DSU systems.

Note: In the TDI-7000 software, a DSU is referred to as a "module".

The digital signal processor provides the interfaces to the TDI-7000 listed below.

10/100/1000 Ethernet

The Host PC communicates with the TDI-7000 over 10/100/1000 (Gigabit) Ethernet using the supplied Graphical User Interface (GUI), or the user;s own software written with the Optiphase supplied Application Program Interface (API) library.

RS232

An RS232 port sends commands to the EDFA and receives status from the EDFA. There is also a spare RS232 port for custom use.

LED's

Four LED driver lines are controlled by the DSP for enabling front panel indicators.

Temperature Sensor

A temperature sensor monitors the temperature of the DSP chip.

ADC

Six 10-bit differential analog to digital converter channels are provided. One is dedicated to the laser temperature and one to the pulser temperature.

Modulation Reference

The DSP can set the reference voltage of the modulation sine digital to analog converter using a 10 bit DAC.

Laser Temperature

A digital potentiometer is programmed by the DSP to fine tune the temperature of the laser thereby adjusting its wavelength.

System Clock

There is a programmable clock chip on the DSU that is programmable by the DSP. However, its function depends on configuration settings that are hard wired on the PCB.

FLASH Memory

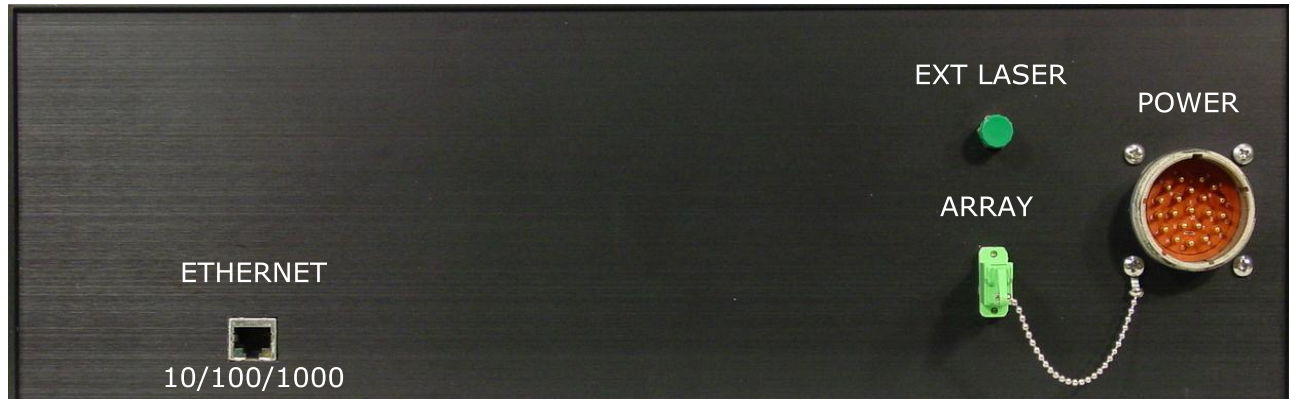
The FLASH memory that holds the image of the gate array can be updated by the DSP as can the boot FLASH of the DSP itself. This provides field upgrades.

Analog Output

The DSP has one channel of buffered/synchronous 24-bit analog output that can be used.

Connections to the TDI-7000

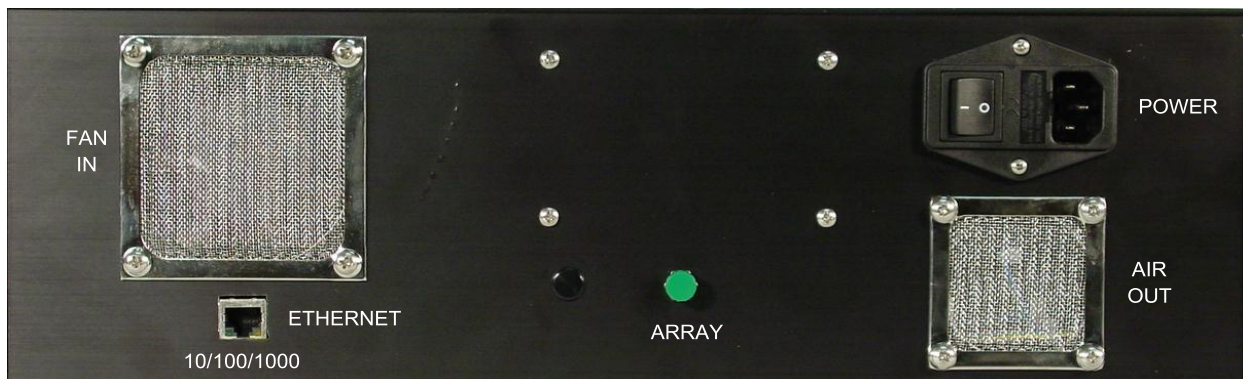
The connections to the TDI-7000 are on the back panel shown below for the external power supply option.



External Power Supply Rear Panel

The sensor array is connected to the ARRAY optical connector which is a Diamond E2000 type connector. The power supply plugs into the POWER connector. The Host PC connects to the Ethernet port, either directly or through a switch.¹ The EXT LASER optical connector is generally not used except when the TDI-7000 is configured internally for a customer's external laser source. Power is applied by turning on the power supply.

The connections to the TDI-7000 for the internal power supply are shown on the rear panel below.



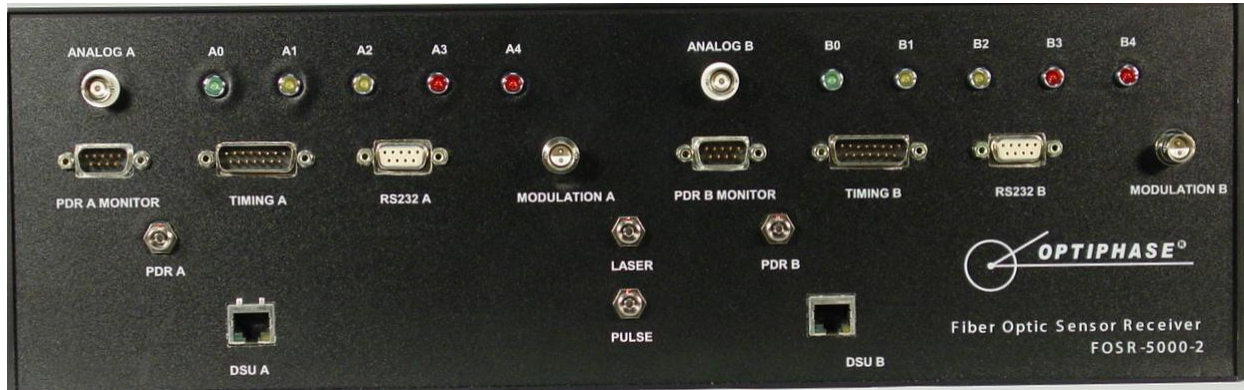
Internal Power Supply Rear Panel

The sensor array is connected to the ARRAY optical connector which is a FC/APC type connector. The power cord connected to the integral fused switch assembly. The Host PC connects to the Ethernet port, either directly or through a switch.¹

¹ CAT-5e or greater cabling and Gigabit Ethernet switches are recommended for optimal performance.

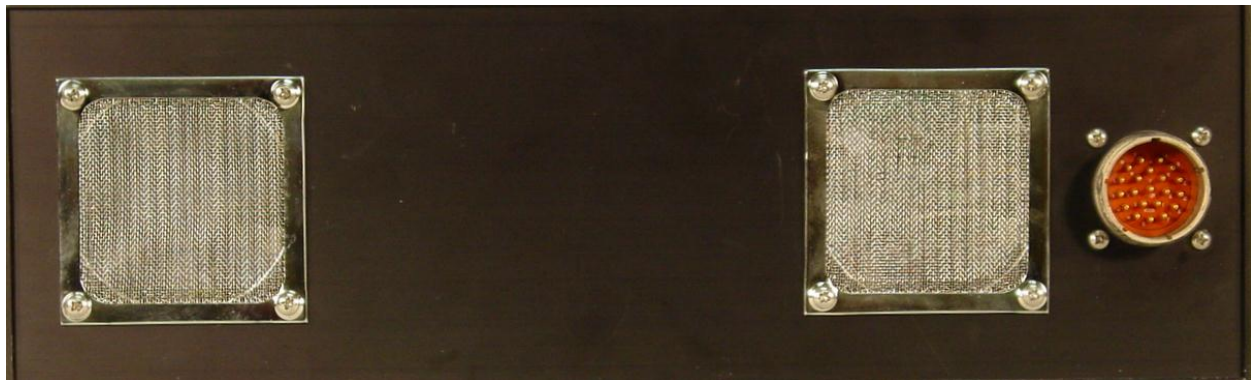
Multiple DSU Systems

Optiphase also supports systems with multiple Dual Slope Units (modules) installed. These systems are custom built. An example of a Dual DSU system is shown below.



Dual DSU Interrogator Front Panel

This system has an external compensator, external EDFA and external sensor mux optics and test connectors on the front panel as well as an external power supply input shown on the rear panel.



Dual DSU Interrogator Rear Panel

The TDI-7000 Control application supports operating multiple DSUs (modules) simultaneously. One may link together any combination of MAIN and AUX DSUs. An AUX must exist in the same configuration as its corresponding MAIN DSU.

Front Panel LEDs

There are five LEDs on the system's front panel:



These LEDs give visual indicators regarding the status of the system. The software provides more detailed status information. The following describes the functionality of the five LEDs. Note: BlockTime is a configuration setting, which is typically 1.0 s.

POWER All LEDs are dark when the power is off.

All are lighted when the AC power is turned on, and then all go off except the power LED when the system starts the boot process.

NET Dark during system boot

Rapidly blinking when system has booted but the Ethernet connection has not been made or when DHCP is enabled and a DHCP server is not available.

Slowly blinking when the Ethernet connection is active and the system is ready to accept a connection from the control software.

Lighted when a connection to the client is established

RUN Dark when system is not active

Blinking during demodulation: $\text{Blink period} = \text{BlockTime (s)}$
 $\text{Duty cycle} = \frac{\text{DemodProcessingTime}}{\text{BlockTime}}$

INPUT While not connected:

Dark when the EDFA is off

Blinking slowly when the EDFA is on

While connected and demodulating:

Lighted when there is an error associated with the optical input during demodulation:

ADC Error Flags

● ADC0 Underflow	● ADC0 Overflow
● ADC1 Underflow	● ADC1 Overflow
● ADC2 Underflow	● ADC2 Overflow

SERVO

While not connected: OFF

While connected and demodulating:

Dark when all servos disabled:

Input servo gain is zero

Modulation servo gain is zero

Mask switching mode is once or initial

Blinking rapidly when any servo is on but not all

Blinking slowly when all servos are on:

Input servo gain is not zero

Modulation servo gain is not zero

Mask switching mode is normal

Lighted when any servo has an error during demodulation:

Modulation Servo Error Flags

-  Neg Overflow (Servo)
-  Pos Overflow (Servo)
-  Neg Overflow (Accum)
-  Pos Overflow (Accum)
-  Servo Multiplier Negative
-  Servo Multiplier Overflow

Receiver Gain Servo Error Flags

-  Neg Overflow (Servo)
-  Pos Overflow (Servo)
-  Neg Overflow (Accum)
-  Pos Overflow (Accum)
-  Rcvr Gain DAC Negative
-  Rcvr Gain DAC Overflow

LED Blink Rates

Mode	Rapid LED period (s)	Slow LED period (s)
Not connected	0.5	2
Connected - demodulating	BlockTime	2 * BlockTime
Connected - other operation	N/A	N/A

CONTROL SOFTWARE

The Optiphase TDI-7000 TDM Fiber Interrogator requires TDI-7000 Control Software to operate. This software is provided by Optiphase.

Installing the software

The software installation package for the TDI-7000 contains the software and firmware necessary to operate the device.

The filename of the software installation package is structured as follows:

TDI-7000-X.YY-setupZZ.exe

Where "X" is the major version number, "YY" is the minor version number, and ZZ is the platform for the software (Either 32 for 32-bit operating systems, or 64 for 64-bit operating systems).

For example, the filename for the software installation package version 1.50 on a 32-bit operating system would be structured as follows:

TDI-7000-1.50-setup32.exe

Perform the following steps to install the software:

1. Acquire the latest software installation package from Optiphase. Ensure the package platform corresponds to your operating system platform.
2. Ensure any previous versions of the software are uninstalled.
3. Run the installation package and install the software.
4. Upgrade the DSP and FPGA firmware in your module(s) if necessary. You may check what versions of firmware are currently programmed using the **Read System Info** Command. The latest firmware is provided in the same location where the software is installed in the "Firmware" directory. Reset or power cycle the module(s) if firmware is upgraded. Refer to section "MODULE MAINTENANCE - Update Firmware" for more details.

Starting the application

After the installation of the software, the icon for the TDI-7000 Control application appears on the desktop. This icon launches the application that operates the unit.



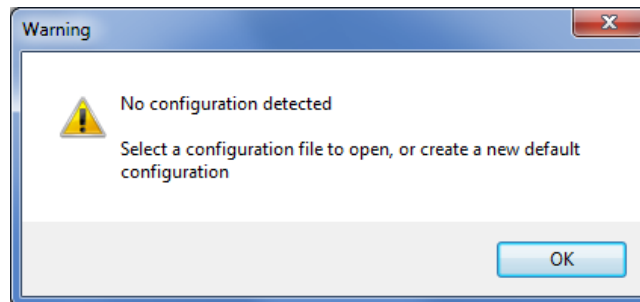
The icon points to the executable file:

C:\Program Files\Optiphase\TDI7000\TDICtrl.exe

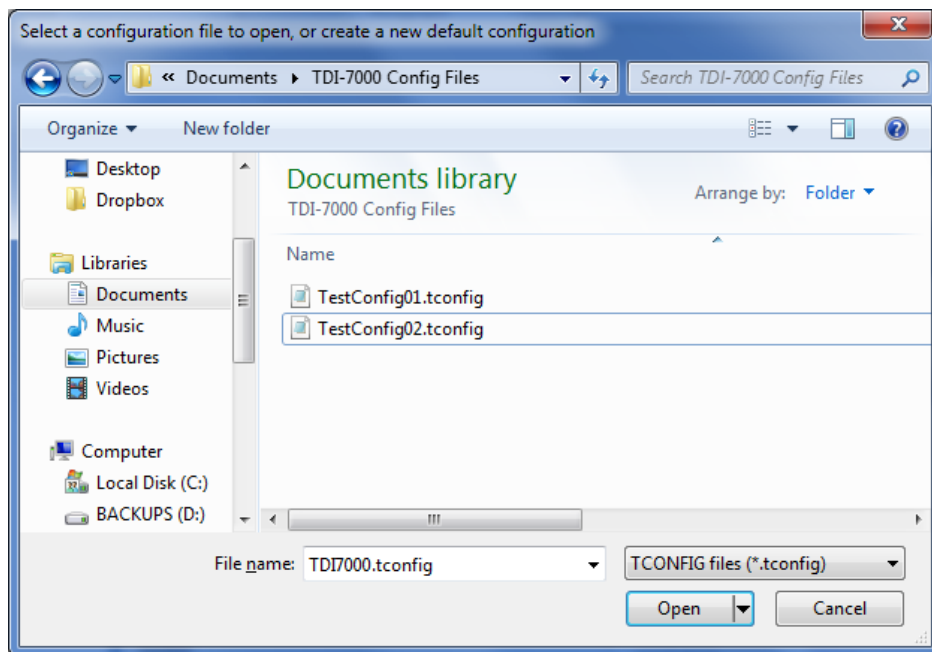
The path may vary depending on the directory into which the installation was made.

Clicking on the icon launches the TDI-7000 Control application.

When running the application for the first time, the following message is displayed:

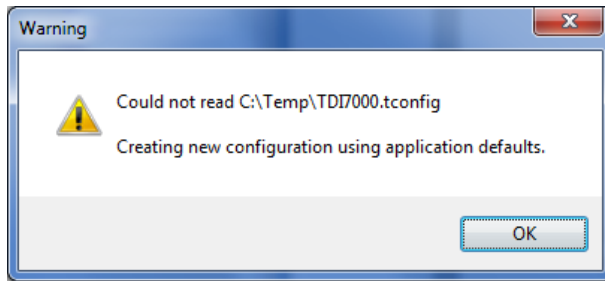


After clicking OK, the following file selector dialog shows up:



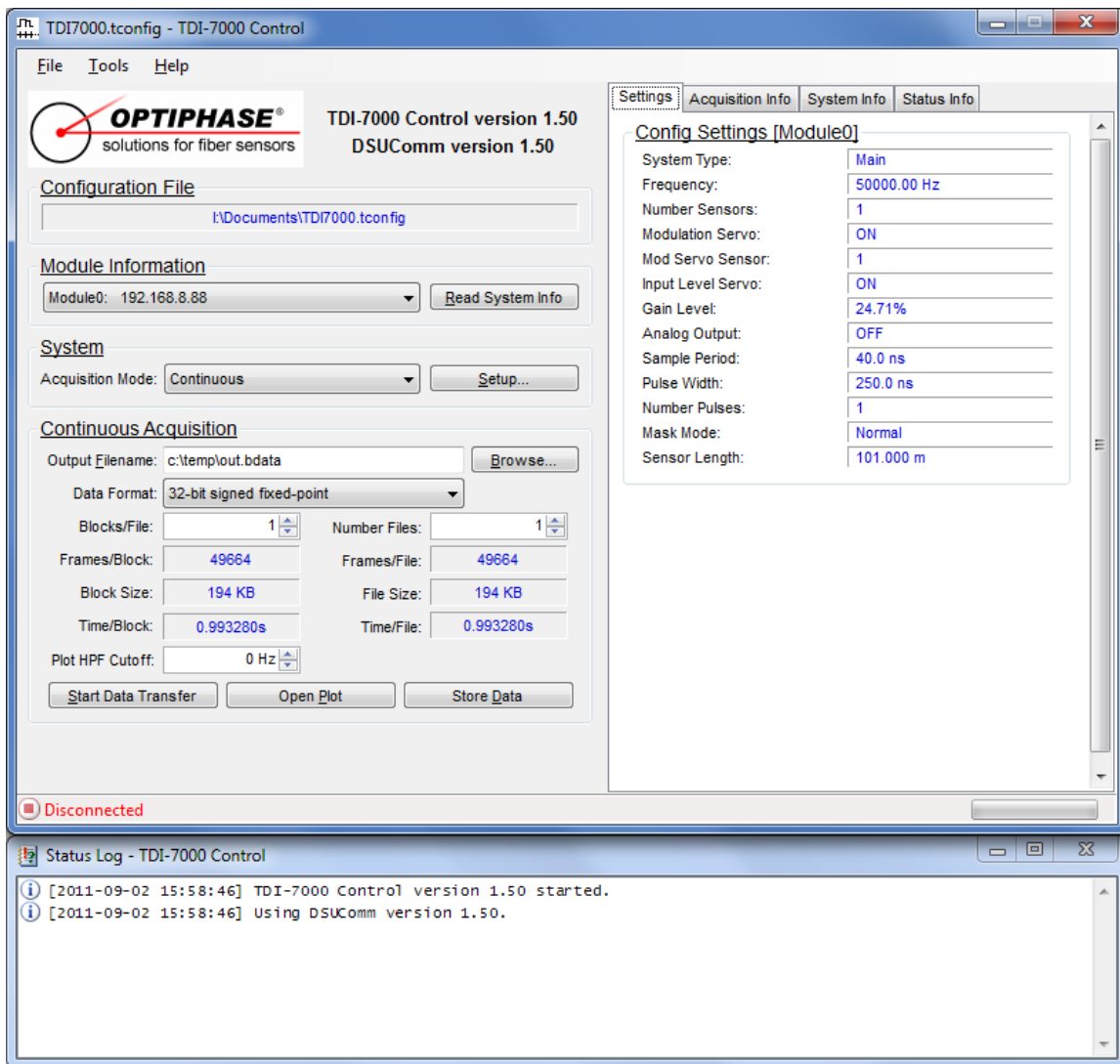
Here, you may browse to select an existing configuration file to load, or you may type in a filename that does not exist. If you type in a filename that does not exist, a new configuration file will be created with application defaults.

If a new file is created, then the following message is displayed:

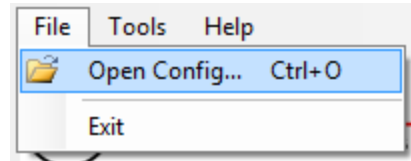


Main Window

After the configuration file is loaded, the main window where you may acquire angle data and view system status information is shown below:



In the main window, you may select a new configuration file by going to the **File** menu and selecting **Open Config**.

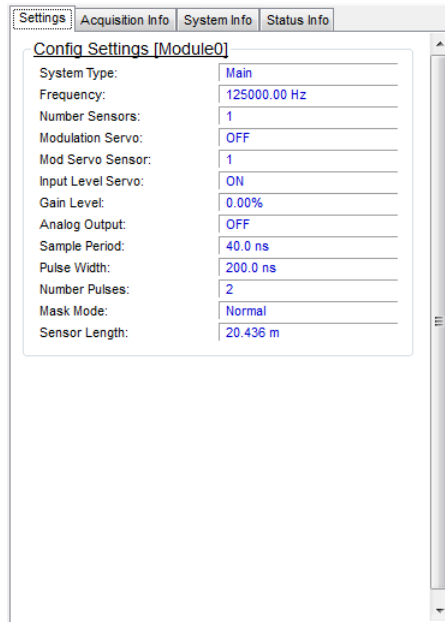


The **Configuration File** box shows the currently loaded configuration file.

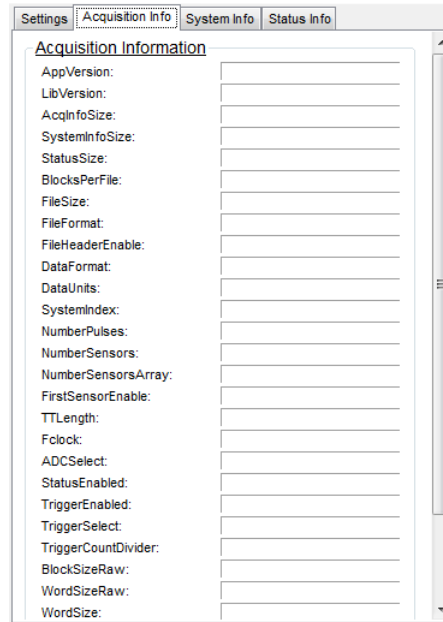
Further information about the Main Window is detailed in the section, **Phase Data Acquisition**.

Acquisition Status

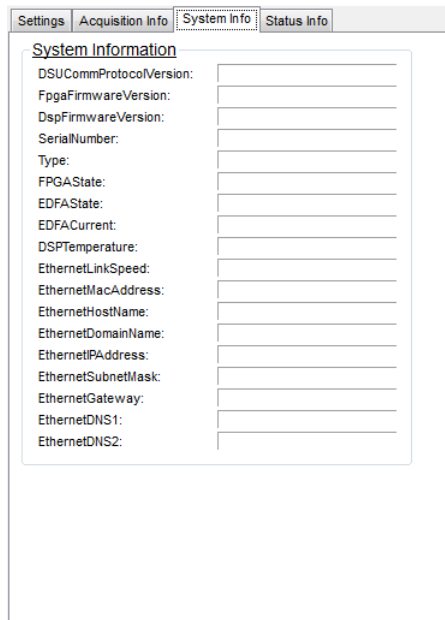
On the right side of the main window, there are tabs which provide detailed configuration and acquisition information. The following shows the 4 tabs on the main window:



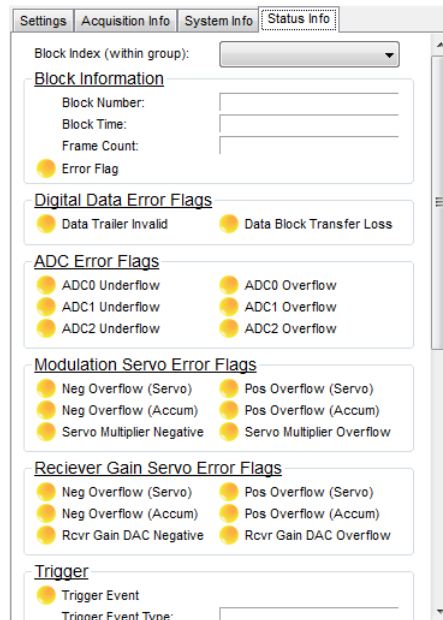
Config Settings



Acquisition Info



System Info



Status Info

The **Config Settings** tab contains key elements of the loaded configuration file.

The **Acquisition Info** tab contains information about the latest acquisition. This tab is populated with information when data acquisition is started during a continuous acquisition.

The **System Info** tab contains information about the module acquired from. This tab is populated with information when data acquisition is started during a continuous acquisition, or when the **Read System Info** button is clicked.

The **Status Info** tab contains detailed status information about the latest acquired block.

A status file may have multiple blocks contained within. If this is the case, the **Block Index** dropdown box will contain multiple blocks to select from in order to view that block's status information.

The **Block Information** section shows the current **Block Number**, the time of the first sample of the block referenced from the beginning of the acquisition (**Block Time**), the **Frame Count**, and an **Error Flag** which displays an error if any other error flags occur.

The **Digital Data Error Flags** section contains flags which tell the user if there were any errors in data transmission. If the **Data Trailer Invalid** flag occurs, then your data is corrupted. If the **Data Block Transfer Loss** flag occurs, then there is a missing block of data within the stream. This error usually happens if you run the system over specification, or if the Host PC reads the data from the Ethernet stream too slow.

The **ADC Error Flags** section contains flags which notify if any ADC is overflowing or underflowing.

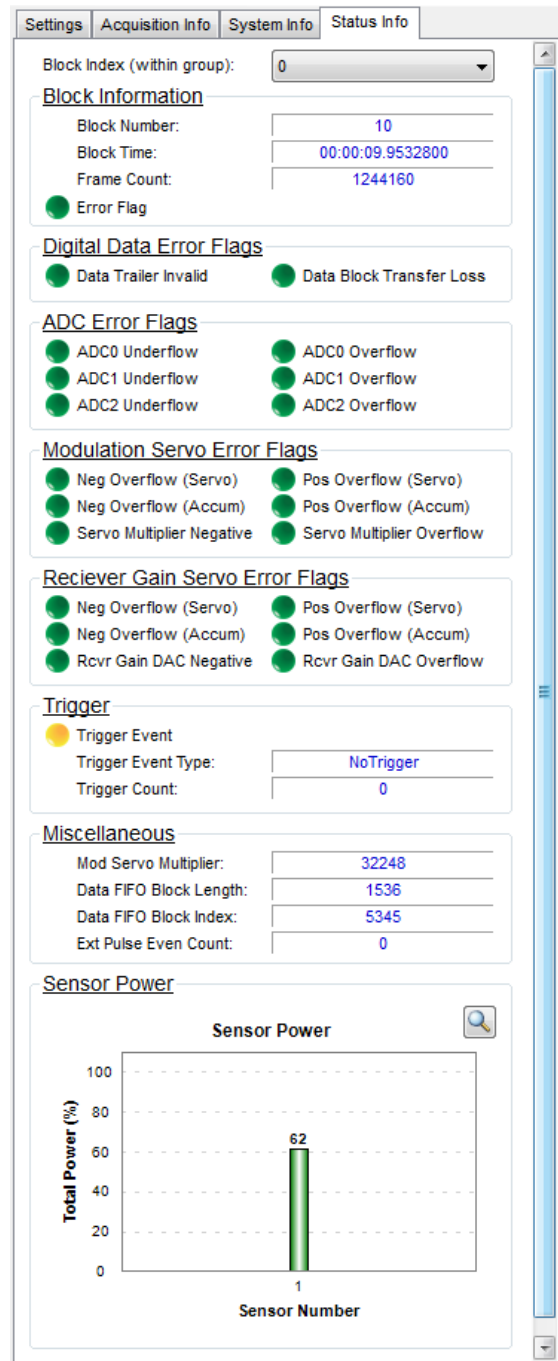
The **Modulation Servo Error Flags** section contains flags which notify whether there are any modulation servo related errors.

The **Receiver Gain Servo Error Flags** section contains flags which notify whether there are any receiver gain servo related errors.

The **Trigger** section contains a **Trigger Event** flags, which notifies whether or not a trigger event occurred in the current data block. If so, the **Trigger Event Type** and **Trigger Count** values are populated with the trigger event's details.

The **Miscellaneous** section contains extra status information. The **Mod Servo Multiplier** display contains the current value of the modulation servo multiplier. The rest of the information in this section are not commonly used.

The **Sensor Power** plot displays a bar graph showing power levels of each sensor. You may pop out the graph by clicking the magnifying glass symbol in the upper right hand corner.

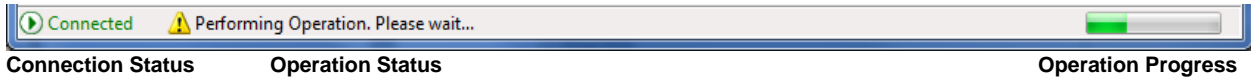


Operation Status Reporting

The application has a few methods to report information to the user to assist setting up and communicating with the device, or to display device status.

Status Bar

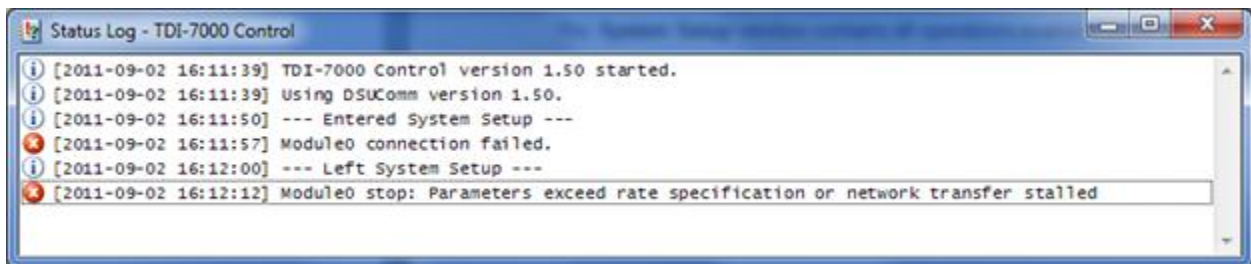
The Status Bar is located at the bottom of windows containing module operations. Status Bar components are described below:



Status Bar Component	Description
Connection Status	When connection is successfully established to module(s), status shows up as "Connected". When "Disconnected", there no module communication.
Operation Status	Shows information pertaining to start and end of operation. If operation is successful or failed, the user will be notified in this area. Further detailed error information may show up in the Status Log.
Operation Progress	Progress bar specifying progress of current operation.

Status Log

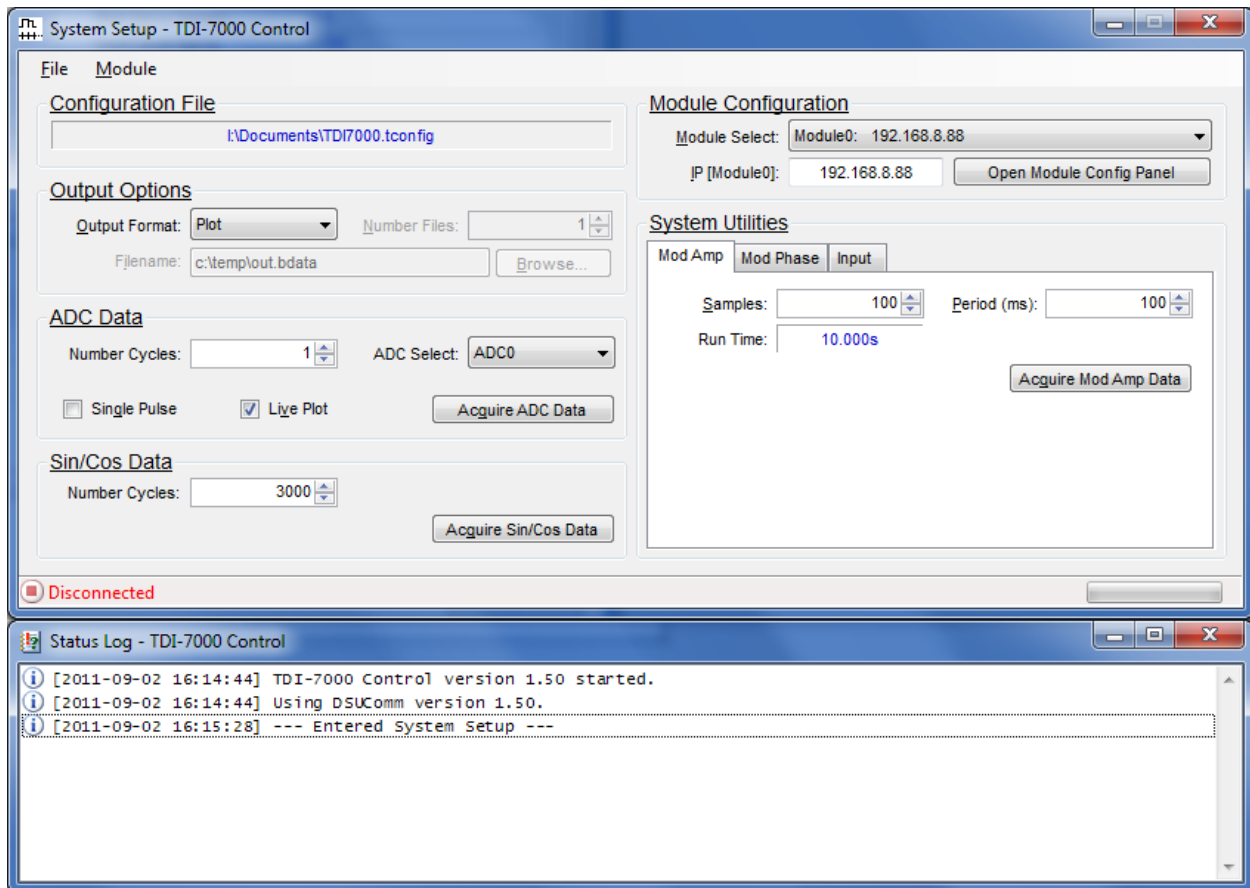
The Status Log contains timestamped entries pertaining to communication errors, search results, and other application status information. An example of the Status Log is shown below.



If any errors occur in the Status Bar, check the Status Log for detailed information, if available.

System Setup

By clicking on Setup button in the main window, one can bring up the System Setup windows shown below.



The **System Setup** window contains all operations available under this mode.

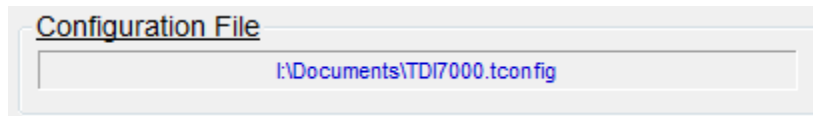
The **Configuration** window at the right contains user-modifiable settings for the selected module in the System Setup window.

The **Status Log** window is also displayed in this mode.

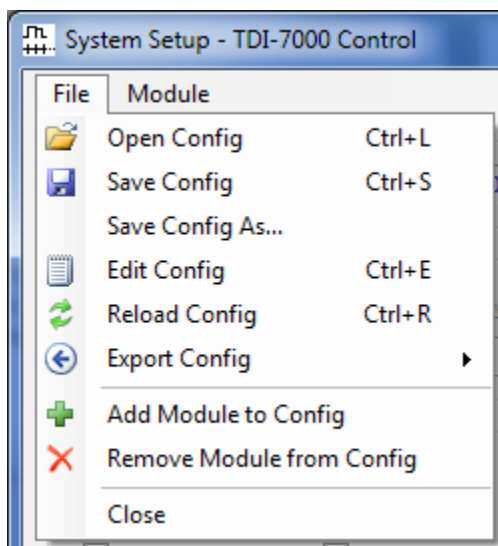
Configuration Files

A configuration file contains settings for the application and for all modules configured. The configuration file format is explained in sections "Optiphase.Settings File Format" and "Configuration File Specification" under DIGITAL DATA. The application and each module have their own sections in the configuration file.

The currently loaded configuration file is displayed at the top of the System Setup window:



If the File menu is clicked, there are several options available:



The following table describes functionality of options under the File menu:

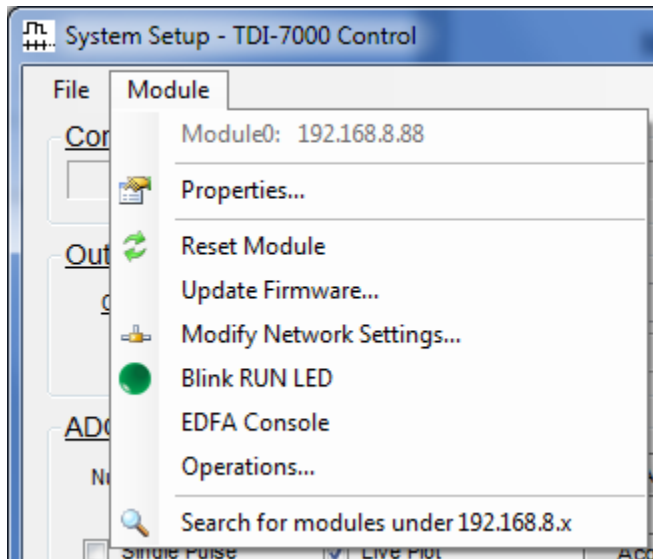
File Menu Option	Description
Open Config	Opens a previously existing configuration.
Save Config	Saves configuration to current file loaded.
Save Config As...	Choose different location to save configuration file to.
Edit Config	Opens currently loaded file in text editor. Note: This only opens the last saved copy.
Reload Config	Reloads current configuration on disk. Useful if manually editing configuration file.
Export Config (Submenu)	Provides operations to export configuration files
-- Select config type	One of the three acquisition types must be selected in this Drop-down box in order to generate the configuration binaries for the following two menu options. There are three acquisition types: Phase , ADC , and Sin/Cos .
-- Binary format (Merged)...	Exports binary configuration (bconfig) and timing table (btiming) files. Binary configuration is saved where the TDI7000AppSettings and DSUConfigN sections are merged together into one file.
-- Binary format (Separated)...	Exports binary configuration (bconfig) and timing table (btiming) files. Binary configuration is saved where the TDI7000AppSettings and DSUConfigN sections are saved to separate bconfig files. Filenames are appended with the corresponding section name.
Add Module to Config	Adds a module to the end of the configuration.
Remove Module from Config	Removes the last module from the configuration.

Close	Closes the System Setup window and returns to the main window.
-------	--

Module Maintenance

Module maintenance functionality is provided through the Module menu in System Setup:

If the Module menu is clicked, there are several options available:



The following table describes functionality of options under the Module menu:

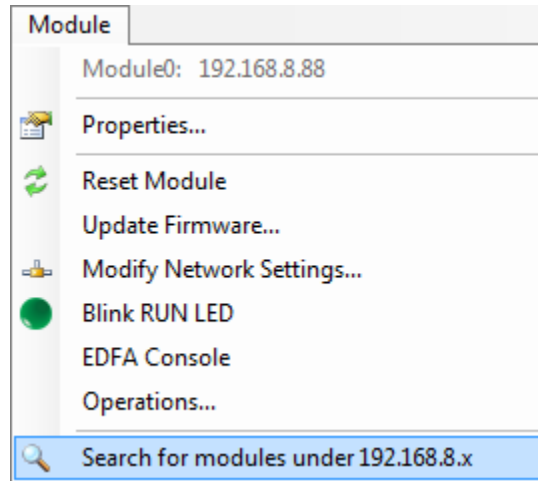
Module Menu Option	Description
ModuleN: NNN.NNN.NNN.NNN	The selected module and its corresponding IP Address are displayed at the top.
Properties...	Displays a page containing various properties of the module.
Reset Module	Performs soft reset on the module.
Update Firmware...	Displays a page that allows updating the firmware for the module's DSP and/or FPGA.
Modify Network Settings...	Displays a page that allows modifying the network settings for the module.
Blink RUN LED	Blinks the RUN LED for 10 seconds. Useful to identify which TDI-7000 is in use.
EDFA Console	Displays a console window to allow communication with the EDFA directly.
Operations...	Displays a page that allows performing several advanced operations.
Search for modules under NNN.NNN.NNN.NNN	Searches for modules through a range of IP addresses.

Most of these options are described under the **MAINTENANCE** section of this manual. The module search functionality is described on the next page.

Searching for modules on the network

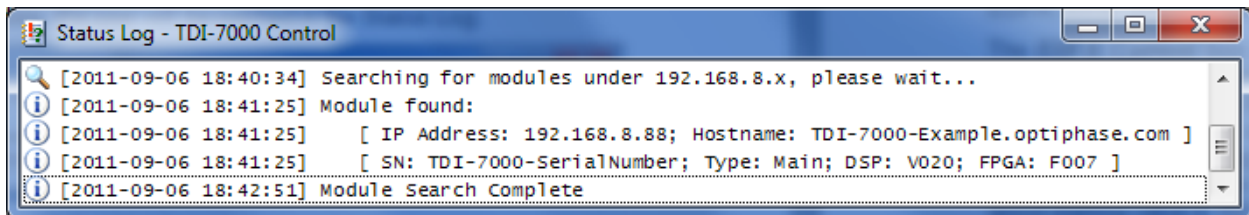
The first requirement is to locate the interrogator on the network.

The module IP address search tool is accessed from the Module menu under the Search selection.



The search tool searches through the fourth octet of the IP address is currently configured for the selected module. For example, if the currently configured IP address is 192.168.10.50, the tool searches the range of IP addresses from 192.168.10.2 through 192.168.10.254.

The results of the search are shown in the Status Log where the IP address, the unit serial number, and other identifying information are shown along with the DSP firmware and the FPGA firmware versions. If the search finds that either firmware is out of date, the user will be notified in the Status Log.

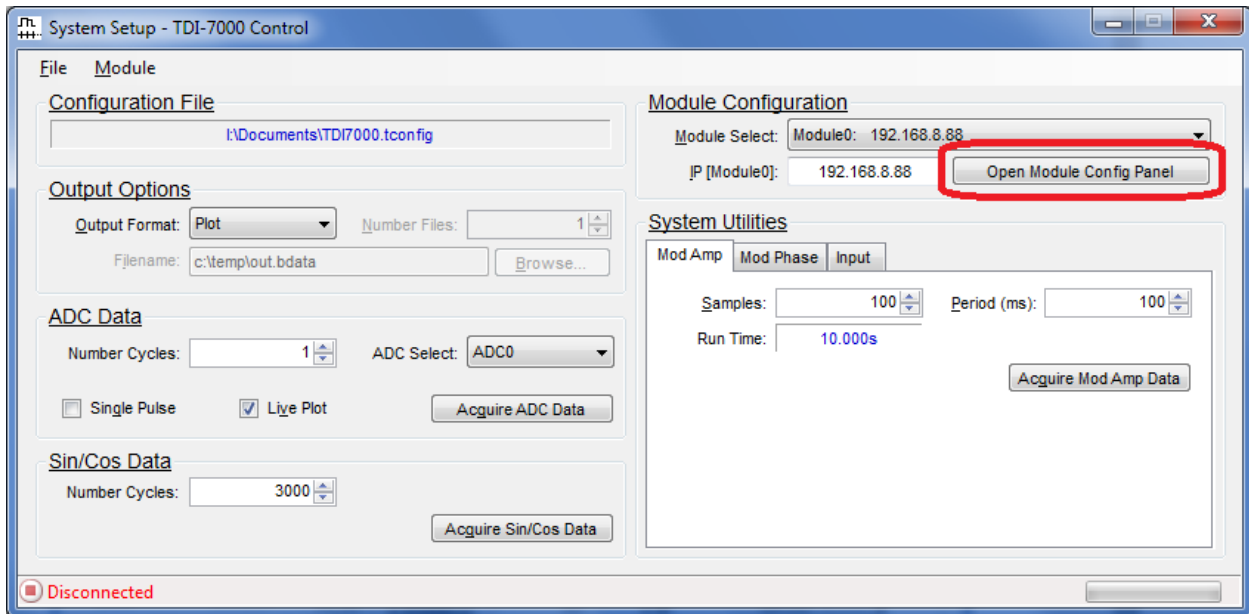


After the unit is found, enter its address into the IP field for the selected module and press enter. At that point it is now ready to set up and use. You may check its Properties under the Module Menu to verify the connection.

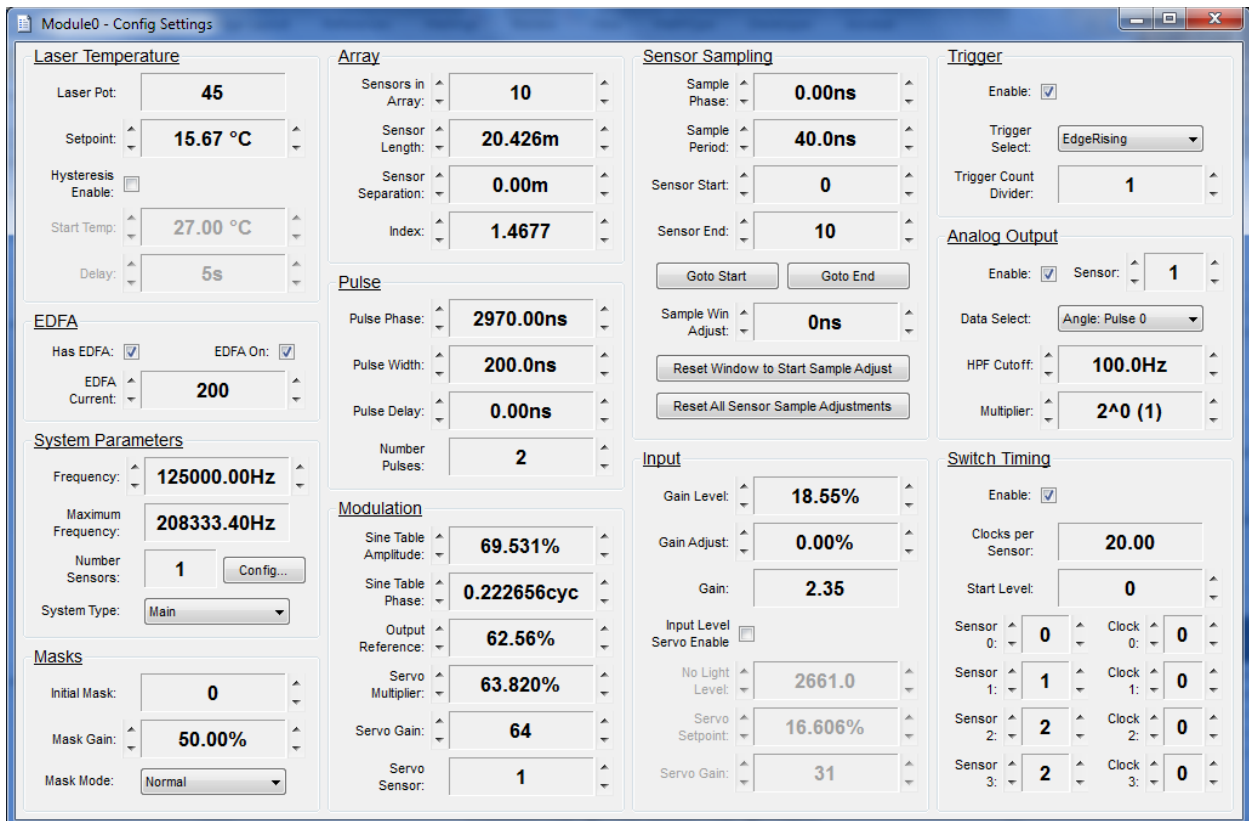
The software supports linking multiple modules together. There are two types of modules, MAIN and AUX. AUX modules only exist in systems that contain more than one DSU processor card that are synced together.

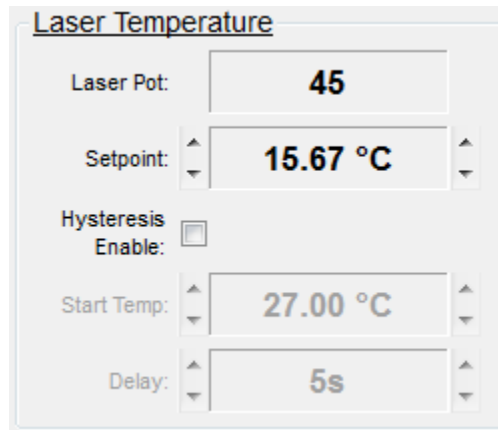
Configuring a module

The parameters that control the operation of the TDI-7000 must be entered by the user for their specific application. The configuration parameters are entered using the right pane shown in the **Setup** menu.



When the **Open Module Config Panel** button is clicked, the following window is displayed:

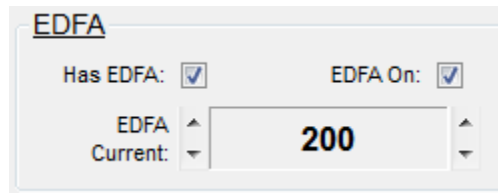


Laser Temperature


The screenshot shows a configuration window titled "Laser Temperature". It contains the following fields:

- Laser Pot:** A text box containing the value "45".
- Setpoint:** A spin box containing the value "15.67 °C".
- Hysteresis Enable:** A checkbox that is currently unchecked.
- Start Temp:** A spin box containing the value "27.00 °C".
- Delay:** A spin box containing the value "5s".

The **Laser Temp** box in this group sets the temperature of the laser optical source. The value goes from 0 to 255. The laser is specified by the customer when the order for the TDI-7000 is placed. The value that should be entered into the box is delivered along with the unit. Some ECL sources have turn on hysteresis and require a turn on temperature profile. A label on the back of the TDI unit specifies the setting for the internal source. The SUM ADC Data mode explained later is useful for this purpose for checking the light level from the array to verify the source has been setup correctly.

EDFA


The screenshot shows a configuration window titled "EDFA". It contains the following fields:

- Has EDFA:** A checked checkbox.
- EDFA On:** A checked checkbox.
- EDFA Current:** A spin box containing the value "200".

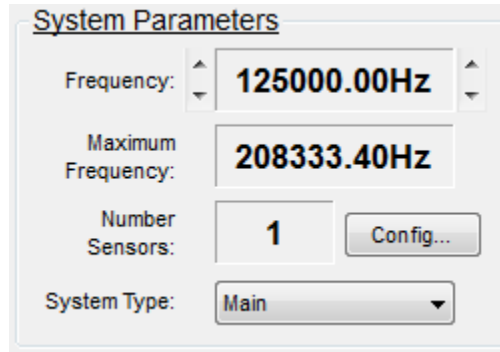
If your system contains an EDFA, ensure the **Has EDFA** checkbox is checked. This tells the module to send EDFA commands.

When **EDFA On** is checked, the EDFA will be enabled when the configuration is sent. To disable the EDFA, either uncheck EDFA On, or set the EDFA Current to 0 mA.

The **EDFA Current** box controls the current to the pump source in the Erbium Doped Fiber Amplifier. The control range is from 0 to 400 milliamps and higher optical gains are achieved with higher currents. The current limit to the EDFA pump is set by the EDFA firmware so entering 400 mA sets the pump current to its maximum value not 400 mA. The EDFA gain must be adjusted so as to not saturate the receivers while providing sensor optical returns that are sufficient for low noise operation. The ADC Data mode enables this value to be adjusted accurately and the procedure to do so is explained below. A good starting value is 150 mA until the adjustment is made.

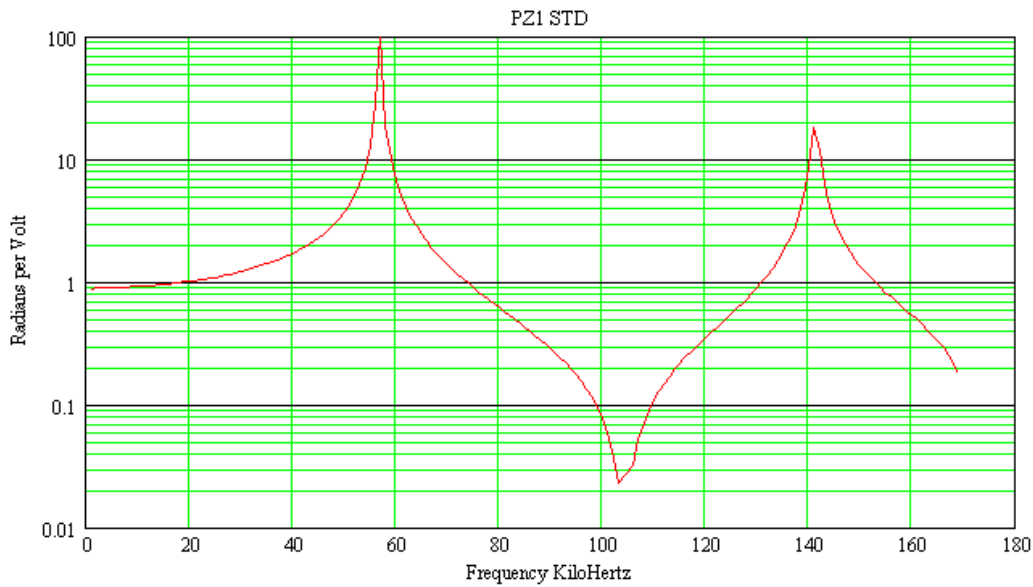
CAUTION: make sure the EDFA current is zero or the **EDFA On** check box is unchecked when connecting or removing the sensor array or working on the array.

System Parameters



The **Frequency** box sets the value of the modulation sine frequency. Only certain discrete frequencies are allowed since the modulation sine is generated by writing the sine table entries in the timing table to the modulation DAC. The frequency of the sine is the 25 MHz timing table sequencing rate divided by the number of number of entries in the sine table. The frequency entry box displays the allowable frequencies as the up and down arrow are clicked.

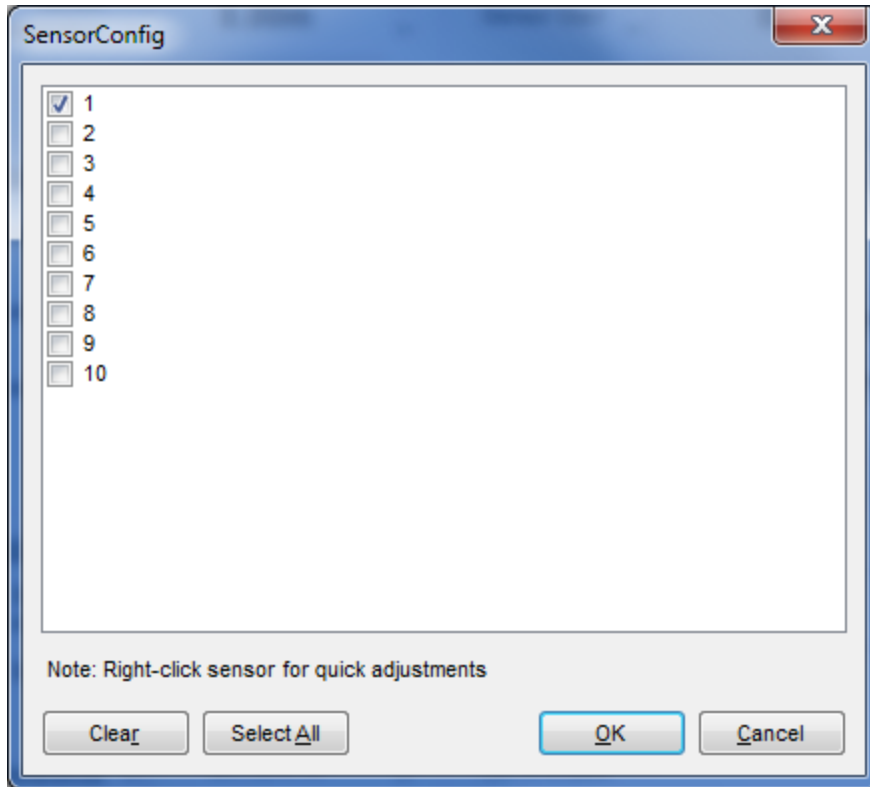
The frequency is also limited by the response of the fiber stretcher in the compensator. The stretch per volt applied changes with frequency as shown in the graph below.



Only certain frequency bands are usable as is described in the modulation section.

The array optical returns for an input pulse cannot overlap with the optical returns from the adjacent input pulse. This condition sets the maximum operating frequency which is displayed in the **Maximum Frequency** box.

The **Number Sensors** box value displays the number of sensors that are sampled. The sampled sensors are set by selecting the **Config** box which brings up the dialog shown below.



All the sensors in the defined array are shown with boxes to check to enable the sampling of a sensor. Sensor 0 (non-interfering sensor) is not shown as it is always sampled along with at least one other sensor.

Mask Parameters

The screenshot shows a control panel titled "Masks". It contains three adjustable parameters:

- Initial Mask:** A numeric input field with the value "0".
- Mask Gain:** A percentage input field with the value "50.00%".
- Mask Mode:** A dropdown menu currently showing "Normal".

The TDI-7000 includes a Polarization Diversity Receiver with three separate optical outputs. Each output has its own receiver preceded by a polarizer. The polarizer for the second receiver is aligned +120 degrees from the first and the third polarizer is aligned -120 degrees from the first. This insures that at least one receiver always has sufficient interference power to operate the TDI-7000 and avoid fading of the sensor signal.

There are three **Mask Mode** selections: **Normal**, **Once** and **Initial** that determine what mask a sensor uses for demodulation.

Normal Mask Mode

In this mode the mask with the most power found during the first demodulation cycle is found for the sensor and demodulation begins with this mask. A mask switch occurs if the highest power mask is **Mask Gain** percent greater than the current mask. Written out, this becomes:

$$100 \cdot \frac{P_{mostpower} - P_{current}}{P_{current}} > \text{Mask Gain}$$

A Mask Gain value of zero causes mask switching every time a mask obtains more power than the current mask and a value of 100% waits until the new mask has twice the power as the current mask. The **Mask Gain** control is limited to 1000% which causes the mask to switch when the new mask power is eleven times the current mask power.

A **Mask Gain** of zero should be avoided when DC stability is desired since the mask switching preserves phase continuity from mask to mask by insuring the phase of the new mask is equal to the phase of the previous mask which inserts an integrator into the phase output. This integrator can cause a walk off drift of the phase due to integrating random noise when the two mask are nearly equal.

Once Mask Mode

This mode uses the mask with the most power found during the first demodulation cycle and never switches again. This is desirable for short acquisitions where the power in a mask does not change much during the run and where the excess mask switching noise is not wanted. The excess mask switching noise is due to the phase continuity requirement where the measurement of the current mask and the previous mask are used together in the phase output which increases the noise by the square root of two for that mask switching output.

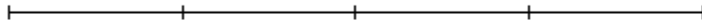
Initial Mask Mode

This mode uses the mask specified in the Initial Mask box for the entire run, never doing a mask switch. It is useful for testing.

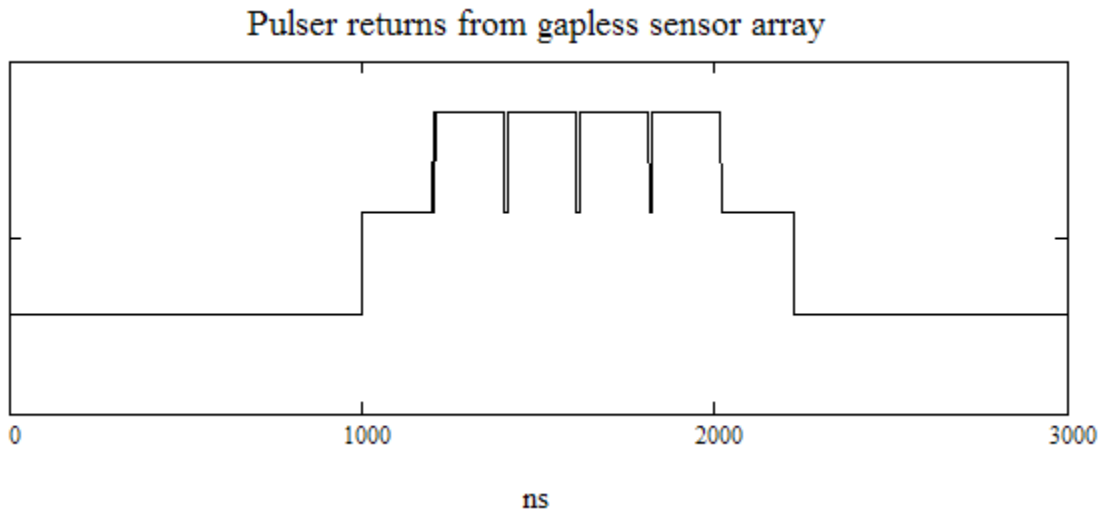
Array Parameters

Array	
Sensors in Array:	10
Sensor Length:	20.426m
Sensor Separation:	0.00m
Index:	1.4677

An array is an optical path that includes spaced optical reflectors. The most common type has a fixed length between the mirrors as shown below.



This array has four sensors as defined as the number of reflector pairs. A pulse with duration equal to the transit time of one sensor length injected into the array via a compensator gives the optical returns shown below.



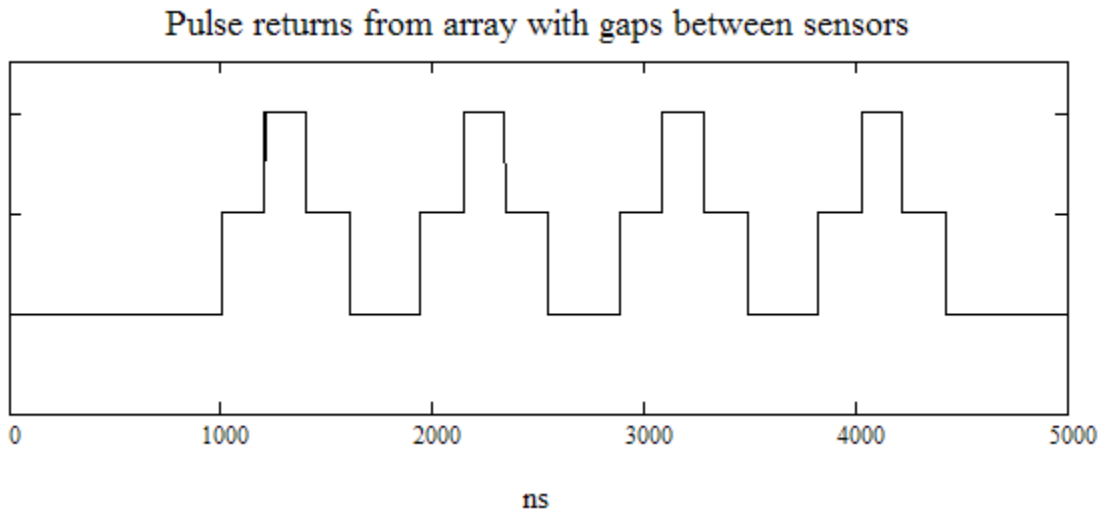
This array is defined by the number of **Sensors In Array** and the **Sensor Length** of each sensor. This specification is used to position the sample strobes for sensor sampling and demodulation. Since the samples occur in time and the sensor is specified in length, the **Index** of refraction of the optical path is also required to change from length to time.

The compensator length must match the sensor length so it is not specified. The TDI-7000 must be ordered to match the desired array.

A sensor array can be constructed with gaps between the sensors as shown in the array below.



This array has four short sensors separated by three equal gaps. A pulse injected into the array has the optical returns shown below.



The **Sensor Separation** length is used to specify this type of array to allow automatic positioning of the sample strobes.

Pulse Parameters

Pulse	
Pulse Phase:	2970.00ns
Pulse Width:	200.0ns
Pulse Delay:	0.00ns
Number Pulses:	2

The **Pulse Phase** value sets the time offset in increments of 10ns from the start of the timing table at which the even slope optical pulse is generated. The odd slope optical pulse is always generated one half of a table length later. If the ADC strobe positions are fixed in the timing table, the **Pulse Phase** is used to move the sensor optical returns so that they overlay the ADC samples.

The **Pulse Width** value controls the duration of the optical pulse and the **Sample Period** value determines the period ($1/F_s$) between samples of the sensor return. Five samples are taken of each sensor return so the pulse width must accommodate this. This means that the **Pulse Width** must be no smaller than four times the **Sample Period** value plus the rise and fall times of the pulse. The minimum **Sample Period** is 40ns so the minimum **Pulse Width** is 160ns plus about 40ns.

The **Pulse Delay** value adds a very fine 0.25ns step size to the optical pulse generation position. In most situations, it is left at zero.

The **Number Pulses** box controls whether one slope or two slopes are demodulated.

Modulation Parameters

Modulation	
Sine Table Amplitude:	69.531%
Sine Table Phase:	0.222656cyc
Output Reference:	62.56%
Servo Multiplier:	63.820%
Servo Gain:	64
Servo Sensor:	1

The modulation parameters control the amplitude and phase of the sine modulation voltage driving the Optiphase PZ1 fiber stretcher in the compensator and setup the automatic amplitude servo operation.

The **Sine Table Amplitude** (A), **Output Reference** (R), and **Servo Multiplier** (M) values together control the amplitude of the sine modulation. The amplitude of the sine voltage is given by:

$$V_A = 160 \cdot \frac{A}{100} \cdot \frac{R}{100} \cdot \frac{M}{100} \text{ volts}$$

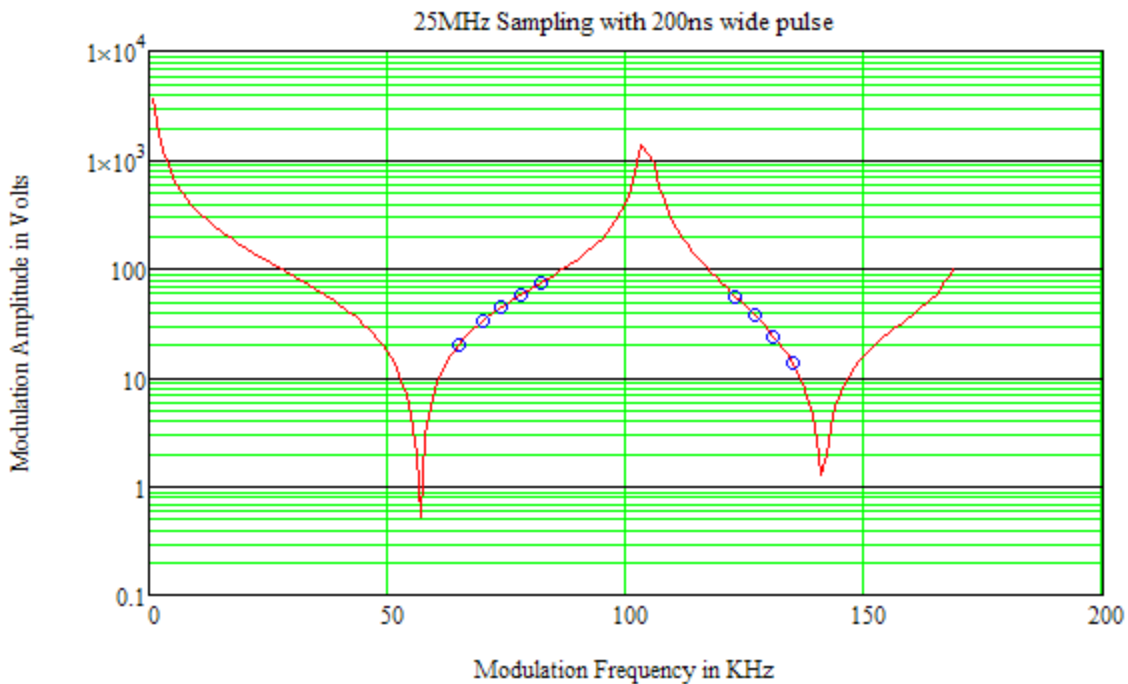
The three sine amplitude multipliers fill different functions. The **Sine Table Amplitude** is the amplitude of the digital sine that is written to the timing table in the FPGA. The **Output Reference** value controls the voltage output of a DAC that sets the reference voltage for the multiplying modulation DAC. The **Servo Multiplier** is a digital value in the FPGA that multiplies the digital sine table to form the digital input for the modulation DAC and is changed by the modulation amplitude servo in the FPGA.

The modulation voltage causes the PZ1 in the compensator to stretch the fiber and change the phase of the light. The required modulation sine to give the needed radian phase change at the zero crossing to separate each of the five sensor samples by $\pi/2$ radians is:

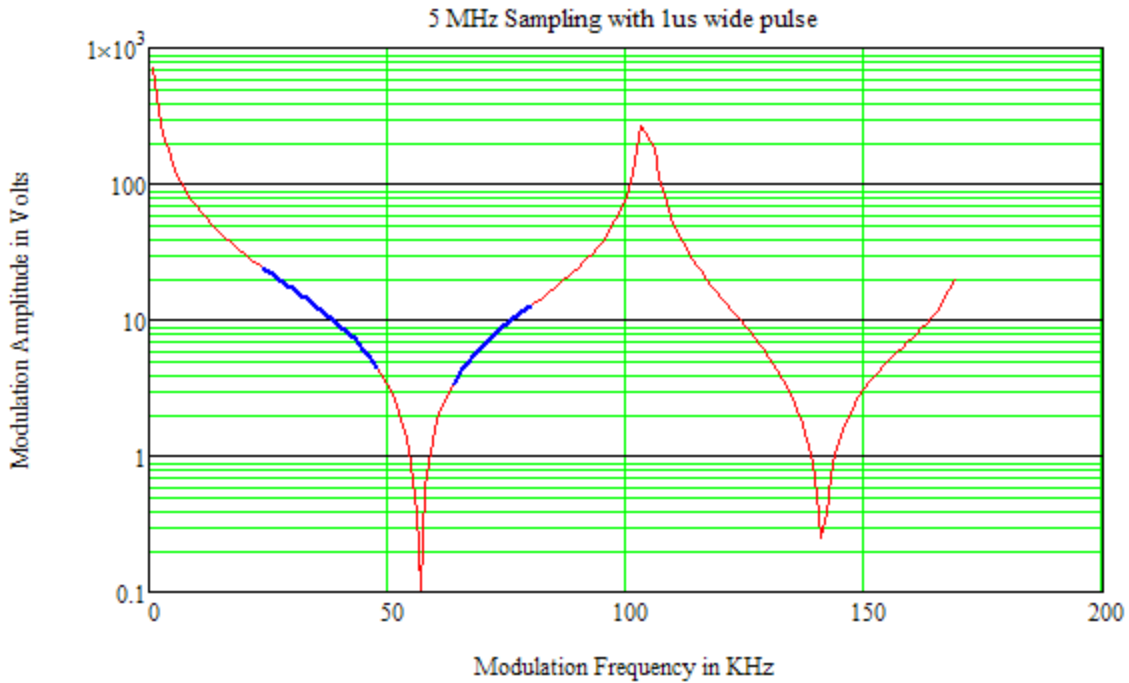
$$\frac{F_s}{4 \cdot F_m} \cdot \sin(2\pi F_m t + W) \quad \text{radians}$$

where F_s is the sensor sample frequency, F_m is the modulation frequency and W is the **Sine Table Phase**. This relation shows that as the modulation frequency decreases, more voltage is needed and as the sample rate increases, the voltage also increases. Additionally, the PZ1 transducer between voltage and induced radians needs to be considered as well.

The graph below shows the required sine amplitude in volts on the compensator modulator for a sample rate of 25MHz as a function of modulation frequency. Since the maximum sine amplitude is 160 volts, a frequency of 100 KHz would not work since it would require a 400 volt amplitude.



The next graph shows the case for a sampling rate of 5 MHz where a 100 KHz frequency would work. However, operating this close to a PZ1 resonance peak is not advised since this region is phase unstable. Operation must be verified for such a frequency.



The modulation **Sine Table Phase** is the last value adjusted after the pulse sampling is set up. It is used to align the optical pulse with the radian modulation sine zero crossing.

The FPGA has a built in servo that adjusts the modulation amplitude using a sensors samples. The modulation servo insures that there is $\pi/2$ radians between each sensor sample as required for correct measurement of the sensor signal. The modulation **Servo Sensor** value sets up the FPGA to use the samples from sensor the specified in this value. Normally, sensor 1 is used (sensor 0 is the non-interfering return and cannot be used). The modulation **Servo Gain** value determines the modulation servo bandwidth. A zero value in the **Servo Gain** box turns off the servo while a higher value makes the servo close faster. The largest value allowed is 255.

Sensor Sampling

The screenshot shows a control panel titled "Sensor Sampling". It contains the following elements:

- Sample Phase:** A numeric input field with up/down arrows, set to 0.00ns.
- Sample Period:** A numeric input field with up/down arrows, set to 40.0ns.
- Sensor Start:** A numeric input field with up/down arrows, set to 0.
- Sensor End:** A numeric input field with up/down arrows, set to 10.
- Goto Start:** A button.
- Goto End:** A button.
- Sample Win Adjust:** A numeric input field with up/down arrows, set to 0ns.
- Reset Window to Start Sample Adjust:** A button.
- Reset All Sensor Sample Adjustments:** A button.

The **Sample Period** value determines the period ($1/F_s$) between samples of the sensor return. It also determines the modulation cycle sample fraction. The sensor return is sampled five times so the duration of the sampling is four times the **Sample Period** value and the modulation cycle period is one over the modulation frequency. The TDI-7000 works by inducing a linear modulation phase change during the duration of the sensor sampling using the zero crossing of the sine modulation which is approximately linear only if the cycle fraction (sampling period divided by modulation period) is small. Larger cycle sampled fractions extend into the nonlinear region of the modulation sine and create distortion of the sensor measurement.

The **Sample Phase** sets the 10ns position in the timing table of the first sample of the non-interfering pulse return and thus the offset into the timing table of all the samples. It is used to move the sample set forward with respect to the start of the timing table to align with the pulse returns. For single DSU systems with a main DSU only, the **Sample Phase** is always zero since the **Pulse Phase** performs exactly the same function by moving the sensor returns instead of the sensor samples. The **Sample Phase** is required in dual DSU systems where the main DSU generates the optical pulse. The **Pulse Phase** adjusts the sensor returns for the main DSU sampling and the **Sample Phase** adjusts the sensor samples for the aux DSU sampling since the pulse control on the aux DSU is not used in dual systems.

The **Sample Win Adjust** parameter advances or retards one or more sensors' samples by 10ns increments. The **Sample Win Adjust** values are normally zero for all the array sensors and are only changed when the array cannot be constructed according to the specification in the Array parameter section. The minimum **Sample Period** of 40ns cannot be violated by the **Sample Adjust** value so it is only useful for arrays where the sensors are longer than about 20m or there exists gaps between the sensors.

The sample adjust parameter applies to the samples for the sensors listed in the **Sensor Start** to **Sensor End** boxes. A sensor window can be set to move a group of samples. The color of the sensor sample points are green when they are included in this adjust window, and they are red if not. This is useful for systems with separated sub-arrays. The sample adjust can be reset to zero for all the sensors in the window or all the sensors in the array to provide a known initial state.

Input Parameters

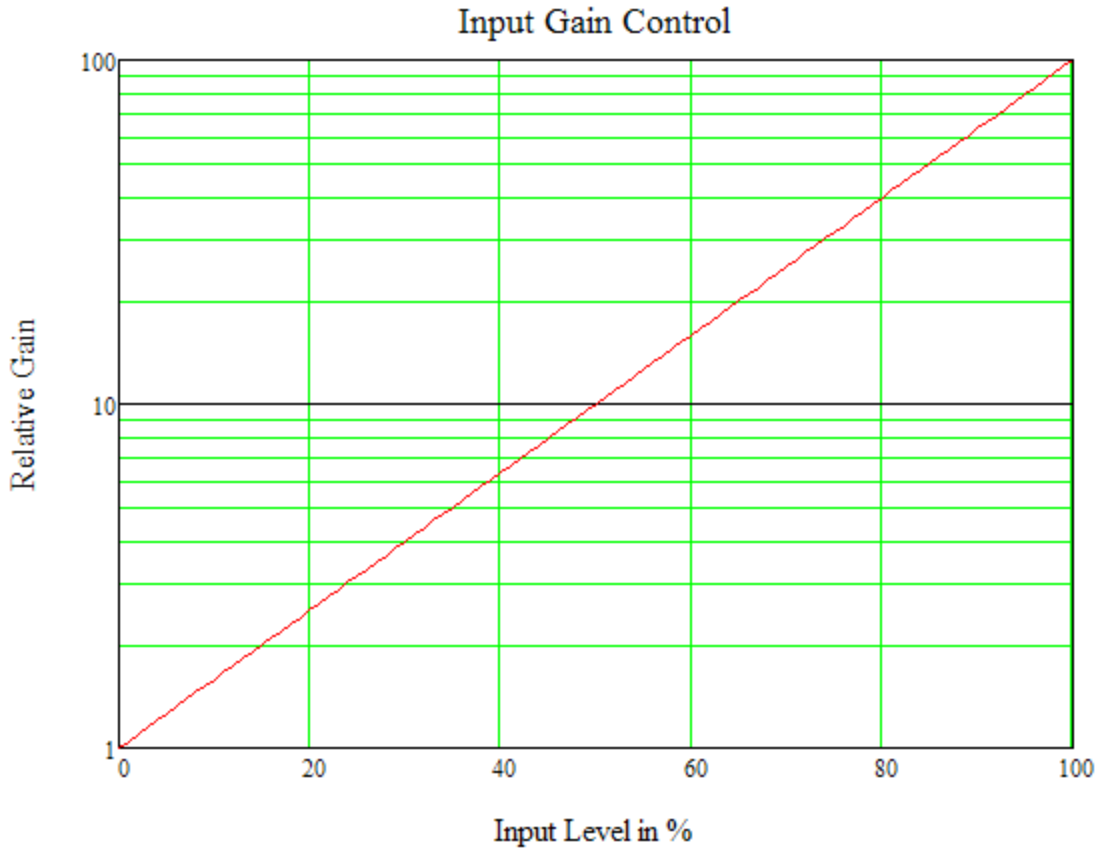
Input	
Gain Level:	18.55%
Gain Adjust:	0.00%
Gain:	2.35
Input Level Servo Enable	<input type="checkbox"/>
No Light Level:	2661.0
Servo Setpoint:	16.606%
Servo Gain:	31

The three optical receivers have a common amplification control line whose analog level is set by a digital to analog converter Gain DAC. The digital input to the Gain DAC is the sum of the timing table 10-bit gain array and the **Gain Adjust** register in the FPGA. A sequential entry from the timing table gain array is added to the Gain Adjust register and output to the Gain DAC every four master clock periods which is about 40ns.

The amplification control line goes to amplifiers which have a logarithmic gain response. The linear gain is calculated and displayed in the **Gain** box. The relative gain is computed from the relation:

$$\frac{G}{G_0} = 10^{0.02*((GainLevel+GainAdjust) \text{ in } \%)}$$

The maximum relative gain is 100. This means that the programmable input gain can vary from 1 to 100 times the input level.



The **Gain Level** value sets the timing table gain array entries to a constant value. Dynamic gain profiles are not yet supported by the Setup interface but are supported by the hardware. The Gain Adjust value can be changed when the **Input Level Servo Enable** box is unchecked.

Checking the **Input Level Servo Enable** box activates the optical level servo in the FPGA.

Input

Gain Level:

Gain Adjust:

Gain:

Input Level Servo Enable

No Light Level:

Servo Setpoint:

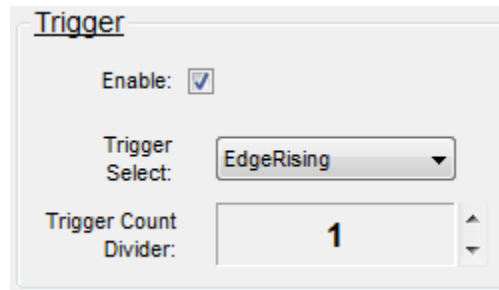
Servo Gain:

The input level servo works by maintaining the optical power in sensor 0 at a target level. Sensor 0 is the non-interfering reflection from the first mirror in the array. The optical power in sensor 0 is measured by summing the three mask samples of the non-interfering pulse return and changing the receiver gain to make the sum equal to a target value called the servo setpoint. The mask samples include the no light level offset which must be removed to obtain the true non-interfering pulse height which can be compared to the setpoint target. The servo operation follows the equation below where the values are the FPGA register names.

$$RcvrOffset = RcvrOffset + RcvrServoGain \cdot \left[RcvrServoSet - \sum_{i=0}^2 (S2_i^0 - RcvrZLL) \right]$$

The **No Light Level** is entered by taking an ADC Data plot acquisition in the sum mode with the EDFA disabled and using the *Calculate Mean* function in the plot to find the height of the base line. This value is entered into the **No Light Level box**. The **Servo Setpoint** should normally be no larger than 25% to avoid ADC saturation. Small values of the **Servo Gain** insure a less noisy input level control line that takes longer to close and larger values close faster but with higher resulting input level gain noise.

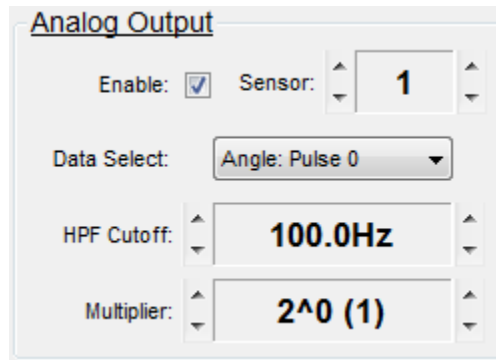
Trigger Parameters



A BNC on the rear panel of the TDI provides an opto-isolated input for an external trigger. Enabling the trigger circuit causes the rising edge, the falling edge or both to latch a 32-bit counter in the gate array. This counter starts from an initial value of zero and counts up to record the time from the TDI reset at which the external trigger occurred. The Trigger Count Divider can be specified and 25 is a reasonable initial value giving a clock maximum time of 1.2 hours and a 1us trigger time resolution. The counter overflows after this time which can be easily detected allowing the trigger time to be adjusted for longer periods of operation.

The latched counter value is transferred to the status trailer in the gate array and sent to the host. Multiple triggers are supported but cannot occur closer in time than the gate array data frame period or the trigger counter status is over written.

Analog Output



Analog output, when enabled, sends 24 bits data to the analog out BNC on the rear panel. Internal to the module, the sensor selected is extended to 64 bits then the high pass filter is applied.

The **Sensor** option selects which sensor to send via analog out if an Angle **Data Select** is chosen. Note: Sensor 0 is the non-interfering sensor, so its data may not be useful. Sensors in the array begin with Sensor 1.

Data Select gives an option of the following outputs:

Data Select	Description
Angle: Pulse 0	Output angle data from first pulse derived from odd modulation slope
Angle: Pulse 1 ²	Output angle data from second pulse derived from even modulation slope
Angle: Average ²	Output the average angle of both measurements

In addition, there are three test modes:

Data Select	Description
Test: Zero	Sends a digital zero to the DAC
Test: Square Wave	Full scale square wave with modulation cycle period
Test: Sinusoid	Full scale sinusoid; frequency may vary slightly depending on configuration

The **HPF Cutoff** parameter specifies the cutoff frequency of the single pole high pass filter for the analog output. A value of zero means that the filter is not applied to the data.

The **Fringe** setting shifts the 64-bit angle by the specified value. A positive value shifts left, and a negative value shifts right. The analog out DAC is 24-bits. So a Fringe value of 0 would give 20 bits of angle and 4 bits of fringe, for 16 fringes full scale.

² Option is only available if NumberPulses = 2

Switch Timing

Switch Timing

Enable:

Clocks per Sensor: 20.00

Start Level: 0

Sensor 0:	0	Clock 0:	0
Sensor 1:	1	Clock 1:	0
Sensor 2:	2	Clock 2:	0
Sensor 3:	2	Clock 3:	0

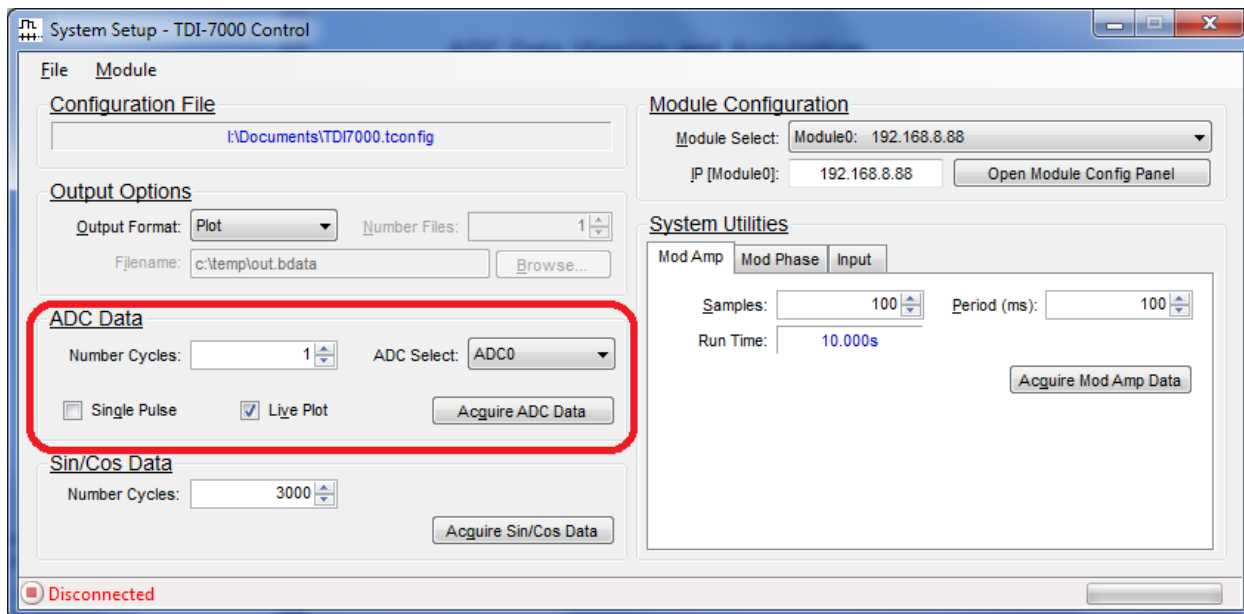
A general purpose programmable digital output is provided from the DSU for synchronizing external events to the sampling of sensors. The timing of the Switch output is setup by the section shown below.

The **Clocks per Sensor** is the number of master clocks between sensor returns and the **Start Level** is the initial voltage (low or high) of the **Switch** output line. From the initial level, the Switch state can be changed four times during even half of a modulation cycle. The switch pattern is repeated for the odd half of the modulation cycle.

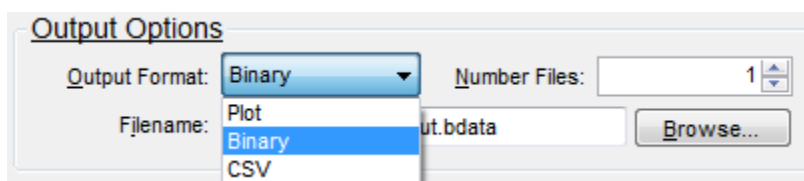
The time that a state change takes place is specified as the start time of a sensor return plus some fraction of the **Clocks per Sensor**. The **Switch Sensor** number is entered first and then the **Switch Clock** which is the number of master clocks into the **Switch Sensor**. The switch timing is entered in this way to synchronize the state transition of the switch output with the sensor returns.

ADC Data Viewing and Acquisition

The **ADC Data** section provides a 50MHz oscilloscope-style capture of the raw sensor returns:



The **Output Format** box directs the output of the ADC Data acquisition to one of the following: a plot, a binary file or a CSV (spreadsheet) file. The file's name is entered in the **Filename** box. Multiple files can be written using the **Number Files** setting.



Since there are three receivers, one must be selected by using the **ADC Select** pull down menu. The choices are:

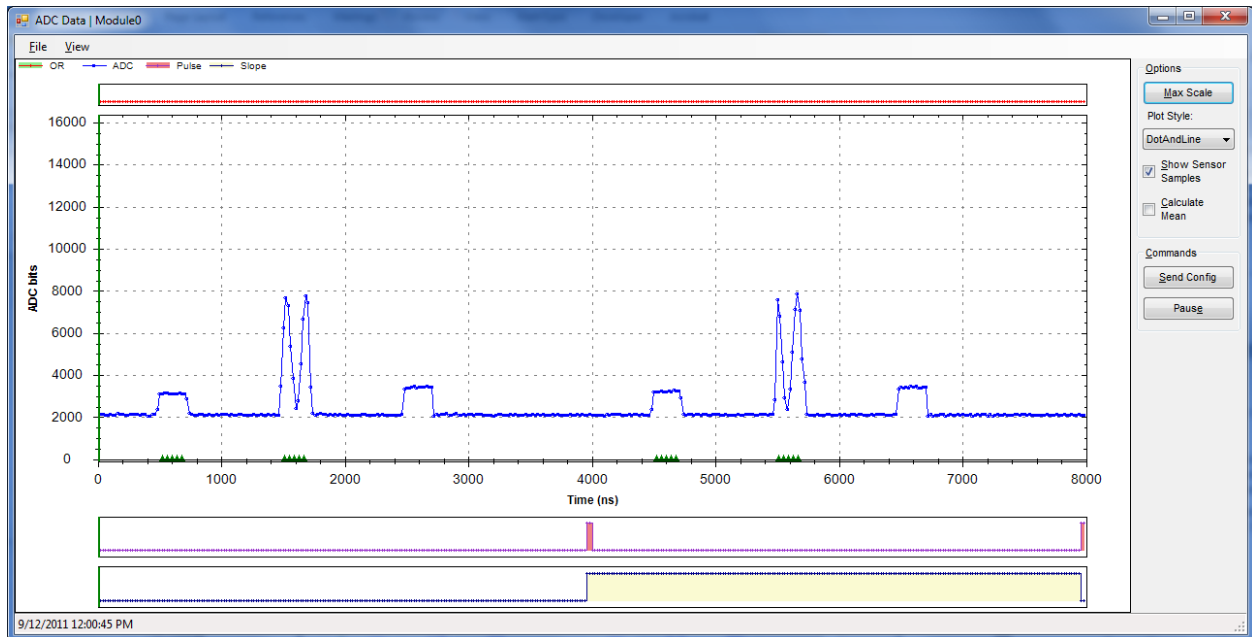
ADC Select	Description
ADC0	Receiver 0
ADC1	Receiver 1
ADC2	Receiver 2
SUM	Their sum divided by two: [ADC0 + ADC1 + ADC2] / 2

The **Number Cycles** box enters the number of modulation cycles to acquire per file or plot.

The EDFA must be on to see a sensor return on the receiver. When the **Single Pulse** check box is checked, only one pulse is activated before the ADC Data capture instead of one or two pulses per modulation cycle. This allows the operator to check for lead in fiber reflection discontinuities without multiple sensor returns hiding defects.

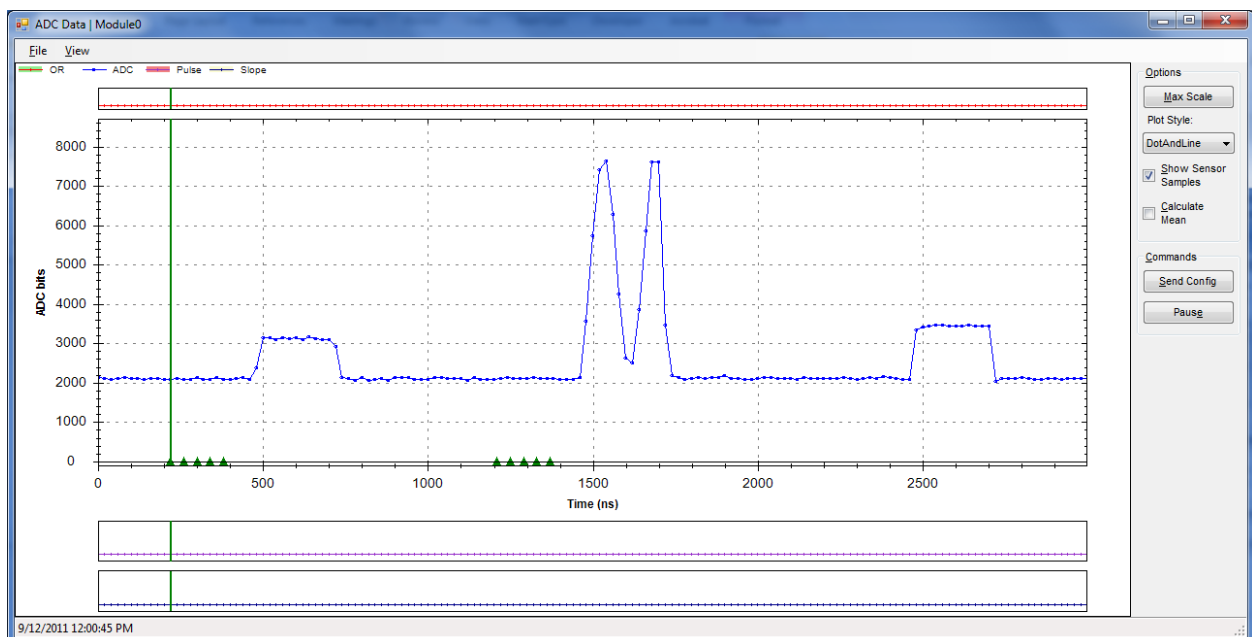
TDI-7000 TDM Fiber Interrogator

The example below shows receiver 0 for a single sensor. The blue curve is the sampled receiver data. The red line at the top is the overflow bit from the ADC indicating ADC saturation when not level. The violet line below the graph is the pulse bit shown when the pulse was generated. The bottom graph is the slope bit showing even and odd modulation regions.



The graph utility has many functions. By clicking and moving the cursor, a region can be selected for expansion as shown below. The red triangles on the time axis show the location of the sensor samples. The **Plot Style** can graph dots, lines or dots and lines. The **Calculate Mean** checkbox finds the average value of the graph.

In the graph below, the sensor samples are in the wrong location. This fact is easy to see and shows why the ADC Data mode is very useful.



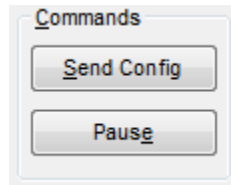
If you click the display, a cursor shows up. This cursor assists the user in aligning the sensor samples.

The ADC Data display graphs the FPGA FIFO read data versus time and also the timing table sample strobe positions. This information is used to align the ADC samples with the array returns.

One issue is that the FIFO data starts from the first sample read which is called time zero, while the timing table first ADC strobe starts at time **SamplePhase**, from the start of the timing table. In order to plot the timing table ADC strobes on the same graph as the FIFO data reads, it is necessary to subtract the **SamplePhase** offset time from the timing table strobes. Doing this forces the first ADC sample to occur at the zero time of the FIFO data read graph.

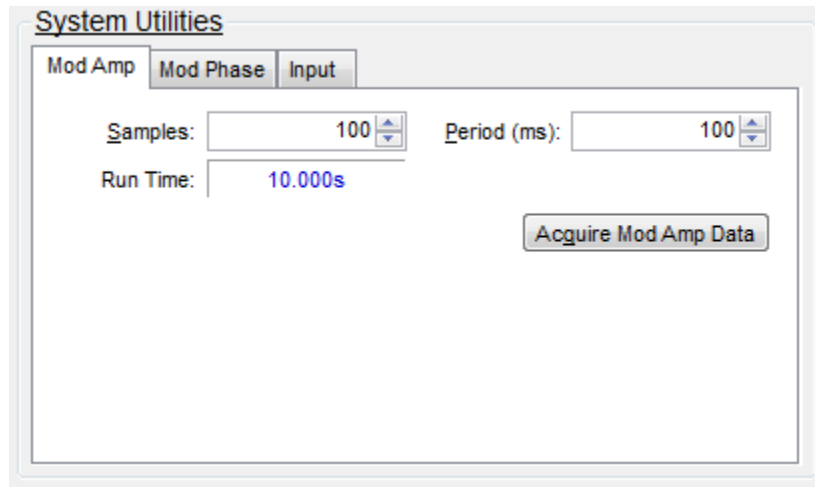
This makes the graph look the same for pulse adjustments and sample phase adjustments except that the directions are different. When the pulse position is increased, the returns move to the right. When the ADC phase is increased, the returns move to the left since the first sample remains at zero.

Two commands appear in the ADC Data display window:



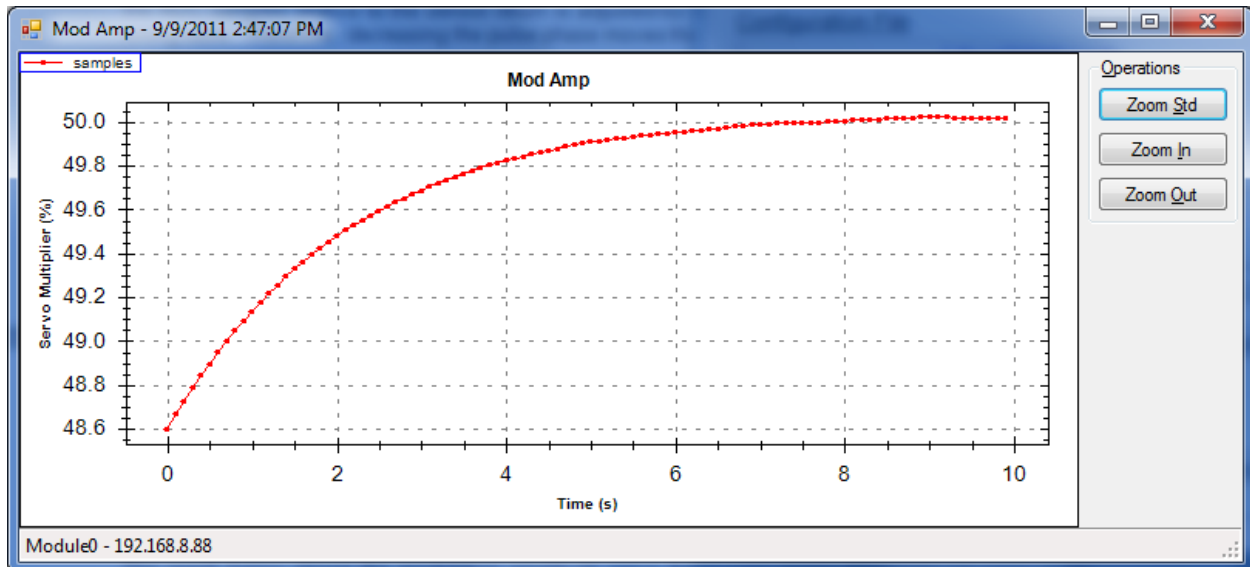
If a configuration setting is changed while the Live Plot is running, one may send the configuration to the module using the **Send Config** button. To pause the Live Plot, click the **Pause** button.

Modulation Amplitude Servo Utility



The modulation amplitude servo mode acquires the modulation Servo Multiplier status register from the FPGA at a set time interval after a reset. It starts with the current **Servo Multiplier** entry and writes that to the FPGA and then reads the FPGA register for the number of **Samples** where there is **Period** milliseconds between the gate FPGA register reads. The **Servo Gain** sets the time constant of the modulation amplitude servo and the **Servo Sensor** specifies the sensor to be used.

The starting value of the modulation amplitude must be close to the correct value and also leave room for the servo to operate. Also, the **Sine Table Phase** must be approximately correct. The graph below appears when the acquisition is complete and shows the modulation amplitude servo closing to the setpoint value of about 50%. This value is used by the operator to update the set point in the **Servo Multiplier** box.



Sine Table Phase Adjust Utility

The center of the optical pulse must be aligned with the zero crossing of the induced radian modulation. This is accomplished using this utility. The approach is to automatically step the **Sine Table Phase** about a center value and let the modulation servo operate at each phase. The endpoints of the modulation servo are read for each value of the **Sine Table Phase**. A graph of the **Servo Multiplier** versus the **Sine Table Phase** has a minimum where the center of the optical pulse aligns with the zero crossing of the modulation.

The screenshot shows the 'System Utilities' window with the 'Mod Phase' tab selected. The configuration parameters are as follows:

Parameter	Value	Parameter	Value
Sweep Samples:	11	Sweep Center (Module0):	0.222656cyc
Sweep Increment:	0.006256cyc	Sweep Range:	±0.031281cyc
Sample Delay:	10s	Run Time:	220.000s
Sample Averaging:	100		
Averaging Period:	100ms		

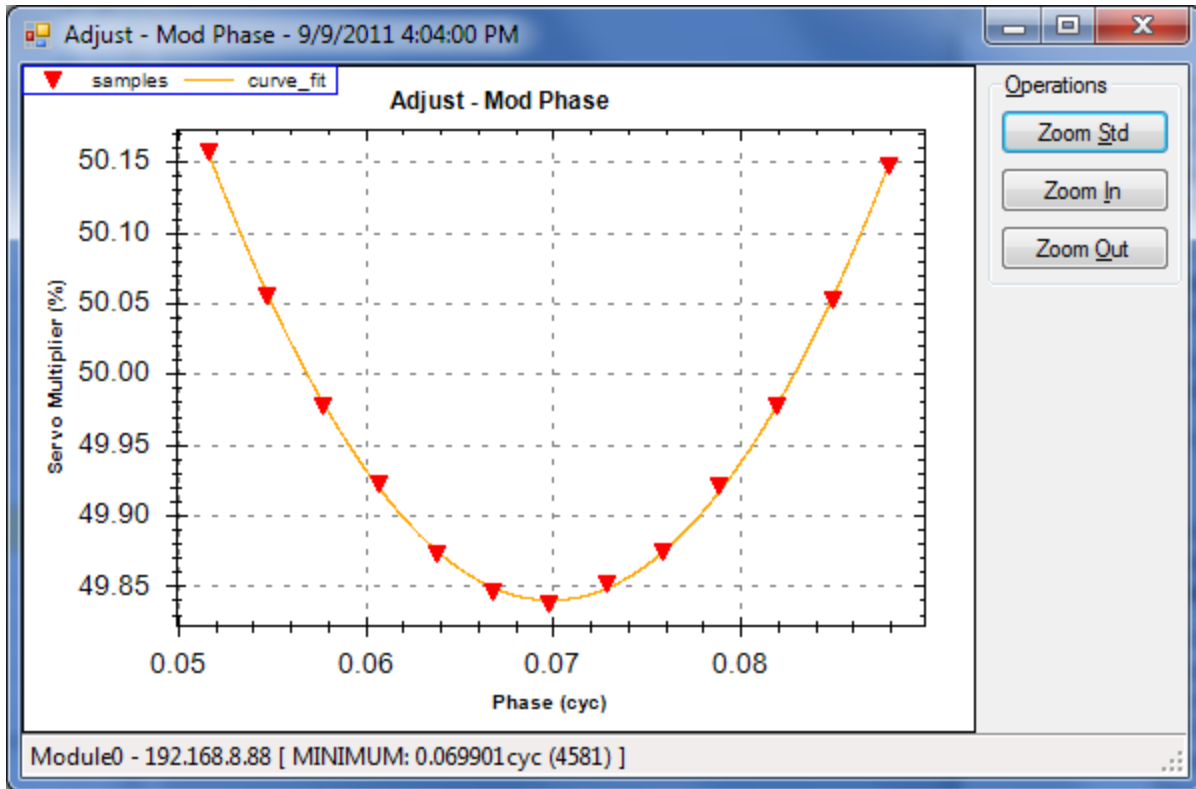
A 'Run' button is located at the bottom right of the configuration area.

The phase sweep is setup using the **Sweep Samples** to designate the number of times to run the modulation amplitude servo, the **Sweep Increment** to specify the fraction of a modulation cycle between each trial, the **Sample Delay** to wait for the modulation servo to settle before reading the endpoint, the **Sample Averaging** to compute the endpoint by averaging this number of points and the **Averaging Period** to specify the time period between each **Servo Multiplier** point in the average.

The **Sweep Center** is copied from the **Sine Table Phase** value in the modulation section and the **Sweep Range** is computed from the **Sweep Samples** and the **Sweep Increment**. The resulting **Run Time** is also computed and displayed.

The **Servo Gain** must be setup to match the acquisition parameters which can be verified with the Modulation Amplitude Mode.

The acquisition is started by clicking the **Run** box. The graph on the next page is the result.



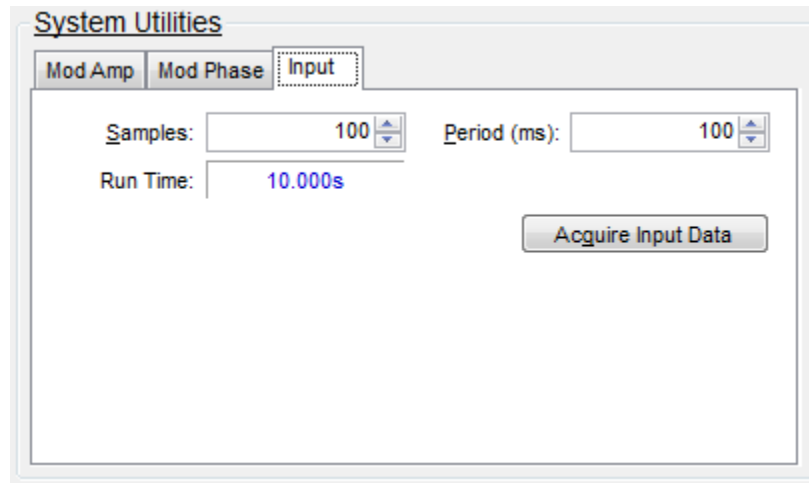
The Phase at which the minimum of this curve occurs is displayed at the bottom left of the graph box and found by least squares fitting a parabola to the data. This value should be entered into the **Sine Table Phase** box as the correct value to use.

This utility must be used with the amplitude utility to setup the modulation. The procedure is to use the ADC Data mode first to get a full cycle of modulation by changing the modulation amplitude and the modulation phase all with the modulation gain set to zero. The next step is to enter a non-zero servo gain (100 works well in most cases) and run the amplitude utility using its end value to update the servo multiplier field. Next, run the phase utility and enter the minimum phase into the sine table phase field. Then, go back and run the amplitude utility again and enter the new endpoint into the servo multiplier field. Finally, run the phase utility again and get a new sine phase value. Several cycles of this procedure may be needed to insure calibrated operation.

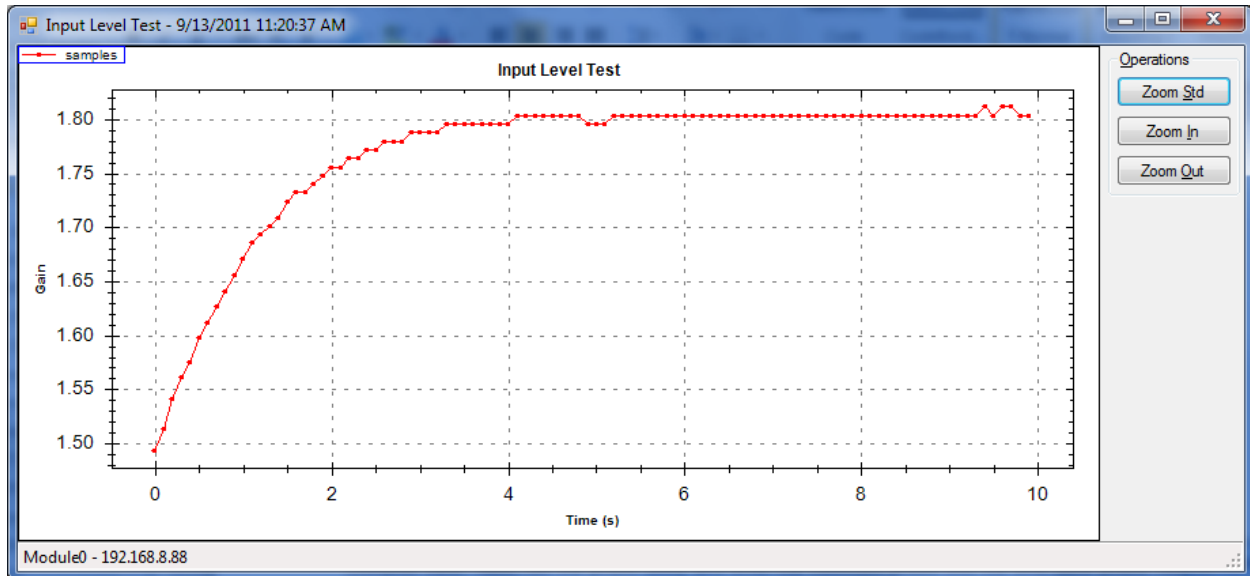
The modulation parameters change only if the modulation frequency is changed or the pulse phase is changed. Then the above procedure must be run again.

Input Level Utility

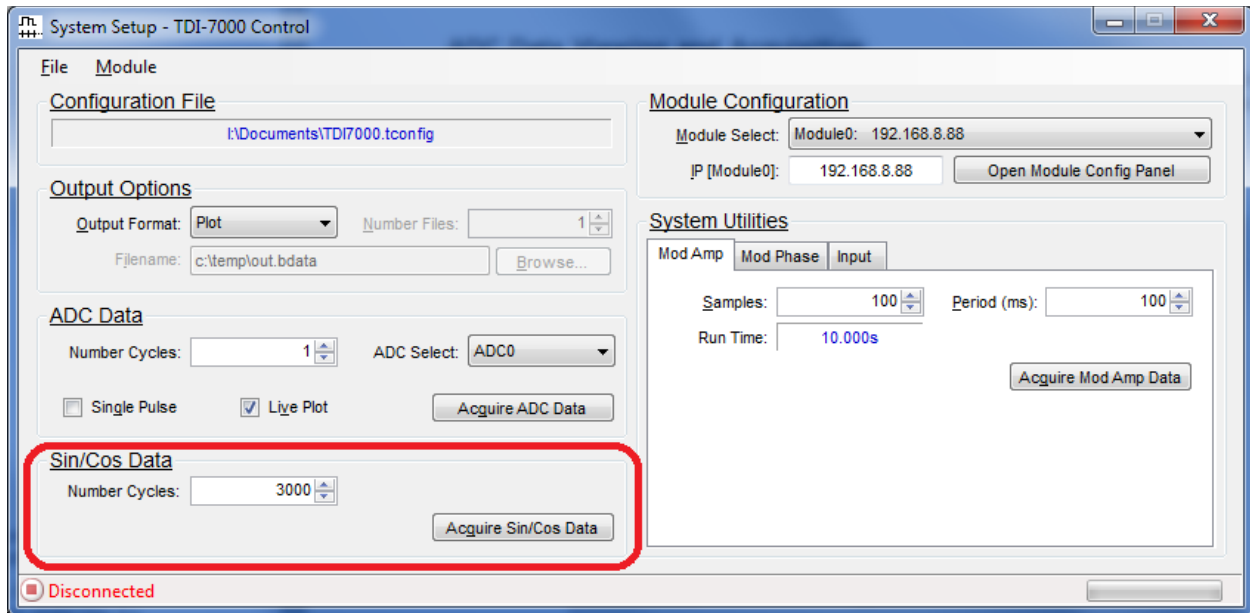
The **Input** mode reads the receiver gain level at **Period** time intervals for the number of **Samples** specified.



The output graph displays the receiver gain at a function of time. The input level servo closes on the input **Servo Setpoint** when the input level **Servo Gain** is non-zero. The graph below indicates that the initial non-interfering sensor 0 pulse height sum is too high for the specified **Servo Setpoint** since the receiver gain is being reduced to meet the **Servo Setpoint** requirement. The EDFA pump current or the input **Gain Level** can be reduced and the Input utility mode repeated until there is no change in the graph. This gives the correct gain setting.

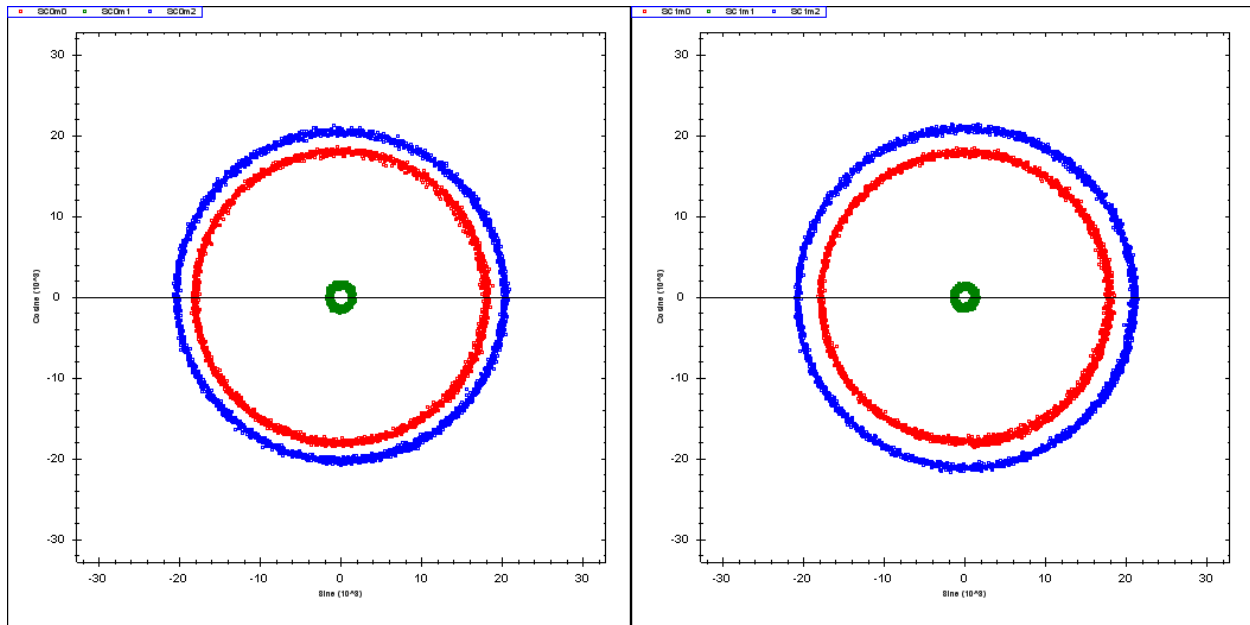


Sine / Cosine Data Viewing and Acquisition



When the TDI-7000 is set up correctly, the FPGA computes the Sine and Cosine of the phase of each sensor. Plotted against each other in a Lissajous graph, they should produce perfectly centered circles, one circle for each receiver when a sufficient sensor input signal is present. The Sin/Cos mode acquires the sine and cosine data from the FPGA and generates the Lissajous plot or writes the data to a file.

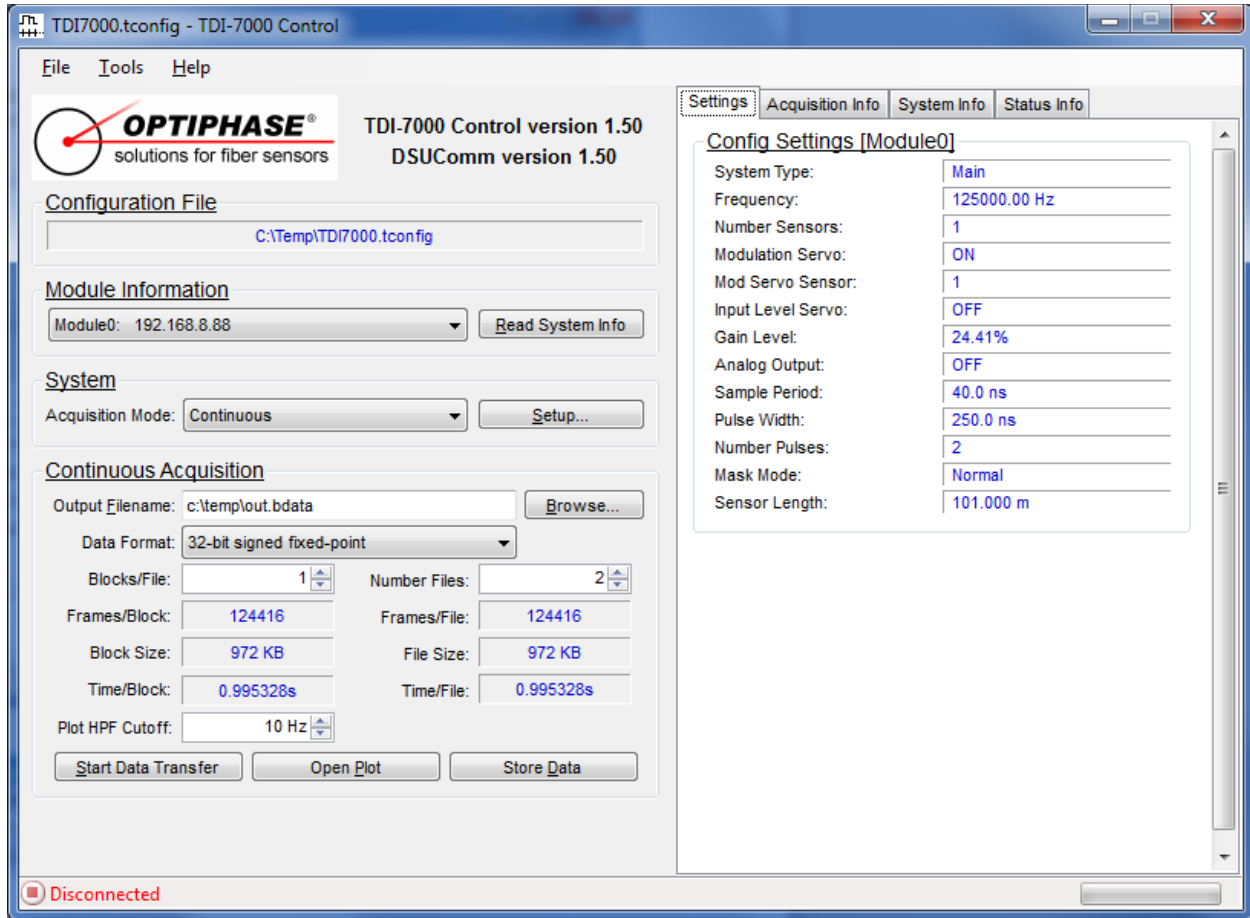
The number of modulation cycles to read from the FPGA is designated in the **Number Cycles** box and clicking the Acquire button starts the acquisition. The resulting plots are shown below.



There are two plots because the even slope demodulator is completely separate from the odd slope demodulator so there are actually six demodulators, two for each of the three receivers. The radius of each circle indicates the power on each receiver.

Phase Data Acquisition

The Main window's Phase Data tab collects the phase measurements of each sensor after the TDI-7000 has been configured properly. Each sample is a modulation cycle so includes the even and odd demodulator outputs for each sensor if pulses per cycle is set to two.

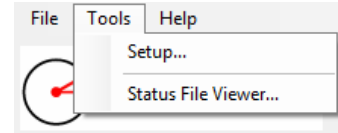


There are three acquisition modes:

Acquisition Mode	Description
Fixed	Fixed-length data acquisition
Continuous	Data is streamed to the host
Continuous (Triggered)	Data is streamed to the host and captured on the a trigger event

Tools menu

The Tools menu contains the following options:

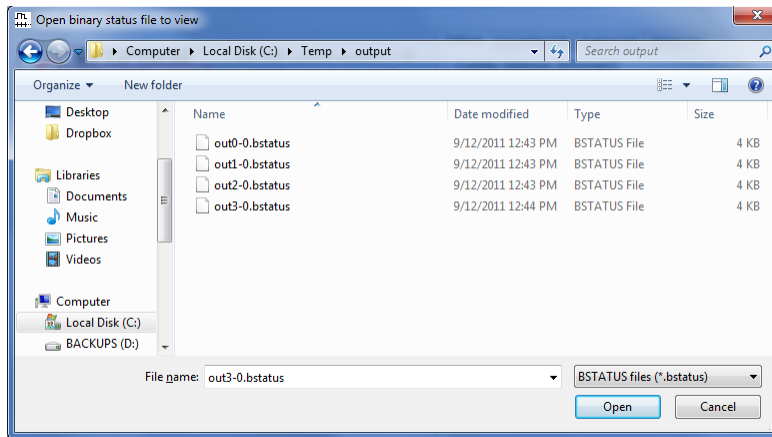


Setup

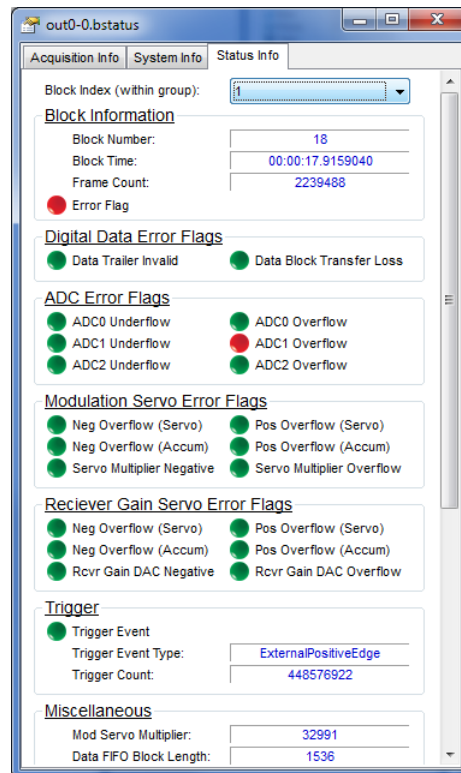
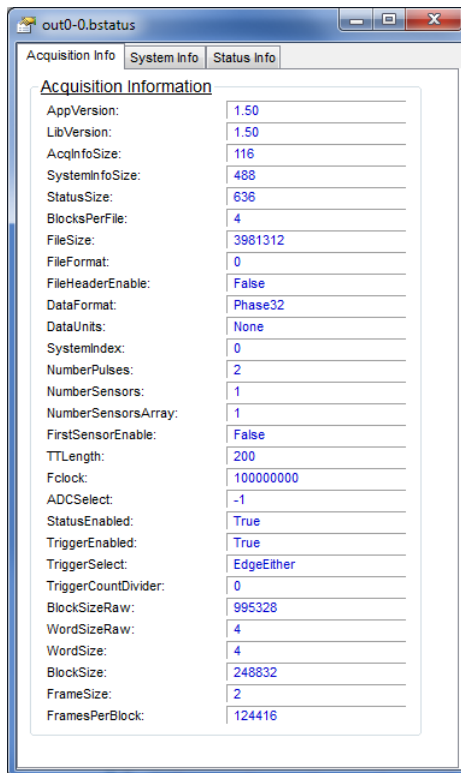
If **Setup** is clicked, the System Setup window appears to allow module configuration. The Setup button on the main window also takes you to the System Setup window.

Status File Viewer

If **Status File Viewer** is clicked, a dialog box which allows selection of a saved status file pops up:



After selection of a status file, the status file viewer loads, and all the status information in the file is presented in the same fashion as during live acquisition:



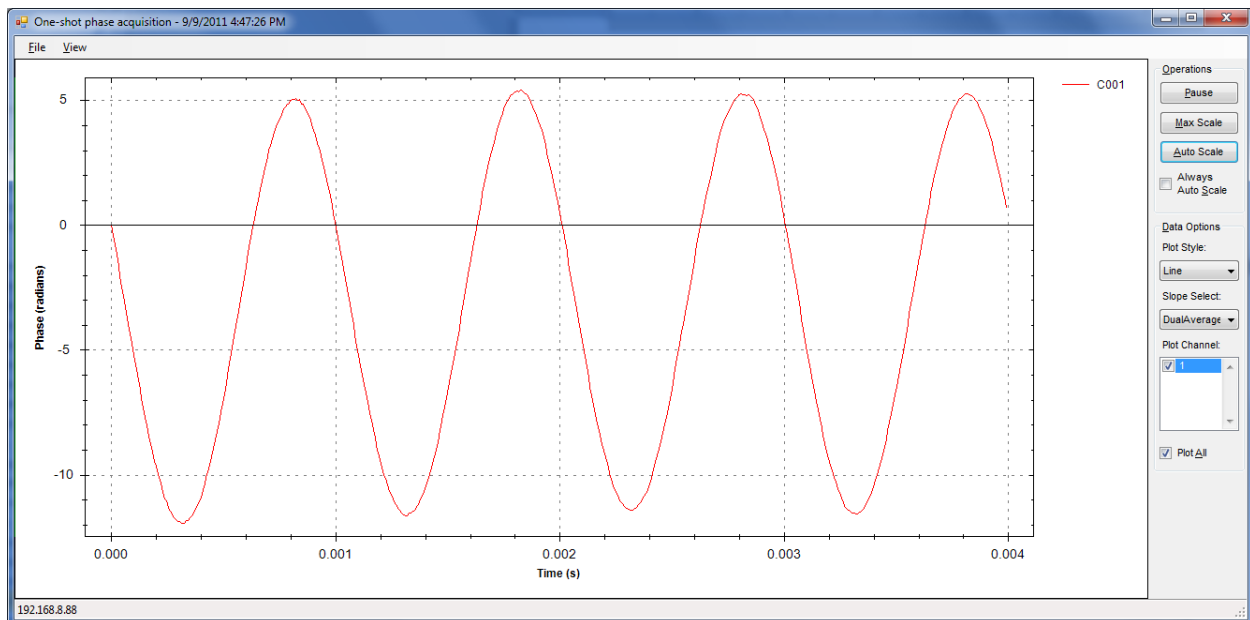
Fixed Data Acquisition

Reads a fixed amount of samples (**Number Frames**) and sends to selected **Output Format**. The **Output Format** can be set to the plotting utility, a two's complement binary integer file or a spreadsheet CSV format file. This mode is limited to the amount of RAM in the client PC.

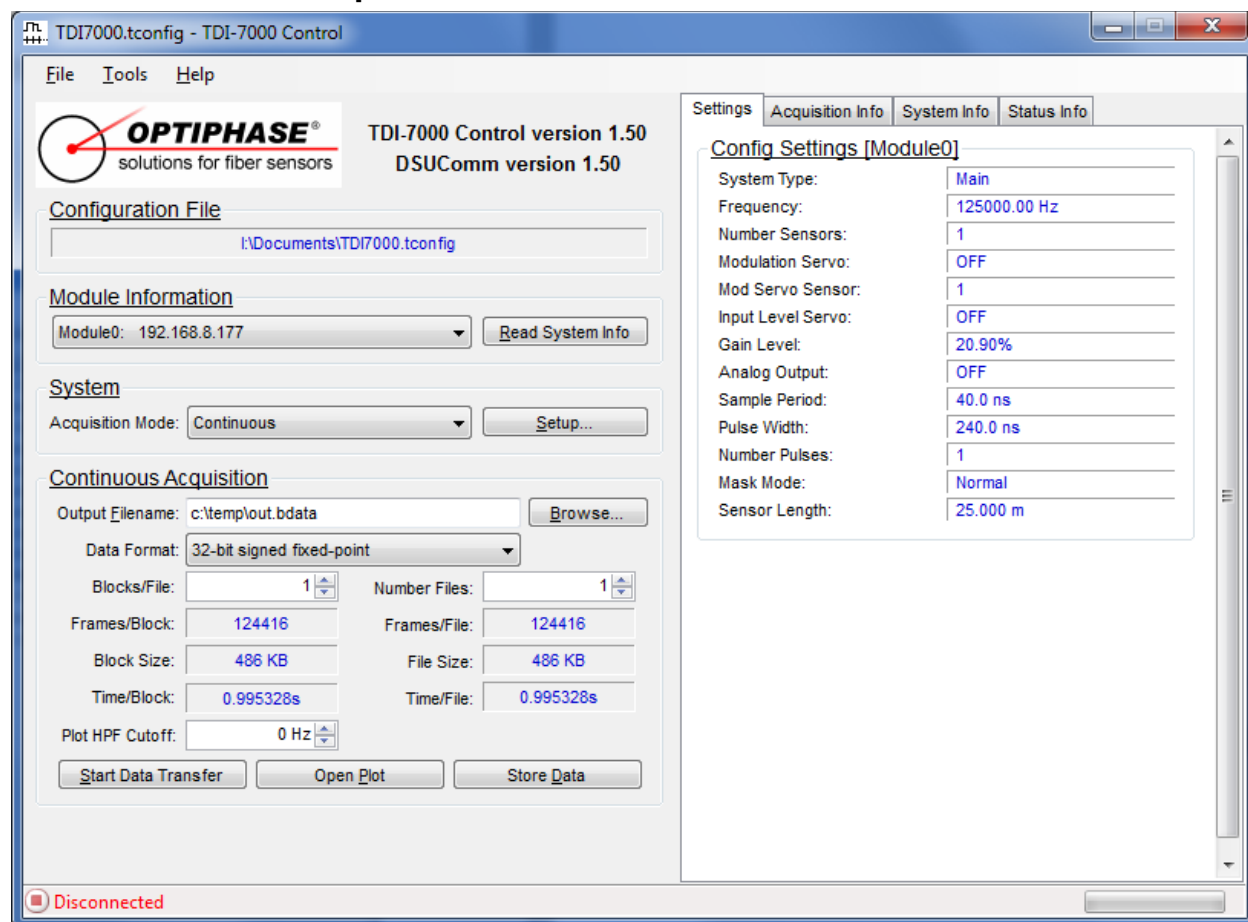
The number of files, **Number Files** can be specified for multiple acquisitions. A file index is added to the filename of each file appended to the end to mark multiple files.

Refer to section DATA FORMAT for information regarding the organization of saved data.

The following screenshot is an example of a one-shot phase acquisition with Plot output selected.



Continuous Data Acquisition



Real-time acquisitions are performed in a block-by-block basis. The block size will vary depending on configuration parameters, such as number of sensors.

To perform a data acquisition or to plot data real-time, data transfer must first be started. Click **Start Data Transfer** to begin the process. To view real-time data coming in, click **Open Plot**. Upon the start of the data transfer, the Acquisition Info and System Info tabs are populated with their corresponding information for the transfer. The Status Info tab contains a live view of real-time health (status) of the system.

To store data to the selected **Output Filename**, click **Store Data**. The **Block Size** is displayed and its size depends on the configuration. You may select the number of blocks stored in each file (**Blocks/File**) and the number of files saved to disk (**Number Files**). For each file, a file index number is appended to the filename (before the file extension).

Note: Storing data to disk and viewing the Real-time plot may be performed concurrently.

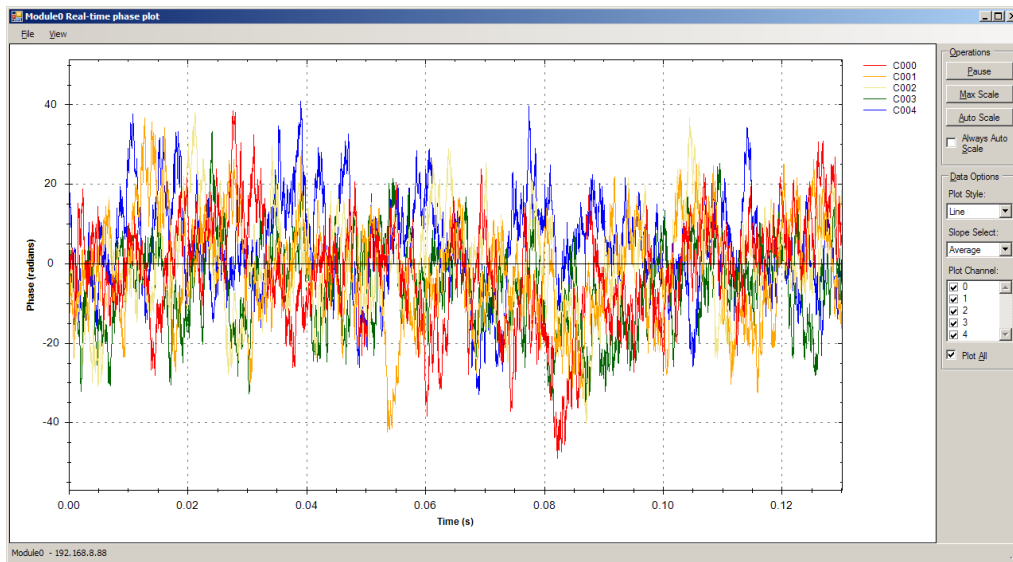
The following table details the data formats supported:

Data Format	Description
32-bit signed fixed point	Uses Angle32 data format ³
64-bit signed fixed point	Uses Angle64 data format ³

³ Refer to section *DIGITAL DATA* for a description of the format.

TDI-7000 TDM Fiber Interrogator

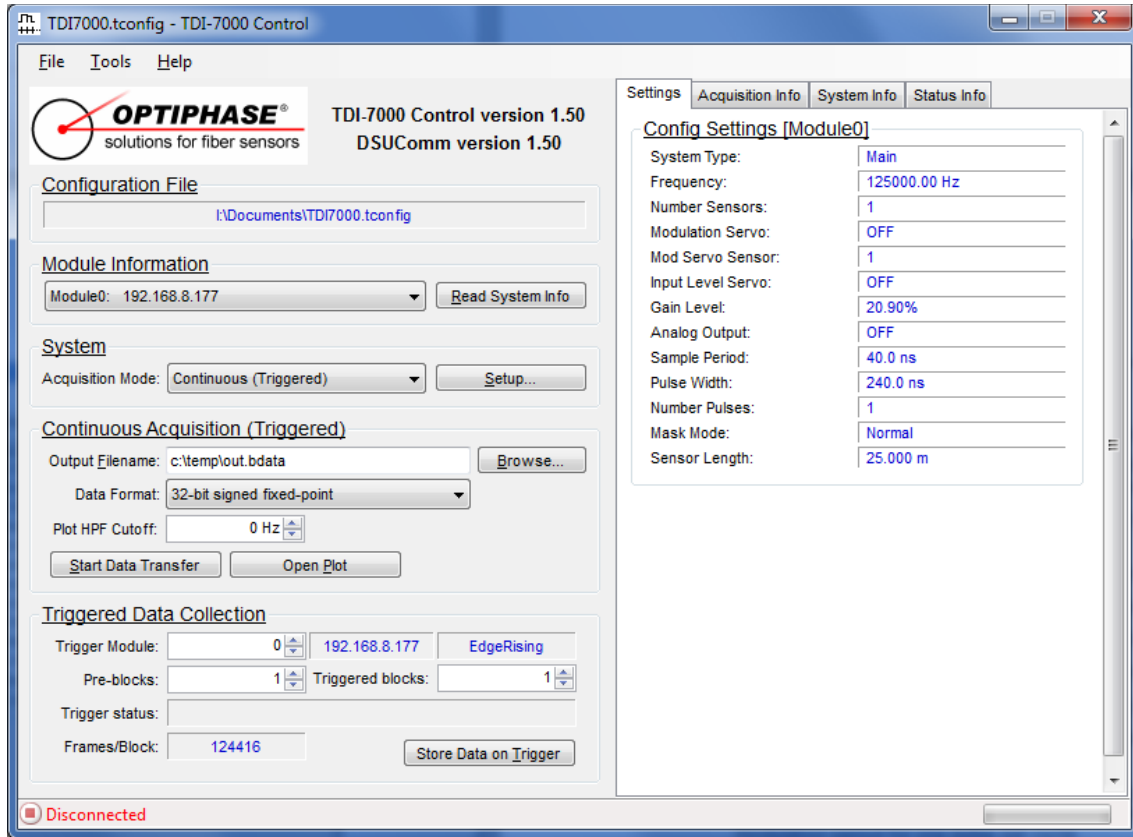
The following screenshot is an example of the real-time plot in action:



Status Information

Status information is saved along with each data block. This status information is viewable through the status info tab:

Continuous Triggered Data Acquisition



The continuous triggered data acquisition invokes the continuous acquisition mode but does not save data until the occurrence of a trigger event. The data capture includes the requested number of Pre-blocks before the trigger event and then the requested number of Triggered block after the trigger event. The plotting functionality is available in this mode.

Store Data On Trigger begins the acquisition which then waits for a trigger event. It is necessary to inspect the block status to find the position of the trigger event.

OPERATION

Requirements

For proper operation of the TDI-7000, the following conditions must be met.

- 1 The Laser temperature must be set so that the source is at the proper wavelength that matches the sensor reflectors and the source power is sufficient to interrogate the array.
- 2 The PZ1 radian per volt response must support the modulation frequency at the desired ADC sample rate
- 3 The optical pulse width must work with the sensor lengths
- 4 The optical pulse power level cannot cause saturation of the receiver
- 5 The five ADC strobes must latch each sensor return
- 6 The modulation sine amplitude must give a $\pi/2$ modulation separation of the samples
- 7 The midpoint of the optical pulse must be at the zero crossing of the radian modulation sine
- 8 The number of sensors read must be less than or equal to the number of sensors in the array.
- 9 The array optical return must be short enough to fit into the time between the sensor measurements which is either a half modulation cycle period for two pulses or a full modulation period for one pulse per cycle.

Laser Wavelength

The laser wavelength range is determined by the source that is installed in the TDI-7000. The nominal wavelength is usually specified to be an entry in the ITU grid. The standard wavelengths are shown in the table below.

ITU	λ nm	THz	ITU	λ nm	THz
1	1577.03	190.10	37	1547.72	193.70
2	1576.20	190.20	38	1546.92	193.80
3	1575.37	190.30	39	1546.12	193.90
4	1574.54	190.40	40	1545.32	194.00
5	1573.71	190.50	41	1544.53	194.10
6	1572.89	190.60	42	1543.73	194.20
7	1572.06	190.70	43	1542.94	194.30
8	1571.24	190.80	44	1542.14	194.40
9	1570.42	190.90	45	1541.35	194.50
10	1569.59	191.00	46	1540.56	194.60
11	1568.77	191.10	47	1539.77	194.70
12	1567.95	191.20	48	1538.98	194.80
13	1567.13	191.30	49	1538.19	194.90
14	1566.31	191.40	50	1537.40	195.00
15	1565.50	191.50	51	1536.61	195.10
16	1564.68	191.60	52	1535.82	195.20
17	1563.86	191.70	53	1535.04	195.30
18	1563.05	191.80	54	1534.25	195.40
19	1562.23	191.90	55	1533.47	195.50
20	1561.42	192.00	56	1532.68	195.60
21	1560.61	192.10	57	1531.90	195.70
22	1559.79	192.20	58	1531.12	195.80
23	1558.98	192.30	59	1530.33	195.90
24	1558.17	192.40	60	1529.55	196.00
25	1557.36	192.50	61	1528.77	196.10
26	1556.55	192.60	62	1527.99	196.20
27	1555.75	192.70	63	1527.22	196.30
28	1554.94	192.80	64	1526.44	196.40
29	1554.13	192.90	65	1525.66	196.50
30	1553.33	193.00	66	1524.89	196.60
31	1552.52	193.10	67	1524.11	196.70
32	1551.72	193.20	68	1523.34	196.80
33	1550.92	193.30	69	1522.56	196.90
34	1550.12	193.40	70	1521.79	197.00
35	1549.32	193.50	71	1521.02	197.10
36	1548.51	193.60	72	1520.25	197.20

A further adjustment to the wavelength can be made by changing the operating temperature of the source using the **Laser Temp** entry box. The source wavelength and its tuning range are given at delivery along with the **Laser Temp** setting to give the ITU nominal wavelength. The source wavelength must work with the wavelength of the sensor reflectors which may be broadband mirrors in some cases. The optical amplifier must also work at the selected ITU wavelength. The default source is an ECL at ITU35. There is also an option to use a customer supplied external source, multiple sources at several wavelengths or a DFB with an adjustable wavelength.

Modulation and Sample Frequencies

The modulation frequency F_m and the sample frequency F_s must satisfy the relation:

$$\frac{F_s}{F_m} \cdot Z(F_m) \cdot \frac{1.3}{1.9} \leq 160 \text{ volts}$$

where the compensator PZ1 modulator scale factor is $Z(F_m)$ as a function of the modulation frequency F_m , as given in the table below. These are only approximate values.

F_m KHz	Z	F_m KHz	Z	F_m KHz	Z	F_m KHz	Z
1	0.144	43	0.063	85	0.298	129	0.156
2	0.143	44	0.059	86	0.323	130	0.140
3	0.143	45	0.055	87	0.349	131	0.126
4	0.143	46	0.051	88	0.378	132	0.111
5	0.142	47	0.048	89	0.409	133	0.097
6	0.142	48	0.043	90	0.445	134	0.085
7	0.140	49	0.039	91	0.487	135	0.072
8	0.139	50	0.034	92	0.533	136	0.060
9	0.138	51	0.030	93	0.586	137	0.048
10	0.138	52	0.025	94	0.654	138	0.037
11	0.137	53	0.020	95	0.719	139	0.025
12	0.136	54	0.015	96	0.820	140	0.015
13	0.136	55	0.010	97	0.953	141	0.007
14	0.134	56	0.004	98	1.103	142	0.011
15	0.132	57	0.001	99	1.313	143	0.020
16	0.132	58	0.007	100	1.644	144	0.031
17	0.130	59	0.013	101	2.146	145	0.041
18	0.129	60	0.019	102	3.036	146	0.051
19	0.127	61	0.025	103	5.570	147	0.062
20	0.124	62	0.031	106	3.938	148	0.073
21	0.123	63	0.038	107	2.408	149	0.084
22	0.121	64	0.045	108	1.792	150	0.095
23	0.120	65	0.052	109	1.427	151	0.106
24	0.119	66	0.059	110	1.153	152	0.119
25	0.116	67	0.067	111	0.977	153	0.130
26	0.113	68	0.075	112	0.854	154	0.143
27	0.112	69	0.083	113	0.736	155	0.158
28	0.110	70	0.092	114	0.647	156	0.171
29	0.106	71	0.101	115	0.583	157	0.186
30	0.104	72	0.110	116	0.525	158	0.202
31	0.102	73	0.120	117	0.475	159	0.220
32	0.099	74	0.130	118	0.431	160	0.239
33	0.096	75	0.142	119	0.394	161	0.261
34	0.093	76	0.153	120	0.360	162	0.285
35	0.091	77	0.166	121	0.328	163	0.315
36	0.088	78	0.181	122	0.297	164	0.347
37	0.084	79	0.191	123	0.274	165	0.385
38	0.081	80	0.205	124	0.251	166	0.435
39	0.078	81	0.221	125	0.228	167	0.507
40	0.074	82	0.237	126	0.209	168	0.582
41	0.071	83	0.257	127	0.191	169	0.705
42	0.067	84	0.275	128	0.174		

Optical Pulse Width

The time taken for the optical pulse to transit the sensor length and reflect back to the start of the sensor sets the maximum optical pulse width. The maximum optical pulse width: T , can be calculated from the sensor length: L , by the relation:

$$T \leq 9.78677 \cdot L \quad ns$$

where the sensor length is expressed in meters. The convention is to design the sensor length to give integer maximum pulse width times expressed in nanoseconds. A 1000ns maximum width pulse would correspond to a sensor length of 102.179 meters.

The maximum pulse width must also accommodate the five sensor samples including the pulse rise and fall times. This means that the sample frequency must be such that:

$$\frac{4}{F_s} \cdot 10^9 + 40 \leq T \quad ns$$

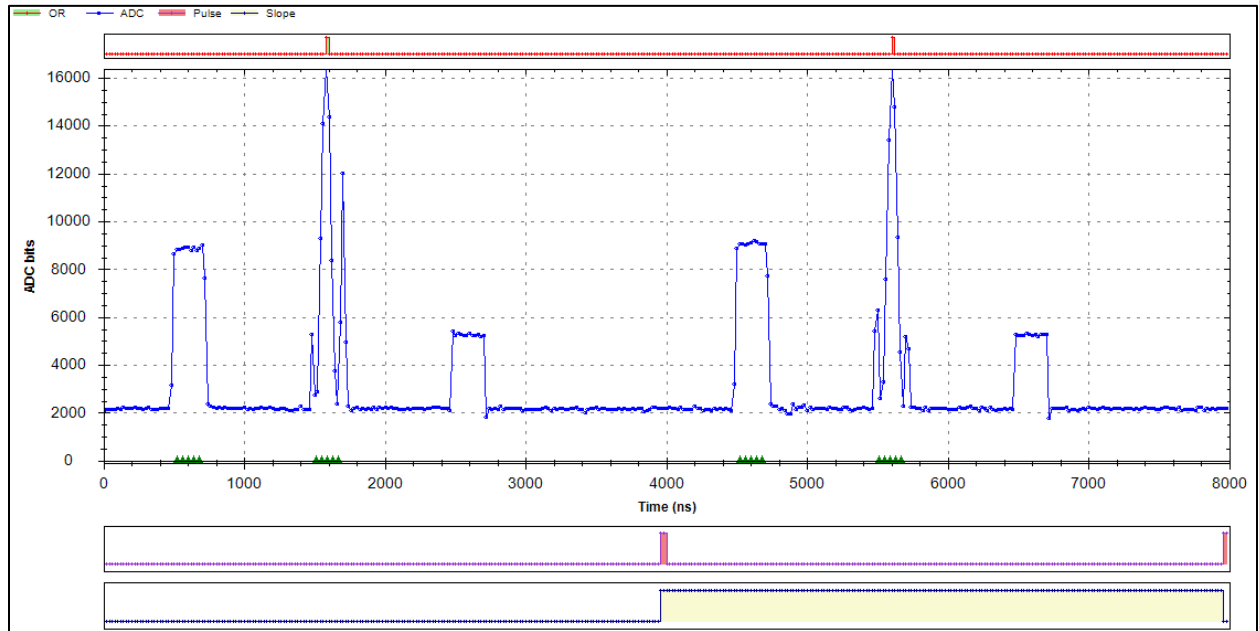
The optical width also determines the minimum distortion which is set by the linear region of the sine modulation at its zero crossing. A shorter pulse relative to the modulation period means higher sample frequencies and higher linearity. A measure of the distortion is given by the sample cycle fraction which is the time interval between the first sample and the fifth sample divided by the modulation period.

$$cycle \ fraction = \frac{4 \cdot F_m}{F_s} \cdot 100 \quad \%$$

The maximum cycle fraction allowed depends upon how much distortion the application can tolerate. A cycle fraction of 16% keeps the total harmonic distortion below 0.1%

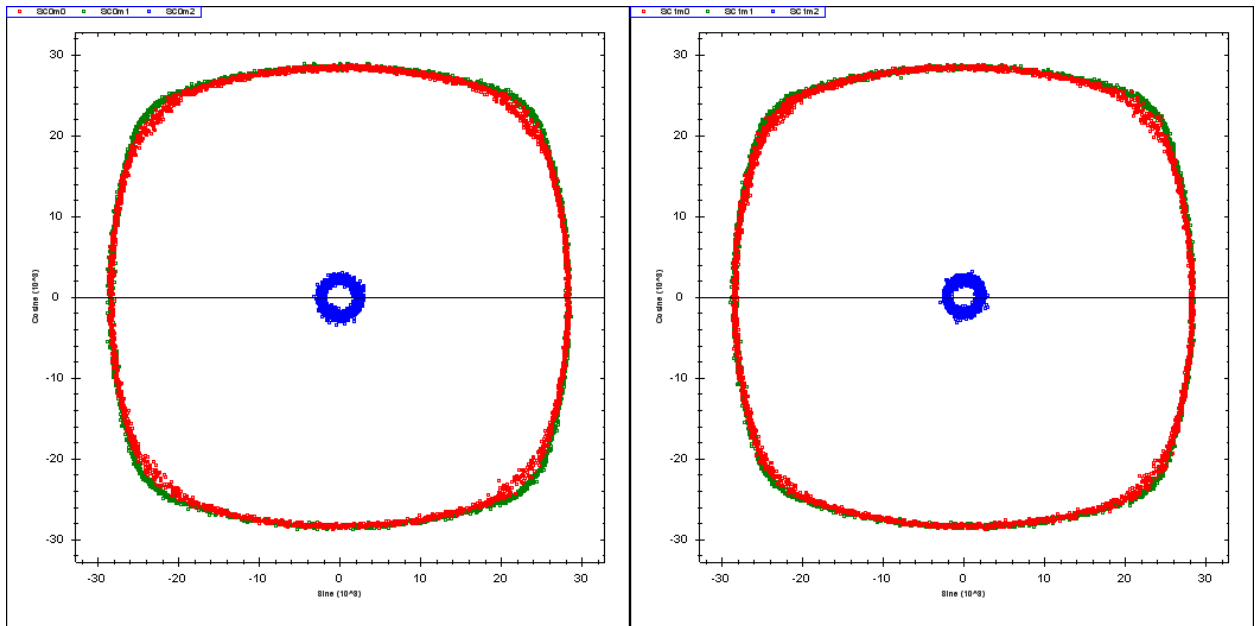
Receiver Saturation

The ADC Data plot displays the receiver output from a single mask or their sum. Too much optical power saturates the ADC input and results in an unusable sensor reading as shown below by the appearance of the ADC overflow bit active at the top of the graph.



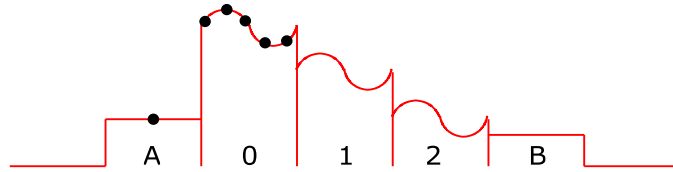
The maximum value of a displayed ADC reading is 16384 bits.

Receiver saturation also shows up in the sin/cos plot as a distorted circle as shown below.



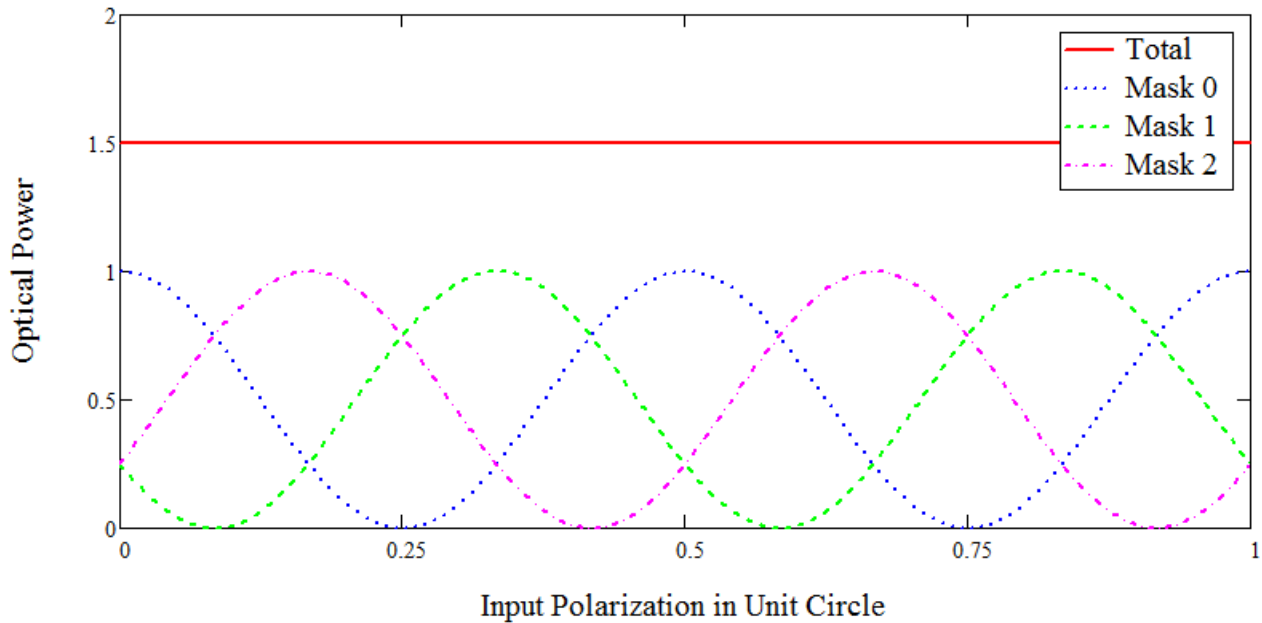
The system optical gain must be adjusted to prevent ADC saturation and this requires looking at all three masks or the sum. The ADC Data SUM mode is useful for setting this system gain.

The two pulses from the compensator reflect from the sensors and produce a pulse train consisting of the first non-interfering A pulse by itself, then each sensor interferometer pulse and finally the last non-interfering B pulse by itself. A mask return is diagrammed below.



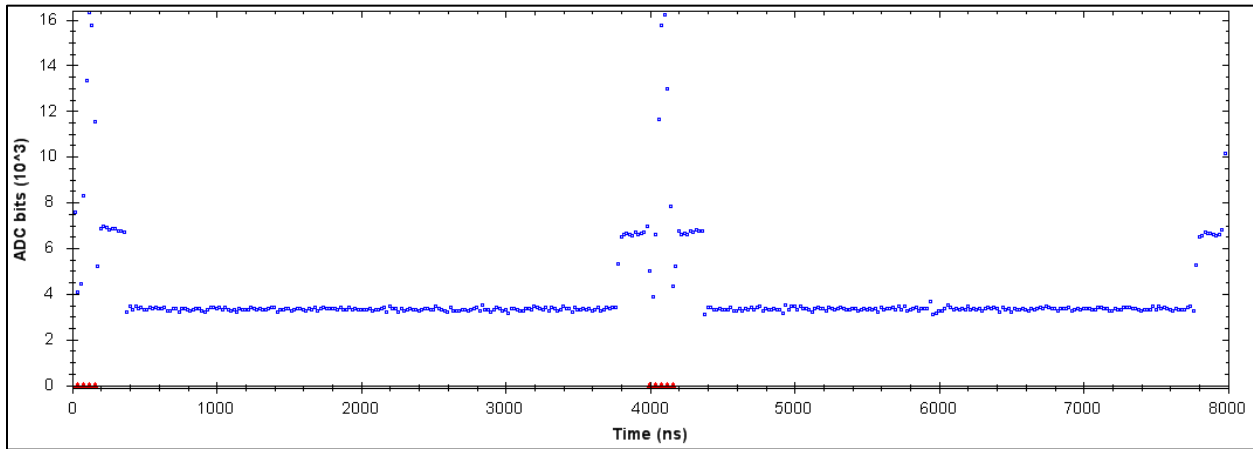
The level of the first A pulse return measures the power level of the first pulse from the compensator. The sum of the first pulse return from all three masks is shown below for equal A and B pulses.

First Non-Interfering Pulse Power Total and each Mask

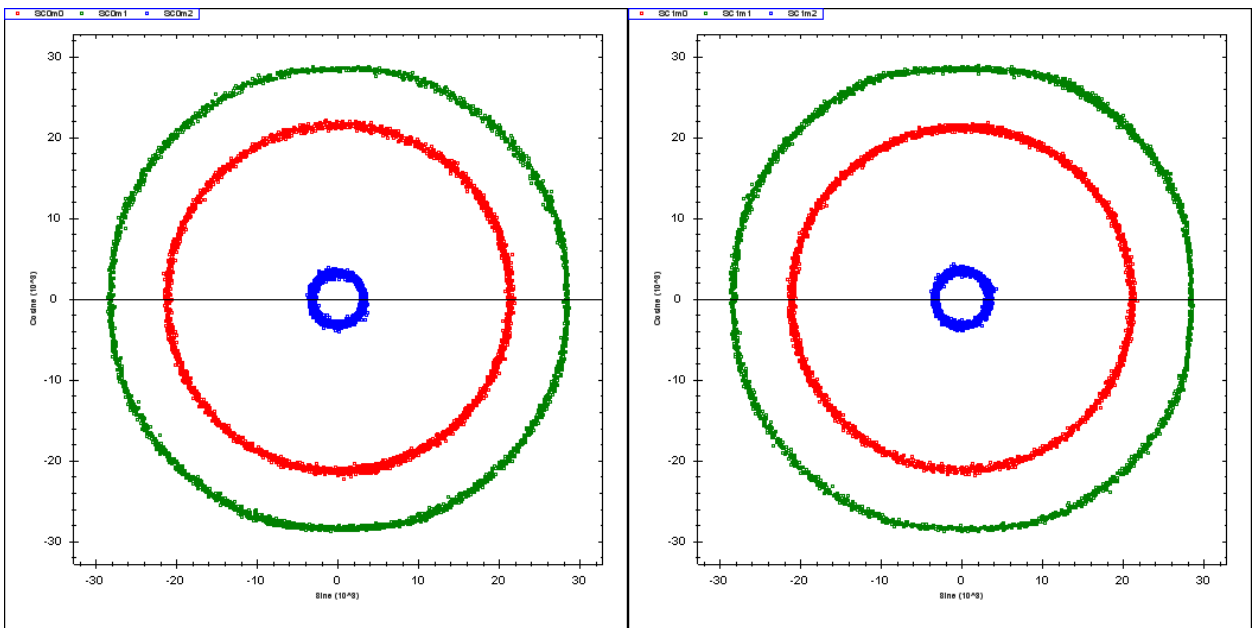


The graph shows the power in each mask for the A pulse and the sum of all the masks as the linear polarization of the A optical pulse goes around the PDR. Even though the power in each mask varies, the sum is constant at 1.5. The maximum interference power is 6 so maintaining the sum at 25% of the ADC full scale minus the zero light level would insure that no sensor return ever saturates the ADC. For elliptical polarization states the visibility can decrease.

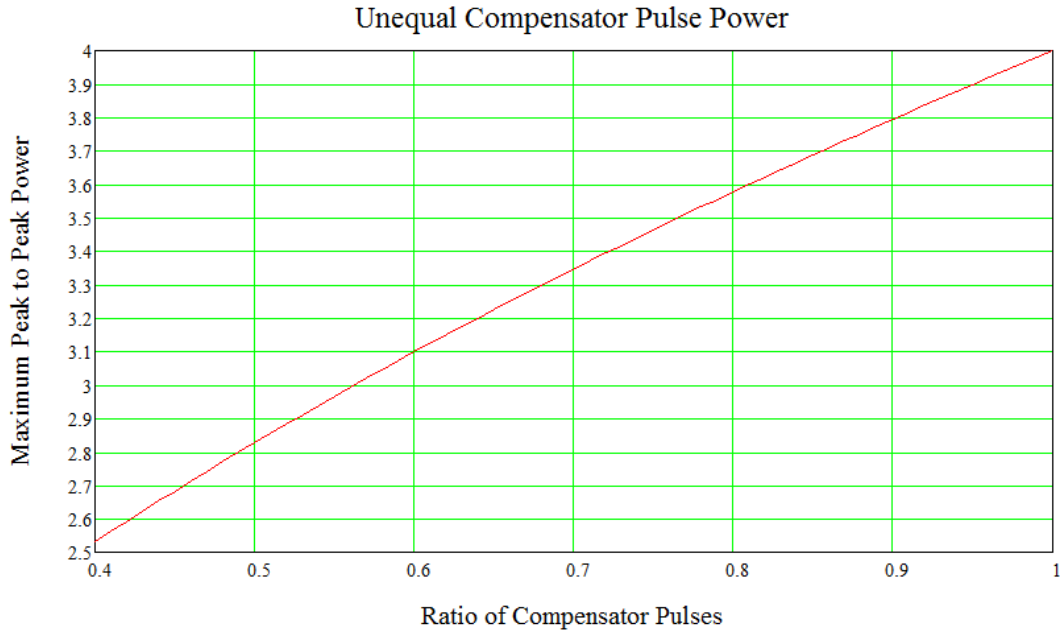
The SUM ADC Data acquisition below shows near saturation optical power. The zero light level is at 3353 bits and A pulse sum is at 6643 bits so the A sum pulse height above the zero light level is 3290 bits. The full scale is 16384 minus 3353 which is 13031 so that the A sum pulse height is 3290/13031 or 25.2% of full scale.



The sine/cosine plot shows that the receiver is still slightly saturated so it is best to use a lower value than the maximum of 25%.



It is possible that the A and B compensator pulses may not have the same optical level. In that case the graph below shows the effect the peak to peak sensor power for best alignment between the A and B pulses.



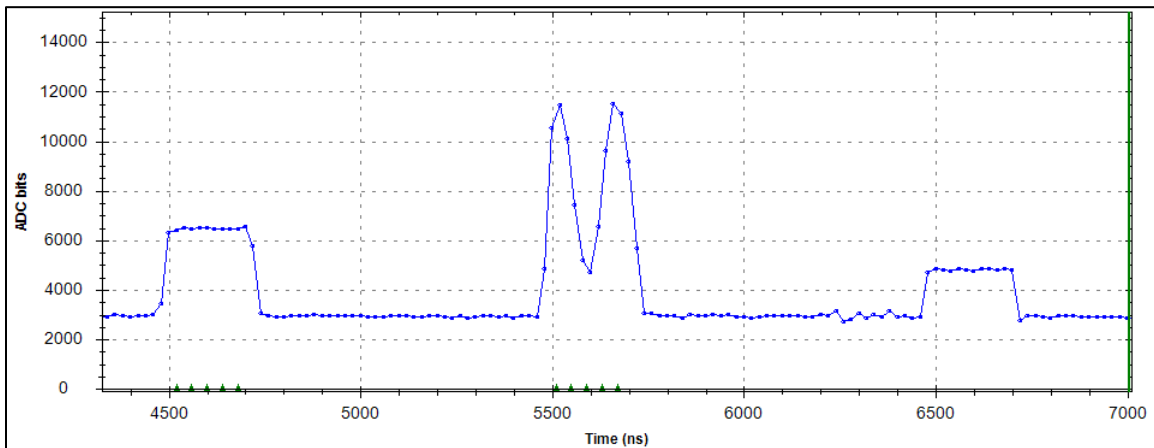
Since the maximum peak to peak power decreases from the ideal value of four, the target ADC fill level can be increased to compensate for the lower interferometer power.

The optical power level can be changed by adjusting the EDFA current. The receiver gain level can be adjusted by changing the **Gain Level** or **Gain Adjust** values. Noise is optimized when the receiver level is low and the optical power is high.

Checking the **Input Level Servo Enable** box maintains the non-interfering pulse total power level to the set value but this cannot be used until the modulation and ADC samples are correct.

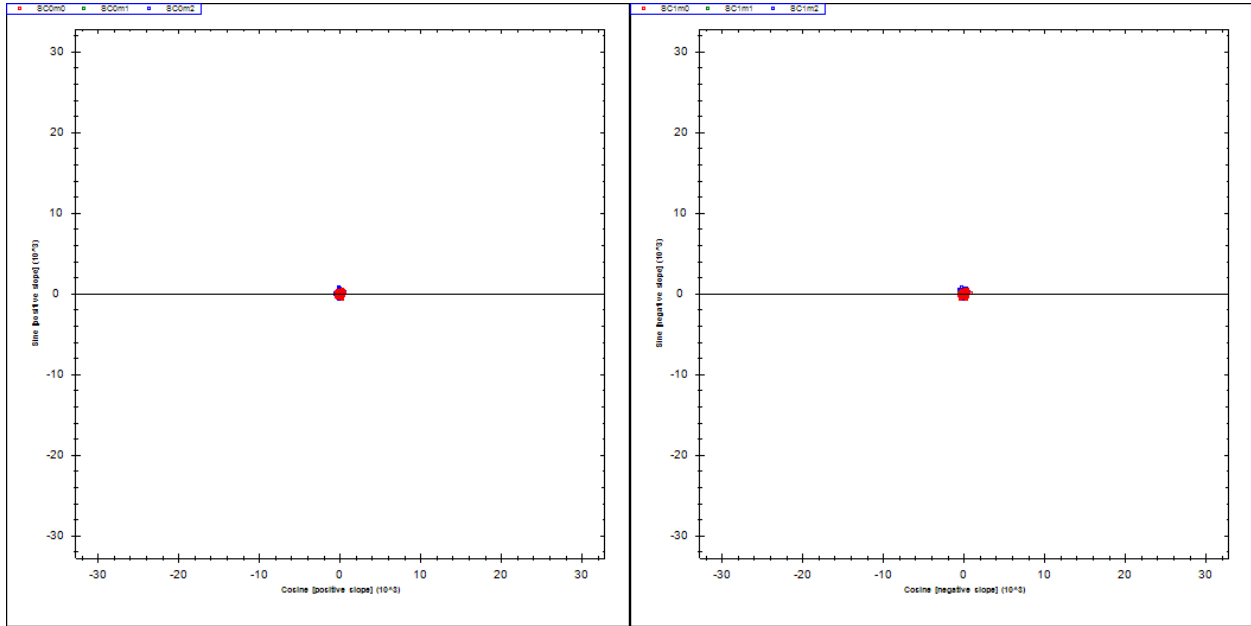
ADC Samples

The graph below shows the ADC samples along with a single sensor return.

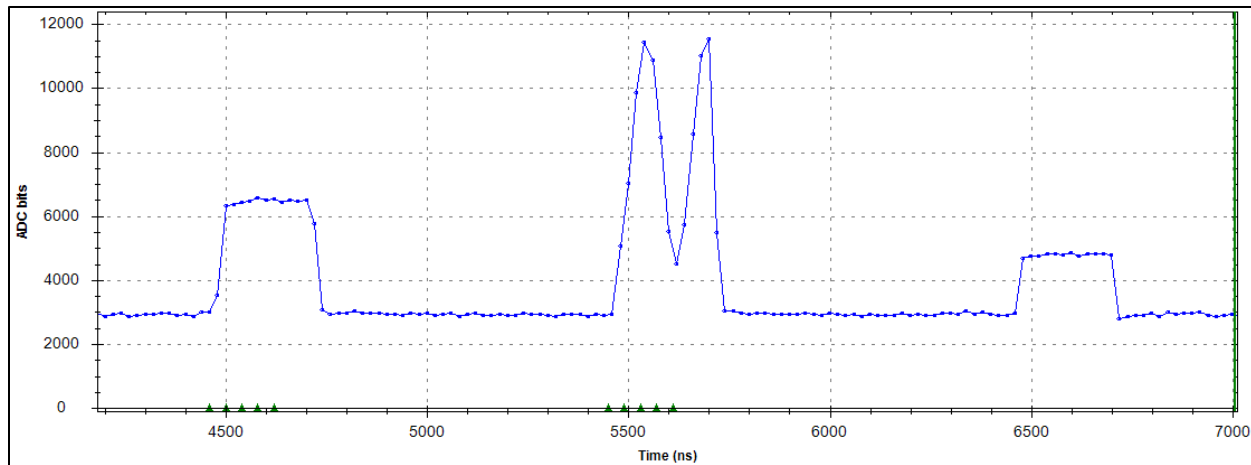


This is the correct ADC sample alignment as the first five ADC samples are on the top of the first non-interfering return and five more are aligned with the sensor 1 return.

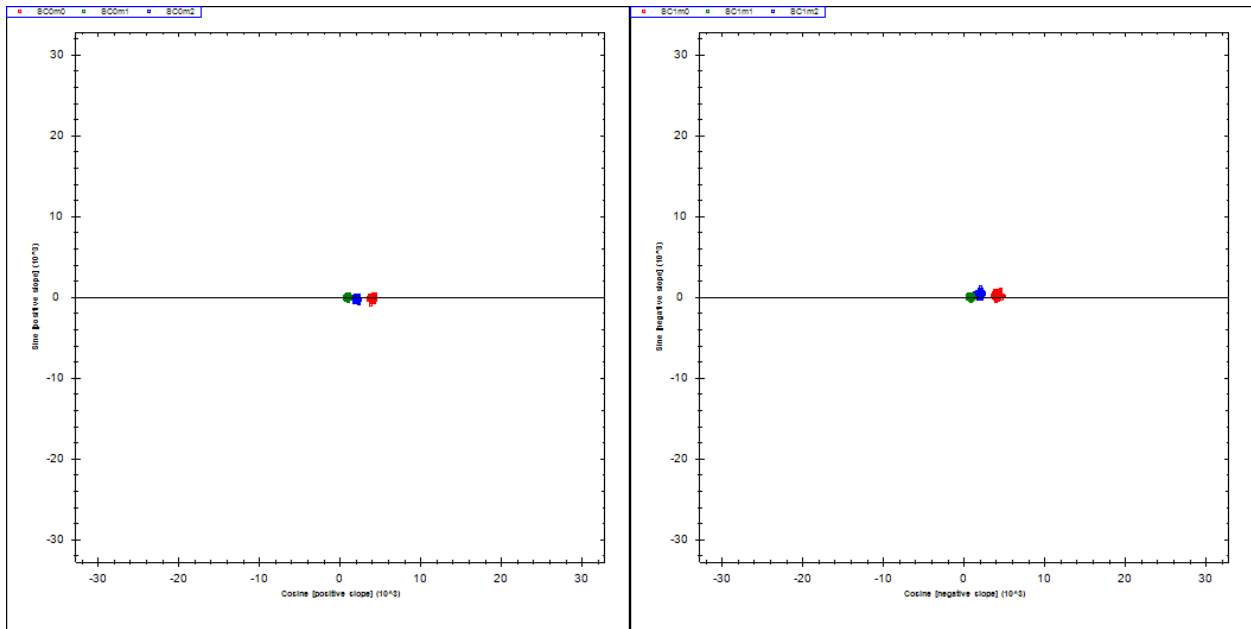
The sine/cosine graph shows a point in the center for sensor 0 (non-interfering return) because all the samples are the same except for noise so their sum and difference in the sine/cosine calculation just returns the noise of the ADC samples.



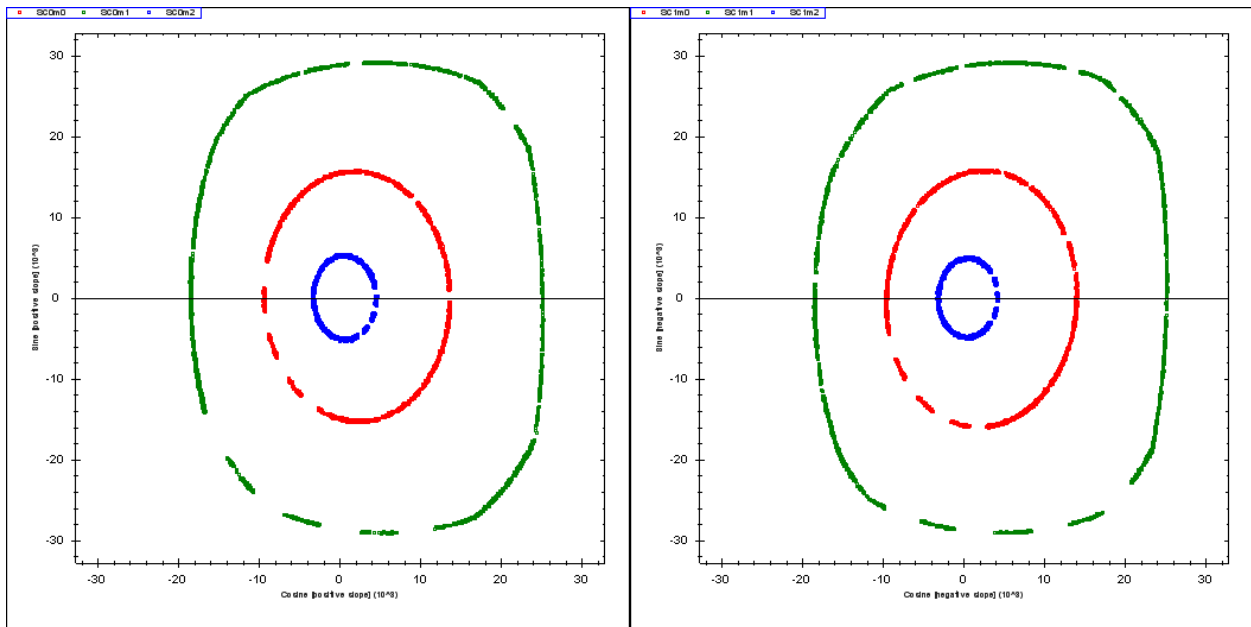
Moving the ADC sample positions by 60ns towards the start of the sensor 0 pulse shows that the first sample is no longer on the flat top.



The sine/cosine graph now shows a line instead of a point since the first sample introduces an offset into the sine/cosine calculation.



The position of the ADC samples can thus be verified looking at the sine/cosine plot of sensor 0 which must be a blurry point when the ADC samples are in the correct position. The resulting Sin/Cos plot of sensor 1 is shown below which no longer produces circles.

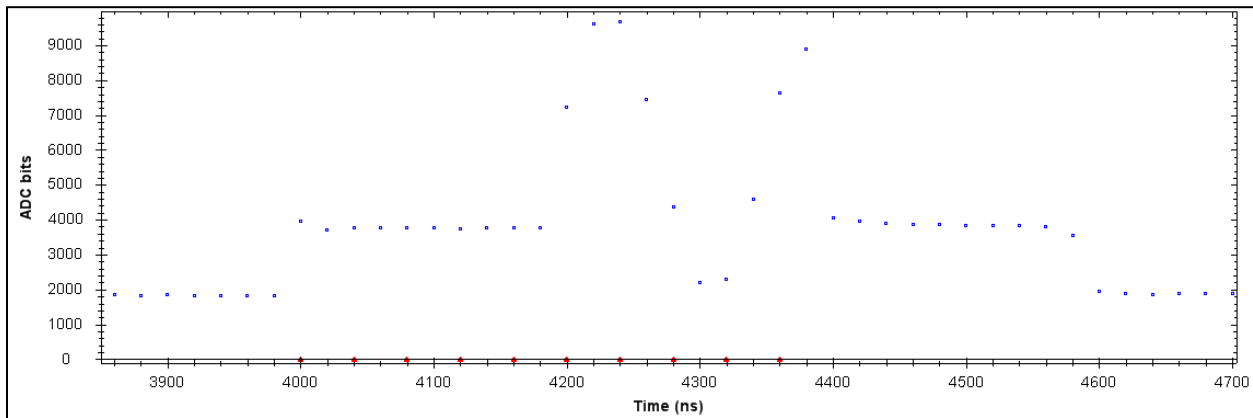


The TDI system works by placing the ADC samples in a fixed location in the timing table. The position of the ADC samples relative to the sensor return is adjusted by moving the pulse generation position in the timing table. Therefore, decreasing the pulse phase moves the sensor pulse to the left on the graph.

It is often better to turn off the modulation to when adjusting the ADC sample positions so that all the sensors return flat top pulses. This makes aligning the sensor samples easier.

Modulation Amplitude

The ADC Data plot below shows a correct modulation level where the last ADC sample has the same level as the first ADC sample and between them is one cycle of a sinusoid.

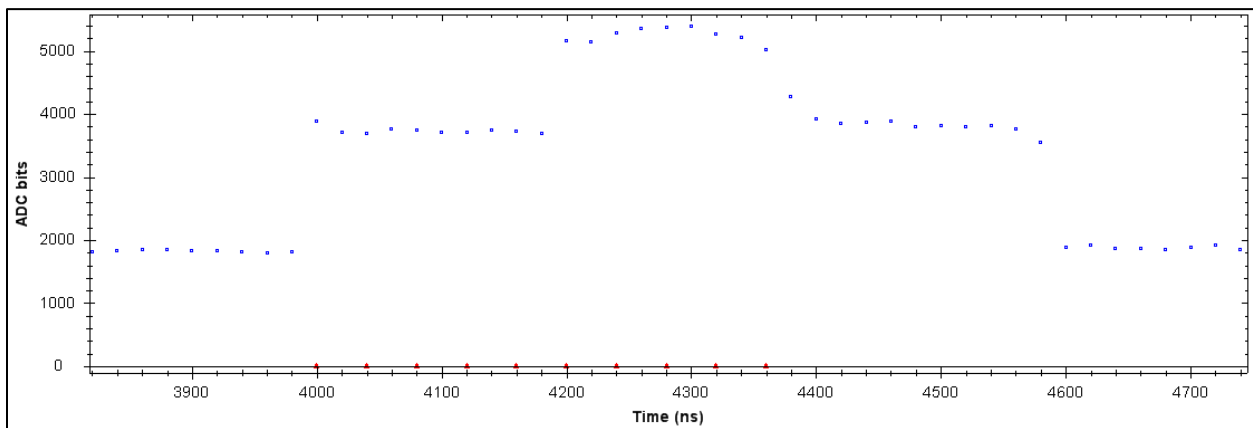


Getting to this correct state requires a course setting and then a fine setting of both the modulation amplitude and the modulation phase. This is because the correct modulation amplitude also requires that the compensator induced modulation occur at the zero crossing of the modulation sinusoid and this position is controlled by the modulation phase.

When starting with a new system or a new frequency, the starting modulation amplitude and phase must be searched for. This is done by setting the sine table amplitude to the 50% level and the sine table phase to zero and setting the modulation servo gain to zero. Then take repeated ADC Data plots of the highest power ADC while incrementing the sine table phase by about 0.05 cycle steps. The idea is to either maximize or minimize the induced modulation in the sensor. The maximum modulation occurs at the zero crossing and the minimum modulation occurs when the pulse is at the flat top of the modulation sinusoid.

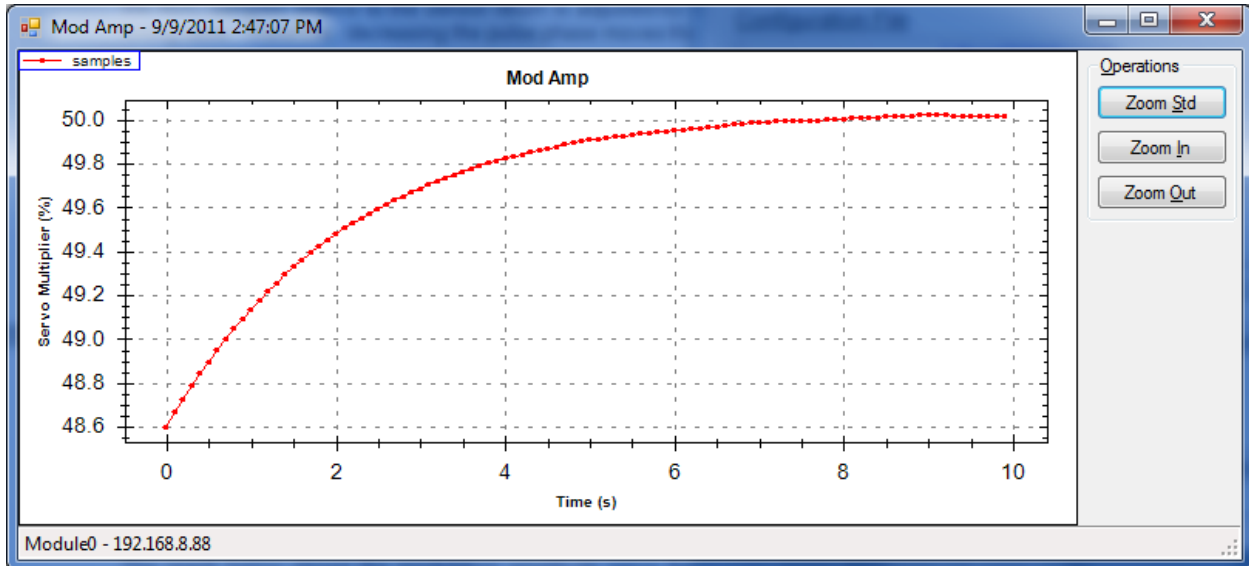
The graph below shows what happens to the graph above when the sine table phase is increased by 0.25 cycles so that the modulation is now on the peak of the sinusoid. The induced modulation is now a line and not sine as it should be. Once this sine table phase is found by minimizing the modulation, the correct phase is 0.25 cycles less or more than this value.

The next step once the modulation phase is approximately found is to change the modulation sine table amplitude until one cycle of a sinusoid appears in the sensor return pulse. Now, both the modulation table amplitude and phase are coarsely set.



A fine adjustment of the modulation sine table amplitude is made by setting the modulation servo gain to 30 and running the modulation acquisition mode **Mod Amp** utility with 100 samples with 100ms periods.

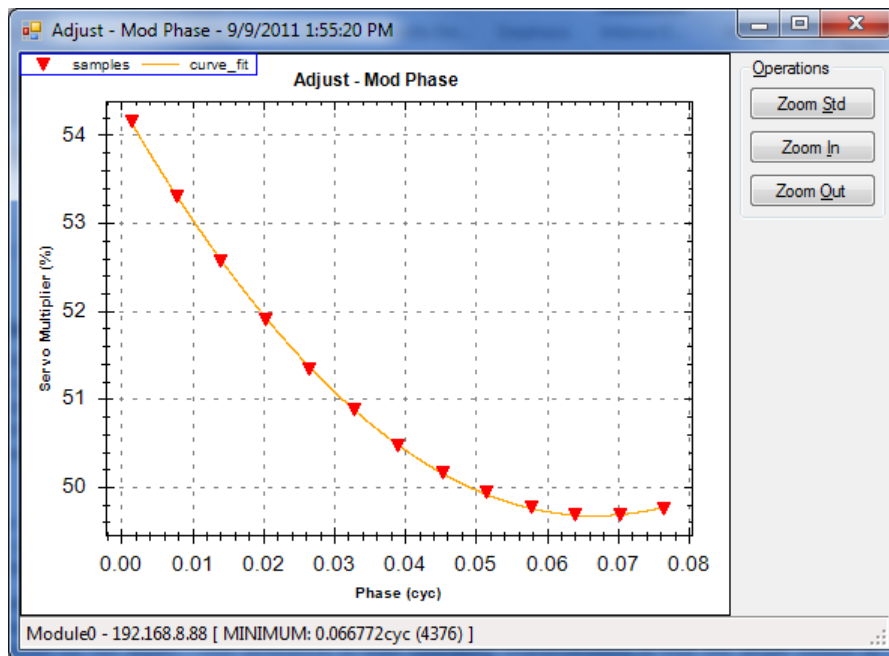
The **Mod Amp** utility reads the status of the **Servo Multiplier** so this is the value that is adjusted, not the **Sine Table Amplitude**. The graph below shows the modulation amplitude servo finding the **Servo Multiplier** set point to be 50%.



The product of the **Sine Table Amplitude**, **Output Reference** and **Servo Multiplier** controls the final modulation amplitude. It is best to keep the **Output Reference high**, the **Sine Table Amplitude** above 50% and use the **Servo Multiplier** as the control value.

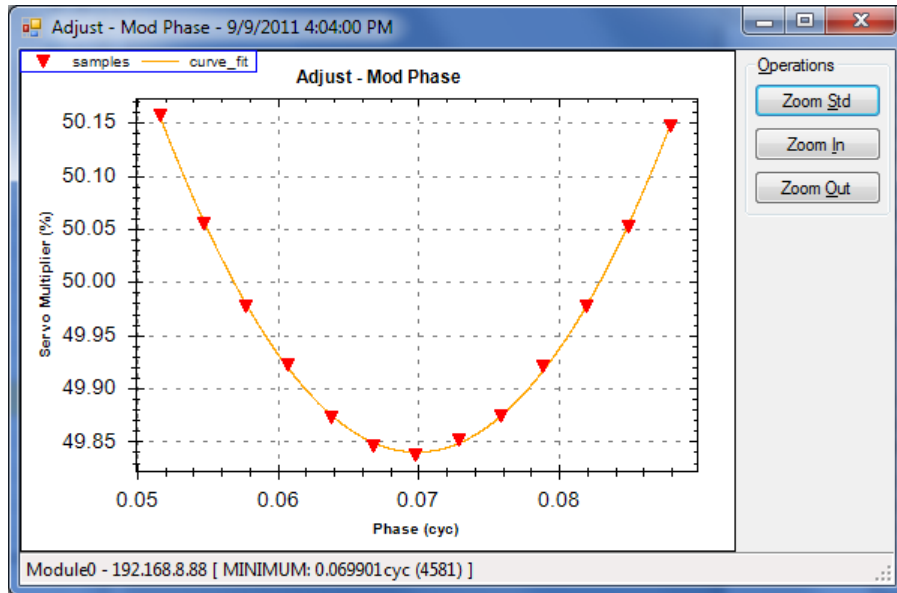
Modulation Phase

The modulation amplitude is not set correctly by the above procedure alone since the induced modulation may not be at the zero crossing of the modulation sine. The final adjustment is made by running the **Mod Phase** utility which runs the amplitude servo at selected modulation phase steps points. The steps are symmetric about the starting phase and so the output curve should be a centered parabola if the starting phase was correct. If it was not, the parabola center is off as in the graph below.



The new starting modulation phase value is displayed in the lower left corner and should be entered as the new modulation phase. It is often necessary to run the Mod Phase utility several times in between runs of the Mod Amp utility to get both value correct and produce a centered parabola.

The following graph shows a centered parabola:



The modulation **Sine Table Phase** has two correct set points. Finding one and then adding 0.5 cycles gives the other set point. There may be a small difference in the optimum value and the **Mod Phase** utility should be run again. The difference in their operating characteristics is that the angle output is inverted by switching between the two.

Sensors

The number of sensors in the array sets the maximum number of active sensors. Sensor 0 is the first non-interfering pulse and is always active. At least one other sensor must be active. A sensor is enabled by checking its box in the **Sensor Config** section which lists all the possible sensors except sensor 0. Suppose in a ten sensor array, sensor 3, 4 and 7 were active. Then the **Sin/Cos** mode would report four sensors: 0, 3, 4 and 7 but they would be numbered: 0,1,2 and 3. The **Angle** mode would report just three sensor: 3, 4 and 7 but number them: 1, 2 and 3. Sensor 0 is not measured in the **Angle** mode since it would not produce a phase measurement.

Setup Procedure

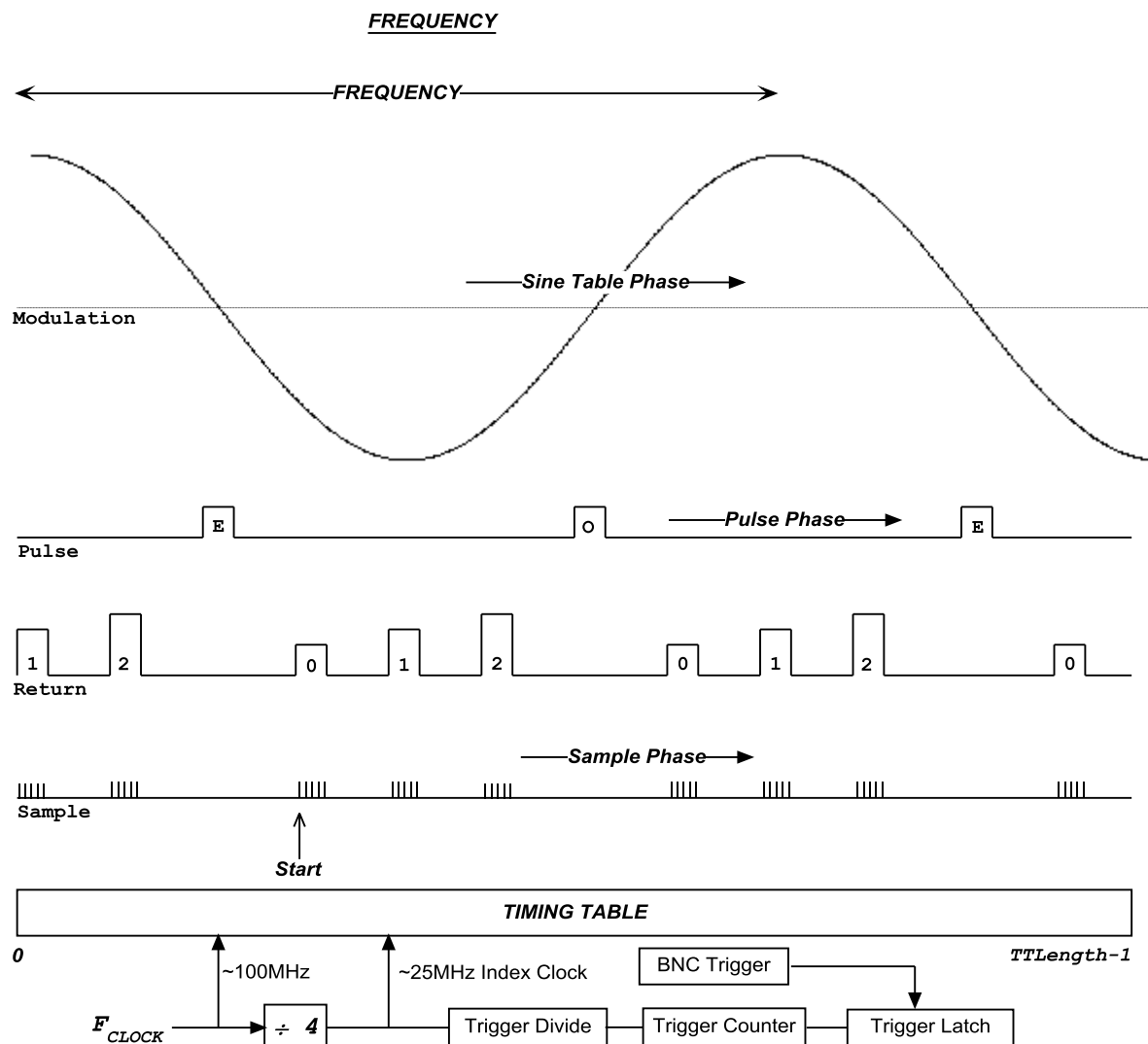
1. Referring to the contents of this manual, perform the following steps to measure sensor angles on a new array.
2. Attach an array to the TDI-7000 with reflectors that match the source wavelength and lengths that match the installed compensator.
3. Connect the TDI-7000 to the power supply for the external version.
4. Connect an Ethernet cable to the TDI-7000, 10/100/1000 is preferred.
5. Turn on the power supply.
6. Launch the TDI-7000 applet.
7. Search for the TDI-7000 by clicking **Tools/Search** and enter the **IP Address** of the unit found.
8. Initialize the following values (values are suggestions only)

Laser Temp	=	supplied value
Hysteresis	=	from back panel
EDFA Current	=	150ma
Frequency	=	desired value
Sine Table Amplitude	=	100%
Sine Table Phase	=	0
Output Reference	=	75%
Servo Multiplier	=	50%
Servo Gain	=	0
Servo Sensor	=	0
Pulse Phase	=	0
Pulse Width	=	five times sample period
Pulse Delay	=	0
Sample Phase	=	0 for single DSU systems
Sample Period	=	>= 40ns
Sample Adjust	=	0 for all sensors
Input Gain Level	=	20%
Input Gain Adjust	=	0%
Input Level Servo Enable	=	unchecked (check briefly to set parameters below)
No Light Level	=	from (SUM) ADC Data plot
Input Servo Setpoint	=	25%
Input Servo Gain	=	30
Initial Mask	=	0
Mask Gain	=	50%
Mask Mode	=	Normal
Sensors in Array	=	one or number in array
Sensor Length	=	from array construction (must match compensator)
Sensor Separation	=	from array construction
Index	=	1.4677

9. Use the ADC Data plot to survey the mask inputs and verify the presence of a sensor return.
10. Adjust the EDFA and receiver gains to prevent receiver saturation using the (SUM) ADC Data plot.
11. Align the ADC samples with the first non-intefering return, verify sensor sample positions.
12. Set the approximate modulation amplitude.
13. Set the approximate modulation phase.
14. Enable modulation servo and run the modulation amplitude utility.
15. Run the modulation phase utility.
16. Enable input servo and run the Input Level utility.
17. Verify circles in the Sin/Cos mode
18. Take sensor data.

Timing Summary

The timing of the TDI-7000 is diagrammed below.

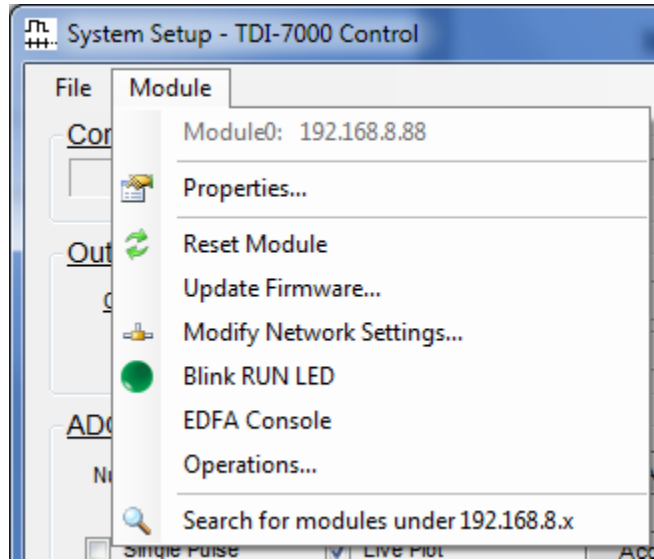


The Start bit in the timing table is where the clocks begin and it is aligned with the first sample of the even returns. The Start position is always the zero time on the ADC Data plots. The Start bit position can be moved in the timing table by the **Sample Phase** setting. The pulse position can be moved in the timing table by the **Pulse Phase** setting. The modulation sine phase can be changed with respect to the timing table by filling the timing table with a new sine with a phase set by the **Sine Table Phase** setting. The sensor returns can be moved by changing the fiber length to the first sensor.

The timing has more degrees of freedom than needed since the pulse can be aligned with the modulation zero crossing by changed the **Sine Table Phase** or the **Pulse Phase** or the samples can be aligned with the sensor returns by changing the **Pulse Phase** or the **Sample Phase**.

Any approach to setting up the timing must align the pulse with the modulation zero crossing and the sample strobes with the sensor returns.

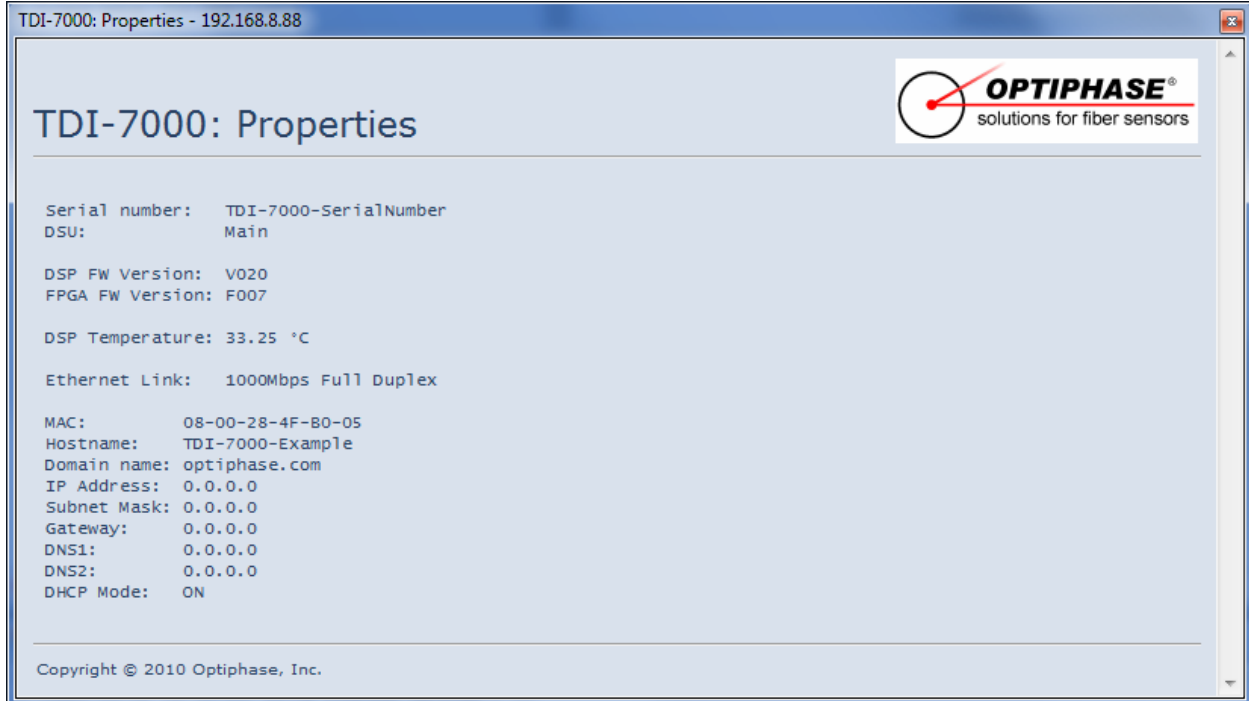
MAINTENANCE



When the Module menu in System Setup is clicked, a menu of module related operations pops up.

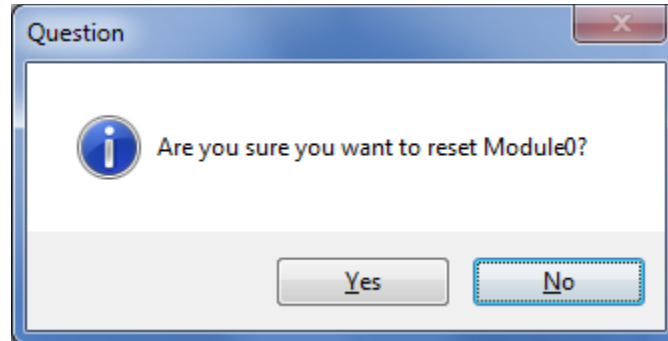
Properties

The properties selection brings up the screen shown below which contains general information regarding the TDI-7000 including the Optiphase assigned serial number, the firmware versions, the DSP temperature and the Ethernet parameters.



Reset Module

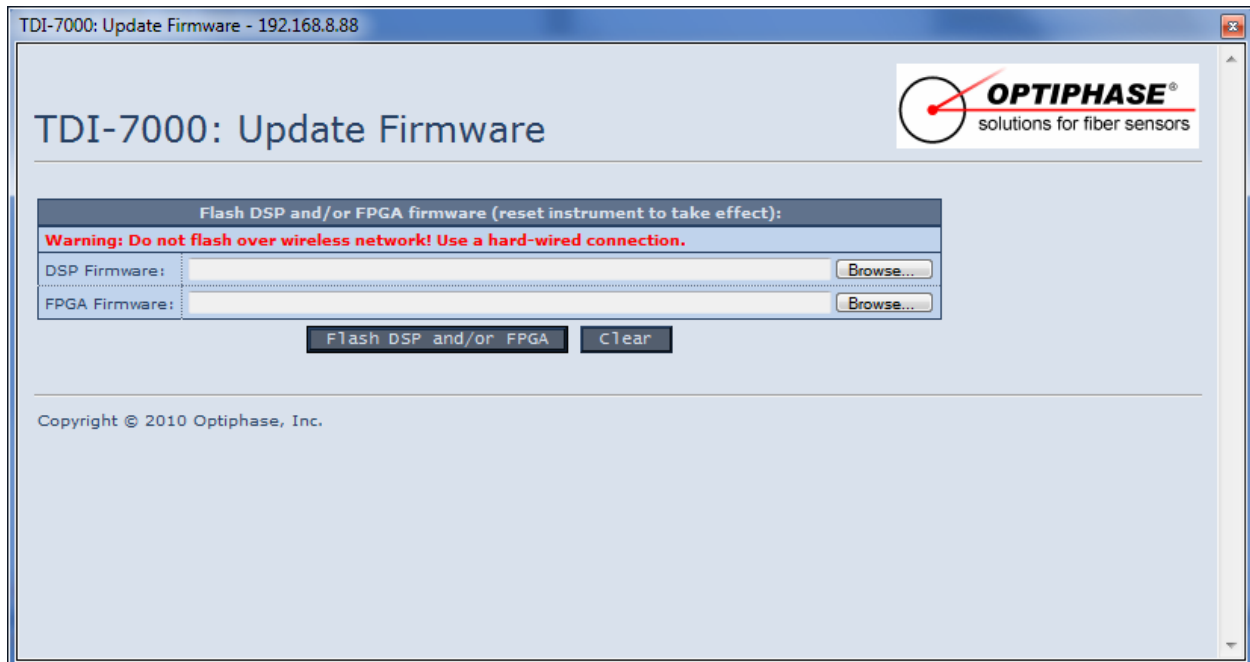
The reset instrument selection asks if you want to proceed with resetting the corresponding module. Make sure to wait until the NET LED on the front panel is lit and blinking slowly before using the TDI-7000 after a reset. If the device is remote, wait approximately 40-50 seconds for the device to boot up.



Update Firmware

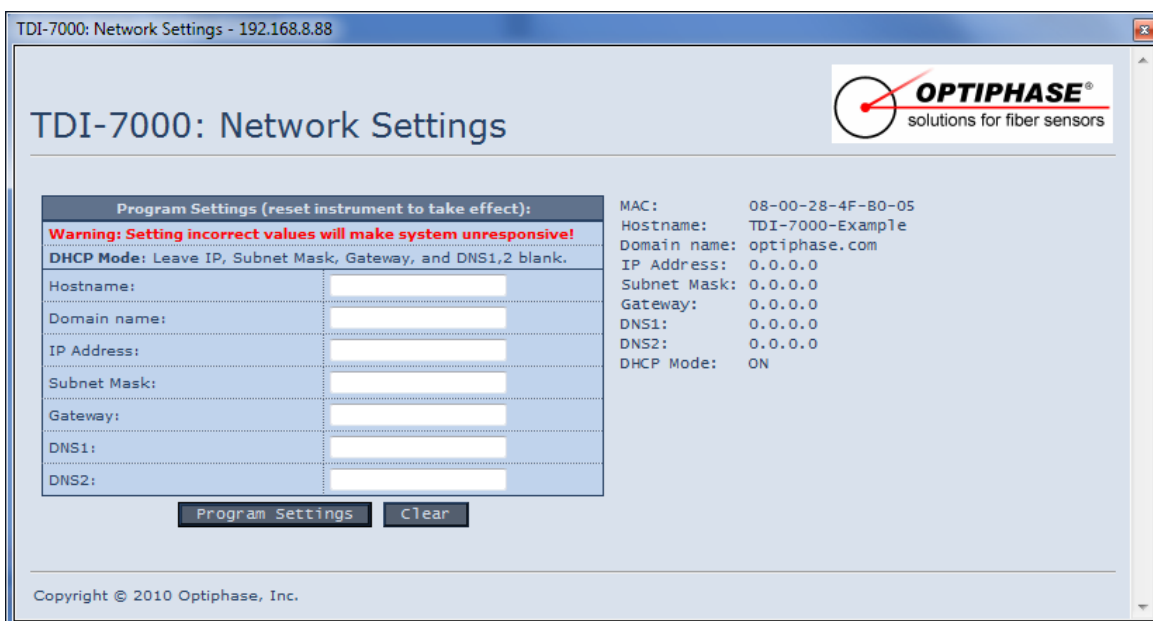
There are two firmware packages installed in the TDI-7000. One is the software that runs on the Texas Instruments Digital Signal Processor (DSP) on the Optiphase DSU electronics board and the other is the configuration file for the Xilinx Field Programmable Gate Array (FPGA). Both can be field upgraded if the need arises. Optiphase may release new firmware to add features in the future. Check with support at Optiphase.

The TDI-7000 must be reset after the installation of new firmware.



Modify Network Settings

The network settings can be manually changed by the operator with extreme caution. Take care not to use this operation unless knowledgeable or instructed by Optiphase.



The TDI-7000 supports either a **static** or **dynamic (DHCP)** IP address configuration. DHCP is the default.

If you would like to configure the system to use a **Static IP**, configure at a minimum the *Hostname*, *Domain name*, *IP Address*, and *Subnet Mask*.

If you would like to configure the system for **DHCP**, configure the *Hostname* and *Domain name*, but leave all other settings blank (or set to 0.0.0.0). When the system is configured for DHCP, the TDI-7000 requests an IP address from the DHCP server on the network.

The following table describes each network setting:

Network Setting	Description
Hostname	Host name label of module. 256 characters max. Alphanumeric string, underscores and dashes ok, no spaces. Example: tdi-7000-module-001
Domain name	Domain name label of module. 64 characters max. Alphanumeric string, underscores and dashes ok, no spaces. Example: optiphase.com
IP Address	IP Address of module. Example: 192.168.1.100
Subnet Mask	Subnet mask of module. Example: 255.255.255.0
Gateway	Gateway of module. Example: 192.168.1.1
DNS1	First DNS server of module. Example: 192.168.1.10
DNS2	Second DNS server of module. Example: 192.168.1.20

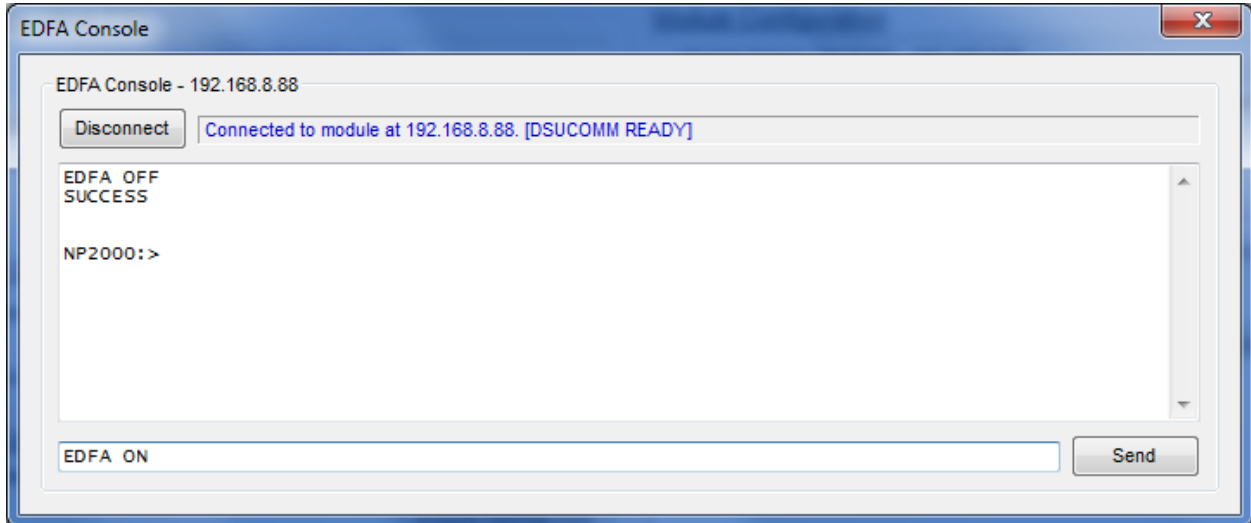
The MAC address is displayed to the right, and is set at the factory. It is not user-changeable.

Blink RUN LED

This operation simply causes the RUN LED on the front panel to blink for 10 seconds. It is useful to identify which TDI-7000 is in use.

EDFA Console

When the EDFA Console is loaded, you can initiate a connection to the module and send commands directly to the EDFA. The EDFA response is displayed in the window. This utility can only return limited length strings so some responses from the EDFA may be truncated.



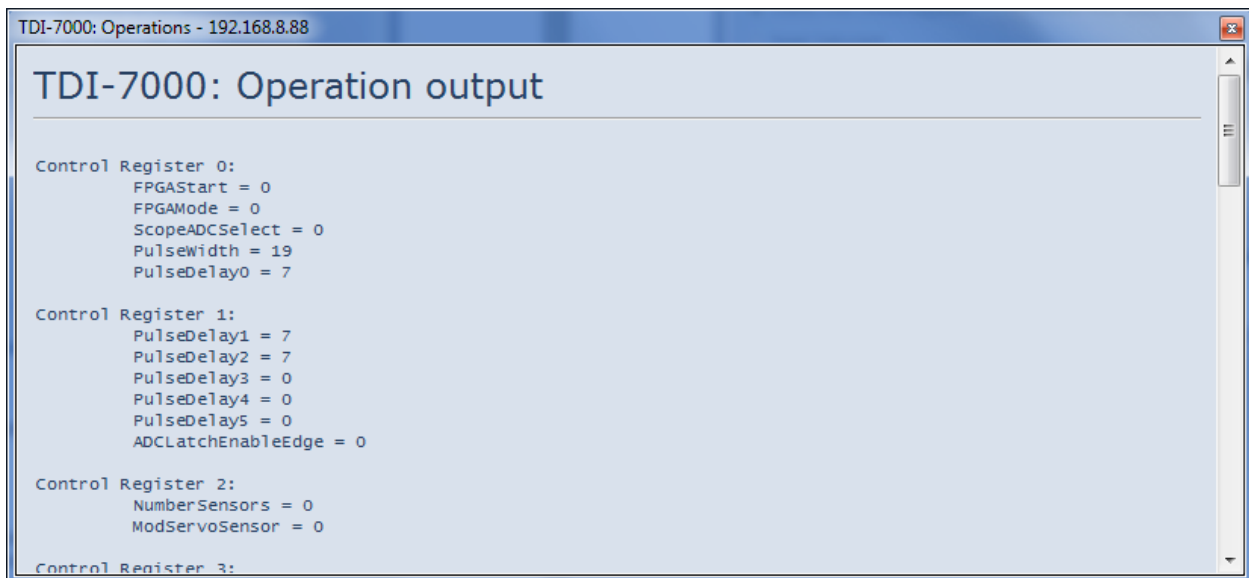
Operations

This selection brings up the screen shown below. The available operations are explained next.



Display FPGA Control Registers

This operation displays the current contents of the control registers in the FPGA. Refer to the software interface manual for register definitions.



Display FPGA Status Registers

This operation displays the current contents of the status registers in the FPGA. Refer to the software interface manual for register definitions.

The screenshot shows a window titled "TDI-7000: Operations - 192.168.8.88" with a sub-header "TDI-7000: Operation output". The content displays the following status register information:

```

Status Register 0:
  FPGAID = F0

Status Register 1:
  FPGARevision = 07

Status Register 2:
  StatusRegister02 = 52

Status Register 3:
  StatusRegister03 = 53

Status Register 4:
  eADC0_UF = 0
  eADC0_OF = 0
  eADC1_UF = 0
  eADC1_OF = 0
  eADC2_UF = 0
  eADC2_OF = 0
  eModServoErr_NOE = 0
    
```

Display Timing Table

This operation displays the current contents of the timing table in the FPGA. Refer to the software interface manual for the definition of the timing table bits.

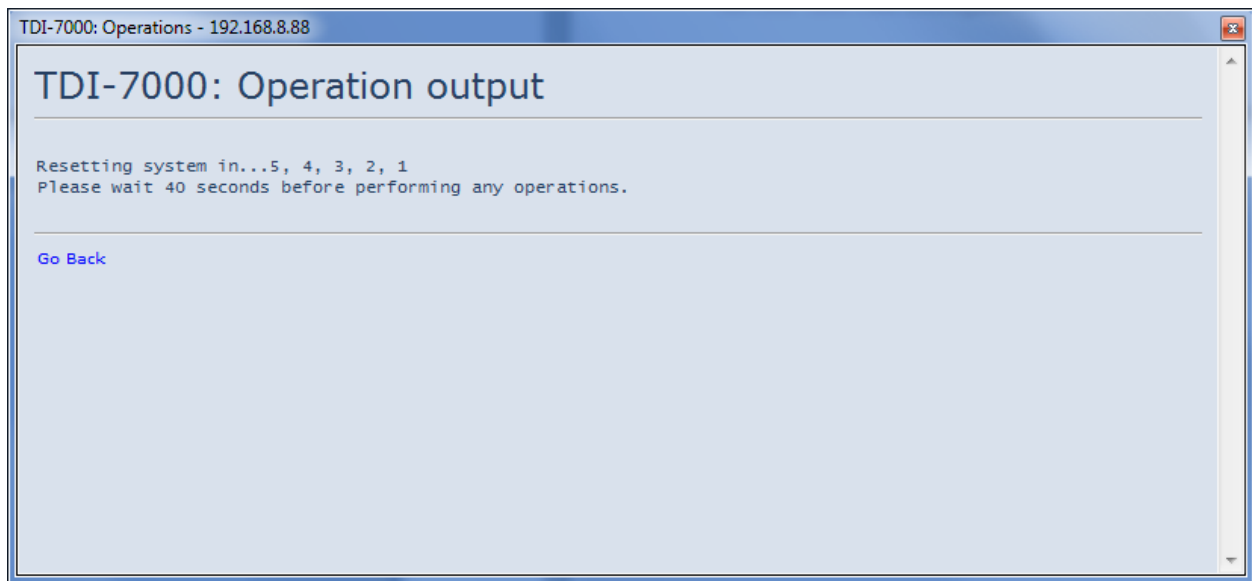
The screenshot shows a window titled "TDI-7000: Operations - 192.168.8.88" with a sub-header "TDI-7000: Operation output". The content displays the following timing table data:

Timing Table read from FPGA:

Address	Read	Address	Read
0000	00000000	2000	00
0001	00000000	2001	00
0002	00000000	2002	00
0003	00000000	2003	00
0004	00000000	2004	00
0005	00000000	2005	00
0006	00000000	2006	00
0007	00000000	2007	00
0008	00000000	2008	00
0009	00000000	2009	00
000A	00000000	200A	00
000B	00000000	200B	00
000C	00000000	200C	00
000D	00000000	200D	00
000E	00000000	200E	00
000F	00000000	200F	00

Reset Instrument

This operation resets the TDI-7000. Be sure to wait about 40-50 seconds before using the instrument after a reset.



DIGITAL DATA

Introduction

The software for the TDI-7000 provides several data and file formats. All of the formats are summarized in the following list:

1. Optiphase.Settings File Format
 - Structured human-readable text file format
 - Functionality provided through Optiphase.Settings API
2. Configuration Files
 - Contains application settings along with module settings
 - Supports multiple modules within a single configuration file
 - File formats: Binary (.bconfig) and Text (.tconfig)
3. Timing Table Files
 - Contains module timing information.
 - Useful for end-user application development.
 - File formats: Binary (.btiming) and CSV (.timing)
4. Status Files
 - Contains status information for corresponding data acquisition
 - File formats: Binary (.bstatus) and Text (.tstatus)
5. Phase sensor measurement data
 - Data types:
 - Phase32: 32-bit signed fixed-point [20-bit angle + 12-bit fringe (incl. sign bit)]
 - Phase64: 64-bit signed fixed-point [20-bit angle + 44-bit fringe (incl. sign bit)]
 - PhaseFloat: Double precision floating point (radians, wavelengths, or nanometers)
 - File formats: Binary (.bdata) and CSV (.tdata)
 - Status information can be optionally tagged onto data:
 - Saved to separate status file and/or embedded within binary data.
6. Sin/Cos data
 - Contains Sine and Cosine pairs of the phase of each sensor for all polarization masks
 - File formats: Binary (.bdata) and CSV (.tdata)
 - Status information can be optionally tagged onto data.
 - All status information is saved to separate status file
7. ADC data
 - Select ADC output of one polarization mask (of three) at a time
 - File formats: Binary (.bdata) and CSV (.tdata)
 - Some status information is embedded with the data
 - Optional separate status file is supported

The following pages describe the data formats in detail.

Optiphase.Settings File Format

Optiphase.Settings is an API used by TDI-7000 Control which provides a structured human-readable text file setting format. Each file contains various settings, optionally grouped. The elements of the file format are as follows:

Settings

A setting has the form:

```
name = value;
```

The trailing semicolon is required. Whitespace is not significant. The value may be a numeric value (integer or floating-point), a string, a boolean value (true/false), or an array of these values.

Arrays

An array has the form:

```
[ value, value ... ]
```

An array is a special value that contains multiple values. It is enclosed by brackets with commas separating the sub-values.

Groups

A group has the form:

```
GroupName : { settings ... };
```

Each group must be named properly or else its settings will not be read. Any group not recognized will be ignored. Any setting within the group that is not recognized will be ignored. If a setting is missing within a group, the setting will be loaded with its default value.

Comments

There are three types of comments supported within a configuration:

1. Script-style comments: Text starting with '#' to the end of the line is ignored.
2. C-style comments: Text, including line breaks, between '/*' and '*/' is ignored.
3. C++-style comments: Text starting with '//' to the end of the line is ignored.

Comments are ignored when the configuration is read by the software.

Example

```
/* This is a C-style comment
*/
SettingOutsideOfGroup = 1;
GroupName :
{
  SettingName1 = 100;
  SettingName2 = 10.2222;
  SettingName3 = false;
  SettingName4 = "a string";
  SettingName5 = [ 500, 1000, 1500, 2000 ];
};
# this a script-style comment
// this is a C++-style comment
// integer value
// floating-point value
// boolean value
// string value
// array of integer values
```

Configuration File Specification

A TDI-7000 Configuration File contains TDI-7000 application settings and module settings. There are two formats, binary and text. The file extension for a binary configuration file is `.bconfig`. The file extension for a text configuration file is `.tconfig`. The TDI-7000 software uses text-based config (tconfig) files. It can generate bconfig files for automation or development purposes. A configuration file can support 1 or more modules.

Groups

A Configuration File contains the following groups of settings:

Group name	Description
TDI7000AppSettings	Contains TDI-7000 Control application-specific settings. TDI7000AppSettings settings are linked to properties within a TDI7000AppSettings object. The setting TDI7000AppSettings : NumberModules must equal the number of DSUConfigN groups.
DSUConfigN	Contains module-specific settings, where N is the module index. DSUConfigN settings are linked to properties within a DSUConfig object (specification is provided with the DSUComm API). N is an integer starting from zero.

The following table demonstrates what groups should exist in the configuration file depending on the number of modules linked together within the TDI-7000 Control application:

NumberModules	Groups
1	TDI7000AppSettings, DSUConfig0
2	TDI7000AppSettings, DSUConfig0, DSUConfig1
3	TDI7000AppSettings, DSUConfig0, DSUConfig1, DSUConfig2
N	TDI7000AppSettings, DSUConfig0, DSUConfig1, ..., DSUConfig[N-1]

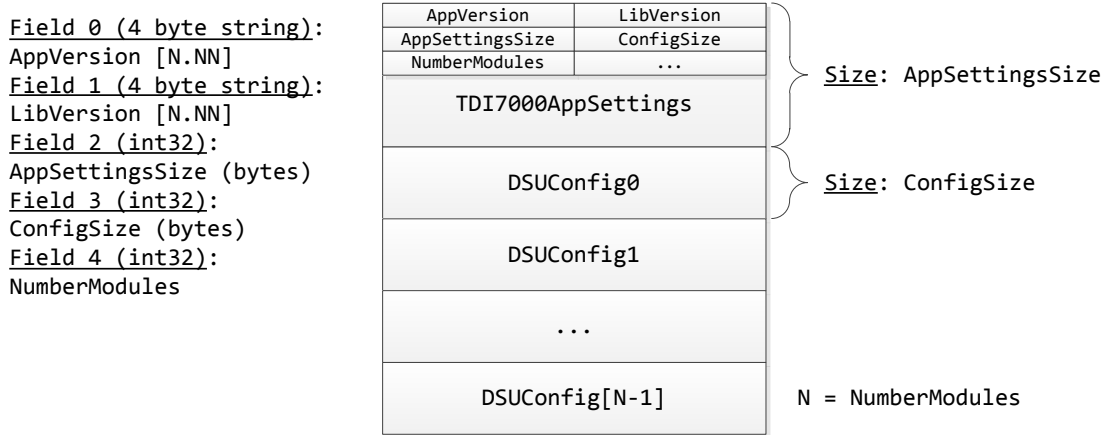
Some configuration settings are required to be arrays, even if only one element exists in the array. For example, the TDI7000AppSettings : IPAddress setting is an array of strings where its number of strings depends on the TDI7000AppSettings : NumberModules setting.

When the configuration is saved, comments that were present in the original configuration file are lost.

Binary Configuration File Layout

The layout for a binary configuration file containing N modules is shown below:

Binary Configuration File Layout



Total Size (bytes): AppSettingsSize + NumberModules * ConfigSize

Text Configuration File Layout

A sample configuration text file for two modules is shown below.

Note: The following configuration is shown to demonstrate the configuration file grouping specification. Please refer to TDI-7000 Control application documentation and DSUComm Library documentation for a list of settings within TDI7000AppSettings and DSUConfig structures.

```
// TDI-7000 Control Application Settings
TDI7000AppSettings :
{
  AppVersion = 1.50;
  LibVersion = 1.50;
  AppSettingsSize = 512;
  ConfigSize = 1024;
  NumberModules = 2;
  IPAddress = [ "192.168.1.20", "192.168.1.21" ];
  // additional settings clipped
};
// Configuration for Module0 @ 192.168.1.20
DSUConfig0 :
{
  // Module0 settings located here
};
// Configuration for Module1 @192.168.1.21
DSUConfig1 :
{
  // Module1 settings located here
};
```


Timing Table Files

Timing table files contain module timing information and are linked to specific configuration. This timing information is able to be exported along with configuration files from the TDI-7000 application software. It is useful to generate these files in order to allow a developer to create their own software utilizing the TDI-7000 API (DSUComm). These files may be in binary format (.btiming) or text format (.ttiming).

Status File Specification

A TDI-7000 Status File contains TDI-7000 status information during a data acquisition. There are two formats, binary and text. The file extension for a binary status file is .bstatus, and the file format is described under the section *Binary Status File Layout*. A text configuration file uses the file extension .tstatus, and it follows the Optiphase.Settings file format specification. A configuration file can support 1 or more modules.

Groups

A Status File may contain the following groups of settings:

Group name	Description
DSUAcquisitionInfo	Contains key configuration settings and information about the acquisition
DSUSystemInfo	Contains system information about the module used for the acquisition
DDBStatusN	Contains status information, such as errors, flags, and trigger data for a single data block acquired, where N is the block index. Settings are linked to properties within a DDBStatus object provided by the DSUComm API. N is an integer starting from zero. Note: This group only exists in the status file for the Phase and SinCos data formats, and if AcquisitionInfo : StatusEnabled is set to true.

The following table demonstrates what groups should exist in the configuration file depending on the number of modules linked together within the TDI-7000 Control application:

BlocksPerFile	Groups
1	DSUAcquisitionInfo, DSUSystemInfo, DDBStatus0
2	DSUAcquisitionInfo, DSUSystemInfo, DDBStatus0, DDBStatus1
3	DSUAcquisitionInfo, DSUSystemInfo, DDBStatus0, DDBStatus1, DDBStatus2
N	DSUAcquisitionInfo, DSUSystemInfo, DDBStatus0, DDBStatus1, ..., DDBStatus[N-1]

DSUAcquisitionInfo Group Elements

The following table lists the information provided through the DSUAcquisitionInfo group:

Setting	Format	Description
AppVersion	string	Application version (4 byte string)
LibVersion	string	Library version (i.e. DSUComm) (4 byte string)
AcqInfoSize	int32	Size of DSUAcquisitionInfo structure
SystemInfoSize	int32	Size of DSUSystemInfo structure
StatusSize	int32	Size of DDBStatus structure
BlocksPerFile	int32	Number of blocks per file
FileSize	int32	Size of data file in bytes: Includes embedded headers if FileHeaderEnable is true
FileFormat	int32	File format of saved block data: 0: Binary 1: CSV

FileHeaderEnable	bool	<p>Headers embedded in data file: If true, headers are embedded in data file. If FileFormat is Binary and FileHeaderEnable is true, then a status header (in status file format) will precede each block. If FileFormat is CSV, FileHeaderEnable is always false.</p>																
DataFormat	int32	<p>Data format of saved block data:</p> <table border="1"> <thead> <tr> <th>DataFormat</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>-1: None</td> <td>No format / undefined format</td> </tr> <tr> <td>0: RAW</td> <td>Raw bytes</td> </tr> <tr> <td>1: Phase32</td> <td>Phase (32-bit signed fixed-point)</td> </tr> <tr> <td>2: Phase64</td> <td>Phase extended (64-bit signed fixed-point)</td> </tr> <tr> <td>3: PhaseFloat</td> <td>Phase float (64-bit floating-point)</td> </tr> <tr> <td>4: SinCos</td> <td>Sin/Cos pairs for 3 masks</td> </tr> <tr> <td>5: ADC</td> <td>ADC Data for selected mask</td> </tr> </tbody> </table>	DataFormat	Description	-1: None	No format / undefined format	0: RAW	Raw bytes	1: Phase32	Phase (32-bit signed fixed-point)	2: Phase64	Phase extended (64-bit signed fixed-point)	3: PhaseFloat	Phase float (64-bit floating-point)	4: SinCos	Sin/Cos pairs for 3 masks	5: ADC	ADC Data for selected mask
DataFormat	Description																	
-1: None	No format / undefined format																	
0: RAW	Raw bytes																	
1: Phase32	Phase (32-bit signed fixed-point)																	
2: Phase64	Phase extended (64-bit signed fixed-point)																	
3: PhaseFloat	Phase float (64-bit floating-point)																	
4: SinCos	Sin/Cos pairs for 3 masks																	
5: ADC	ADC Data for selected mask																	
DataUnits	int32	<p>Units for saved block data:</p> <table border="1"> <thead> <tr> <th>DataUnits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>-1: None</td> <td>No units</td> </tr> <tr> <td>0: radians</td> <td>units are in radians</td> </tr> <tr> <td>1: wavelengths</td> <td>units are in wavelengths</td> </tr> <tr> <td>2: nanometers</td> <td>units are in nanometers</td> </tr> </tbody> </table>	DataUnits	Description	-1: None	No units	0: radians	units are in radians	1: wavelengths	units are in wavelengths	2: nanometers	units are in nanometers						
DataUnits	Description																	
-1: None	No units																	
0: radians	units are in radians																	
1: wavelengths	units are in wavelengths																	
2: nanometers	units are in nanometers																	
SystemIndex	int32	<p>System Index Contains module number. If not used, then equals -1.</p>																
NumberPulses	int32	<p>Number of pulses per modulation cycle (1 or 2)</p>																
NumberSensors	int32	<p>Number of sensors (channels) enabled in data Note: If FirstSensorEnable = true, NumberSensors count includes non-interfering pulse.</p>																
NumberSensorsArray	int32	<p>Number of sensors in array</p>																
FirstSensorEnable	bool	<p>Non-interfering pulse data enable</p>																
TTLLength	int32	<p>Timing table length</p>																
Fclock16	int32	<p>Master clock frequency $Fclock16 = MasterClock * 16$</p>																
ADCSelect	int32	<p>ADC mask selection</p> <table border="1"> <thead> <tr> <th>ADCSelect</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>-1: None</td> <td>N/A (DataFormat != ADC)</td> </tr> <tr> <td>0: ADC0</td> <td>ADC 0 data</td> </tr> <tr> <td>1: ADC1</td> <td>ADC 1 data</td> </tr> <tr> <td>2: ADC2</td> <td>ADC 2 data</td> </tr> <tr> <td>3: SUM</td> <td>$SUM = (ADC0 + ADC1 + ADC2) / 2$</td> </tr> </tbody> </table>	ADCSelect	Description	-1: None	N/A (DataFormat != ADC)	0: ADC0	ADC 0 data	1: ADC1	ADC 1 data	2: ADC2	ADC 2 data	3: SUM	$SUM = (ADC0 + ADC1 + ADC2) / 2$				
ADCSelect	Description																	
-1: None	N/A (DataFormat != ADC)																	
0: ADC0	ADC 0 data																	
1: ADC1	ADC 1 data																	
2: ADC2	ADC 2 data																	
3: SUM	$SUM = (ADC0 + ADC1 + ADC2) / 2$																	

<p>StatusEnabled</p>	<p>bool</p>	<p>Digital status enabled:</p> <table border="1"> <thead> <tr> <th>DataFormat</th> <th>StatusEnabled</th> </tr> </thead> <tbody> <tr> <td>-1: None</td> <td>false</td> </tr> <tr> <td>0: RAW</td> <td>false</td> </tr> <tr> <td>1: Phase32</td> <td rowspan="3">true: status is enabled and status file contains DDBStatusN groups</td> </tr> <tr> <td>2: Phase64</td> </tr> <tr> <td>3: PhaseFloat</td> </tr> <tr> <td>4: SinCos</td> <td>false: status is disabled and status file does not contain DDBStatusN groups</td> </tr> <tr> <td>5: ADC</td> <td>true: saved data has embedded status false: saved data does not have status</td> </tr> </tbody> </table>	DataFormat	StatusEnabled	-1: None	false	0: RAW	false	1: Phase32	true : status is enabled and status file contains DDBStatusN groups	2: Phase64	3: PhaseFloat	4: SinCos	false : status is disabled and status file does not contain DDBStatusN groups	5: ADC	true : saved data has embedded status false : saved data does not have status		
DataFormat	StatusEnabled																	
-1: None	false																	
0: RAW	false																	
1: Phase32	true : status is enabled and status file contains DDBStatusN groups																	
2: Phase64																		
3: PhaseFloat																		
4: SinCos	false : status is disabled and status file does not contain DDBStatusN groups																	
5: ADC	true : saved data has embedded status false : saved data does not have status																	
<p>TriggerEnabled</p>	<p>bool</p>	<p>Trigger enabled</p>																
<p>TriggerSelect</p>	<p>int32</p>	<p>Trigger select:</p> <table border="1"> <thead> <tr> <th>TriggerSelect</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Software activated</td> </tr> <tr> <td>1</td> <td>Rising edge on external trigger</td> </tr> <tr> <td>2</td> <td>Falling edge on external trigger</td> </tr> <tr> <td>3</td> <td>Either edge on external trigger</td> </tr> </tbody> </table>	TriggerSelect	Description	0	Software activated	1	Rising edge on external trigger	2	Falling edge on external trigger	3	Either edge on external trigger						
TriggerSelect	Description																	
0	Software activated																	
1	Rising edge on external trigger																	
2	Falling edge on external trigger																	
3	Either edge on external trigger																	
<p>TriggerCountDivider</p>	<p>int32</p>	<p>Trigger count divider: 32-bit counter resolution set to: $F_{clock} / (4 * (TriggerCountDivider + 1))$</p>																
<p>BlockSizeRaw</p>	<p>int32</p>	<p>Block size in bytes (raw data transferred from module)</p>																
<p>WordSizeRaw</p>	<p>int32</p>	<p>Word size in bytes (raw data transferred from module):</p> <table border="1"> <thead> <tr> <th>DataFormat</th> <th>WordSizeRaw</th> </tr> </thead> <tbody> <tr> <td>-1: None</td> <td>-1</td> </tr> <tr> <td>0: RAW</td> <td>1</td> </tr> <tr> <td>1: Phase32</td> <td>4</td> </tr> <tr> <td>2: Phase64</td> <td>4</td> </tr> <tr> <td>3: PhaseFloat</td> <td>4</td> </tr> <tr> <td>4: SinCos</td> <td>2</td> </tr> <tr> <td>5: ADC</td> <td>2</td> </tr> </tbody> </table>	DataFormat	WordSizeRaw	-1: None	-1	0: RAW	1	1: Phase32	4	2: Phase64	4	3: PhaseFloat	4	4: SinCos	2	5: ADC	2
DataFormat	WordSizeRaw																	
-1: None	-1																	
0: RAW	1																	
1: Phase32	4																	
2: Phase64	4																	
3: PhaseFloat	4																	
4: SinCos	2																	
5: ADC	2																	
<p>WordSize</p>	<p>int32</p>	<p>Word size in bytes (saved data):</p> <table border="1"> <thead> <tr> <th>DataFormat</th> <th>WordSize</th> </tr> </thead> <tbody> <tr> <td>-1: None</td> <td>-1</td> </tr> <tr> <td>0: RAW</td> <td>1</td> </tr> <tr> <td>1: Phase32</td> <td>4</td> </tr> <tr> <td>2: Phase64</td> <td>8</td> </tr> <tr> <td>3: PhaseFloat</td> <td>8</td> </tr> <tr> <td>4: SinCos</td> <td>2</td> </tr> <tr> <td>5: ADC</td> <td>2</td> </tr> </tbody> </table>	DataFormat	WordSize	-1: None	-1	0: RAW	1	1: Phase32	4	2: Phase64	8	3: PhaseFloat	8	4: SinCos	2	5: ADC	2
DataFormat	WordSize																	
-1: None	-1																	
0: RAW	1																	
1: Phase32	4																	
2: Phase64	8																	
3: PhaseFloat	8																	
4: SinCos	2																	
5: ADC	2																	

BlockSize	int32	Block size in words (number of elements per block):	
		DataFormat	BlockSize
		-1: None	-1
		0: RAW	BlockSizeRaw / WordSizeRaw
		1: Phase32	BlockSizeRaw / WordSizeRaw
		2: Phase64	BlockSizeRaw / WordSizeRaw
		3: PhaseFloat	BlockSizeRaw / WordSizeRaw
FrameSize	int32	Frame size in words (number of words in one modulation cycle):	
		DataFormat	FrameSize
		-1: None	-1
		0: RAW	1
		1: Phase32	NumberPulses * NumberSensors
		2: Phase64	NumberPulses * NumberSensors
		3: PhaseFloat	NumberPulses * NumberSensors
FramesPerBlock	int32	Number of frames per block:	
		DataFormat	FramesPerBlock
		-1: None	-1
		0: RAW	BlockSize / FrameSize
		1: Phase32	BlockSize / FrameSize
		2: Phase64	BlockSize / FrameSize
		3: PhaseFloat	BlockSize / FrameSize
4: SinCos	BlockSize / FrameSize		
5: ADC	BlockSize / FrameSize		

SystemInfo Group Elements

The following table lists the information provided through the SystemInfo group:

Setting	Format	Description
DSUCommProtocolVersion	string	DSUComm Protocol Version, 4 bytes: x.yy Where x is major version and yy is minor version
FpgaFirmwareVersion	string	FPGA Firmware Version, 4 bytes First character: 'F' for release version; 'X' for experimental version Second character: FPGA index in system Third and fourth characters: FPGA revision Examples: "F104" - Release FPGA firmware, index 1, revision 04 "X012" - Experimental FPGA firmware, index 0, revision 12

DspFirmwareVersion	string	DSP Firmware Version, 4bytes First character: 'V' for release version; 'X' for experimental version Last three characters: DSP revision Examples: "V018" - Release DSP firmware, revision 018 "X013" - Experimental DSP firmware, revision 013
SerialNumber	string	DSU Serial Number, 32 bytes Contains serial number of module set at factory. Example: 0035-1234-B000
Type	int32	DSU System Type 0: Main DSU 1: Auxiliary DSU (only used in Dual-DSU configurations)
FPGAState	int32	FPGA State 0: Error 1: Ready
EDFAState	int32	EDFA State 0: Error 1: Ready (on) 2: Ready (off) 3: NoEDFA
EDFACurrent	int32	EDFA Current (mA)
DSPTemperature4	int32	DSP Temperature $\text{DSP temperature (}^\circ\text{C)} = \frac{\text{DSPTemperature4}}{4}$
EthernetLinkSpeed	int32	Ethernet Link Speed (of module) 0: No Link 1: 10Mbps Half Duplex 2: 10Mbps Full Duplex 3: 100Mbps Half Duplex 4: 100Mbps Full Duplex 5: 1000Mbps Full Duplex
EthernetMacAddress	string	MAC Address, 20 bytes Contains MAC Address of module set at factory. Example: 00-BB-EE-CC-11-FF
EthernetHostName	string	Host Name, 256 bytes Host name label of module. Alphanumeric string, underscores and dashes ok, no spaces Example: tdi-7000-module-001
EthernetDomainName	string	Domain Name, 64 bytes Domain name label of module. Alphanumeric string, underscores and dashes ok, no spaces Example: optiphase.com
EthernetIPAddress	string	IP Address, 16 bytes IP Address of module. Example: 192.168.1.100
EthernetSubnetMask	string	Subnet Mask, 16 bytes Subnet mask of module. Example: 255.255.255.0
EthernetGateway	string	Gateway, 16 bytes Gateway of module. Example: 192.168.1.1

EthernetDNS1	string	<u>DNS 1</u> , First DNS server of module. Example: 192.168.1.10	16 bytes
EthernetDNS2	string	<u>DNS 2</u> , Second DNS server of module. Example: 192.168.1.20	16 bytes

Block Status Group Elements

The DDBStatus group provides status information corresponding to a data block.

The following table lists the information provided through the DDBStatus group:

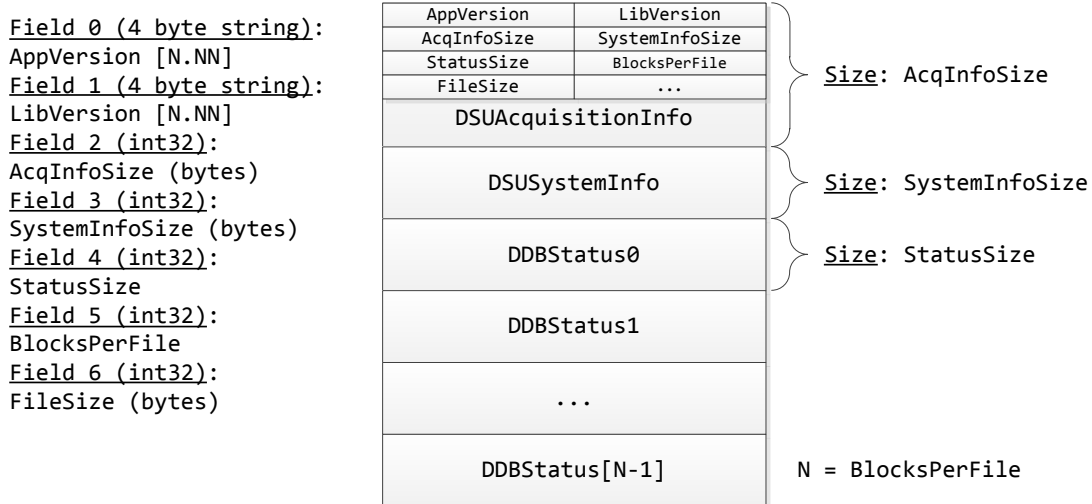
Setting	Format	Description
Error	bool	An error has been flagged
eDataTrailerInvalid	bool	FPGA data trailer invalid; data is not in expected format
eDataBlockTransferLoss	bool	Data transfer stalled or parameters exceed rate limitation
eADC0_UF	bool	ADC0 Underflow Error
eADC0_OF	bool	ADC0 Overflow Error
eADC1_UF	bool	ADC1 Underflow Error
eADC1_OF	bool	ADC1 Overflow Error
eADC2_UF	bool	ADC2 Underflow Error
eADC2_OF	bool	ADC2 Overflow Error
eModServoErr_NOF	bool	Mod Servo Error Negative Overflow Error
eModServoErr_POF	bool	Mod Servo Error Positive Overflow Error
eModServoAccErr_NOF	bool	Mod Servo Accum Error Negative Overflow Error
eModServoAccErr_POF	bool	Mod Servo Accum Error Positive Overflow Error
eModServoMult_Neg	bool	Mod Servo Multiplier Negative Error
eModServoMult_OF	bool	Mod Servo Multiplier Overflow Error
eRcvrGainServo	bool	Rcvr Gain Servo Error (if any error from the next 6 are true)
eRcvrServoErr_NOF	bool	Rcvr Servo Error Negative Overflow Error
eRcvrServoErr_POF	bool	Rcvr Servo Error Positive Overflow Error
eRcvrServoAccErr_NOF	bool	Rcvr Servo Accum Error Negative Overflow Error
eRcvrServoAccErr_POF	bool	Rcvr Servo Accum Error Positive Overflow Error
eRcvrGainDAC_Neg	bool	Rcvr Gain DAC Negative Error
eRcvrGainDAC_OF	bool	Rcvr Gain DAC Overflow Error
fTriggerEvent	bool	Trigger Event Flag
TriggerEventType	int32	Trigger Event Type 0: No trigger 1: Software 2: External Positive Edge 3: External Negative Edge
TriggerCount	uint32	Trigger Count (range: [0,2 ³² -1])
ModServoMult	int32	Mod Servo Multiplier, unsigned
DataFIFOBlockLength	int32	Data FIFO Block Length (words) (without trailer)
DataFIFOBlockIndex	int32	Data FIFO Block Index
ExtPulseEvenCount	int32	External Pulse Even 100MHz Divided down Count

<p>FrameCount</p>	<p>int64</p>	<p>64-bit unsigned frame counter.</p> <p>With this value, you may calculate the block index:</p> $N = \text{block index} = \frac{\text{DDBStatus.FrameCount}}{\text{DSUAcquisitionInfo.FramesPerBlock}}$
<p>Power</p>	<p>int16[256]</p>	<p>Power from last sample (NumberPulses=1) or last even sample (NumberPulses=2) for each sensor of the corresponding block. Range: [0,49149]</p> <p>Power[0] is non-interfering pulse, and Power[1] is the first sensor, Power[2] is the second sensor, and so on. The array has 256 elements, of which NumberSensors elements are used.</p>

Binary Status File Layout

The layout for a binary status file containing N blocks is shown below:

Binary Status File Layout



Total Size (bytes): AcqInfoSize + SystemInfoSize + BlocksPerFile * StatusSize

Text Status File Layout

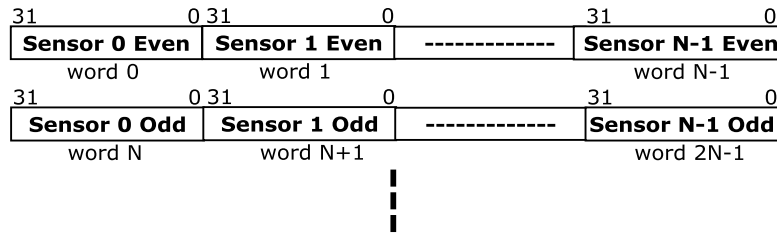
A sample Status File for a 2-block data acquisition is shown below:

```
//
// Status File Example for a 2-block acquisition
//
AcquisitionInfo :
{
  AppVersion = 1.50;
  LibVersion = 1.50;
  AcqInfoSize = 116;
  SystemInfoSize = 488;
  StatusSize = 636;
  BlocksPerFile = 1;
  FileSize = 995328;
  // Additional elements clipped
};
SystemInfo :
{
  // System information elements located here
};
DDBStatus0 :
{
  // Block 0 status information elements located here
};
DDBStatus1 :
{
  // Block 0 status information elements located here
};
```

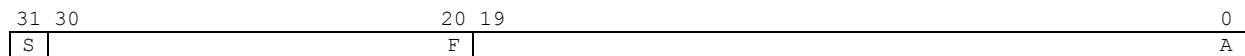

Phase Sensor Data Formats and Calculations

Phase32: Normal Sensor Mode

This is the normal data output format where a reading is provided for each sensor. The data is 32 bits where the lower 20 bits cover the unit circle and the upper 12 bits are the signed fringe counter.



The 32-bit signed fixed point number, also known as Phase32, is in the following format:

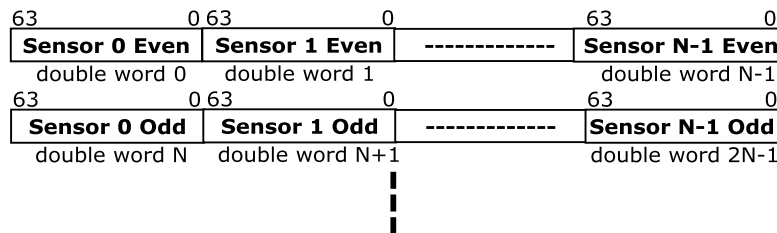


S = Sign bit [0 = positive; 1 = negative]
 F = 11-bit fringe
 A = 20-bit angle

This integer is composed of the unit circle angle A added to the fringe counter F. This integer is not in radians, but in bits. It is the raw angle data format provided by the module.

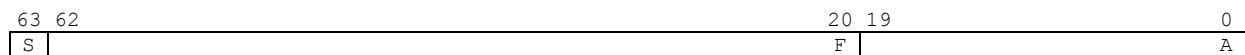
Phase64: Extended Sensor Mode

In this mode the normal data is extended to 64 bits with the addition of an external fringe counter. This format prevents rollovers from occurring. It is normally calculated after Ethernet transfer in the host.



Averaging should only be performed on extended data since rollovers would corrupt the data.

The 64-bit signed fixed point number, also known as Phase64 (Extended Angle), is in the following format:

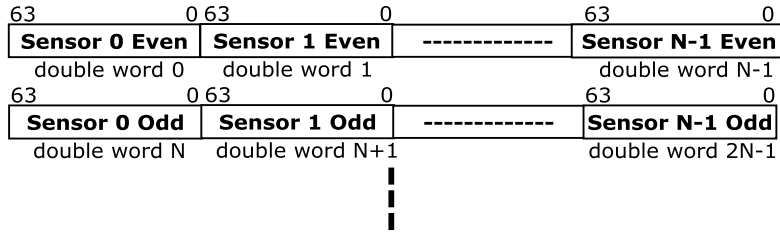


S = Sign bit [0 = positive; 1 = negative]
 F = 43-bit fringe
 A = 20-bit angle

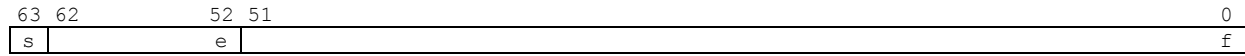
This integer is composed of the unit circle angle A added to the fringe counter F. This integer is not in radians, but in bits. It is calculated utilizing an external fringe counter.

PhaseFloat

In this mode, the data is arranged in a similar fashion to Phase64, but the data is stored as double precision floating point values.



The 64-bit double precision floating point number, also known as PhaseFloat, is in the following format:



s = Sign bit [0 = positive; 1 = negative]
 e = 11-bit exponent
 f = 52-bit fraction

The units for the values may be radians, wavelengths, or nanometers, depending on the DataUnits setting. All units are converted from Phase64 (see **Conversions** below for scale factors).

Conversions

A sensor measurement is a 32 bit digital value (Phase32) where the lower 20 bits cover one wavelength of the optical source and the upper 12 bits are a signed number of integer wavelengths (fringe counter). The digital value taken as a whole ranges from -2,147,483,648 to 2,147,483,647. The sensor measurement can be converted to other units using the scale factors shown below.

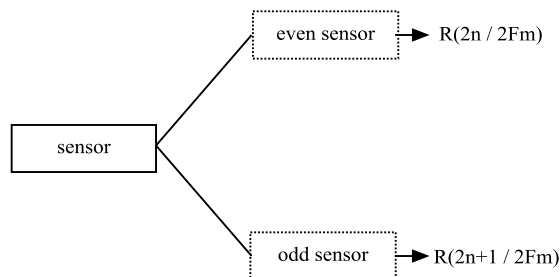
$$R = \text{sensor measurement (Angle32 format)}$$

$$\text{wavelengths} = R \cdot \frac{1}{2^{20}}$$

$$\text{radians} = R \cdot \frac{2\pi}{2^{20}}$$

$$\text{nanometers} = R \cdot \frac{\lambda}{n \cdot 2^{20}} \quad (\lambda = \text{wavelength of source in nm, } n = \text{index of fiber})$$

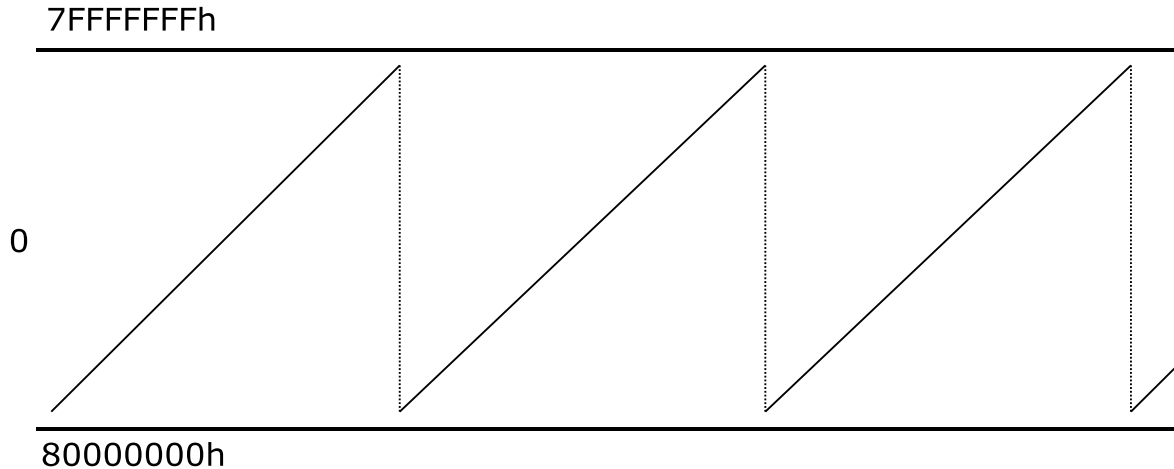
The TDI-7000 measurement technique uses an odd optical pulse and an even optical pulse as explained in the introduction. These two pulses make independent measurements of a sensor in the respect that they have separate fringe counters and may be different by a constant amount. This occurs because modulation uses a different slope of the sine for each type of pulse and changes in the slope condition directly change the measured phase. For the constant difference to be zero, the center of both optical pulses would have to be exactly on the respective zero crossings of the modulation, an unlikely event. The effective DSU sensor measurement is diagramed below.



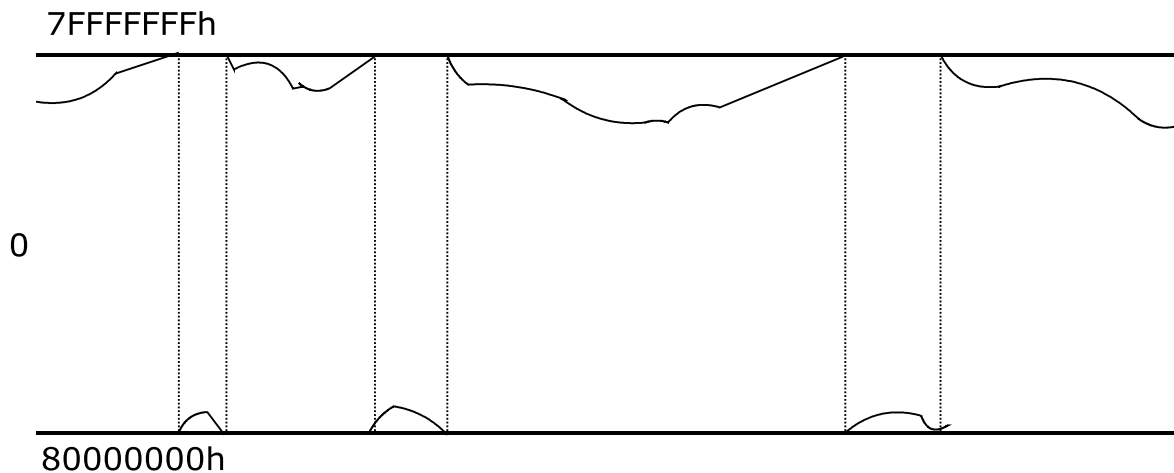
The odd slope sensor reading occurs once per modulation cycle as does the even slope sensor reading. Combined, the sensor is read at a rate of twice the modulation frequency but with a square wave at the modulation frequency present due to the different offsets between the odd and even readings.

Rollovers

The digital representation of the sensor output is limited in dynamic range by the number of bits allocated to the fringe counter. Stretching the sensor fiber or changing its temperature can induce very large changes in the phase that will exceed the digital word size. The situation is diagrammed below for a constantly increasing temperature.



Note that the sensor output never hits the limits since the change in angle between samples has some finite step size from 0 to just less than π radians so the output flies back to the rolled over value. A more problematic situation is shown below.



Here, the temperature has stabilized but now the sensor output is near the digital boundary. Small input signals will cause boundary crossings which inject fly backs into the digital output data.

External Fringe Counter

The rollover problem occurs because the digital output cannot represent the full dynamic range of the sensor. The direct solution is to increase the number of bits in the digital representation. The bandwidth of the interface between the host PC and the DSU demodulator limits the size of the digital representation of the sensor output. The default solution is to use a 32-bit representation with a 12-bit fringe. In this case, the host can extend the sensor output to 64 bits by recognizing a rollover event.

A rollover event changes the sensor output by almost a full 32-bit word value which vastly exceeds the half circle change that is allowed between each sensor sample. The host processing shown below will add an external fringe counter to the sensor measurement R.

D_e 64 bit signed fringe in Host, initially 0
H_e 64 bit extended output in Host
h_e previous extended output in Host

```

He = R2n + De
if( ( He - he ) < 0 )
    if( (2^30 + (He - he )) < 0 )) De = De + 2^32
else
    if( (2^30 - (He - he )) < 0 )) De = De - 2^32
He = R2n + De
he = He

```

The 64 bit measurement H represents a value that never has the rollover problem. The external fringe counter must be done separately for the even measurement and the odd measurement so that the two data streams are processed separately. The even external fringe counter is shown above and the odd is shown below.

D_o 64 bit signed fringe in Host, initially 0
H_o 64 bit extended output in Host
h_o previous extended output in Host

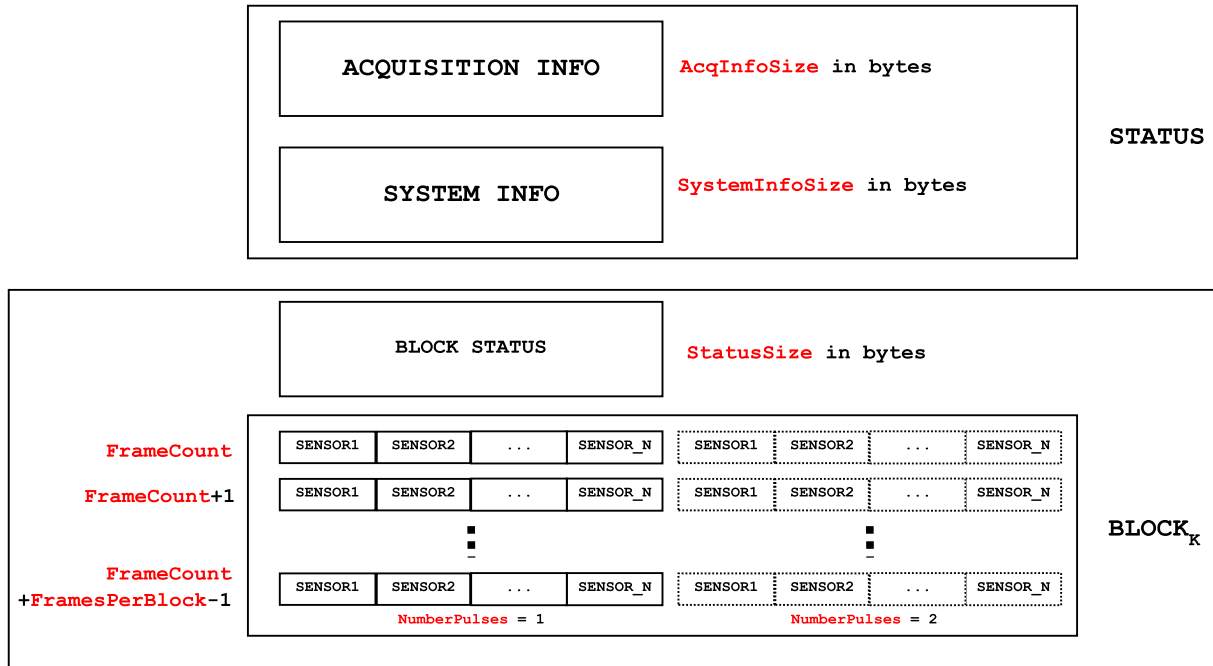
```

Ho = R2n+1 + Do
if( ( Ho - ho ) < 0 )
    if( (2^30 + (Ho - ho )) < 0 )) Do = Do + 2^32
else
    if( (2^30 - (Ho - ho )) < 0 )) Do = Do - 2^32
Ho = R2n+1 + Do
ho = Ho

```

Trigger Detection

The TDI-7000 data stream includes status information and sensor readings. This data stream is organized as **Frames** which are the sensor readings from one timing table pass, a **Block** which is a group of Frames, **Block Status** which applies to the block and a **Status** header which applies to the entire acquisition. One pass through the timing table can generate one or two optical pulses and every optical pulse returns one reading from every sensor.



This data and status information can be written to one or multiple files and grouped in different ways.

The status information includes the parameters and data required to detect and use the trigger. The numbers used are listed below.

<code>Fclock16</code>	master clock frequency times 16 in Hz.
<code>TLength</code>	number of timing table entries
<code>NumberPulses</code>	number of pulses per timing table pass
<code>fTriggerEvent</code>	trigger occurred flag
<code>TriggerCount</code>	time of trigger event in trigger clock counts modulo 2^{32}
<code>TriggerCountDivider</code>	trigger clock source
<code>FrameCount</code>	time of first frame in block in modulation cycle counts
<code>FramePerBlock</code>	number of frames in a block

Block Time

The Block Time is the time after RESET that is associated with the first frame of a block. It is simply the `FrameCount` in the block status divided by the modulation frequency.

The Modulation Frequency is derived from the master clock frequency and the timing table length.

$$F_{mod} = \frac{F_{clock16}/16}{4 \cdot TTLength} \quad \text{Hz}$$

The Block Time can now be computed.

$$BlockTime = \frac{FrameCount}{F_{mod}} \quad \text{seconds}$$

The Frame Count is a 64 bit integer and never overflows.

Frame Time

The Frame Time is the time after RESET that is associated with a frame in a block. It is the Block Time of the block which contains the frame plus the time offset of the frame in the block. The time between frames is the period of the modulation. Given frame J in block K , the Frame Time can be computed.

$$FrameTime_{KJ} = BlockTime_K + \frac{J}{F_{mod}} \quad \text{seconds}$$

Trigger Time

When the `fTriggerEvent` flag is set to one the `TriggerCount` value in the Block Status is the count modulo 2^{32} from RESET of the Trigger Clock. The Trigger Clock is the master clock divided by four divided by the `TriggerCountDivider` plus one.

$$TriggerClock = \frac{F_{clock16}}{64} \cdot \frac{1}{TriggerCountDivider + 1} \quad \text{Hz}$$

Since the `TriggerCount` is contained in a 32-bit counter it may overflow and lose track of the time from the RESET. The trigger time for a number of Over Flows is given by the lost number of overflows plus the trigger counter time.

$$TriggerTime = \frac{OverFlows \cdot 2^{32}}{TriggerClock} + \frac{TriggerCount}{TriggerClock} \quad \text{seconds}$$

The Trigger Time measures the number of seconds after RESET that the trigger event occurred.

The Trigger Clock is normally 1Mhz for a `TriggerCountDivider` equal to 24 which makes the maximum time before a `TriggerCount` overflow equal to 1.2 hours. For many applications, the Over Flows are zero.

In the general case, the number of Over Flows that have occurred can be found using the Block Time of the block in which the trigger event flag is set. The time difference between the Block Time and the Trigger Time cannot be more than the duration of a few blocks which is much shorter than the time lost from an Over Flow miscount.

$$|TriggerTime - BlockTime| \ll \frac{2^{32}}{TriggerClock}$$

This condition is met by finding the Over Flow count that meets it.

$$\left| \frac{OverFlows \cdot 2^{32}}{TriggerClock} + \frac{TriggerCount}{TriggerClock} - \frac{FrameCount}{Fmod} \right| \ll \frac{2^{32}}{TriggerClock}$$

The best case is that the Trigger Time is equal to the Block Time which can be forced by allowing a fractional Over Flow count solution.

$$OVERFLOWS = \frac{\frac{FrameCount}{Fmod} \cdot TriggerClock - TriggerCount}{2^{32}} \quad \text{Fractional count}$$

The Over Flows cannot be negative so if the fractional answer is less than zero, no overflows occurred.

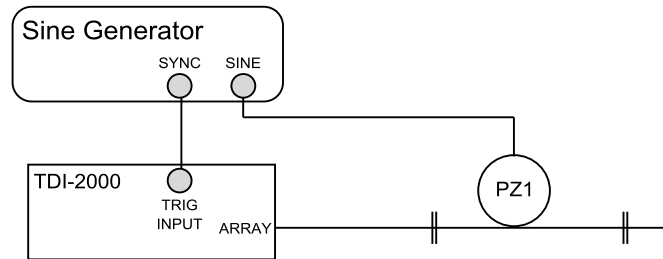
```
IF OVERFLOWS <= 0 THEN OverFlows = 0
```

A positive result implies the Over Flow count is either the integer floor or the integer floor plus one. The answer is the one that gives the lessor time difference between the Block Time and the Trigger Time.

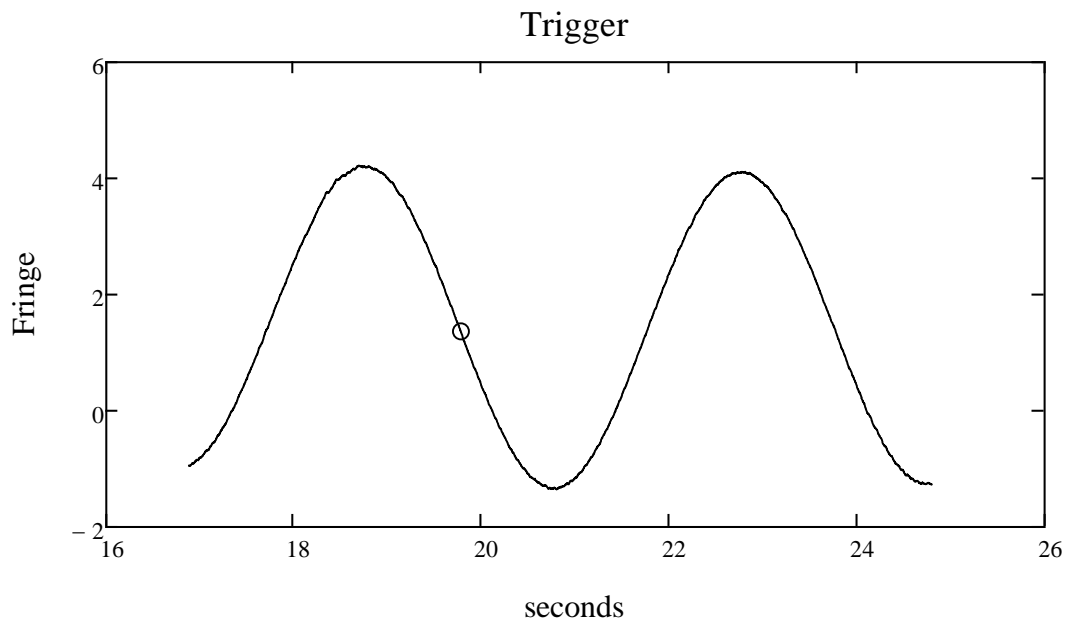
Sensor Time

The Frame Time is the seconds after RESET of a given set of sensor measurements associated with one or two optical pulses. The Trigger time is the seconds after RESET of the trigger event. The desire is to find the time at which a sensor encoded its measurement relative to the time of the trigger occurrence. This is a complex problem since it involves time of flight of the optical pulse to a sensor, the encoding time averaged over the pulse modulation, and various electrical offsets in the TDI system. The initial approach is to plot a sensor output versus the Frame Time and then superimpose the Trigger Time event on that graph and ignore the optical and electrical delays involved.

An example of a trigger setup is to have a SINE generator driving the trigger input of the TDI-7000 as its output voltage drives a PZ1 that is part of an interferometer.

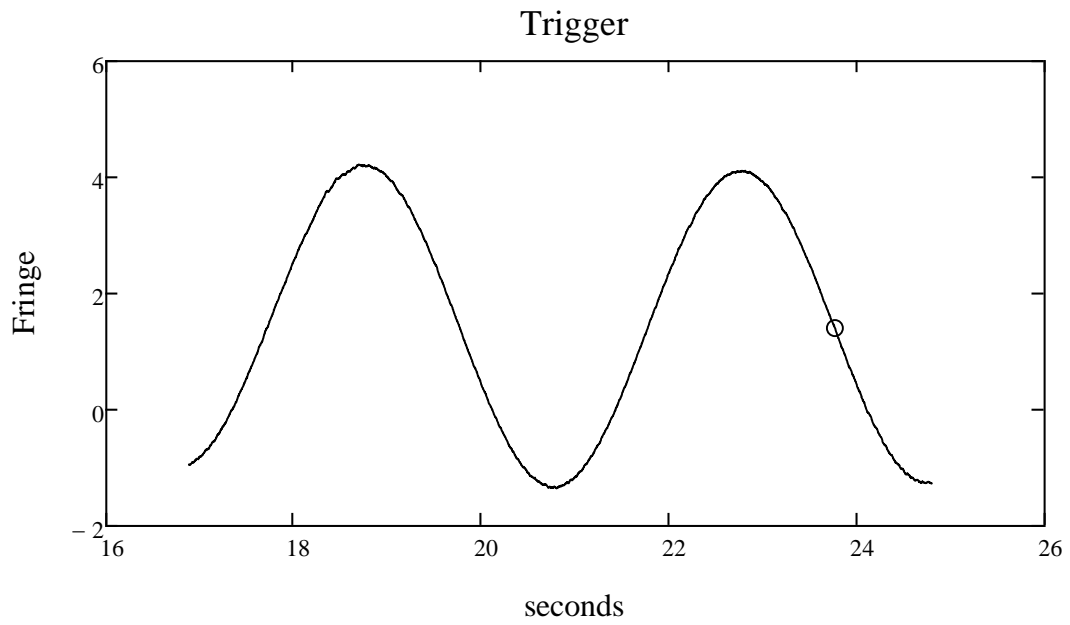


Such a setup was used to verify the trigger by collecting data and status which is shown below. The circle is the location of the trigger event which coincides with the generator SYNC position.



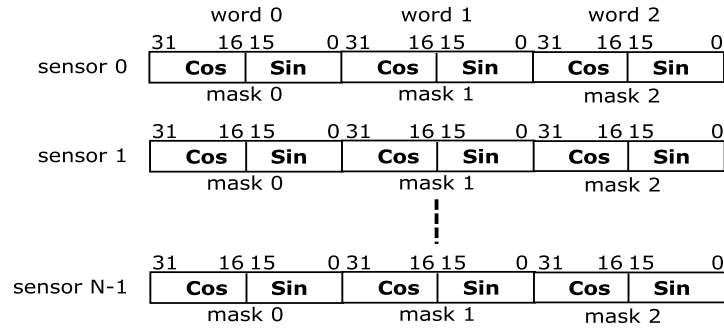
Eight blocks were taken with a 50Khz modulation frequency and 49664 frames per block which makes the data collection duration 7.95 sec. The trigger event flag was found to be set in the third block. The time base is the seconds from the RESET of the TDI. The data measurements started at 16.89 seconds and the trigger occurred at 19.8 sec.

The same data also had a trigger occur in the 7th block which is shown below. It occurred at 23.8 sec which is exactly 4 sec after the first trigger which is correct given that the sine had a $\frac{1}{4}$ Hz frequency.



Sine/Cosine Data Format

This is the raw data mode where the sine and cosine pairs for each mask are sent to the data interface.



This mode triples the required bandwidth of the Ethernet interface.

ADC Data Format

The ADC Data mode is a test mode used to align the timing table to the optical pulses. ADC data from the selected mask or sum is sampled at 50MHz and since two ADC readings are packed into one 32 bit word, the final data rate 25MHz. Since this is too fast to transmit continuously over the Ethernet link, the data is buffered in DDR memory up to the number of modulation cycles requested. Thus, the ADC Data data collection is performed on a one trigger basis. The 32 bit word for the ADC Data is shown in the table below.

Bit	Value
31	Timing table pulse enable bit from latch bit location
30	ADC out of range bit for this conversion
29:16	ADC 14 bit value next 2 master clocks from first value
15	Timing table slope bit from latch bit location
14	ADC out of range bit for this conversion
13:0	ADC 14 bit value latched by timing table

ELECTRICAL/OPTICAL INTERFACE

POWER **DC power inputs to TDI-7000**

Connector	MIL-DTL-38999, series III Subminiature cylindrical,D38999/20MH21PN box mount
Mate	D38999/26MH21PN cable plug Supplier: Avnet D38999/26MH21SN M85049/38S23N Cable Strain Relief Supplier: Avnet M85049/38S23N

Pin Assignments

PIN	SIGNAL	COLOR	AMPS	DESCRIPTION
L	RS232 RX	Orange		
M	RS232 TX	Violet		
N	RS232 COMMON	Black		
K	+24V_Fan	Blue	1.5	Fan supply
V	+24V_Fan RETURN	White		
J	+24V_Piezo	Blue		Piezo driver supply
H	+24V_P RETURN	White		
W	+5V_Digital RETURN	Black		
U	+5V_Digital RETURN	Black		
X	+5V_Digital	Red	3.2	DSU, EDFA, Laser and Pulser Digital
G	+5V_Digital	Red		
P	+5V_TEC RETURN	Gray		
T	+5V_TEC RETURN	Gray		
R	+5V_TEC	Orange	2.5	DSU, EDFA, Laser and Pulser TEC
S	+5V_TEC	Orange		
A	+5V_Analog	Red	2.75	DSU, Laser, Pulser and Piezo Analog
B	+5V_Analog	Red		
C	+/-5V_A RETURN	Black		
D	+/-5V_A RETURN	Black		
E	-5V_Analog	Brown	1.0	DSU, Pulser and Piezo
F	CHASSIS GROUND	Green		

EXTERNAL SOURCE **optional external light source input**

Connector	FC/APC Bulkhead receptacle narrow key
Mate	FC/APC Plug narrow key, SMF28 single mode fiber
Description	An external source can be connected to the TDI-7000 using this input connector when modifications have been made internally to route this optical input to the input of the internal optical pulser. Normally, this connector is not used.

SENSOR ARRAY **connection to sensor array**

Connector	E2000 to FC/APC Adapter Narrow Key, Green, Diamond USA PN 1018962 (old 165-311-907V022)
Mate	Diamond E2000 APC
Description	The sensor array is connected to this optical port. The sensor array must have spaced reflectors that match the internal compensator configuration and match the source wavelength.

RS232 auxiliary RS232 communications port

Connector DB9 bulkhead female with jackscrews

Mate DB9 male

Pin Assignments

PIN	SIGNAL	DESCRIPTION
1	NC	Not used
2	RxD	Receive data
3	TxD	Transmit data
4	NC	Not used
5	common	ground
6	NC	Not used
7	NC	Not used
8	NC	Not used
9	NC	Not used

Description The DSU controller board has an auxiliary RS232 port that is brought to this connector. Its use depends on the application.

TIMING MONITOR (optional) system timing signals for monitoring operation

Connector DB15 bulkhead male with jackscrews

Mate DB15 Female

Pin Assignments

PIN	SIGNAL	DESCRIPTION
1	Timing monitor 0-	Pulse-
2	Timing monitor 0+	Pulse+
3	Timing monitor 1-	Slope-
4	Timing monitor 1+	Slope+
5	Timing monitor 2-	Sample-
6	Timing monitor 2+	Sample+
7	Timing monitor 3-	Switch-
8	Timing monitor 3+	Switch+
9	Timing monitor 3	Switch
10	Digital ground	GND
11	Timing monitor 2	Sample
12	Timing monitor 1	Slope
13	Digital ground	GND
14	Timing monitor 0	Pulse
15	Digital ground	NC

Description LVDS and 3.3V logic digital timing signals monitor the operation of the DSU

Pulse high when optical pulse is on

Slope low when even slope, high when odd slope

Sample ADC sample strobe low to high

Switch Table Timing output bit for general use

PDR AND MODULATION (optional) receiver monitors and modulation sine monitor

Connector DB9 bulkhead male with jackscrews

Mate DB9 Female

Pin Assignments

PIN	SIGNAL	DESCRIPTION
1	Not used	Not used
2	Modulation out	-4V to +4V modulation sine monitor
3	PDR CH0	Receiver 0 monitor 0 to 2V
4	PDR CH1	Receiver 1 monitor 0 to 2V
5	PDR CH2	Receiver 2 monitor 0 to 2V
6	Modulation return	Ground for pin 2
7	PDR CH0 return	Ground for pin 3
8	PDR CH1 return	Ground for pin 4
9	PDR CH2 return	Ground for pin 5

Description The three receivers of the polarization diversity receiver are brought out to this connector for monitoring along with the modulation sine monitor.

10/100/1000 Ethernet communications interface

Connector RJ45 CAT5 receptacle

Mate CAT5 Plug

Description The main communication interface for the TDI-7000 is Ethernet 10/100/1000.

ANALOG ANGLE (optional) analog monitor of one sensor

Connector Isolated BNC jack

Mate BNC male plug

Pin Assignments

Center is analog output and shell is ground

Description One sensor can be assigned to this output and the digital angle is scaled by software selection to this -3.2V to +3.2V analog output.

SPECIFICATIONS

General Characteristics	
Operational Wavelength	ITU35 [1549.32] standard for single grating array, C-band or L-band option for multiple wavelength arrays
Optical Source Type	External Cavity Laser, spectral width ~5 – 10 KHz
Interrogation Method	Interferometric TDM, Optiphase Dual Slope method
Interrogation Controls	Frequency, Pulse Width, Receiver Gain
Sensor Array Interface	Single, there is a practical limit for IIR configurations due to crosstalk as to how many sensors can be on a single fiber at the same wavelength. Array fan-out approaches are assumed for high channel counts and dual arrays
Sensor Array Length	Up to 25Km, assuming sensor return loss >37dB
Sensor Resolution	Minimum length 20.44m / maximum length 200m, all sensors must be same length. Longer lengths are an option
Compensator	Path mismatched Michelson, must match sensor length
Interrogator Sensor Count	Up to 256 sensors
Concurrent Event Detection	Sensors work independently
Background Noise	Constrained to sensor length

Measurement Characteristics	
Interrogator Read Rate	1.75 million per second total for interrogator
Data Interface	10/100/1000 Ethernet required for high rates
Modulation Frequency	12KHz to 150KHz in restricted bands (PZ1 response)
Sensor Read Rate	One or two reads per modulation cycle
Data Formats	32-bit or 64-bit per sensor, ADC, sin/cos
Self Noise	40urad/rt-hz, does not include noise caused by telemetry, Coherent Rayleigh or Laser
Linearity	Better than 0.5%

Interfaces	
Optical Interface	Array interface FC/APC or E2000 optional
Electronic Interfaces	Gigabit Ethernet for control and I/O, optional analog and digital monitors
Software Interface	Microsoft .NET Framework based Application Program Interface (API) for setup, control and data acquisition; Windows based GUI for device configuration, data acquisition, and system maintenance.

Physical & Environmental	
Power	Internal Universal or external DC supply Box
Operating Temperature	0 to 60 C
Storage Temperature	-40 to 85C
Dimensions	3U Full Rack 20" deep excluding power supply
Weight	22 Lbs (10Kg) excluding power supply
Requirements	Windows PC with Gigabit Ethernet