

OIF Tunable Laser Projects

Abstract: *This paper describes the effort at the OIF to develop electrical, optical, and mechanical specifications for continuous wavelength (CW) tunable laser modules.*

Executive Summary

The mission of the Optical Internetworking Forum (OIF) is to foster the development of standards enabling deployment of interoperable products and services that use optical networking technologies.

Tunable lasers are an enabling technology that will produce savings for service providers in both operating expenses and capital expenses. The early deployments of tunable laser technologies have generally focused upon cost savings resulting from an inventory reduction of line cards for sparing. The next generation of networks will rely heavily upon tunable lasers. These new transparent dynamic network architectures will result in further expense reduction as well as enabling significant new revenue streams through rapid provisioning.

Over the past several years, tunable lasers have matured to the point where they meet the performance and reliability requirements for network deployment.

The OIF has completed two tunable laser projects. The first project resulted in the *Tunable Laser Implementation Agreement*, OIF-TL-01.1 released in November 2002. It addressed the communication protocol, electrical interface and mechanical form factor interoperability for tunable continuous wavelength (CW) lasers. The document serves as a roadmap for future tunable device implementation agreements.

In February 2003, the OIF began a new fast track project, the *Tunable Laser MSA Implementation Agreement*. This MSA-IA builds upon the existing *Tunable Laser Implementation Agreement*, generating a more comprehensive specification of the optical, electrical, mechanical, and communication protocols.

An OIF interoperability demonstration of the tunable laser communication protocol was performed at OFC 2003 and at Supercomm 2003.

The Tunable Laser MSA Implementation Agreement has been adopted by a consortium of tunable laser vendors and users. This effort can be viewed at <http://TunableLaserMSA.com>.

Overview of the Tunable Laser Implementation Agreements

An overview of the implementation agreement features is presented in the following sections. For more information, please consult the OIF website for the latest version of the tunable laser implementation agreements.

Design Objectives and Features

The Tunable Laser-IA was created with the following design guidelines in mind.

- **Configurable:**

The tunable laser's behavior shall be configurable such that it can be used in a number of different network equipment manufacturer's line card architectures and design philosophies.

Some examples:

1. Three physical interface options are provided: I2C, RS232, SPI
2. Alarms and service request behaviors are configurable as to trigger conditions as well as latching/non-latching behavior.
3. Module reset on power on condition is configurable.

- **Efficient and Robust Communication Interface:**

The command interface shall be robust and consume a relatively small fraction of the bus bandwidth. This allows interfaces such as 9600 baud RS232 to provide timely control of the module in the event of a SONET/SDH protection switch. The efficient interface also provides for high bus utilization for highly shared buses such as SPI and I2C.

Some examples:

1. The command interface consists of only 4 bytes per command. For instance, ~4ms is required to send a set channel command over RS232 and ~8 μ s for a 4Mb/s SPI interface. Many other command interfaces require 7 or more bytes to transmit the same command.
2. The interface keeps all communications transactions relatively short such that commands can be interlaced. For example, a SONET/SDH protection event can be handled in a timely manner while the module's firmware is being uploaded.

- **Extensible Platform for Communication Interfaces:**

The Tunable Laser-IA communication protocol is designed around 3 independent layers:

1. Physical Layer: There are 3 physical interfaces to choose from. (I2C, RS232, SPI)
2. Transport Layer: Reliable data exchange (framing, checksums and error reporting, pacing)
3. Application Layer: General and specific commands

These features allow other tunable devices (such as tunable transmitters or receivers) to use the same protocol.

Optical Application Spaces

The implementation agreement support a number of application spaces. The application spaces are divided into two separate vectors:

- Tuning speed
- Optical performance requirements.

Each of the application space vectors contains three options. The application space matrix is shown in Table 1.

Table 1: Optical Specification Requirement Matrix

Optical Specification Matrix			Application Requirement (Tuning Speed)		
			A	B	C
Application Requirement	1	Ultra Long Haul	SONET/SDH Protection	SONET/SDH Restoration	Provisioning & Sparing
	2	Long Haul	A1	B1	C1
	3	Metro	A2	B2	C2
			A3	B3	C3

Physical Interfaces

The interface to be used is determined at power up or hard reset by the state of the IOMode and IOMode1 pins for the tunable laser module. Note that for a given application, the manufacturer need only implement the physical interface required by the user application.

The module uses a forty pin connector as shown in Figure 1 and pin assignments in Table 2.

Table 2: Pin Assignments

Pin #	Pin	Pin #	Pin	Pin #	Pin	Pin #	Pin
1	Vcc	11	GND	21	TDI	31	Mfg Spec
2	DIS*	12	ALM*	22	IOMODE	32	Reserved
3	Vcc	13	GND	23	TMS	33	Mfg Spec
4	MS*	14	TxD	24	IOMODE1	34	Reserved
5	Vcc	15	GND	25	TCK	35	Mfg Spec
6	RST*	16	RxD	26	A0	36	Reserved
7	Vcc	17	TRST (JTAG)	27	ADITHER	37	Mfg Spec
8	SRQ*	18	IRDY*	28	A1	38	Reserved
9	GND	19	TDO	29	Mfg Spec	39	Mfg Spec
10	FATAL*	20	IOCLK	30	A2	40	Reserved

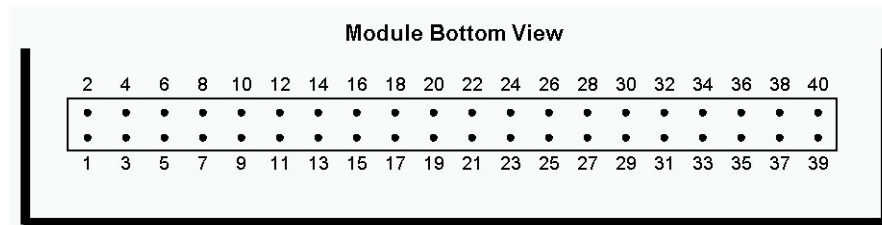


Figure 1: Connector Pin Numbering

Module Signaling Lines

The module has three hardware lines to signal its status.

- FATAL*
- SRQ*
- ALM*

The FATAL* line is used to signal fatal conditions which typically will cause shutdown of the optical output. The FATAL*, once asserted, remains asserted until the status register is cleared.

The SRQ* line is used to signal fatal conditions, warning conditions, or other module service request needs such as an execution error (XE) for a command processing in the background (pending operation) or a communication error (CE) which occurs on the SPI interface. The SRQ* line, once asserted, remains asserted until the status register is cleared.

The ALM* line is used to signal a warning condition. It remains asserted only during the time that the specified conditions occur. Due to the transient nature of the ALM* line, it is recommended to use the SRQ* to signal errors. The default module configuration specifies the ALM* to function as a LOCKED line.

Transport Layer

The transport layer encapsulates the command and response packets to form a 32-bit frame. Figure 2 and Figure 3 depict the in-bound and out-bound frames. The transport layer is responsible for the fields in white. The application layer is responsible for the shaded fields below.

Figure 2: In-Bound (Host to Module) Frame

31	30	29	28	Bits 27:0			
Checksum				Command packet being framed			

Figure 3: Out-Bound (Module to Host) Frame

31	30	29	28	27	26	Bits 25:0	
Checksum				CE	Response	Response packet being framed	

Each in-bound and out-bound packet contains a four bit checksum. The checksum is computed over all the bits being encapsulated using a BIP-4 checksum. A CRC-16 is also available.

Application Layer

Communication Overview

The application layer also provides the ability for registers to hold two object types: either 16 bit integers or an arbitrarily long sequence of bytes (multi-byte response). The command response identifies the response as either a 16-bit integer or multi-byte response.

The following diagram (Figure 4) depicts the communication process. The application layer also provides the ability for registers to hold two object types: either 16 bit integers or an arbitrarily long sequence of bytes (multi-byte response). The command response identifies the response as either a 16-bit integer or multi-byte response.

Assume the host has a request to transmit to the module (Host's Request). The request is first encoded as a 28-bit command packet in the Host Driver's application layer. The command is then framed as a 32-bit packet in the host driver's transport layer. The framing operation includes the addition of a BIP¹-4 checksum. Finally, the host driver's physical interface (RS232 shown) encodes the 32 bit packet as four, ten bit RS232 "characters" and transmits in across the TxD line to the module.

The module's physical layer receives 40 bits and de-codes them by removing the RS232 start and stop bits. The resulting 32-bit frame is delivered to the transport layer where checksum is checked for consistency. Assuming no error is generated, the 28-bit command packet is delivered to the module's application layer where the command is decoded and executed.

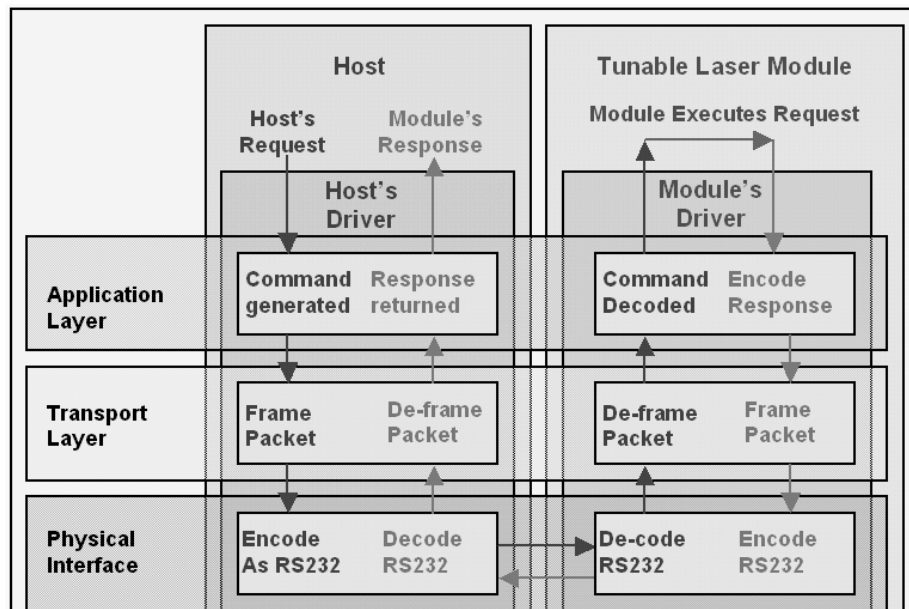


Figure 4- Three Layer Communication Diagram

The command execution will generate a response when complete. The response packet consists of 26 bits.

The response packet is delivered to the module's transport layer which frames the packet by pre-pending a checksum, communication error (CE), and a response flag. The resulting 32-bit packet is then delivered to the module's physical layer where it is then encoded as 40 bits.

The host then receives the four RS232 characters and performs the inverse operations as the packet moves up the host's layer hierarchy.

Command Overview

The application layer's command set has been divided into two groupings:

- The first group of commands deals with general module behavior as well as providing a way to determine what kind of module is present (tunable laser, tunable transmitter, etc.). These commands are independent of device type.

¹ Bytes in parallel.

- The second group of commands is device type specific. This OIF-IA describes the “CW Laser” device type.

The commands to the module consist of a 9-bit operation followed by 2 bytes of optional data. Alternatively, the command can be thought of as one read/write bit followed by an 8-bit register number followed by 2 bytes of optional data. See Figure 5. The register paradigm is used in the implementation agreements.

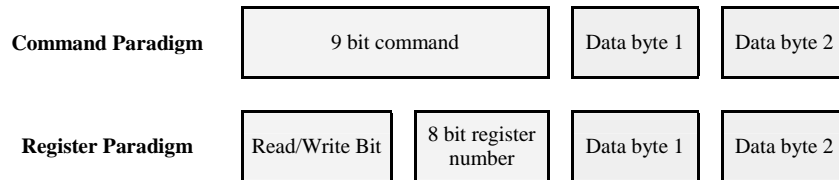


Figure 5: Paradigms for Module Control

There are 256 directly accessible registers (0x00 to 0xFF) in the primary register address space. The OIF-IA allocates the first 32 registers (0x00 to 0x1F) for generic module operations for all module types. Another 96 registers (0x20-0x7f) are reserved for device type “CW Laser”. Finally, the remaining 128 registers (0x80-0xFF) are provided as manufacturing specific registers.

Command Execution Overlap

The application layer provides support for pending operations especially useful for operations that can take a significant period of time to complete.

If a command is issued to the module that results in a long time to complete, the module will return a response packet within the specified time out period for the module and flag the operation as pending. The interface is now free to respond to additional commands. The host can determine when the pending operation completes by polling the NOP register (0x00). The NOP register returns the pending operation status as well as any error conditions.

Extended Addressing

Extended addressing provides an additional memory space (22 address bits) in addition to the primary 256 registers (8-bit address space).

The extended addressing feature consists of three registers described in Table 3.

Once the configuration and address registers are configured, the host may issue a series of read or write commands to the (indirect contents register) thereby accessing the memory location pointed to by the indirect register. The locations may map to physical or virtual memory spaces.

Table 3: Extended Address Register Description

Register	Description	Fields
Configuration	Defines basic configuration for the extended address	Defines the <ul style="list-style-type: none"> ▪ address space ▪ high order address bits
Address	Address of field in either physical or virtual memory space	Defines the 16 low order address bits
Contents	Reading from this register returns data stored in this field 16 bits at a time Write to this register stores data into this field 16 bits at a time	16 bit data value

The configuration register and address registers are usually pre-configured when one of the primary registers is accessed which holds an object longer than a 16-bit integer.

Data Types

All of the general registers hold 16-bit data values or serve as pointers to a sequence of bytes (extended addressing mode). All values are stored as big endian, two's complement .

- Two Byte Data Values
Data is represented in the registers as either signed or unsigned 16 bit integers. Note that single byte values would be stored with the appropriate leading zeros. Real values are stored with an implied decimal point location. For instance, the value "12.3 dBm" would be stored as 12310 in a field and has an implied formatting of one decimal place.
- Multi-byte Fields
Fields holding data longer than 16 bits are stored as a sequence of bytes and accessed through the extended addressing register. ASCII strings are terminated with a null. Note that the extended address register allows the host to read beyond a null termination but not beyond the maximum field size. Integers, floats, or structures are stored as a sequence of bytes .

Execution Error Field Conditions

The reason for an execution error (XE) can be determined by reading the NOP/Status register (NOP 0x00). Bits 3:0 encode the error field value.

Register Lockout for Write Access

Registers are classified as falling into one of four lockout levels as shown in Table 4. The lockout feature provides a degree of protection from altering registers which impact network traffic.

Table 4: Register Lock Levels

Lock Level	Description	Key Value	Key Length
0	Fully locked (all lockable registers are write protected)	0x0	1
1	Partially unlocked – Lockable registers set 1 are unlocked Available for channel tuning etc.	MFG Specific	<20
2	Partially unlocked – Lockable registers set 2 are unlocked Available for Reset, firmware downloads, alarm thresholds, etc. (Basic module configuration)	MFG Specific	<20
3	All lockable registers are unlocked Normally reserved for manufacturer	MFG Specific	<20

Registers which are locked out can be unlocked for writing by writing a key (a series of unsigned characters) to the Lock register (0x16) through the AEA mechanism.

Communication Error Detection

Communication error detection occurs on the module and host sides of the communication interface.

The module examines the in-bound packets (host to module) to see if the checksum or the optional CRC-16 is consistent. An inconsistency results in an unprocessed response packet with the CE flag asserted in the out-bound packet. When the host observes the CE flag, the last out-bound packet should be resent.

The host examines the response packets for consistency by checking the checksum and the optional CRC-16 for the out-bound packet (module to host). If either the CRC or checksum is inconsistent, the host may request the module's last response to be retransmitted by reading the LstResp (0x13) register.

Execution Error Detection

Execution errors occur when the module is unable to execute the requested command. The module encodes the XE flag bit (execution error flag) in the response packet. When the host detects an XE flag in the response packet, it can read the NOP (0x00) register to determine the error field condition.

Non-Volatile Default Configuration

The command interface allows the current module configuration to be saved as the default configuration. The default configuration is restored upon hard reset or upon power up. In the event of loss of power or hard reset during a save configuration request, the module's default configuration will remain unchanged.

*Author:: Jeff Hutchins
Iolon, Inc.
jeff@iolon.com
408-952-6923*