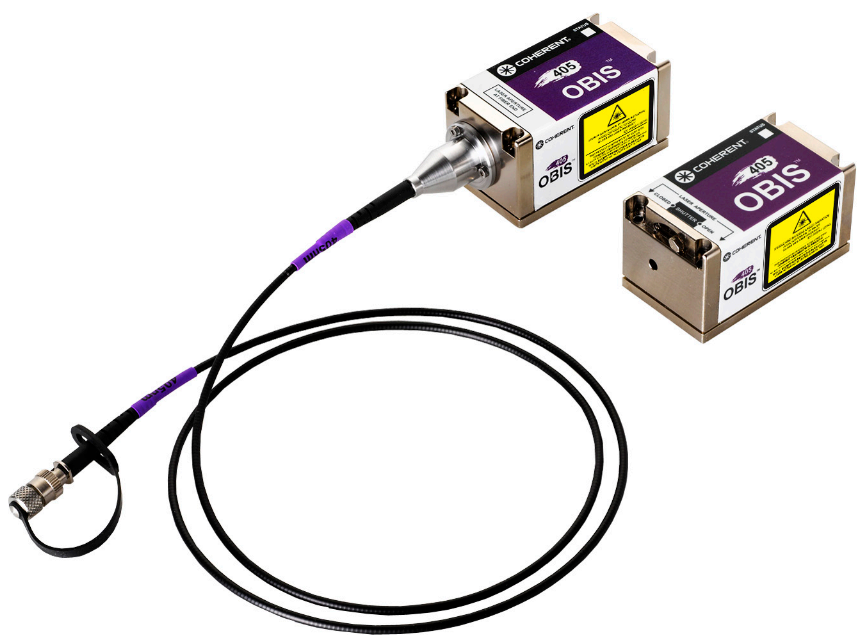
OBIS LX/LS Lasers Description & Operation Part 3 of 3

Operator's Manual



Superior Reliability & Performance

Operator's Manual
OBIS LX/LS Lasers Description & Operation
Part 3 of 3 - Interface & Commands

Coherent, Inc.
26750 SW 95th Ave.
Wilsonville, Oregon 97070
USA

Copyright

Patents referenced in this manual are active as of the date this manual is printed. For a list of current patents, see www.coherent.com/patent.

Every effort has been made to make sure that the data shown in this document is accurate. The information, figures, tables, specifications, part numbers, and schematics contained herein are subject to change without notice. Coherent makes no warranty or representation, either expressed or implied, with respect to this document. In no event will Coherent be liable for any direct, indirect, special, incidental, or consequential damages caused by any defects in its documentation.

Coherent product information and related software is now available in one easily accessible location on the Coherent website. Filter your search by product type, document category, or both. To download manuals and software, go to:

https://www.coherent.com/resources

Should you experience any difficulties with your laser or need product or technical information, contact Coherent as follows:

- By email: customer.support@coherent.com
- Visit our website: www.Coherent.com
- By phone: +**1-(734)-456-3100**

See "Appendix - Service & Support" (p. 105) for more information and detailed instructions.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1 PREFACE

The Operator's Manual presents the following information for OBIS LX and OBIS LS laser systems:

- **Part 1** — Description of the laser, installation instructions, operations, and information related directly to the laser module
- **Part 2** — Descriptions of the components, controllers, and accessories for the OBIS LX/LS laser systems
- **Part 3** — Supporting information about the interface protocol and host commands for OBIS LX/LS laser systems

WARNING!
Use all controls, adjustments, and procedures for set-up and operations as specified in Parts 1 and 2 of the *OBIS LX/LS Operator's Manual*. Failure to do so can cause dangerous radiation exposure.

This user information reported in this manual is in compliance with the following standards for Light-Emitting Products EN/IEC 60825-1 *"Safety of laser products – Part 1: Equipment classification and requirements"* 21 CFR Title 21 Chapter 1, Sub-chapter J, Part 1040 *"Performance standards for light-emitting products"*.

1.1 Safety Warnings

Anyone setting up or operating the an OBIS laser must first read and understand safety information prior to beginning any tasks.

CAUTION!
Read about laser safety features before operating the laser for the first time.
Pay special attention to the "Laser Safety" section in Part 1 of the *OBIS LX/LS Operator's Manual*.

This section provides information about signal words and safety symbols that you need to know before you begin.

This documentation may contain sections in which particular hazards are defined or special attention is drawn to particular conditions. These sections are indicated with signal words in accordance with ANSI Z-535.6 and safety symbols (pictorial hazard alerts) in accordance with ANSI Z-535.3 and ISO 7010.

1.1.1 Signal Words

Four signal words are used in this documentation: **DANGER**, **WARNING**, **CAUTION** and **NOTICE**. These signal words designate the degree or level of hazard when there is the risk of injury, as described in Table 1:

Preface Table-1. Signal Words

Signal Word	Description
DANGER	Indicates a hazardous situation that, if not avoided, WILL result in ***death or serious injury***. This signal word is to be limited to the most extreme situations.
WARNING	Indicates a hazardous situation that, if not avoided, COULD result in ***death or serious injury***.
CAUTION	Indicates a hazardous situation that, if not avoided, could result in ***minor or moderate injury***.
NOTICE	Indicates information considered important, but not hazard-related. The signal word “NOTICE” is used when there is the ***risk of property damage***.

Messages relating to hazards that could result in both personal injury and property damage are considered safety messages and not property damage messages.

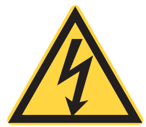
1.1.2 Symbols

The signal words are always emphasized with a safety symbol that indicates a special hazard, regardless of the hazard level. The icons are intended to alert the operator as described in Table 2:

Preface Table-2. Safety Symbols

Icon	Alerts the operator to…
	This symbol is intended to alert the operator to important operating and maintenance instructions.
	Important notes or instructions for operation and maintenance.

Preface Table-2. Safety Symbols

Icon	Alerts the operator to…
	Danger of exposure to hazardous visible and invisible laser radiation.
	Danger of susceptibility to Electro-Static Discharge (ESD).
	Dangerous voltages within the product enclosure that may be of sufficient magnitude to constitute a risk of electric shock.

1.2 Export Control Laws

It is the policy of Coherent® to comply strictly with export control laws of the United States of America (USA).

Export and re-export of lasers manufactured by Coherent are subject to U.S. Export Administration Regulations, which are administered by the Commerce Department. In addition, shipments of certain components are regulated by the State Department under the International Traffic in Arms Regulations (ITAR).

The applicable restrictions vary depending on the specific product involved and its destination. In some cases, U.S. law requires that U.S. Government approval be obtained prior to resale, export or re-export of certain articles. When there is uncertainty about the obligations imposed by laws in the USA, clarification must be obtained from Coherent or an appropriate agency of the U.S. Government.

For products manufactured in the European Union, Singapore, Malaysia, Thailand: These commodities, technology, or software are subject to local export regulations and local laws. Diversion contrary to local law is prohibited. The use, sale, re-export, or re-transfer directly or indirectly in any prohibited activities are strictly prohibited.

Declaration of Conformity certificates are available upon request.

1 OBIS INTERFACE & COMMANDS

This section introduces the interface protocol and host commands for OBIS LX/LS laser systems.

Information is provided in the following sections for these topics:

- Coherent Connection software (p. 2-17)
- Host Interface Commands (p. 3-29)
- OBIS RS-485 Interface (p. II-69)

Coherent product information and related software is now available in one easily accessible location on the Coherent website. Filter your search by product type, document category, or both. To download manuals and software, go to:

https://www.coherent.com/resources

To contact Coherent Technical Support, see “Appendix - Service & Support” (p. I-105).

2 Coherent Connection

This section describes how to set up and install the Coherent Connection software and related drivers for the OBIS Laser system.

Coherent Connection software supports the following laser products: OBIS LX, OBIS LX-SF (Single-Frequency), OBIS LS, OBIS CORE LS, OBIS LG, OBIS XT, OBIS CellX, StingRay, and BioRay.

Coherent product information and related software is available in one easily accessible location on the Coherent website. To download, go to:

https://www.coherent.com/resources

Through this software, you can control laser power or other parameters directly through a USB or RS-232 connection.

- Coherent Connection software (p. 2-19)
    - System requirements (p. 2-20)
    - Main tabs (p. 2-25)
- Remote control via USB and RS-232 (p. 2-17)
    - Connect USB/RS-232 for remote control (p. 2-18)
    - Connect USB at the laser (p. 2-18)

For information about using a terminal program, see 'OBIS Communications through a Terminal Program' in Part 1 of the *OBIS LX/LS Operator's Manual*.

2.1 Remote Control via USB or RS-232

Through the Coherent Connection software, you can control laser power or other parameters directly through a USB or RS-232 connection.

- USB and RS-232 use the same syntax, commands, and queries.
- When both USB and RS-232 are connected to the OBIS Remote, the USB overrides the RS-232.

To install Coherent Connection software, you must first connect the OBIS laser system to a workstation (personal computer or laptop) using a USB cable or a standard DB9F RS-232 connection.

- The USB cable is included in the OBIS Laser System.
- The RS-232 cable is a standard PC serial cable (not included with the laser).

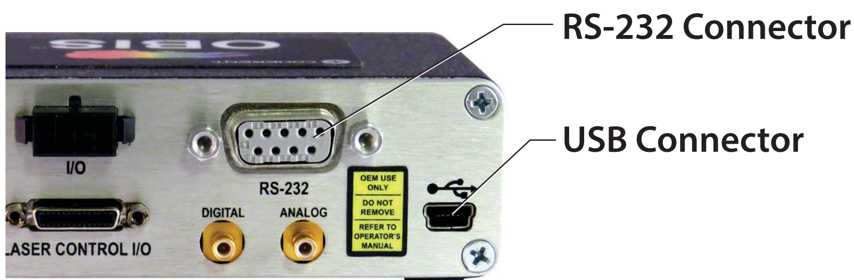
For information about RS-232 pin-out or communication settings, see Part I of the *OBIS LX/LS Operator's Manual)*.

Using the OBIS USB driver allows communication with the OBIS using a terminal program or a custom-developed program. The driver creates a virtual OBIS COM device in the host computer that gives access to its controls.

NOTICE
When installing Coherent Connection software, drivers are automatically loaded onto the host computer as part of the installation process.

2.1.1 Connect USB/RS-232

Connectors for a USB or RS-232 cable are located on the back panel of an OBIS Remote, as shown in Figure 2-1.



OBIS Remote Back Panel

Figure 2-1. Connectors for a USB or RS-232 Cable

Connect a standard serial cable from he back of the OBIS Remote to the host workstation (PC or laptop).

2.1.2 Connect USB

Figure 2-2 shows the USB connector on the back panel of the OBIS laser. This is a standard Mini-B USB connector that supports USB 2.0 communications.

Connect the OBIS LX/LS laser from the back panel of the laser to a USB port on the host computer.

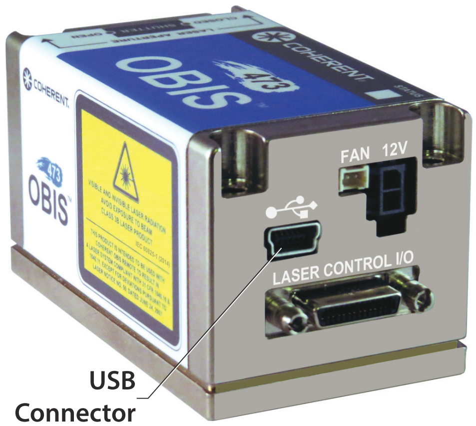


Figure 2-2. USB Connection at the Laser

IMPORTANT

DO NOT make a connection to the USB connector on the back panel of a OBIS Remote. Instead, the connection must be made to the USB connector on the OBIS *Laser*.

2.2 Coherent Connection Software

Coherent Connection provides an easy-to-use interface between a Coherent OBIS Laser or mini-controller and a PC.

Coherent Connection software lets a user set modes, change laser output power, and get laser status and information in its graphical user interface (GUI). The software supports both OBIS LS and OBIS LX lasers.

This section lists the system requirements, introduces the main tabs in the software, and provides instructions to install the software.

NOTICE

When installing Coherent Connection, drivers are automatically loaded onto the host computer as part of the installation process.

2.2.1 System Requirements

It is recommended that you use the most current and robust systems possible. Support for the OBIS laser system is provided on Windows v10 (32- and 64-bit) operating systems: In addition, the workstation must meet the following minimum requirements:

- 512 MB of RAM
- Microsoft .NET Framework 4.0 or higher. If no version (or an older version) is found on the workstation, then the installation program installs a version of Microsoft .NET Framework.
- USB or RS-232 port

2.2.2 Install Software

NOTICE
Before you install Coherent Connection software, it is recommended that you first close all other applications. The installation requires that you restart the workstation when installation is complete.

To install the Coherent Connection software and related drivers:

1. Coherent product information and related software is available in one easily accessible location on the Coherent website. Go to Coherent.com/Resources to search for and download the software:

   https://www.coherent.com/resources

2. Close all programs.
3. Double-click the following file to start the installation process. The last two digits represent the number for the current software build.

   `Coherent_Connection_Setup_v4.0.0.xx`

   The following message is displayed. Available languages include English, Italian, French, German, Hebrew, and Japanese. Note that the language selection applies only to software set-up instructions on-screen, and not to the Coherent Connection software itself (available in English only).
4. From the drop-down menu shown in Figure 2-3, select the language in which to display the software and click OK.

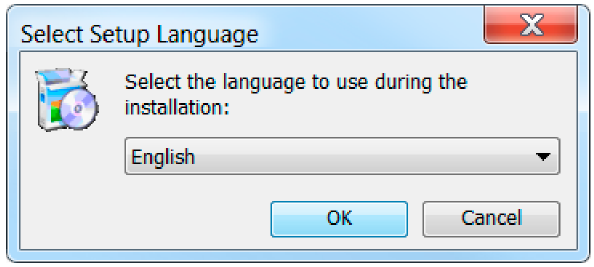


Figure 2-3. Select Language for Software

5. If you had previously installed the Coherent Connection software, the message shown in Figure 2-4 is displayed. Click Yes to proceed.

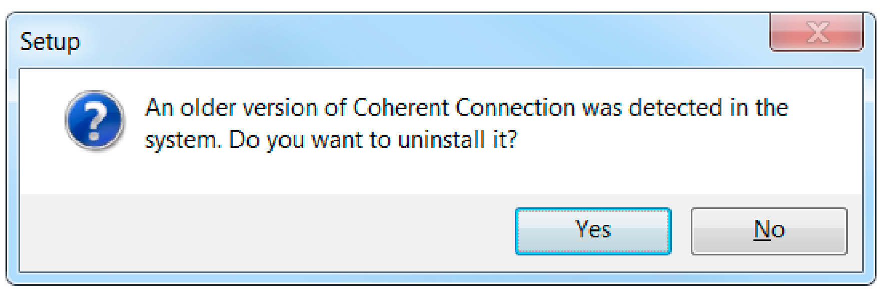


Figure 2-4. Uninstall Old Version of Software

6. The Welcome screen like the one shown in Figure 2-5 displays.

Figure 2-5. Welcome Screen for Installation

7. Read the instructions, then click Next. The License Agreement shown in Figure 2-6 is displayed.

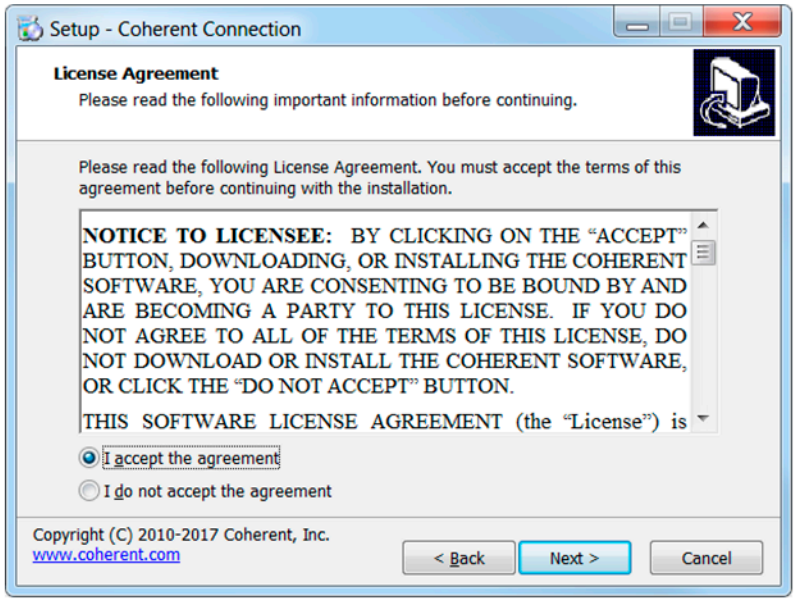


Figure 2-6. Coherent Connection 4 License Agreement

8. Scroll down to read the agreement. Note that the Next button is grayed out until you click the radio button to **Accept** the terms and conditions. When you do that, the button is activated; click Next.

9. The window shown in Figure 2-7 is displayed. Accept the selection, or browse to select the directory on the workstation where you want to install the software, and click Next.

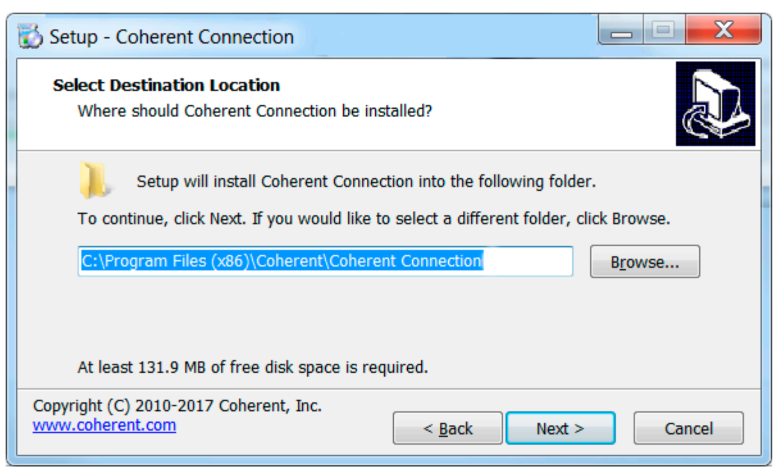


Figure 2-7. Select Directory to Install Software

10. You can create an icon for the software either on your desktop or for a Quick Launch (or both). As shown in Figure 2-8, click the appropriate check box, and then click Next.

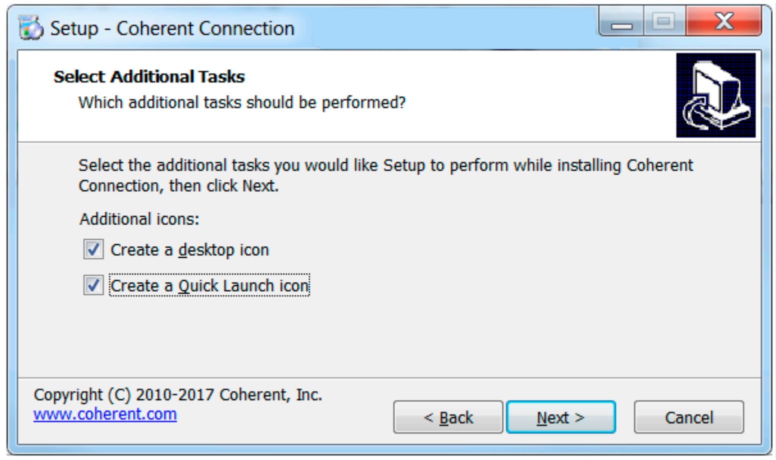


Figure 2-8. Set Desktop or Quick Launch Icon

11. The set-up utility is now ready to begin installing Coherent Connection 4 software on your workstation. Review the location and icons, as shown in the example in Figure 2-9, and then click Next.

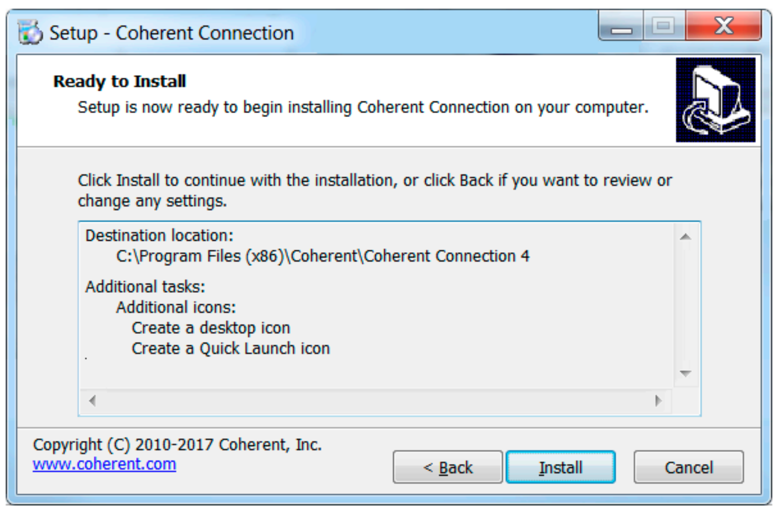


Figure 2-9. Review Set-Up before Installation Begins

A progress bar displays, as shown in Figure 2-10.

12. During the installation process, some files are extracted, as shown in the example in Figure 2-11.

13. After all files are extracted, click Finish. The screen shown in Figure 2-12 closes and the software is ready to be launched.

The software and USB driver are now installed.

If you selected a short-cut (icon) to be set up during installation, that is now displayed on the desktop of your workstation and/or in the Quick Launch menu, as shown in Figure 2-13:

To access complete operating instructions, open the Coherent Connection software and click **Help**.

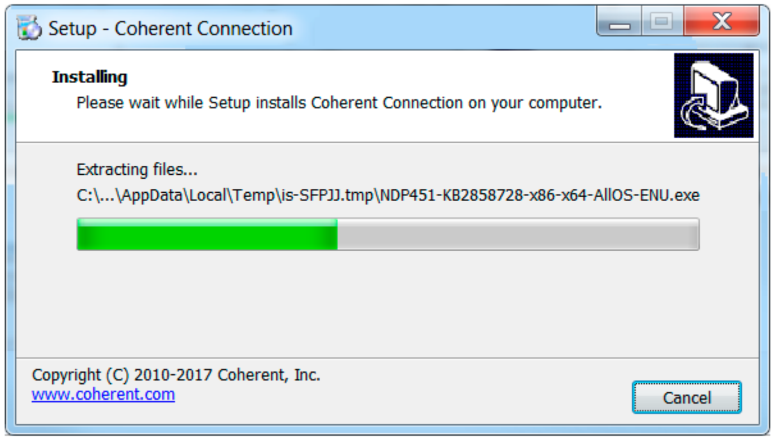


Figure 2-10. Progress of Installation

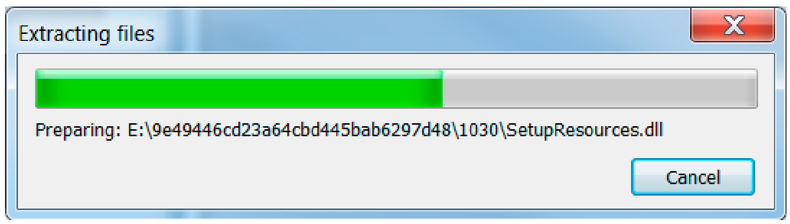


Figure 2-11. Extracting Files

Figure 2-12. Finish the Software Installation

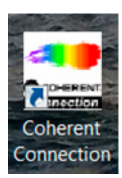


Figure 2-13. Desktop Icon for Coherent Connection Software

- Click on the icon for the Main menu to display the options in the drop-down menu.
- Click the Help icon to display the embedded Help file.

The Help menu option is shown in Figure 2-14.

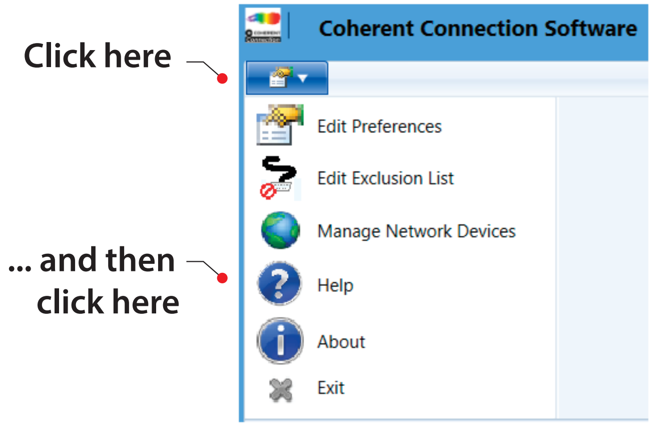


Figure 2-14. Coherent Connection HELP Menu Option

2.3 Overview of the Main Tabs

This section provides a brief description of each of the tabs in the Coherent Connection Software.

Figure 2-15 shows the Operating Power tab. On this page of the software, you can set power levels.

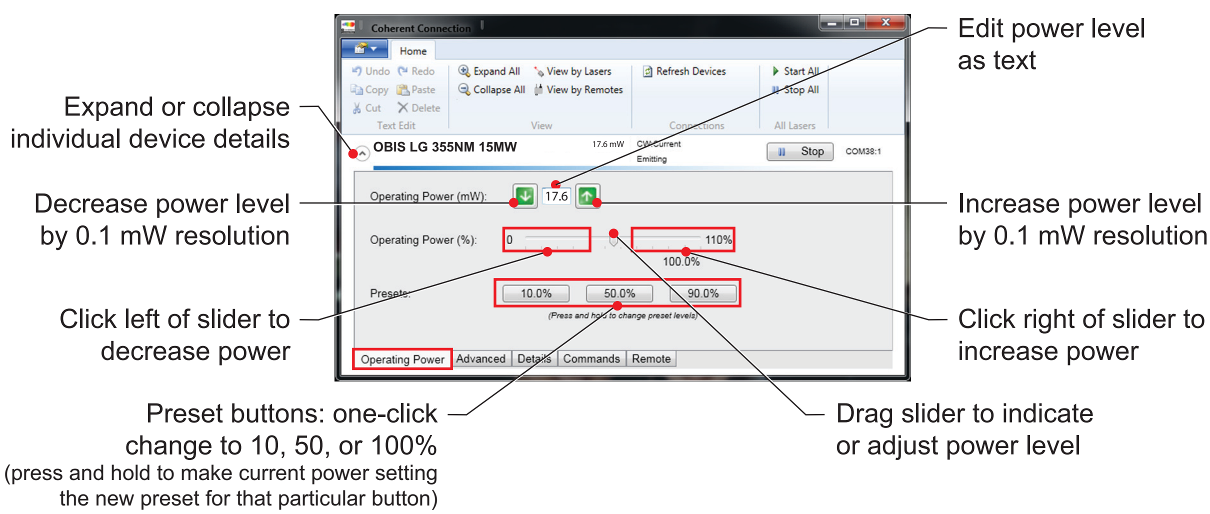


Figure 2-15. Coherent Connection - Operating Power Tab

Figure 2-16 shows the Advanced tab. On this page of the software, you can select the Operating mode, enable or disable the CDRH delay, Auto Start, Blanking, as well as reset the laser or factory calibration settings,

Note that recalibration requires the laser to be updated to the most current firmware.

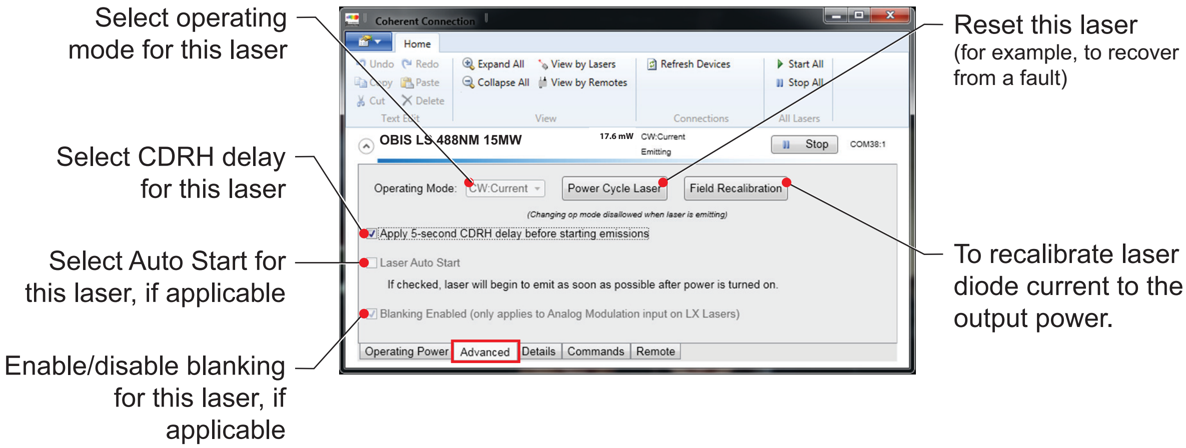


Figure 2-16. Coherent Connection - Advanced Tab

Figure 2-17 shows the Details tab. On this page of the software, you can view the model, serial number, and other information specific to the laser.

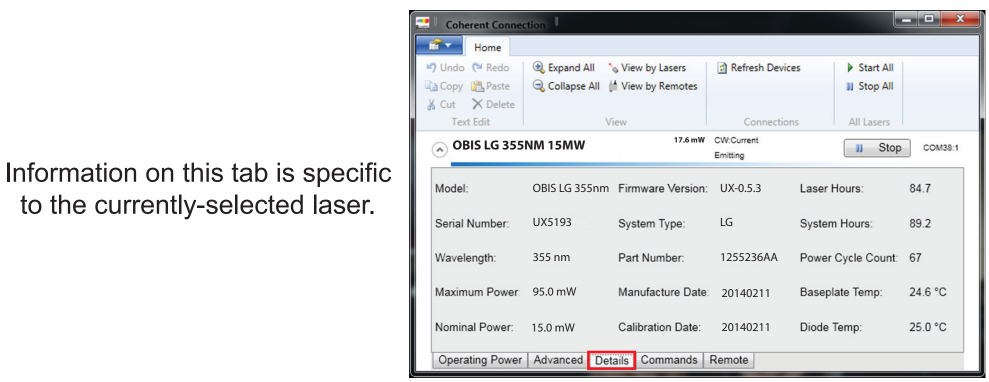


Figure 2-17. Coherent Connection - Details Tab

Figure 2-18 shows the Commands tab. You can view commands and responses, or enter commands to control the laser.

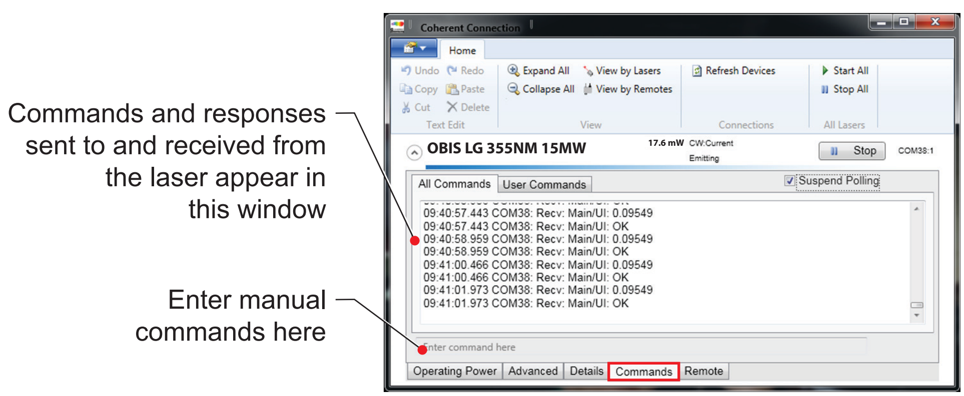


Figure 2-18. Coherent Connection - Commands Tab

Figure 2-19 shows the Remote tab. On this page of the software, you can enable settings to start all lasers on power-up, as well as select the input impedance for Analog Modulation mode.

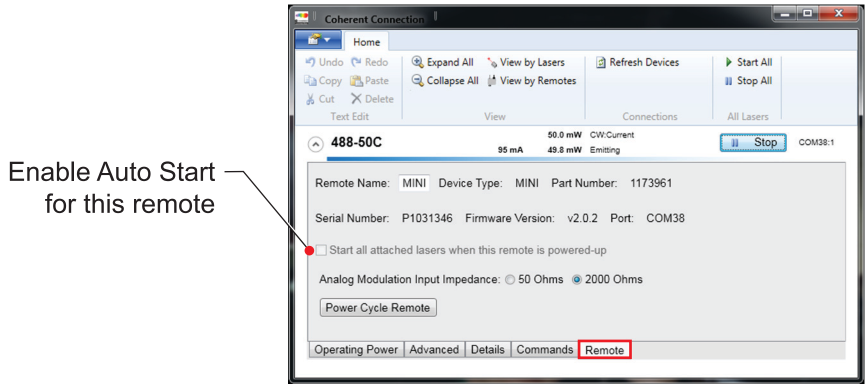


Figure 2-19. Coherent Connection - Remote Tab

2.3.1 OBIS LX Single-Frequency and Coherent Connection

To fine-tune the wavelength of the OBIS LX-SF (Single-Frequency) laser, use the Power tab to:

- Adjust the diode temperature
- Adjust the diode current by adjusting the operating power
- Or a combination of both

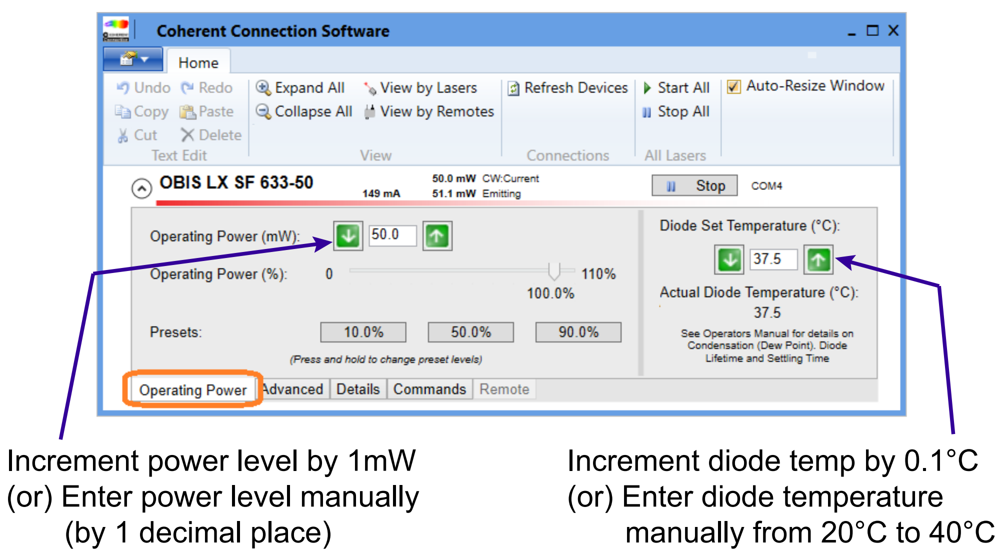


Figure 2-20. OBIS LX-SF Tune Wavelength

NOTICE

Ensure that the laser is running in ‘CW:Current’ (constant current mode).

Figure 2-21 shows the emission status of the laser.

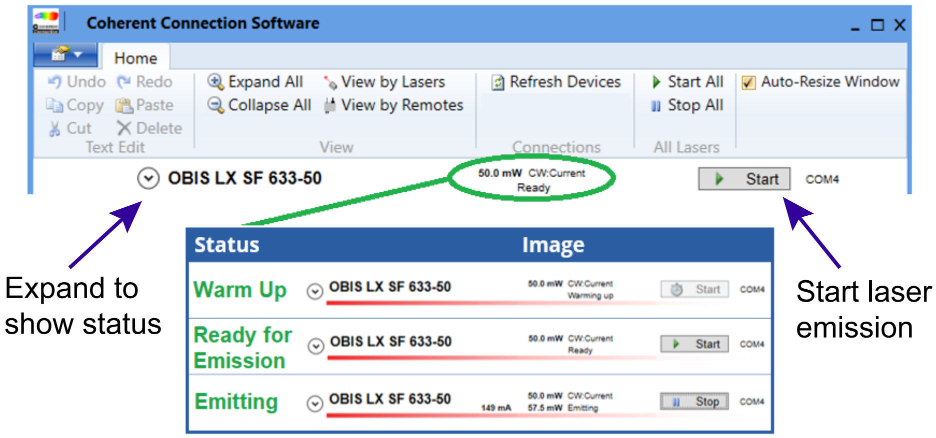


Figure 2-21. OBIS LS-SF Emission Status

3 HOST INTERFACE

This section describes:

- Host command quick reference (this page)
- Message considerations (p. 3-32)
- Master commands and queries (p. 3-36)
- Controls and queries (p. 3-57)
- System standby and sleep mode (p. 3-63)
- OBIS RS-232 interface (p. 3-63)

NOTICE
When a command is sent to the OBIS system, the parameter for the command is stored in internal persistent memory.

Internal persistent memory has a logic cell life of one million cycles for the laser or 10,000 cycles for the OBIS Remote. The cell life sets the limits for repetitive commands sent to the OBIS system.

This information only applies to commands and not queries.

3.1 Host Command Quick Reference

Coherent Connection software that supports OBIS LX and LS lasers is available in an easily accessible location on the Coherent website. To download, go to:

https://www.coherent.com/resources

Table 3-1 briefly describes all host commands and queries. For specific command or query details, refer to the page listed in the right column.

Table 3-1. Host Command Quick Reference

Command	Description	Page No.
Mandatory Commands/Queries		
IEEE-488.2		
*IDN?	Gets the laser's identification string	3-37
*RST	Causes a device to warm boot if implemented	3-38

Table 3-1. Host Command Quick Reference (continued)

Command	Description	Page No.
*TST?	Runs a laser self-test procedure, if implemented	3-38
Session Control		
SYSTem:COMMunicate:HANDshaking	Toggles the system handshaking	3-41
SYSTem:COMMunicate:HANDshaking?	Queries the system handshaking	3-41
SYSTem:COMMunicate:PROMpt	Toggles the system command prompt	3-41
SYSTem:COMMunicate:PROMpt?	Queries the system command prompt	3-41
SYSTem:AUTostart	Enables or disables the laser Auto Start feature	3-41
SYSTem:AUTostart?	Queries the laser Auto Start feature	3-41
SYSTem:INFormation:AMODulation:TYPe	Sets the analog modulation type	3-42
SYSTem:INFormation:AMODulation:TYPe?	Queries the analog modulation type	3-42
SYSTem:STATus?	Queries the system status	3-42
SYSTem:FAULt?	Queries current system faults	3-43
SYSTem:INDicator:LASer	Turn ON/OFF laser status indicator(s)	3-44
SYSTem:INDicator:LASer?	Queries laser status indicator(s)	3-44
SYSTem:ERRor:COUNt?	Queries the number of error records in the error queue	3-45
SYSTem:ERRor:NEXT?	Queries the next error record(s) in the error queue	3-45
SYSTem:ERRor:CLEar	Clears all error records in the error queue	3-46
OBIS Common Commands/Queries		
System Information		
SYSTem:INFormation:MODel?	Retrieves the model name of the laser	3-47
SYSTem:INFormation:MDATe?	Retrieves the manufacture date of the device	3-47
SYSTem:INFormation:CDATe?	Retrieves the calibration date of the device	3-47
SYSTem:INFormation:SNUMber?	Retrieves the serial number of the laser	3-48
SYSTem:INFormation:PNUMber?	Retrieves the manufacturer part number of the laser	3-48
SYSTem:INFormation:FVERsion?	Retrieves the current firmware version	3-48
SYSTem:INFormation:PVERsion?	Retrieves the current OBIS protocol version	3-48
SYSTem:INFormation:WAVelength?	Retrieves the wavelength of the laser	3-49
SYSTem:INFormation:POWer?	Retrieves the power rating of the laser	3-49
SYSTem:INFormation:TYPe?	Retrieves the device type	3-49
SOURce:POWer:NOMinal?	Returns the nominal CW laser output power	3-49
SOURce:POWer:LIMit:LOW?	Returns the minimum CW laser output power	3-49
SOURce:POWer:LIMit:HIGH?	Returns the maximum CW laser output power	3-50
SYSTem:INFormation:USER	Enters and stores user-defined information	3-50
SYSTem:INFormation:USER?	Queries user-defined information	3-50
SYSTem:INFormation:FCDate	Enters and stores date of last field calibration	3-50
SYSTem:INFormation:FCDate?	Queries date of last field calibration	3-50
System State		
SYSTem:CYCLes?	Returns the number of ON/OFF power cycles	3-51
SYSTem:HOURs?	Returns the hours the laser has been powered on	3-51
SYSTem:DIODe:HOURs?	Returns the hours the laser diode has operated	3-51
SOURce:POWer:LEVel?	Returns the present output power of the laser	3-51
SOURce:POWer:CURRent?	Returns the present output current of the laser	3-51

Table 3-1. Host Command Quick Reference (continued)

Command	Description	Pag e No.
SOURce:TEMPerature:BASeplate?	Returns the present laser base plate temperature	3-52
SYSTem:LOCK?	Returns the status of the system interlock	3-52
Operational		
SOURce:AM:INTernal	Sets the laser operating mode to internal CW	3-53
SOURce:AM:EXTernal	Sets the laser operating mode to external modulation	3-53
SOURce:AM:SOURce?	Queries the current operating mode of the laser	3-53
SOURce:POWer:LEVel:IMMediate:AMPLitude	Sets present laser power level	3-53
SOURce:AM:STATe	Turns the laser ON or OFF	3-54
SOURce:AM:STATe?	Queries the current laser emission status	3-54
SYSTem:CDRH	Enables or disables the CDRH laser emission delay	3-54
SYSTem:CDRH?	Queries the status of the CDRH laser emission delay	3-54
OBIS Optional Commands/Queries		
SOURce:TEMPerature:APRobe	Enables/disables temperature control of the laser diode	3-54
SOURce:TEMPerature:APRobe?	Queries temperature control of the laser diode	3-54
OBIS LX-Specific Commands/Queries		
SOURce:POWer:CALibration	Starts a self-laser power calibration	3-55
SOURce:POWer:UNCalibration	Undoes the filed calibration	3-55
SOURce:AModulation:BLANKing	Enables/disables Blanking in Analog Modulation mode	3-55
SOUR:AM:BLAN?	Queries present state of Analog Modulation Blanking	3-55
SOURce:TEMPerature:PROTec-tion:INTernal:HIGH?	Queries the high internal temperature limit settings	3-55
SOURce:TEMPerature:PROTec-tion:INTernal:LOW?	Queries the low internal temperature limit settings	3-55
SOURce:TEMPerature:DIODe?	Queries the present laser diode temperature	3-56
SOURce:TEMPerature:DSETpoint?	Queries the diode set point temperature	3-56
SOURce:TEMPerature:INTernal?	Queries the present internal laser temperature	3-57

The following table shows an example of the command/query for the indicator status light on the top cover of the OBIS:

Table 3-2. Example Command/Query

Command	Description
SYST:IND:LAS ON	Turns the OBIS top cover indicator ON. Do not use SYST:IND:LAS=1. Do not use SYST:IND:LAS 1.
SYST:IND:LAS OFF	Turns the OBIS top cover indicator OFF. Do not use SYST:IND:LAS=0. Do not use SYST:IND:LAS 0.
SYST:IND:LAS?	Returns the value of the indicator as ON or OFF. It will not return a 1 or a 0. The reply will be ON or OFF.

In a similar fashion, use commands and queries with the appropriate value after the command and a space between the command and the value. To query the laser, add a question mark (?) at the end of the command.

3.2 Message Considerations

This section describes the following topics:

- Communication port selection
- Message completion handshaking
- Message terminators
- Message syntax
- Device selection syntax
- The command prompt
- Broadcast commands
- Types of commands and queries

3.2.1 Communication Port Selection

The laser design includes both USB and Coherent Connection Bus (CCB) communication ports.

The communication protocol described within this section works identically on either port; however, the ports are mutually exclusive and cannot be used simultaneously.

When both USB and CCB connections are connected, the laser gives the CCB port precedence and ignores any input received from the USB port. Note that certain information on the laser/controller communications—such as one controller talking to multiple lasers—is part of future expansion protocol and is not applicable to the OBIS Remote.

3.2.2 Message Completion Handshake

Standard commands for programmable instrument (SCPI) message round trip handshaking is implemented on every message sent by the laser firmware; however, the handshaking may be disabled using an SCPI command. Change of the setting will be saved in non-volatile memory.

This handshake serves several purposes:

1. It provides an indication to the host/controller that the message was received.
2. It provides a synchronization mechanism to the host/controller so it will know when a message has been processed to completion so a new message may be sent.
3. It provides the host/controller with an indication of any errors that may have occurred.

The handshake is a short message string that is sent as the last action performed when handling a received message. The handshake string represents either an OK response or an error response if a received message raises an error condition. Note that quotation marks as depicted here are never included in the handshake string.

- The OK response is formatted as: **OK\r\n**
- Error responses are formatted as follow, where <n> represents the error code number: **ERR<n>\r\n**

Negative numbers are permitted in the error string.

When handshaking is enabled, OBIS devices transmit one of the following handshake reply strings in response to each received command or query:

- Valid commands with valid data parameters will reply with **OK\r\n**.
- Valid queries with any optional valid data reply as explicitly defined elsewhere in this section, followed by **OK\r\n**. For example, if querying the model name string, the laser will transmit the model name string followed by the **OK\r\n** string.
- Valid commands or queries which result in an error reply with **ERR<n>\r\n**.
- Unrecognized or unsupported commands or queries reply with **ERR<n>\r\n.**

Note that the message completion handshake is not transmitted in response to a command that has been broadcast to all devices.

3.2.3 Message Terminators

Messages between the laser and the host computer or controller are comprised entirely of ASCII string characters; no binary messages are supported. All message strings passing through the host interface are terminated to signal the end of a message string. The maximum message length supported is 255 bytes, which includes all terminating characters.

3.2.3.1 Messages Received by the Laser

Messages received by the laser must be terminated by a carriage return (decimal 13). A line feed (decimal 10) following the carriage return is ignored so messages may be terminated with a carriage return and line feed pair. A command or query is considered incomplete without proper termination.

3.2.3.2 Messages Sent by the Laser

All messages sent by the laser are terminated by a carriage return (decimal 13) and line feed (decimal 10) pair. The maximum length of any message sent by the laser is limited to 255 bytes, including all terminating characters.

3.2.4 Message Syntax

Syntax specified by the SCPI and IEEE 488.2 Standards is followed unless otherwise specified. Refer to the SCPI and IEEE 488.2 Standards for more information.

Notably, the base-10 numeric data format specification is used heavily in this document and covered in the IEEE 488.2 Standard. Unless otherwise specified, numeric data items referred to as NRf (IEEE flexible numeric representation) are interchangeable and may be represented in any of these formats:

- integer values
- non-scientific notation floating point values
- scientific notation floating point values (uppercase or lowercase E)

For example, the following data values are functionally equivalent:

- 31256
- 31256.0
- 3.1256E4
- 31.256E3
- +3.1256E+4.

Unless otherwise specified, non-numeric data items (typically referred to as strings) are not quoted.

Devices interpret hexadecimal data using the following rules:

- Uppercase and lowercase are accepted ("**`FE`**" is the same as "**`fe`**")
- Leading zeroes are required and accepted ("**`0A`**" is the same as "**`A`**")
- The data string may optionally be preceded by a "**`0x`**" or "**`0X`**" C hexadecimal notation idiom (**`0xD2C4`** is the same as **`D2C4`**)
- Following the optional "**`0x`**" prefix, the acceptable characters are from the list: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, A, B, C, D, E, and F

Enumerated values must match exactly, using the long form/short form comparison rules defined under the SCPI Standard.

Dates (manufacturing date, calibration date, etc.) will use the **`YYYYMMDD`** format. Using this format, dates may be stored as ASCII strings or as numeric long integers and converted easily from one format to the other.

3.2.5 Device Selection Syntax

Many common commands are supported by all OBIS devices. When such a command is transmitted by a host computer to a system of devices (a controller and one or more lasers), an ambiguity exists where the exact destination device is not clear.

The SCPI protocol provides a method to communicate with multiple virtual devices within an instrument. Since a complete OBIS system could be considered an “instrument” (controller) with multiple “virtual devices” (lasers) this mechanism is used to disambiguate the destination of a command.

A SCPI command consists of one or more words separated by delimiters. The first word in a command string is called the base word.

SCPI channel selection is performed by appending a numeric suffix to the base word in any command string. When the numeric suffix is left off or has a value of zero, the command refers to the first connected device.

For example, **`*idn?*`** and **`*idn0?`** query strings both refer to the first connected device. If a host computer is connected to a controller and this query is issued, it is responded to by the controller. If the host is connected directly to a laser, without going through a controller, the first connected device is the laser and it should respond.

Consider the scenario where the host computer is connected to a mini-controller which, in turn, is connected to a laser.

If the host issues the **`*idn?*`** query, the OBIS Remote should respond. If, however, the host appends a numeric suffix to the base word of the query, then the suffix specifies the device which should respond. In this scenario **`*idn?*`** and **`*idn0?`** would be responded to by the OBIS Remote, **`*idn1?*`** would be responded to by the laser, and **`*idn2?*`** would receive no response since device number 2 does not exist.

If the host is connected to a master controller with four connected lasers, then a missing or zero suffix would apply to the master controller and suffixes 1…4 would refer to lasers 1…4.

The numeric suffix mechanism may be applied to the base word of any command or query.

As an implementation detail, lasers should always respond to commands with either no suffix or suffixes of 0 or 255 to accommodate connections to a bus and also directly to a host computer.

When host commands are routed through an OBIS controller, the numeric suffix will be stripped off before the command is transmitted to a laser. In this instance, the laser does not have to deal with the numeric suffix at all and can behave as if it were connected directly to a host.

However, when a laser is connected directly to a host, then it is valid for the host to append a 0 to the command base word to refer to the first connected device. Since that device could be a laser, the laser will support a numeric suffix of 0.

The numeric suffix of 255 refers to a command (not query) that is broadcast to all devices on the bus. Queries cannot be broadcast since a stream of query results won't make any sense to the host. Therefore, the command `SYSTem255:PROMpt ON` enables the system prompt for all devices while `SYSTem0:PROMpt ON` enables the prompt for the first connected device.

3.2.6 Command Prompt

Each device implements the ability to output a command prompt to support interactive operation by an operator typing commands in a terminal program. A command has been specified to describe the command prompt behavior.

Note that the command prompt will not be transmitted in response to a command that has been broadcast to all devices.

3.2.7 Broadcast Commands

It is possible that a host message could be broadcast to all attached devices. Generally the broadcast capability is best used when a command is needed to synchronize the action of a group of devices (such as turning all connected lasers on or off simultaneously). The ability to broadcast device queries is prohibited.

Lasers silently ignore any query that is broadcast and will act upon a broadcast command, if possible, without transmitting anything in response (including error, handshake, or command prompt strings).

Controllers respond to queries that are broadcast by returning error 103 along with optional handshake and command prompt strings.

Controllers respond to commands that are broadcast by rebroadcasting them to all devices on the bus, performing the broadcast action locally if appropriate for the command, and then returning any optional handshake and command prompt strings to the host.

This method allows the host to receive a single handshake and/or command prompt when a command is broadcast to several devices.

3.2.8 Commands and Queries

The OBIS Laser Protocol supports three types of laser devices:

- OBIS LX direct diode lasers (DDL)
- OBIS LS optically pumped semiconductor lasers (OPSL)
- Other similar accessories

Each of these laser types support the common command sets and zero or more of the extension command sets, as shown in Table 3-3.

Table 3-3. Supported Commands by Laser Type

Command Set	OBIS LX (DDL)	OBIS LS (OPSL)	Other
SCPI Common Command Set	X	X	X
OBIS Common Command Set	X	X	X
OBIS LX Extension Command Set	X	—	—
OBIS LS Extension Command Set	—	X	—

3.3 Mandatory Commands and Queries

This section describes the mandatory commands and queries.

3.3.1 IEEE-488.2 Mandated Commands and Queries

The SCPI Standard specifies a mandatory set of IEEE-488.2 common commands. All of these commands and queries start with an asterisk. Refer to the IEEE-488.2 specification for more detailed information concerning these commands.

3.3.1.0.1 Identification Query - *IDN?

Gets the laser's identification string, such as model name, firmware version, and firmware date.

QUERY	`*IDN?`
REPLY	`"Coherent, Inc" + "-" + <model name> + "-" + <firmware version> + "-" + <firmware date>`

The dash sign separates all fields within the reply string.

- The first field will always be “Coherent, Inc”.
- The second field is the model name, which varies based on the laser.
- The third field is the firmware version number, having the format “V<major>.<minor><optional qualifier characters>”.
- The fourth field is the firmware date, having the form YYYYMMDD.

The reply string is not quoted.

For example, a typical identification string might look like:

```
Coherent, Inc-OBIS 405nm 50mW C-1.3- 20090630
```

3.3.1.0.2 Reset Command - *RST

Causes a device to warm boot if implemented. Note that the message handshake is transmitted immediately prior to execution of the reset.

If the command is not implemented, then no error is returned and no response is necessary.

QUERY	`*RST`
REPLY	None

3.3.1.0.3 Self-test Query - *TST?

Runs a laser self-test procedure, if implemented. Any detected faults are set in the laser fault code.

QUERY	`*TST?`
REPLY	`<System Fault Code>`

The returned system fault code is formatted as a 32-bit hex value. A value of 0 indicates no fault conditions. If the self-test is not implemented, a value of `0xffffffff` is returned.

NOTICE

A warm or cold device reboot is required to clear an OBIS Remote or laser fault.

Table 3-4 lists the fault codes for the OBIS remote and OBIS lasers.

Table 3-4. Fault Codes—OBIS Remote and Laser

Code Bit[a]	Error Value	OBIS Remote	Laser	Error Description	Cause and Possible Solution
0	00000001		X	Baseplate temperature fault	**Cause**: Baseplate temperatures is greater than 40°C or lower than 10°C. **Solution**: Improve heatsink to reduce baseplate temperature or adjust the ambient temperature where the laser operates.
1	00000002		X	Diode temperature fault	**Cause**: Diode temperature is greater than 40°C or lower than 10°C. **Solution**: Make sure the TE cooler is on and/or adjust the ambient temperature where the laser operates.
2	00000004		X	Internal temperature fault	**Cause**: Microprocessor temperature exceeds factory set limit. **Solution**: Make sure the TE cooler is on and the ambient temperature is within the specified range.
3	00000008		X	Laser power supply fault	**Cause**: There is no electrical power to the laser diode. **Solution**: Make sure the SDR cable is plugged in and secured properly on both ends.

Table 3-4. Fault Codes—OBIS Remote and Laser (continued)

Code Bit[a]	Error Value	OBIS Remote	Laser	Error Description	Cause and Possible Solution
4	00000010		X	Device internal I2C bus error	**Cause**: An error was encountered in internal I2C bus communications. **Solution**: Perform a warm or cold reboot of the laser system. If the problem persists, contact Coherent technical support.
5	00000020		X	Laser diode over-current error	**Cause**: Laser diode current exceeds the specified upper limit. **Solution**: Turn off laser emission and reboot the device. If the problem persists, contact Coherent technical support.
6	00000040		X	Laser checksum error	**Cause**: An error occurred that is associated with persistent memory where critical data is stored. **Solution**: Reboot the laser system. If the problem persists, contact Coherent technical support.
7	00000080		X	Checksum recovery error	**Cause**: An error occurred when trying to recover from checksum error via host command. **Solution**: Contact Coherent technical support.
8	00000100		X	Message buffer overflow	**Cause**: An overflow error associated with message buffer was encountered in the firmware. **Solution**: Perform a warm or cold reboot of the laser system. If the problem persists, contact Coherent technical support.
9	00000200		X	Warm-up limit fault	**Cause**: The 5-minute warm-up limit was exceeded. **Solution**: Make sure the TE cooler is enabled. If the laser was started in a very low temperature environment, keep the laser powered for 10-15 minutes, then reboot the device.
10	00000400		X	TEC control error	**Cause**: An error associated with the TEC operation was encountered. It can be caused by insufficient heatsink. **Solution**: Make sure heatsink is sufficient, then perform a device reboot. If the problem persists, contact Coherent technical support.
11	00000800		X	Coherent Connection bus error	**Cause**: An error associated with RS485 bus communications between the laser and OBIS Remote was encountered. **Solution**: Make sure the SDR cable is plugged in and secured properly on both ends.
12	00001000		X	Diode temperature limit error	**Cause**: Laser diode temperature deviates from the temperature set point by more than 3°C. **Solution**: Make sure the TE cooler is turned on. If the laser warm-up process is disabled, keep the laser running for 10-15 minutes, then perform a device reboot.
13	00002000		X	Laser ready fault	**Cause**: Laser fails to emit within ± 2% of the requested power. **Solution**: If the problem persists, contact Coherent technical support for a system recalibration.

Table 3-4. Fault Codes—OBIS Remote and Laser (continued)

Code Bit[a]	Error Value	OBIS Remote	Laser	Error Description	Cause and Possible Solution
14	00004000		X	Photodiode fault	**Cause**: Readings from the internal photodiode for power control were negative. **Solution**: Reboot the laser. If the problem persists, Contact Coherent technical support.
15	00008000		X	Device fatal error	**Cause**: A device error not recoverable in the field if persistent. **Solution**: If the problem persists, contact Coherent technical support.
16	00010000		X	Startup error	**Cause**: Errors were encountered during firmware start-up. **Solution**: Perform a cold or warm device reboot.
17	00020000		X	Watchdog timer reset	**Cause**: Firmware was resumed from a processor watchdog reset. **Solution**: Contact Coherent technical support.
18	00040000		X	Field calibration error	**Cause**: Errors were encountered while running power field calibration. **Solution**: Re-run field calibration. If the problem persists, contact Coherent technical support.
20	00100000		X	Overpower fault	**Cause**: Error occurs when actual power is 10% over the maximum power setting. **Solution**: Perform a cold or warm device reboot. If the problem persists, contact Coherent technical support.
...	...	...	...	...	...
30	40000000	X		Min-controller checksum error	**Cause**: An error associated with persistent memory was encountered. **Solution**: Reboot the OBIS Remote. If the problem persists, contact Coherent technical support.
31	80000000	X		Fault status from OBIS Remote	**Cause**: A firmware or hardware fault was encountered in the OBIS Remote. **Solution**: Reboot the OBIS Remote. If the problem persists, contact Coherent technical support.

a. Unspecified bits are reserved and will be zero.

3.3.2 OBIS Mandatory Commands and Queries

The OBIS Mandatory Command set is implemented by all OBIS compatible devices.

3.3.2.1 Session Control Commands

3.3.2.1.1 Handshaking

Toggles the system handshaking on and off. Setting is saved in persistent memory. The factory default is ON.

COMMAND	`SYSTem:COMMunicate:HANDshaking {ON|OFF}`
QUERY	`SYSTem:COMMunicate:HANDshaking?`
REPLY	`ON|OFF`

When enabled, the device transmits, in response to each received command or query, one of the handshaking strings described under “Message Completion Handshake” (p. 3-32).

Note that the handshaking reply is not transmitted in response to a command that has been broadcast to all devices, except by a controller device.

3.3.2.1.2 Command Prompt

Toggles the system command prompt on and off. Setting is saved in persistent memory. The factory default is OFF.

COMMAND	`SYSTem:COMMunicate:PROMpt {ON|OFF}`
QUERY	`SYSTem:COMMunicate:PROMpt?`
REPLY	`ON|OFF`

When enabled the device outputs a command prompt after each reply string. The command prompt is preceded by a carriage return and line feed, and consists of a '>' character and a space character.

Note that the command prompt is not transmitted in response to a command that has been broadcast to all devices, except by a controller device.

3.3.2.1.3 Laser Auto Start

Enables or disables the laser Auto Start feature. Setting is saved in persistent memory. The factory default is OFF.

COMMAND	`SYSTem:AUTostart {ON|OFF}`
QUERY	`SYSTem:AUTostart?`
REPLY	`ON|OFF`

If Auto Start is enabled, the device, when powered up, will automatically start emission at a previously-set level.

The Auto Start setting is saved in the non-volatile memory of the laser. If the laser is connected to a OBIS Remote through a SDR cable, this setting is overridden by the hardware switch of the min-controller; however, the ON/OFF position of the switch will not overwrite the setting in the laser memory.

3.3.2.1.4 Analog Modulation Type

Sets the analog modulation type that provides unique electrical impedance on the analog interface of the OBIS Remote. The factory default is 50Ω.

COMMAND	`SYSTem:INFormation:AMODulation:TYPe {1|2}`
QUERY	`SYSTem:INFormation:AMODulation:TYPe?`
REPLY	`1|2`

The input impedance is 50Ω and 2 kΩ for type 1 and 2, respectively.

3.3.2.1.5 System Status Query

Gets the system status code. The status code is returned in a string expressed in uppercase hexadecimal integer form. The 32-bit word represents a bit-mapped status indicator.

The MSB of the code is used to indicate if the code represents the status of a controller or a laser. If the MSB is set, the code represents controller status. This is important since the meaning of some bits is subtly different for a controller.

Table 3-5 describes status code bit mapping. The "Controller" column specifies the meaning of each bit when the status word is read from the controller and the 'Laser' column specifies the bit meaning when the status word is read from a laser.

Table 3-5. Status Code Bit Definitions

Bit	Mask	Bit Label	Controller	Laser
0	00000001	Laser Fault	Logical OR from all lasers	Laser faults—that is, fault words shown in Table 3-6 (p. 3-44)
1	00000002	Laser Emission	Logical OR from all lasers	Laser emission status
2	00000004	Laser Ready	Logical OR from all lasers	Laser ready status
3	00000008	Laser Standby	Logical OR from all lasers	Laser standby status
4	00000010	CDRH Delay	Logical OR from all lasers	Laser CDRH delay status
5	00000020	Laser Hardware Fault	Logical OR from all lasers	Hardware related faults in Table 3-6 (p. 3-44)
6	00000040	Laser Error	Logical OR from all lasers	Laser error is queued
7	00000080	Laser Power Calibration	Logical OR from all lasers	Laser power is within factory calibration specification
8	00000100	Laser Warm Up	Logical OR from all lasers	Laser warm-up status
9	00000200	Laser Noise	Logical OR from all lasers	Noise level is over 30
10	00000400	External Operating Mode	Logical OR from all lasers	External operating mode is selected

Table 3-5. Status Code Bit Definitions (continued)

Bit	Mask	Bit Label	Controller	Laser
11	00000800	Field Calibration	Logical OR from all lasers	Field calibration is in progress when set
12	00001000	Laser Power Voltage	Logical OR from all lasers	12V laser power voltage is present when set
...	…	…		
25	02000000	Controller Standby	Keyswitch is in “STANDBY” position	Always 0
26	04000000	Controller Interlock	“INTERLOCK” is open.	Always 0
27	08000000	Controller Enumeration	One or more lasers have been enumerated	Always 0
28	10000000	Controller Error	Controller error flag	Always 0
29	20000000	Controller Fault	Controller fault status	Always 0
30	40000000	Remote Active	Host is connected	Always 0
31	80000000	Controller Indicator	Status word is from controller.	Always 0

Unspecified bits are reserved and are zero.

The status word MSB indicates whether a status word is from a laser or from a controller.

COMMAND	`None`
QUERY	`SYSTem:STATus?`
REPLY	`<status word>`

As an example, if the laser is turned on, but is being delayed by the CDRH required delay, the system status query returns:

`00000012`(*Laser emission enabled but delayed by CDRH*)

3.3.2.1.6 System Fault Query

Gets the system fault code. The fault code is returned in a string expressed in uppercase hexadecimal integer form. The 32-bit word represents a bit-mapped fault indicator.

COMMAND	`None`
QUERY	`SYSTem:FAULt?`
REPLY	`<fault word>`

As an example, if the base plate and laser diode temperature limits are both exceeded, the system fault query will return:

`00000003`(*Base Plate & Laser Diode Temp. Limits Exceeded*)

The Most Significant Bit (MSB) of the code is used to indicate if the code represents the status of a controller or a laser. If the MSB is set, the code represents controller fault status. This is important since the meaning of some bits is subtly different for a controller.

Table 3-6 describes fault code bit mapping.

Table 3-6. Fault Code Bit Definitions

Bit[a]	Mask	Bit Label	Controller	Laser
0	00000001	Base Plate Temp. Fault	Logical OR from all lasers	Base plate temperature out of range
1	00000002	Diode Temp. Fault	Logical OR from all lasers	Diode temperature out of range
2	00000004	Internal Temp. Fault	Logical OR from all lasers	Internal temperature out of range
3	00000008	Laser Power Supply Fault	Logical OR from all lasers	No electrical power to laser diode
4	00000010	I2C Error	Logical OR from all lasers	I2C bus error
5	00000020	Over Current	Logical OR from all lasers	Diode over current
6	00000040	Laser Checksum Error	Logical OR from all lasers	EEPROM checksum error in at least one section
7	00000080	Checksum Recovery	Logical OR from all lasers	EEPROM was restored to default settings
8	00000100	Buffer Overflow	Logical OR from all lasers	Bus message buffer overflow
9	00000200	Warm-up limit fault	Logical OR from all lasers	Warm-up time limit exceeded
10	00000400	TEC Driver Error	Logical OR from all lasers	TE controller driver failure
11	00000800	CCB Error	Logical OR from all lasers	RS-485 bus error
12	00001000	Diode Temp Limit Error	Logical OR from all lasers	Diode temperature off by > 5°C from set point
13	00002000	Laser Ready Fault	Logical OR from all lasers	Fail to emit at set power level
14	00004000	Photodiode Fault	Logical OR from all lasers	Negative photodiode readout
15	00008000	Fatal Fault	Logical OR from all lasers	Irrecoverable system failure
16	00010000	Startup Fault	Logical OR from all lasers	Errors encountered during firmware start-up
17	00020000	Watchdog Timer Reset	Logical OR from all lasers	Firmware resumed from watchdog reset
18	00040000	Field Calibration	Logical OR from all lasers	Errors encountered during field calibration
20	00100000	Over Power	Logical OR from all lasers	Output power above limit
...	...	...		
30	40000000	Controller Checksum	Controller checksum error	Always 0
31	80000000	Controller Status	Fault word is from controller	Always 0

a. Unspecified bits are reserved and are zero.

3.3.2.1.7 Turn On/Off Laser Status Indicator

Turns on (or turns off) the status indicator(s) associated with the laser. Setting is saved in persistent memory. The factory default is ON.

COMMAND	`SYSTem:INDicator:LASer {ON|OFF}`
QUERY	`SYSTem:INDicator:LASer?`
REPLY	`ON|OFF`

This command is used to turn on (or turn off) the Status LED indicator(s) that is visible to the user. The status bits returned by `SYSTem:STATus?`, however, are not affected by this command.

3.3.3 Error Record Reporting

Programming and system errors will occasionally occur while testing or debugging remote programs and during measurement. Error strings follow the SCPI Standard for error record definition:

`<error code>,<quoted error string><CR><LF>`

The host queries for errors in two steps.

1. First, the host queries for the number of error records available (N).
2. Secondly, the host queries N times for the error records.

Errors are stacked up to 20 deep. In the case of error overflow, the last error in the error list is an indication of error overflow.

Note that the error records defined in this section are the errors generated in response to external commands or queries. Any errors generated from the internal operation of the laser or controller will be reflected in the fault code displayed in Table 3-6 (p. 3-44).

3.3.3.0.1 Error Count Query

Gets the number of error records in the error queue at the time of the query.

COMMAND	None
QUERY	**SYSTem:ERRor:COUNt?**
REPLY	**<integer count of error records stored>**

3.3.3.0.2 Error Query

Gets the next error record(s) in the error queue.

- More than one error record may be queried using this optional parameter, which must be an integer value: **<error record count>**.
- A single error record is returned if **<error record count>** is not specified.

COMMAND	None
QUERY	**SYSTem:ERRor:NEXT?**
REPLY	**<next available error record>**

No reply is transmitted if there are no available error records.

As the device transmits each error record:

- The error record is permanently removed from the error queue.
- The queued error record count is decremented by one.

3.3.3.0.3 All Error Clear

Clears all error records in the error queue.

COMMAND	`SYSTem:ERRor:CLEar`
QUERY	None

Table 3-7 lists possible error codes and briefly describes the text strings returned. Any errors generated from the internal operation of the lasers or controller are reflected in the fault code—see Table 3-4 on page 3-38.

Table 3-7. Error Codes and Description Strings

Error Code	Error String	Error Description
-400	Query Unavailable	Broadcast of query is prohibited. Occurs when sending a query as a broadcast message. Queries may not be broadcast.
-350	Queue overflow	Error queue is full. Non-**`Queue overflow`** errors are replaced by **`Queue overflow`** errors when there is exactly one available storage location available in the error queue. No additional errors are added to the error queue if the error queue is full.
-321	Out of memory	Internal memory is exhausted. Occurs when there is an internal memory related error. This error could be caused by exhaustion of the memory heap, overflow of a fixed memory buffer, or similar type of problem.
-310	System error	Unexpected/unrecoverable hardware or software fault. Occurs when the device firmware detects an unexpected or unrecoverable error. This error condition includes unrecoverable hardware faults.
-257	File to open not named	The file open is not possible because the file has not been named. Occurs when an attempt is made to open a file without specifying a file name.
-256	File does not exist	The specified file does not exist. Occurs when an attempt is made to open a file that does not exist.
-241	Device Unavailable	Command was sent to a device that is not available. Occurs when sending a message to a device that is not currently available.
-221	Settings conflict	Command not executed due to current device state. Occurs when a command is received that is at odds with the current device settings.
-220	Invalid parameter	The command or query parameter is invalid. Occurs when an invalid parameter has been specified.
-203	Command protected	Command is password protected. Occurs when an attempt is made to execute a password protected command when in user mode.
-200	Execution error	Command is out of order. Occurs when an order-dependent command sequence is issued out of order (for example, when trying to read from a file before the file has been opened).
-109	Parameter missing	No or fewer parameters were received. Occurs when there are no or fewer parameters for a received command or query.
-102	Syntax error	Unrecognized command or data type was encountered. Occurs when command or data type is not recognized.
-100	Unrecognized command or query	The command or query is not recognized. Occurs when the device receives an unrecognized command or query. This is a generic syntax error for devices that cannot detect more specific errors.
0	No error	No error.

Table 3-7. Error Codes and Description Strings (continued)

Error Code	Error String	Error Description
500	CCB fault	A Coherent Connection bus error was encountered.
510	I2C bus fault	A device internal I2C bus error was encountered.
520	Controller Time Out	No response was received within 0.7 seconds from a slave device and the message was resent three (3) times by the controller.
900	CCB Message Timed Out	A A Coherent Connection bus message timed out after three 93) retries.

3.4 OBIS Common Commands and Queries

OBIS Common Commands and Queries is implemented by all OBIS devices that support the features contained in this section. If a device does not support a given feature, the command may be ignored.

3.4.1 System Information Queries

The System Information commands allow a host to retrieve static information describing the characteristics of the laser.

3.4.1.1 System Model Name Query

Retrieves the model name of the laser.

QUERY	`SYSTem:INFormation:MODel?`
REPLY	`<model name>`

3.4.1.2 System Manufacture Date Query

Retrieves the manufacture date of the device.

QUERY	`SYSTem:INFormation:MDATe?`
REPLY	`<manufacture date>`

3.4.1.3 System Calibration Date Query

Retrieves the calibration date of the device.

QUERY	**`SYSTem:INFormation:CDATe?`**
REPLY	**`<calibration date>`**

3.4.1.4 System Serial Number Query

Retrieves the serial number of the laser.

QUERY	SYSTem:INFormation:SNUMber?
REPLY	<serial number>

3.4.1.5 System Part Number Query

Retrieves the manufacturer part number of the laser.

QUERY	**`SYSTem:INFormation:PNUMber?`**
REPLY	**`<manufacturer part number>`**

3.4.1.6 System Firmware Version Query

Retrieves the current firmware version from the laser firmware. The format of the returned firmware version number string is identical to that described in the *IDN? Query.

QUERY	**`SYSTem:INFormation:FVERsion?`**
REPLY	**`<current firmware version>`**

3.4.1.7 System Protocol Version Query

Retrieves the current OBIS protocol version from the laser firmware.

QUERY	**`SYSTem:INFormation:PVERsion?`**
REPLY	**`<current protocol version>`**

The format of the returned firmware version number string is in the format:

```
P<major>.<minor><optional qualifier characters>
```

For example, **`P1.0a`** is a valid firmware version format.

3.4.1.8 System Wavelength Query

Retrieves the actual wavelength (in nanometers) of the laser.

QUERY	`SYSTem:INFormation:WAVelength?`
REPLY	`<wavelength>`

3.4.1.9 System Power Rating Query

Retrieves the power rating (in watts) of the laser.

QUERY	`SYSTem:INFormation:POWer?`
REPLY	`<power>`

The power rating is minimum output power under a given set of operating conditions during the laser life. It is generally the same as nominal power

3.4.1.10 Device Type Query

Retrieves the device type. The device includes laser and controller. At this time, the types of lasers supported by this protocol are OBIS LX (Direct Diode), OBIS LS (OPSL) and OTHER. The set of extended laser-specific commands is determined by the response to this query. The type of the controller is hard coded in the controller.

QUERY	`SYSTem:INFormation:TYPe?`
REPLY	`DDL|OPSL|MINI|MASTER|OTHER`

3.4.1.11 CW Nominal Power Query

Returns the nominal CW laser output power in watts.

QUERY	`SOURce:POWer:NOMinal?`
REPLY	`<x.xxxxx>`

The reply string represents the nominal power value in watts.

3.4.1.12 CW Minimum Power Query

Returns the minimum CW laser output power in watts.

QUERY	`SOURce:POWer:LIMit:LOW?`
REPLY	`<x.xxxxx>`

The reply string represents the minimum power in watts.

3.4.1.12.1 CW Maximum Power Query

Returns the maximum CW laser output power in watts.

QUERY	`SOURce:POWer:LIMit:HIGH?`
REPLY	`<x.xxxxx>`

The reply string represents the maximum power value in watts.

3.4.1.13 Set/Query User-Defined ID

Enters and stores user-defined identification or any other information the user desires to store. The information is stored in nonvolatile memory.

COMMAND	`SYSTem:INFormation:USER <item number>, <item>`
QUERY	`SYSTem:INFormation:USER? <item number>`
REPLY	Item stored at the location pointed to by `<item number>`

The user can enter up to four items, with each comprised of up to 31 characters.The item number starts at zero.

3.4.1.14 Set/Query Field Calibration Date

Enters and stores the date on which the last field calibration was performed. This is normally done by the user or Coherent field service personnel.

COMMAND	`SYSTem:INFormation:FCDate <alphanumeric string>`
QUERY	`SYSTem:INFormation:FCDate?`
REPLY	`<alphanumeric string >`

The number of alphanumeric character is limited to a maximum. of 31 characters.

3.4.2 System State Commands and Queries

System State commands allow a host to retrieve dynamic information describing the current operational state of the laser.

3.4.2.1 System Power Cycle Query

Returns the number of ON/OFF power cycles the laser has endured.

QUERY	`SYSTem:CYCLes?`
REPLY	`<integer cycle count>`

3.4.2.2 System Power Hour Query

Returns the number of hours the laser has been powered on.

QUERY	`SYSTem:HOURs?`
REPLY	`<value in x.xx format>`

3.4.2.3 Diode Hour Query

Returns the number of hours the laser diode has operated. This is defined as the accumulation of time while the “Laser Enable” pin is asserted.

QUERY	`SYSTem:DIODe:HOURs?`
REPLY	`<value in x.xx format>`

3.4.2.4 System Output Power Level Query

Returns the present output power of the laser measured in watts.

QUERY	`SOURce:POWer:LEVel?`
REPLY	`<x.xxxxx>`

The reply string is an NRf value representing the present laser output power measured in watts.

3.4.2.5 System Output Current Query

Returns the present output current of the laser measured in amps.

QUERY	`SOURce:POWer:CURRent?`
REPLY	`<x.xxxxx>`

The reply string is an NRf value representing the present laser output current measured in amps.

3.4.2.6 Base Plate Temperature Query

Returns the present laser base plate temperature. An optional unit indicator may be specified. If the 'C' unit indicator is specified, or if the unit indicator is left off, the returned value represents the laser base plate temperature in degrees C. If the 'F' unit indicator is specified, the returned value represents the laser base temperature in degrees F.

QUERY	`SOURce:TEMPerature:BASeplate? {C|F}`
REPLY	<`x.xU` where U is the unit indicator '`C`' or '`F`'>

The reply string represents the base temperature in NRf format, with a unit indicator of 'C' or 'F' appended.

3.4.2.7 System Interlock Query

Returns the status of the system interlock. The method of determining interlock status is device dependent. This feature may not apply to the laser itself.

QUERY	`SYSTem:LOCK?`
REPLY	`ON|OFF`

Query returns the interlock state in string format.

3.5 Operational Commands and Queries

Operational commands and queries are used to configure and operate the laser from a Host or Controller, as well as for user-level applications.

3.5.1 Laser Operating Mode Selection

Seven mutually exclusive operating modes are available:

- CWP (continuous wave, constant power)
- CWC (continuous wave, constant current)
- DIGITAL (CW with external digital modulation)
- ANALOG (CW with external analog modulation)
- MIXED (CW with external digital + analog modulation)
- DIGSO (External digital modulation with power feedback)
  This operating mode is not supported in some device models.

- MIXSO (External mixed modulation with power feedback)
  This operating mode is not supported in some device models.

The exact meaning of the selected mode is device-dependent.

3.5.1.1 Select CW Mode

Sets the laser operating mode to internal CW and deselects external modulation. The setting is saved in non-volatile memory.

The default setting is CW with constant power or CWP.

COMMAND	`SOURce:AM:INTernal CWP|CWC`

3.5.1.2 Select Modulation Mode

Sets the laser operating mode to CW constant current modulated by one or more external sources. MIXED source combines both external digital and external analog modulation. The setting is saved in non-volatile memory.

COMMAND	`SOURce:AM:EXTernal DIGital|ANALog|MIXed|DIGSO|MIXSO`

3.5.1.3 Laser Operating Mode Query

Queries the current operating mode of the laser.

QUERY	`SOURce:AM:SOURce?`
REPLY	`CWP|CWC|DIGITAL|ANALOG|MIXED|DIGSO|MIXSO`

The reply string represents the present laser operating mode, where CWP and CWC are not modulated externally and the other modes imply external modulation.

3.5.1.4 Set/Query Laser Power Level

Sets present laser power level in watts. Setting power level does not turn the laser on.

COMMAND	`SOURce:POWer:LEVel:IMMediate:AMPLitude <value`
QUERY	`SOURce:POWer:LEVel:IMMediate:AMPLitude?`
REPLY	`<x.xxxxx>`

The reply string represents the present laser power level setting as an NRf value in watts.

3.5.1.5 Set/Query Laser Enable

Turns the laser ON or OFF. When turning the laser ON, actual laser emission may be delayed due to internal circuit stabilization logic and/or CDRH delays.

COMMAND	`SOURce:AM:STATe ON|OFF`
QUERY	`SOURce:AM:STATe?`
REPLY	`ON|OFF`

Query returns the present laser ON/OFF state in string format.

3.5.1.6 Set/Query CDRH Delay

NOTICE

Disabling the CDRH delay renders the OBIS system non-CDRH compliant.

Enables or disables the CDRH five-second laser emission delay.

COMMAND	`SYSTem:CDRH ON|OFF`
QUERY	`SYSTem:CDRH?`
REPLY	`ON|OFF`

Query returns the present CDRH setting in string format.

3.6 OBIS Optional Commands

This section describes the optional commands for OBIS lasers.

3.6.1 Set/Query TEC Enable

Enables or disables temperature control of the laser diode (OBIS LX) or resonator (OBIS LS) via the TEC circuit.

COMMAND	`SOURce:TEMPerature:APRobe ON|OFF`
QUERY	`SOURce:TEMPerature:APRobe?`
REPLY	`ON|OFF`

Query returns the present ON/OFF TEC control state in string format.

3.6.2 OBIS LX-Specific Commands

The commands in this section pertain to OBIS LX Direct Diode lasers (DDL) only.

3.6.2.1 Enable/Undo Laser Power Field Calibration

Starts a self laser power calibration using an internal reference. It is used to re-calibrate the laser power in the field against possible degradation of both laser diode and internal reference during its lifetime. You may undo the field calibration if need be.

COMMAND	SOURce:POWer:CALibration
COMMAND	SOURce:POWer:UNCalibration

The calibration process involved in this command may take a few minutes to finish. DO NOT disrupt the power supply until the process is complete. Status bit 11 is set up as a handshaking mechanism for the host program for the progress of calibration process.

3.6.2.2 Enable/Disable Blanking

Enables or disables Blanking in Analog Modulation mode.

COMMAND	SOURce:AModulation:BLANKing ON|OFF
QUERY	SOURce:AModulation:BLANKing?
REPLY	Returns the present Analog Modulation Blanking-enabled state.

3.6.3 Internal Temperature Limit Queries

These queries return the present internal temperature limit settings. An optional unit indicator may be specified. If the 'C' unit indicator is specified, or if the unit indicator is left off, the returned value represents the internal temperature high or low limit in degrees Celsius. If the 'F' unit indicator is specified, the returned value represents the internal temperature limit in degrees Fahrenheit. The internal temperature represents the temperature taken from a built-in temperature sensor of the microprocessor.

The reply string represents the limit value in NRf format with a unit indicator of 'C' or 'F' appended.

3.6.3.1 Internal Temperature High Limit Query

QUERY	`SOURce:TEMPerature:PROTection:INTernal:HIGH?`
REPLY	<**x.xU** where U is the unit indicator '**C**' or '**F**'>

3.6.3.2 Internal Temperature Low Limit Query

QUERY	`SOURce:TEMPerature:PROTection:INTernal:LOW?`
REPLY	<**x.xU** where U is the unit indicator '**C**' or '**F**'>

3.6.3.3 Diode Temperature Query

Returns the present laser diode temperature. An optional unit indicator may be specified. If the 'C' unit indicator is specified, or if the unit indicator is left off, the returned value represents the laser diode temperature in degrees C. If the 'F' unit indicator is specified, the returned value represents the laser diode temperature in degrees F.

QUERY	`SOURce:TEMPerature:DIODe? {C|F}`
REPLY	<**x.xU** where U is the unit indicator '**C**' or '**F**'>

The reply string represents the diode temperature in NRf format with a unit indicator of 'C' or 'F' appended.

3.6.3.4 Diode Set Point Temperature Query

Returns the diode set point temperature that the TEC controller manages to maintain.

QUERY	`SOURce:TEMPerature:DSETpoint? {C|F}`
REPLY	<**x.xU** where U is the unit indicator '**C**' or '**F**'>

An optional unit indicator may be specified.

- If the 'C' unit indicator is specified, or if the unit indicator is left off, the returned value represents the laser diode temperature in degrees C.
- If the 'F' unit indicator is specified the returned value represents the laser diode temperature in degrees F.

The reply string represents the target temperature in NRf format with a unit indicator of 'C' or 'F' appended.

3.6.3.5 Internal Temperature Query

Returns the present internal laser temperature.

QUERY	`SOURce:TEMPerature:INTernal? {C|F}`
REPLY	<`x.xU` where U is the unit indicator 'C' or 'F'>

An optional unit indicator may be specified.

- If the 'C' unit indicator is specified, or if the unit indicator is left off, the returned value represents the internal laser temperature in degrees C.
- If the 'F' unit indicator is specified the returned value represents the laser base temperature in degrees F.

The reply string represents the internal laser temperature in NRf format with a unit indicator of 'C' or 'F' appended.

3.7 Controls and Queries

The OBIS control and query command set confirms to the Standard Commands for Programmable Instruments (SCPI) and IEEE 488.2 standards. In short, a SCPI control command consists of a header built with keyword(s) plus one or more optional parameters. The header and the parameter(s) are separated by a space. A query command is formed by directly appending a question mark to the end of the header. For more detailed information on SCPI commands and syntax, refer to the SCPI standard documentation.

Following is a brief description of the notation conventions for OBIS commands:

- Parameter(s) following a control command is required.
- Item(s) within the angle brackets following a control or query command is required.
- Item(s) within the curly brackets following a control or query command is optional.
- Acceptable parameters or items required for a control or query command are separated by the OR symbol “|”.
- The upper and lower bounds of the range for a parameter or item are given in parentheses.

Table 3-8 lists the OBIS SCPI control commands for the OBIS Remote (MINI), the OBIS LX (DDL) laser, and the OBIS LS (OPSL) laser.

Table 3-8. OBIS Control Commands

Command	OBIS Remote (MINI)	OBIS LX (DDL)	OBIS LS (OPSL)	Description
*RST	X	X	X	Performs a firmware warm reset. Message handshaking, if enabled, is transmitted prior to the execution of reset. This command may be used to clear a fault condition.
SYSTem:COMMunicate:HANDshaking ON|OFF	X	X	X	Enables|Disables host/controller communication handshaking. This setting is stored in persistent memory so that it remains unchanged after a power ON/OFF cycle.
SYSTem:COMMunicate:PROMpt ON|OFF	X	X	X	Enables|Disables command/query prompt (>). This setting is stored in persistent memory.
SYSTem:AUTostart ON|OFF		X	X	Enables|Disables laser power automatic emission. Note: This setting is overridden by interlock switch, keyswitch, or other hardware mechanisms in the OBIS Remote. This setting is stored in persistent memory.
SYSTem:CDRH ON|OFF		X	X	Enables|Disables CDRH delay. This setting is stored in persistent memory.
SYSTem:DIODe:WARMup ON|OFF		X		Enables|Disables laser diode warm-up process. • If this process is disabled, the laser is capable of starting emission as soon as the electronics is up and running. • If this process is enabled, the laser does not emit until after the warm-up process is complete, even if the laser-on command is issued or the Auto Start is enabled. This setting is stored in persistent memory.
SYSTem:RECovery		X	X	Recovers device from persistent checksum error. This command may also be used to restore device to factory default settings. The laser Status LED illumination, if enabled, displays steady green while the device is recovering. Persistent checksum error is an extremely rare event. Contact Coherent before using this command.
SYSTem:INFormation:AMODulation:TYPe 1|2	X			Selects electrical input impedance for the analog modulation channel of the OBIS Remote. Parameter 1 selects 50Ω while parameter 2 selects 2 kΩ. This setting is stored in persistent memory.
SYSTem:INDicator:LASer ON|OFF		X	X	Enables|Disables illumination of Status LED indicator. This setting does not affect the state of device status bits or fault bits. This setting is stored in persistent memory.
SYSTem:ERRor:CLEar	X	X	X	Clears host/controller communication error records.
SYSTem:INFormation:USER <index>, <item>	X	X	X	Enters and stores user-defined identification or other information. <index> = (0, 3). <item> = (0, 31 characters).
SOURce:AM:INTernal CWP|CWC		X	X	Sets laser internal operating mode. Note: The laser internal and external operating modes are mutually exclusive. CWP = CW constant power; CWC = CW constant current.

Table 3-8. OBIS Control Commands (continued)

Command	OBIS Remote (MINI)	OBIS LX (DDL)	OBIS LS (OPSL)	Description
SOURce:AM:EXTernal DIGital|ANALog|MIXed|DIGSO|MIXSO		X	X	Sets laser external operating mode. Note: The laser internal and external operating modes are mutually exclusive and the laser is required to connect to a OBIS Remote to use these modes. DIGital = digital modulation with current feedback (Digital:Current); ANALog = analog modulation with power feedback (Analog:Power); MIXed = digital + analog modulation with current feedback (Mixed: Current); DIGSO = digital modulation with power feedback (Digital:Power); MIXSO = digital + analog modulation with power feedback (Mixed: Power).
SOURce:POWer:LEVel:IMMediate:AMPLitude <laser_power>		X	X	Sets laser output power level in watts. <laser_power> = (0, 110% nominal power). This command itself will not enable laser emission. If laser emission has been enabled, this command will change the laser output power and the new setting is saved in persistent memory. Note: Setting power level to zero watts does not turn off the electrical power to the laser diode.
SOURce:AM:STATe ON|OFF		X	X	Turns laser emission ON or OFF. Actual laser emission may be delayed due to internal electronic circuit stabilization and/or CDRH delay.
SOURce:TEMPerature:APRobe ON|OFF		X	X	Enables|Disables thermoelectric cooler for DDL laser or thermal stabilization for OBIS LS (OPSL).
SOURce:POWer:CALibration		X		Performs field calibration for analog modulation. This command will result in a match of 5V analog input to 100% nominal power.
SOURce:AModulation:BLANKing ON|OFF	X	X		Enables|Disables Blanking in Analog:Power and `Mixed:current` operating mode.

Table 3-9 lists the OBIS SCPI query commands for the OBIS Remote (MINI), the OBIS LX (DDL) laser, and the OBIS LS (OPSL) laser.

Table 3-9. OBIS Query Commands

Command	OBIS Remote (MINI)	OBIS LX (DDL)	OBIS LS (OPSL)	Description
*IDN?	X	X	X	Returns device identification string that includes information about manufacturer name, product name, nominal wavelength, power rating, the firmware version, and firmware release date in the format, as in this example: "Coherent, Inc - OBIS 405nm 50mW C - V1.0.1 - Dec 14 2010".
*TST?	X	X	X	Returns 0xFFFFFFFF for DDL.
SYSTem:COMMunicate:HANDshaking?	X	X	X	Returns communication handshake setting. Reply = ON|OFF.
SYSTem:COMMunicate:PROMpt?	X	X	X	Returns command prompt setting. Reply = ON|OFF.
SYSTem:AUTostart?		X	X	Returns laser auto emission setting. Reply = ON|OFF.

Table 3-9. OBIS Query Commands (continued)

Command	OBIS Remote (MINI)	OBIS LX (DDL)	OBIS LS (OPSL)	Description
SYSTem:CDRH?		X	X	Returns CDRH delay setting. Reply = ON|OFF.
SYSTem:FAULt?	X	X	X	Returns device fault bits in 32-bit hexadecimal format. Refer to Table 3-6 (p. 3-44) for definitions of device fault bits.
SYSTem:CYCLes?		X	X	Returns number of device power-on cycles.
SYSTem:DIODe:HOUR?		X		Returns accumulated laser emission hours. The returned value has a resolution of two digits after decimal point.
SYSTem:DIODe:WARMup ?		X		Returns diode warm-up setting. Reply = ON|OFF.
SYSTem:HOUR?		X	X	Returns accumulated device operating hours. The returned value has a resolution of two digits after decimal point.
SYSTem:LOCK?	X			Returns OBIS Remote interlock status. Reply = ON|OFF, with ON = Close and OFF = Open.
SYSTem:INForma-tion:AMODulation:TYPe?	X			Returns input impedance type for OBIS Remote analog modulation channel. Reply = 1|2, with 1 = 50 Ω and 2 = 2 kΩ.
SYSTem:NOISe?		X		Returns noise level of laser power. The returned integer is a relative measure of laser power stability. It applies to constant power mode only. A level above 30 is considered noisy. It is normal to see a relatively high noise level when the laser is warming up or when the laser power is changed.
SYSTem:INDicator:LASer?		X	X	Returns LED status indicator setting. Reply = ON|OFF.
SYSTem:ERRor:COUNt?	X	X	X	Returns host/controller communication error count.
SYSTem:ERRor:NEXT?	X	X	X	Returns host/controller communication error record.
SYSTem:INForma-tion:MODel?	X	X	X	Returns device model.
SYSTem:INForma-tion:MDATe?	X	X	X	Returns device manufacture date.
SYSTem:INForma-tion:CDATe?	X	X	X	Returns device calibration date.
SYSTem:INForma-tion:SNUMber?	X	X	X	Returns device serial number.
SYSTem:INForma-tion:PNUMber?	X	X	X	Returns device manufacturer part number.
SYSTem:INForma-tion:FVERsion?	X	X	X	Returns device firmware version.
SYSTem:INForma-tion:WAVelength?		X	X	Returns laser nominal wavelength in nanometers based on a diode operating temperature of 25 degrees Celsius.
SYSTem:INForma-tion:POWer?		X	X	Returns laser power rating in watts.

Table 3-9. OBIS Query Commands (continued)

<table>
<tr><th>Command</th><th>OBIS Remote (MINI)</th><th>OBIS LX (DDL)</th><th>OBIS LS (OPSL)</th><th>Description</th></tr>
<tr><td>SYSTem:INFormation:TYPe?</td><td>X</td><td>X</td><td>X</td><td>Returns device type. Reply = MINI|DDL|OPSL.</td></tr>
<tr><td>SOURce:POWer:LIMit:LOW?</td><td></td><td>X</td><td>X</td><td>Returns minimum laser power output in watts available in CW constant current or CW constant power mode.</td></tr>
<tr><td>SOURce:POWer:LIMit:HIGH?</td><td></td><td>X</td><td>X</td><td>Returns maximum laser power output in watts available in CW constant current or CW constant power mode.</td></tr>
<tr><td>SYSTem:INFormation:USER? <index></td><td>X</td><td>X</td><td>X</td><td>Returns user defined identification. <index> = (0,3).</td></tr>
<tr><td>SOURce:POWer:LEVel?</td><td></td><td>X</td><td>X</td><td>Returns present laser output power in watts.</td></tr>
<tr><td>SOURce:POWer:CURRent?</td><td></td><td>X</td><td>X</td><td>Returns present laser output current in amperes.</td></tr>
<tr><td>SOURce:TEMPerature:BASeplate? {C|F}</td><td></td><td>X</td><td>X</td><td>Returns present baseplate temperature.</td></tr>
<tr><td>SOURce:AM:SOURce?</td><td></td><td>X</td><td>X</td><td>Returns present laser operating mode. Reply = CWP|CWC|DIGITAL|ANALOG|MIXED|DIGSO|MIXSO.</td></tr>
<tr><td>SOURce:POWer:LEVel:IMMediate:AMPLitude?</td><td></td><td>X</td><td>X</td><td>Returns laser output power set level in watts.</td></tr>
<tr><td>SOURce:AM:STATe?</td><td></td><td>X</td><td>X</td><td>Returns laser emission status. Reply = ON|OFF.</td></tr>
<tr><td>SOURce:TEMPerature:PROTection:BASeplate:HIGH? {C|F}</td><td></td><td>X</td><td>X</td><td>Returns maximum laser baseplate temperature without triggering a fault condition.</td></tr>
<tr><td>SOURce:TEMPerature:PROTection:BASeplate:LOW? {C|F}</td><td></td><td>X</td><td>X</td><td>Returns minimum laser baseplate temperature without triggering a fault condition.</td></tr>
<tr><td>SOURce:AModulation:BLANKing?</td><td>X</td><td>X</td><td></td><td>Returns the present ON|OFF Blanking-enabled state.</td></tr>
<tr><td>SOURce:TEMPerature:PROTection:DIODe:HIGH? {C|F}</td><td></td><td>X</td><td></td><td>Returns maximum laser diode temperature without triggering a fault condition.</td></tr>
<tr><td>SOURce:TEMPerature:PROTection:DIODe:LOW? {C|F}</td><td></td><td>X</td><td></td><td>Returns minimum laser diode temperature without triggering a fault condition.</td></tr>
<tr><td>SOURce:TEMPerature:DIODe? {C|F}</td><td></td><td>X</td><td></td><td>Returns present laser diode temperature.</td></tr>
<tr><td>SOURce:TEMPerature:DIODe:DSETpoint? {C|F}</td><td></td><td>X</td><td></td><td>Returns TEC temperature set point for the laser diode.</td></tr>
<tr><td>SOURce:TEMPerature:APRobe?</td><td></td><td>X</td><td></td><td>Returns thermoelectric cooler (TEC) status. Reply = ON|OFF.</td></tr>
<tr><td>SOURce:CURRent:LIMit:LOW?</td><td></td><td>X</td><td>X</td><td>Returns laser diode threshold current in amperes.</td></tr>
<tr><td>SOURce:CURRent:LIMit:HIGH?</td><td></td><td>X</td><td></td><td>Returns laser diode upper current level in amperes. Note: Only valid with OBIS LX lasers with firmware 2.x or later.</td></tr>
</table>

Table 3-10 lists the status code bit definitions.

Table 3-10. Status Code Bit Definitions

Bit Code	Mask Value	Bit Label	Description	
			OBIS Remote	**Laser**
0	00000001	Laser Fault	Logical OR from laser	Laser fault
1	00000002	Laser Emission	Logical OR from laser	Laser emission status
2	00000004	Laser Ready	Logical OR from laser	Laser ready status
3	00000008	Laser Standby	Logical OR from laser	Laser standby status
4	00000010	CDRH Delay	Logical OR from laser	Laser CDRH delay status
5	00000020	Laser Hardware Fault	Logical OR from laser	Hardware related fault
6	00000040	Laser Error	Logical OR from laser	Laser error is queued
7	00000080	Laser Power Calibration	Logical OR from laser	Laser power is within factory calibration specification
8	00000100	Laser Warm Up	Logical OR from laser	Laser warm-up status
9	00000200	Laser Noise	Logical OR from laser	Noise level is over 30
10	00000400	External Operating Mode	Logical OR from laser	External operating mode is selected
...	...	...		
25	02000000	Controller Standby	Keyswitch is in "STANDBY" position	Always 0
26	04000000	Controller Interlock	"INTERLOCK" is open	Always 0
27	08000000	Controller Enumeration	Laser has been enumerated	Always 0
28	10000000	Controller Error	OBIS Remote error flag	Always 0
29	20000000	Controller Fault	OBIS Remote fault status	Always 0
30	40000000	Remote Active	A remote host is connected	Always 0
31	80000000	Controller Indicator	Status word is from OBIS Remote	Always 0

3.8 System Standby and Sleep Mode

For users requiring intermittent use of the OBIS Laser System, two levels of non-lasing conditions are available:

- "**Standby**" represents the thermoelectric cooler (TEC), maintaining constant diode temperature with the laser diode off.
- "**Sleep Mode**" represents that both TEC and the laser are off.

With factory default settings, the OBIS Laser is in the "Standby" condition after the system is turned on and the warm-up procedure is complete.

To start the "Sleep Mode" condition, use this command to turn off the TEC in the laser:

```
SOURce:TEMPerature:ARPobe OFF
```

To return to the "Standby" condition, use this command to switch on the TEC and wait for the warm-up procedure to complete:

```
SOURce:TEMPerature:ARPobe ON
```

NOTICE

The "Sleep Mode" is only possible for the OBIS LX (Direct Diode) system and is not available for the OBIS LS (OPSL) system.

OBIS RS-232 Interface

Table 3-11 lists the pin connections for the RS-232 connector:

Table 3-11. RS-232 Pin Connections

Pin	Signal	Pin	Signal
1	DCD (Data Carrier Detect)	6	DSR (Data Set Ready)
2	Rx (Receive)	7	RTS (Request to Send)
3	Tx (Transmit)	8	CTS (Clear to Send)
4	DTR (Data Terminal Ready)	9	Unused
5	GND (Ground)		

Table 3-12 lists the communication settings for the RS-232 connector:

Table 3-12. RS-232 Communication Settings

Setting	Value
Baud	115200
Parity	None
Data Bits	8
Stop Bits	1
Flow Control	None

Table 3-13 lists the factory default settings:

Table 3-13. Factory Default Settings

Setting	Description
OFF	Command prompt
ON	Command handshake

Table 3-13. Factory Default Settings (continued)

Setting	Description
ON	Laser emission Auto Start
ON	CDRH delay
ON	Laser warm-up
Nominal power	Output power level
0 watts	Minimum power output limit
110% nominal power	Maximum power output limit
CW constant power (CWP)	Operating mode
25°C	Laser diode set temperature[a]
ON	Laser Status LED
ON	Laser thermoelectric cooler[b]
50Ω	OBIS Remote analog input impedance
Degrees Celsius	Unit for all temperature settings

a. LX version only
b. LX system only

I APPENDIX - SERVICE & SUPPORT

This section provides information about:

- How to contact Technical Support (p. 65)
- How to obtain service (p. 66)
- Product shipping instructions (p. 66)

I.1 Technical Support

Coherent provides telephone and web-based technical assistance as a service to its customers and assumes no liability for any injury or damage that can occur at the same time with such services.

Operation of any Coherent laser with any of its interlocks (or safety features) defeated is always at the operator's own risk. Under no circumstances do these support services affect the terms of any Warranty agreement between Coherent and the buyer.

Be prepared to provide the following information to the Product Support Engineer responding to your request:

- Model or part number of your unit
- Laser head serial number
- A description of the problem
- Any corrective steps you may have attempted

I.1.1 Support in the USA and North America

Should you experience any difficulties with your laser or need product or technical information, contact Coherent Technical Support as follows:

- By email: customer.support@coherent.com
- Visit our website: www.Coherent.com
- By phone: +**1 (734) 456-3100**

Telephone coverage is available Monday through Friday (except U.S. holidays and company shutdowns). Inquiries received outside of normal office hours will be captured by our automatic answering system and calls will be quickly returned the next business day.

I.1.2 International Support

If you are located outside the U.S., visit www.Coherent.com for technical assistance, or contact your local Service Representative directly:

- Germany: +49–6071–968–0
- Japan: +813–5635–8680

On the Coherent website, you can also view contact information (telephone numbers and addresses) for Service Representatives worldwide.

I.2 Obtain Service

To obtain service under this warranty, Customer must notify the Company of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service.

The Company shall, in its sole discretion, determine whether to perform warranty service at the Customer's facility, at the Company's facility, or at an authorized repair station.

If Customer is directed by the Company to ship the product to the Company or a repair station, Customer shall:

- Package the product (to protect from damage during shipping) as instructed in “Product Shipping Instructions” next.
- Ship it to the address specified by the Company, with shipping prepaid. back to Coherent in conjunction with recalibration and recertification.
- Coherent shall pay the cost of shipping the Product back to the Customer in conjunction with product failures within the first twelve (12) months of time of sale or during an extended 12-month warranty period.

I.3 Product Shipping Instructions

Refer to section “Appendix - Laser Repacking Procedure” (p. 191) for instructions about factory-recommended repacking of OBIS laser systems.

You must include a Returned Material Authorization number (RMA) assigned by the Company on the outside of all shipping packages and containers. Items returned without an RMA number are subject to return to the sender. Detailed instructions to prepare a product for shipping are provided in the next section.

To prepare a product for shipping to Coherent:

1. Contact Customer Service for a Return Material Authorization number.
2. Attach a tag to the product that includes the name and address of the owner, the person to contact, the serial number, and the RMA number you received from Coherent Customer Service. Pack this tag inside the box.
3. Wrap the product with polyethylene sheeting or equivalent material.
4. Using the original shipping and packaging materials, pack the product.
5. Seal the shipping carton with shipping tape or an industrial stapler.
6. Write the RMA number on the shipping label on the outside of the box.
7. Ship the product to the following address:

    Coherent, Inc.
    Attn: RMA #
    27650 SW 95th Ave.
    Wilsonville, OR 97070 USA

4 OBIS RS-485 INTERFACE

This appendix describes the recommended RS-485 serial communication interface between a host micro-controller and several OBIS lasers, with the primary focus on the hardware configuration.

This section describes the following topics:

- Design description (this page)
- Outbound message transmission (p. 4-72)
- Inbound message transmission (p. 4-79)
- Bus management (p. 4-87)

4.1 Design Description

The communication protocol used by OBIS lasers was designed with reliability and simplicity as the driving factors. While the RS-485 hardware layer supports multi-drop, the communication protocol used by OBIS lasers does not.

A host that wants to control more than one laser must either use one UART (Universal Asynchronous Receiver/Transmitter) for each channel, or multiplex a single UART.

This section describes a simple implementation of a two-channel multiplexer driven by a single UART port on the host micro-controller. The information only describes the serial interface on the SDR port of the host. For information about all of the other signals on the SDR port, refer to Part 2 of the *OBIS LX/LS Operator's Manual*.

Table 4-1 lists the signals transmitted to and from the processor.

Table 4-1. Signals to and from the Processor

Signals to the Processor	Signals from the Processor
OBIS-RX	OBIS-TX
/SDR_IN_USE1	TXen1
/SDR_IN_USE2	TXen2 RXen1 RXen2

The schematic presented in Figure 4-1 on page 4-62 shows the implementation of the required interface to two OBIS lasers. To interface a larger number of lasers, the circuits must be replicated. Note that R7 and R8 serve only to guard against bus fighting when more than one TXen line is asserted at the same time.

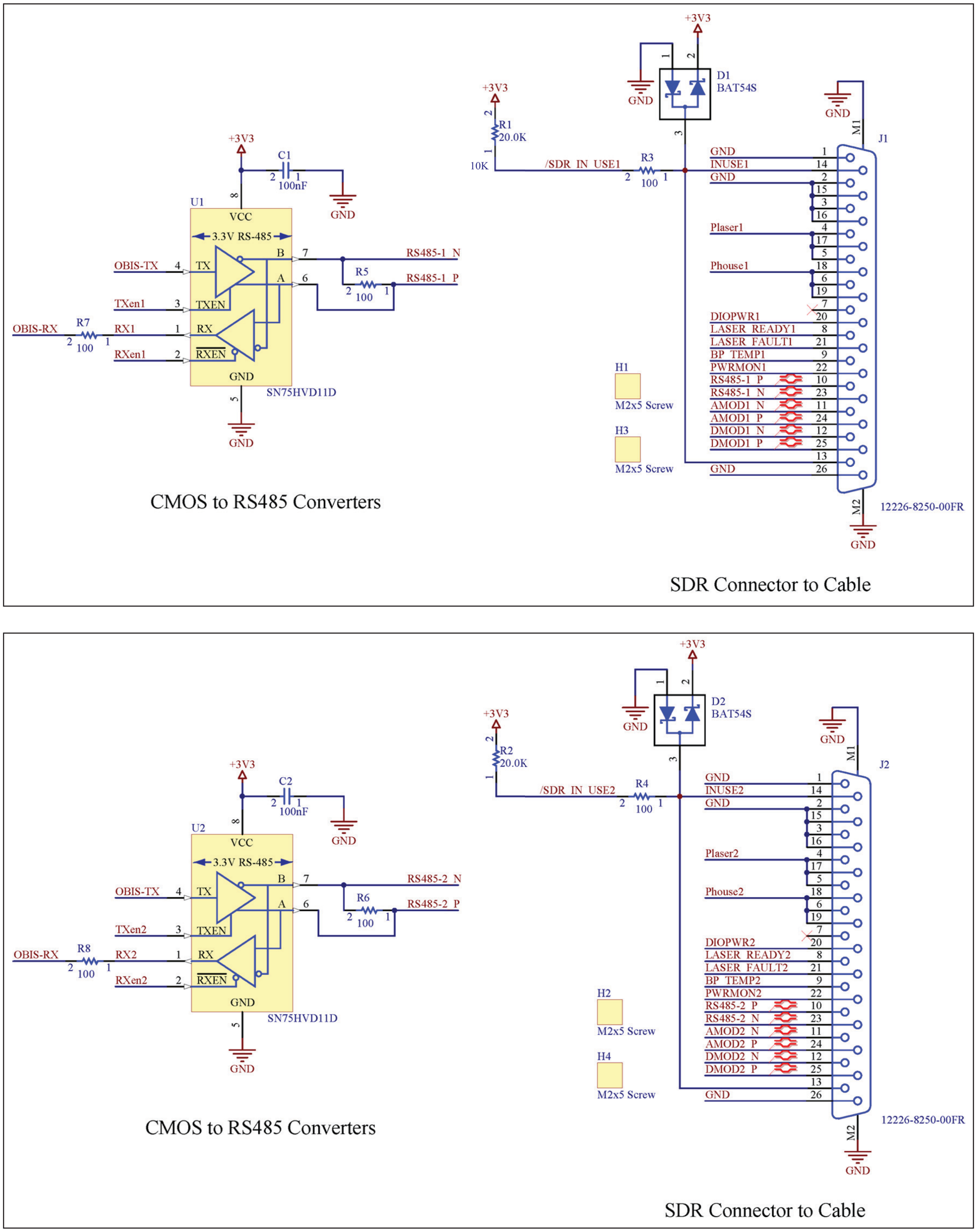


Figure 4-1. OBIS RS-485 Interface Schematic

The required signals between the host and the OBIS Laser are listed in Table 4-2.

Table 4-2. OBIS RS-485 Interface Signals

Signal	Input/ Output	Description	Voltage Level
OBIS-RX	Input	Receive data from OBIS Laser	3.3 VCMOS
/SDR_IN_USE1 , /SDR_IN_USE2	Input	Pull up to 3.3V. Signal grounded by OBIS Laser when plugged into SDR port.	3.3 VCMOS 3.3 VCMOS
OBIS-TX	Output	Transmit Data to OBIS Laser	3.3 VCMOS
TXen1, TXen2	Output	Active High. Signal must go active to address the corresponding OBIS Laser.	3.3 VCMOS
+12V		Power to OBIS Laser	+12V
Ground		Ground	GND

All signals are driven at 3.3 volt CMOS logic levels. This scheme can be expanded to drive as many OBIS lasers as required. The /SDR_IN_USEx lines are pulled up by the host and grounded when an OBIS Laser is connected to its SDR cable. The host should use this line to enable or gate +12V power to the corresponding SDR port.

When the RS-485 bus is idle, all TXen lines must be low. To address a laser, the corresponding TXen line needs to be raised (and the RXen line lowered), and data transmission can start. When the last response is received, the host can drop the TXen line.

If a port's SDR_IN_USE line rises while its TXen line is high (indicating that the laser has been unplugged), the host should terminate transmission. Since the OBIS protocol is a strict master-slave, query-response relationship, there is no interrupt line from the laser to the host to initiate communication. The host should regularly round robin poll all SDR ports with a low logic level on their /SDR_IN_USE pin.

Figure 4-2 shows the timing relationship between communication signals.

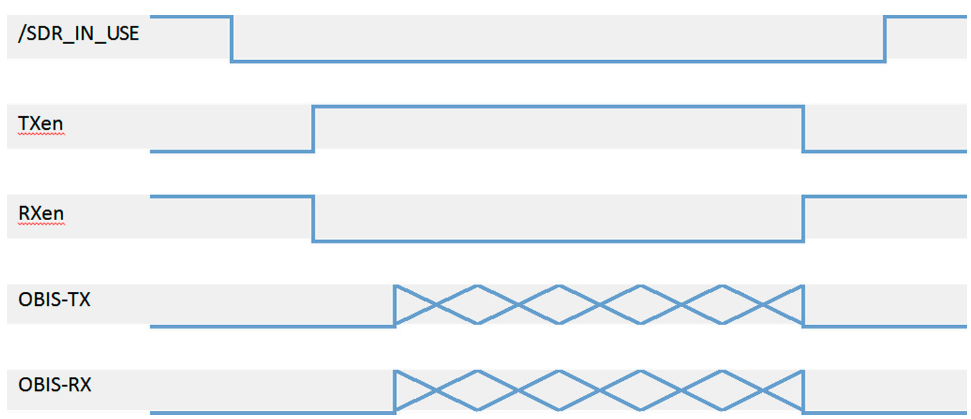


Figure 4-2. Timing Relationship for Communication Signals

4.2 Coherent Connection Bus Functional Overview

The Coherent Connection Bus (CCB) is intended to provide message transmission between a single master device and one or more slave devices connected to the bus.

Two general categories of messages are exchanged:

- *Bus Management Messages* are used to manage bus operation, assign addresses, detect device disconnection, and notify client applications of bus events that may affect the client. These messages are sourced by CCB on both master and slave devices.
- *Standard Messages* are sourced by Client Application Software to implement a specific Application Protocol. Typically the client application running on the master device sends standard messages (commands or queries) to slave devices. After receiving a standard message the slave device responds in a manner defined by the received message, which may include replying to the master with another standard message containing query data or an acknowledgment that a command was received and acted upon.

A slave device must respond to a master's standard message request (non-broadcast) within 700 milliseconds before the master re-sends. The master retries sending up to three times before declaring a device disconnection event. The slave must reply in form of a general acknowledgment or a specific response. This implies that all messages have a 'handshake' mechanism. The master won't send another message type to the slave until it responds to the last message sent.

Message flow is always between master and slave devices, slave devices may not communicate with each other on a peer-to-peer basis.

Each device is assigned a unique address on the bus. The master device is always at fixed address 0. When slave devices power-up they utilize defined Bus Management Messages to request an address from the master device. After an address is acquired, the slave uses that address until the master assigns it a new address (usually at the next power cycle).

Master devices may utilize Bus Management Messages to poll (ping) devices on the bus. Polling is generally used to verify the presence of a device when it has not been heard from for awhile (for the purpose of device disconnection notification).

4.3 OEM RS-485 Hardware Design Requirements

The CCB protocol requires one RS-485 serial port for communication with the bus and a timer with sufficient resolution to detect idle line conditions.

The transmission collision detection function of CCB requires hardware that allows the receiver to be enabled while transmitting data. The software must be able to listen to what is being transmitted.

It is important that the bus be designed to prevent reflections that could cause data corruption, which normally is done with terminating resistors at each end of the bus.

Under some circumstances an RS-485 bus may operate without termination, which is desirable from the standpoint of simplification and power savings. However, elimination of reflections that could corrupt bus data takes precedence.

Bus design and termination requirements are beyond the scope of this document, but need to be taken into account during the hardware design phase.

Due to the nature of 2-wire half-duplex RS-485, it is important that the transmitter be disabled whenever a device is not actually transmitting data. Although a RS-485 bus can tolerate brief periods of time when more than one transmitter is driving the bus, it is extremely important that these time periods are kept to a minimum. It is therefore important that the length of time a transmitter is enabled be kept to a minimum.

If the hardware supports end of transmission interrupts (an interrupt that occurs after the final message bit has been sent), it is recommended that the transmitter is:

- Enabled at the very start of message transmission
- Disabled as soon as the end of transmission interrupt is generated.

Not all processors/UARTs provide for an end-of-transmission interrupt feature. When this is the case, it is acceptable to implement a circuit to perform Automatic Send Data Control (ASDC).

This feature automatically enables the transmitter at the beginning of data transmission and disables it immediately after the last stop bit has been sent. The book, *Serial Port Complete* (see “Applicable Documents” (p. 4-103) describes several simple circuits that may be used to implement this feature.

If ASDC is not implemented, then the software requires another method to determine exactly when the last bit of the last message byte has been clocked out of the transmit register so the transmitter can be immediately disabled.

This requirement could be met with either:

- A UART that can generate an interrupt when the transmitter SHIFT register is empty (a feature that is somewhat rare), or
- A high-resolution timer that can provide an interrupt one character transmission time after the last message byte has been shifted into the transmit shift register (at 921.6kbs this could be as short as 11 µsecs), or
- Simply after receiving the last message byte that was transmitted.

However the transmitter is controlled, it is the responsibility of the implementing engineer to disable the transmitter whenever the device is not actually transmitting. This is necessary to prevent data corruption or seizure of the bus, which would prevent any other device communication.

A software implementation requires access to a timer that has sufficient resolution for detecting idle line conditions and expected event timeouts.

Each device on the bus must have a unique address. Historically assigning RS-485 addresses to devices has been the responsibility of the end user, typically by setting a series of DIP switches or by programming an address into non-volatile storage on the device.

Manual address assignment has some undesirable ramifications, including increased costs for hardware, customer training, and documentation; bus operation and troubleshooting problems caused by duplicate addresses; difficulty in swapping out devices; and problems due to misunderstandings or errors on the part of users.

To eliminate manual address assignment, the CCB protocol allows for a system of automatic address assignment. This relieves the client hardware of any addressing responsibilities.

4.4 Message Structure

All messages sent across the CCB contains a common five-byte header. A message may optionally include up to 255 bytes of data.

The five-byte header is organized as shown in Table 4-3.

Table 4-3. CCB Message Header

Byte	Abbreviated Name	Function
0	Sadd	Source device address
1	Dadd	Destination device address
2	Flags	Message Flags byte
3	Tag	Arbitrary value
4	Len	Data Length

The Flags byte is set by the master device when a command or query transaction is initiated. The responding device returns the Flags byte received in a command or query in any resulting reply message.

The Tag byte is an arbitrary value passed from the original sender through to the destination device. The responding device returns the unmodified Tag byte received in a command or query in any resulting reply message. This byte may be used by the originating node to associate a reply message with the initial command/query message.

One possible implementation would be to use an incrementing counter as the tag value for each initiating message sent. This way, the sender could easily determine if a reply was dropped between consecutive messages.

When the message contains no additional data, the Len byte is set to 0. This type of message may be used exclusively by the Bus Management function and not used by client applications.

When the message contains additional data, limited to 255 bytes, it is appended to the four-byte header and the header Len byte set to the length of the additional data segment.

4.5 Message Framing

Because CCB messages may vary in length, it is necessary to indicate the start and end point of each message, as shown in Figure 4-3.

SOM	Sadd	Dadd	Flags	Tag	Len	Data	EOM

Figure 4-3. CCB Message Framing

The CCB protocol prefixes a SOM (Start of Message) indication to the beginning of the message, populates the source address (Sadd field above) with the local device address, sets any flags appropriate to the type of message being sent, and appends an EOM (End of Message) indication to the end of the message.

Both binary and ASCII data may be transmitted. For this reason, it is not possible to use a single byte to indicate the start or end of a message, since it would be impossible to determine if the byte value represented a start/end of message handshaking character or a valid message data byte.

Assuming that a message packet begins with a single STX character (value 0x02) and ends with a single ETX character (value 0x03), consider the binary message packet shown in Figure 4-4:

0x02	Sadd 0x00	Dadd 0x01	Flag 0x01	Tag 0xnn	Len 0x03	D_0 0x02	D_1 0x10	D_2 0x22	0x03

Figure 4-4. Binary Message Packet

The intent of this message is to represent a single binary message packet. However, it contains ambiguities. The Len byte of 0x03 may be interpreted as an ETX since it has the same value. Similarly, the D0 value of 0x02 may be interpreted as an STX instead of a data byte as was intended. So, instead of a single binary message, this packet could be interpreted as two separate messages and there is no unambiguous method that software can use to guarantee correct interpretation.

To resolve this problem, the framing logic incorporates an escape feature. During transmission, any byte value that is intended to function as a handshake character must be preceded by a DLE character (Data Link Escape, ASCII value 16 decimal or 0x10 hex). This method also requires that any byte in the transmitted data stream with the same value as DLE be sent as two consecutive DLEs.

Using this method, the example above would be framed as shown in Figure 4-5:

		Sadd	Dadd	Flag	Tag	Len	D_0		D_1	D_2		
0x10	0x02	0x00	0x01	0x01	0xnn	0x03	0x02	0x10	0x10	0x22	0x10	0x03

Figure 4-5. Example Using DLE Character

When the leading 0x10 DLE character is read, it “escapes” the very next byte, which is then interpreted unambiguously as an STX byte. In this way, it is easy to identify that sequence as the start of a message. When the Len byte of 0x03 and the D0 byte of 0x02 are read, they are not interpreted as ETX or STX framing characters because they were not immediately preceded by DLE bytes. The D1 value of 0x10, however, is a valid data byte so must be preceded by DLE.

As data is read from the bus, each time the DLE character is encountered the next byte is interpreted as a handshake character. Two consecutive DLE characters are interpreted as one data byte with value equal to DLE. In summary:

- A DLE, STX sequence always represents the start of a message (the SOM sequence)
- A DLE, ETX sequence always represents the end of a message (EOM sequence)
- A DLE, DLE sequence always represents a single data byte with value 0x10 as part of the message body

Table 4-4 lists the CCB protocol framing characters.

Table 4-4. CCB Protocol Framing Characters

ASCII Character	Hex Value	Protocol Function
STX	0x02	Start of message data
ETX	0x03	End of message data
DLE	0x10	Data Link Escape

NOTICE
Any DLE escape characters added to the message data are not reflected in the message length field of the message.

The message length field is only guaranteed accurate at the application layer. The protocol stack does not use the message length field in determining how many bytes to send or receive. A complete message is framed by SOM/EOM sequences.

The last step of message framing involves computing a checksum and appending it to the message. This is covered in more detail under “Outbound Message Framing Function” (p. 4-74).

4.6 Address Allocation

Device addresses are one byte in size allowing a total of 255 possible addresses. Some of these addresses are reserved for specific purposes.

Table 4-5 lists the device address allocated for CCB Addresses.

Table 4-5. CCB Address Allocation

Device Address	Function
0	Master device
0x01-0xfd	Unique addresses assigned to slave devices
0xfe	Power up default slave address
0xff	Broadcast address (all slave devices respond to this)

The master device is always at address 0 and should power-up with this address.

Address 0xfe is the power up default address of all slave devices and also the address a slave reverts to following reception of a Bus Reset message. The automatic address assignment function then assigns a unique address in the 0x01–0xf d range to slave devices.

Address 0xff is a broadcast address. All slave devices receives messages sent to the broadcast address.

4.7 Message Flags Byte

The Message Flags byte contains bits that are used for various purposes, listed in Table 4-6.

Table 4-6. Message Flags Bit Definitions

Bit	Name	Function
0	BUSMGMT	Message is a bus management message
1	SRCCCB	Message originated from CCB stack
2	SRCCONT	Message originated from master device (controller)
3	Reserved	

Table 4-6. Message Flags Bit Definitions

Bit	Name	Function
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved	

The BUSMGMT bit is set on any BUS MANAGEMENT message. When this bit is set on an originating message, the destination protocol stack handles the message response rather than passing it up to the destination application for processing. There is nothing to prevent the application layer on a master device from sending bus management messages, but the protocol stack on the destination device always handles the response. The BUSMGMT flag should be used to do the address assignment, ping, and bus reset commands.

The SRCCCB bit is set on messages that are exchanged between CCB protocol stacks. This bit is used to determine message response routing. If the SRCCCB bit is set on a message, then the message is always handled by the protocol stack.

The SRCCONT bit is set on messages that are sourced by the application layer on a master device (usually a controller). These messages are routed to the application layer on the controller device. *Use the SRCCONT flag to do all the SCPI commands and queries*.

When SRCCCB and SRCCONT are both clear a message is passed on to the host.

4.8 LRC Computation

A checksum is computed on each message during framing and sent as the last byte of a message following the two-byte EOM sequence.

The CCB checksum algorithm uses a simple Longitudinal Redundancy Check with a seed value of 0xff. The entire message frame (including SOM and EOM sequences, the message body, and all inserted DLE escape characters) are included in the calculation. Pseudo code for the algorithm is as follows:

```
Set LRC = 0xff
For each byte c in the message
do
      Set LRC = LRC XOR c
end do
```

4.8.1 Example of a Framed Command and Response Over RS-485

The following example illustrates a framed command and response over RS-485 for checking laser status with handshaking turned on. Non-ASCII data is represented in hexadecimal format delimited by []. For example, decimal 16 is represented as 0x10 in hex and is shown as [10].

```
      (Master) Command TX:
[10][02][00][DF][04][00][0D]SYST:STAT?[0D][0A][00][10][03]
[35]
      (Slave)    Response RX:
[10][02][DF][00][04][00][0F]00000180[0D][0A]OK[0D][0A][00]
[10][03][27]
```

For additional examples (written in C code), refer to the **CCBparser.zip** file. This file is available for download from the Coherent website:

https://www.coherent.com/assets/software/CCBparserExample.zip

4.8.2 Example of a Complete Query and Answer via RS-485

The following example shows what to send over RS-485 to activate the handshaking in the OBIS.

First, do the address assignment to establish communication:

Figure 4-6 shows Binary Data sent over the bus with an OBIS address of 3:

0x10	0x02	0x00	0xFF	0x01	0x00	0x03	0x80	0x03	0x00	0x10	0x03	0x80

Figure 4-6. Activate Handshaking Using RS-485

Now send the handshaking command/query over the bus. The following example is 36 characters.

```
SYSTem:COMMunicate:HANDshaking ON\r\n
```

This results in a binary data sent over the bus, as shown in Figure 4-7:

0x10	0x02	0x00	0x03	0x00	0x00	0x24	0x53	0x59	0x53	0x54	0x65	0x6D
0x3A	0x43	0x4F	0x4D	0x4D	0x75	0x6E	0x69	0x63	0x61	0x74	0x65	0x3A
0x48	0x41	0x4E	0x44	0x73	0x68	0x61	0x6B	0x69	0x6E	0x67	0x20	0x4F
0x4E	0x0D	0x0A	0x00	0x10	0x03	0xE5						

Figure 4-7. Binary Data Sent Over the Bus

The LX responds with “OK\r\n”, as shown in Figure 4-8:

0x10	0x02	0x03	0x00	0x00	0x00	0x05	0x4F	0x4B	0x0D	0x0A	0x00	0x10
0x03	0xFB											

Figure 4-8. Response to Query

IMPORTANT!
Make sure to use Complete Termination `\r\n`.

OBIS RS-485 communication enforces use of both the carriage return AND newline characters at the end of the command/query string.

Not including the '\n' causes the CCB stack to return an 'ACK' because the datalink layer saw a valid message, but the application layer didn't understand it. This results in a return of error code "-220" ("Invalid Parameter") when sending the next command or query.

This event only occurs when talking over RS-485.

4.9 Outbound Message Transmission

4.10 Recommended Outbound Message Functional Flow

Figure 4-9 shows the recommended functional flow for an outbound message.

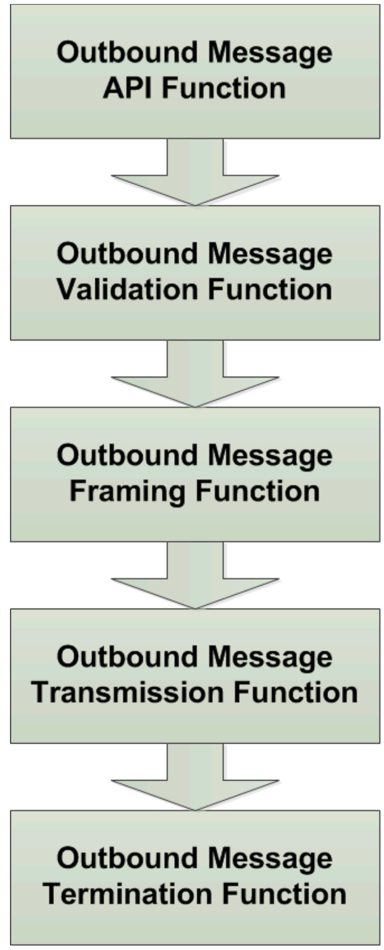


Figure 4-9. Outbound Message Flow

4.11 Outbound Message Validation Function

4.11.1 Input Requirements

An outbound message validation function receives fully assembled outbound messages from the outbound message function.

4.11.2 Processing Requirements

The following validations are performed on outbound messages received from the function.

1. The destination address is a valid address.
2. The destination address does not equal the local device address (a device cannot send a message to itself).
3. If the message is a broadcast message, the local source address must be zero.

4.11.3 Output Requirements

If any of the validation tests fails, the failure event is passed to the Outbound Message Termination Function for return to the client.

If all validation checks are successful, the message is passed on to the Outbound Message Framing Function for further processing.

4.12 Outbound Message Framing Function

4.12.1 Input Requirements

An outbound message framing function receives validated messages from the outbound message validation function.

4.12.2 Processing Requirements

The Outbound Message Framing Function constructs outbound message packet frames for transmission. Figure 4-10 shows how a simple message from the master (source address 0) is framed following the framing function. Notice how the D1 byte, with a value of 0x10 that is equal to DLE, is preceded by another DLE byte.

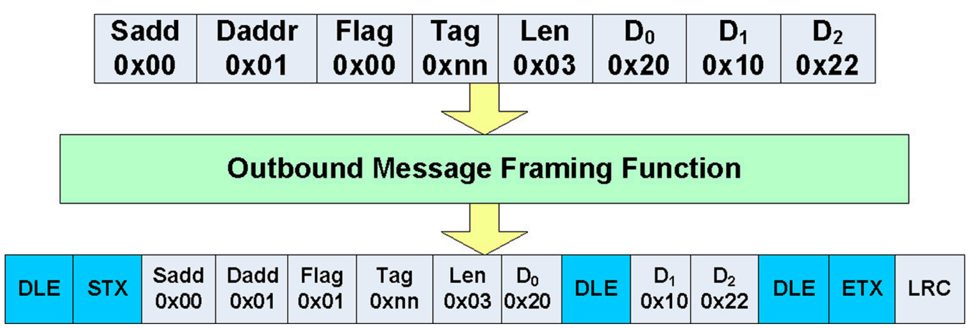


Figure 4-10. Outbound Message Framing

The resulting message packet is fully framed using the following sequence:

1. The first two bytes of the message contain the two-byte SOM marker (DLE, STX)
2. Each byte of the message, as received from the Outbound Message Validation Function, is appended to the SOM.

   As each byte is appended, its value is compared to the DLE value. If the data byte is equal to DLE, then the data byte is appended to the message as two consecutive DLE bytes. **Note that any DLE expansions in the message body after framing cause the Message Body Length byte to be invalid during transmission. After the DLEs are stripped by the receiver, the Length byte is again accurate.**
3. Append the two-byte EOM sequence (DLE,ETX) to the message modified message data.
4. Compute the LRC for all message bytes and append it to the message.

At this point, the assembled packet may be passed to the Outbound Message Transmission Function for further processing.

NOTICE

The LRC byte is computed and transmitted by the Outbound Message Transmission Function, but the LRC is NOT appended to the message by the framing logic.

4.12.3 Output Requirements

The fully framed message assembled by the Outbound Message Framing Function is passed to the Outbound Message Transmission Function.

4.13 Outbound Message Transmission Function

4.13.1 Input Requirements

The Outbound Message Transmission Function receives a fully assembled and framed message from the Outbound Message Framing Function.

4.13.2 Processing Requirements

The Outbound Message Transmission Function transmits the complete message packet to the destination address and returns the transmission results back to the Outbound Message Termination Function.

Message transmission involves the following steps performed in sequence; however, the task is more complex when error checking and retries are performed.

1. Wait for idle bus condition
2. Transmit the entire message. Each byte transmitted is echoed back and buffered as received.

NOTICE
Devices that use idle frame detection to detect the end of a received message are very sensitive to interbyte timing.

For this reason the maximum time between any two message bytes cannot exceed the time it takes to transmit one character frame at the current baud rate. Doing so causes invalid End-Of-Message (EOM) detection on the receiving device.

3. Compare transmitted message and received message byte-by-byte, looking for a mismatch
4. If the received message does not match the transmitted message, perform collision-based retries

The flow chart in Figure 4-11 (p. 4-77) depicts the logical flow of message transmission including collision detection and retries.

The following sections describe these operations in more detail.

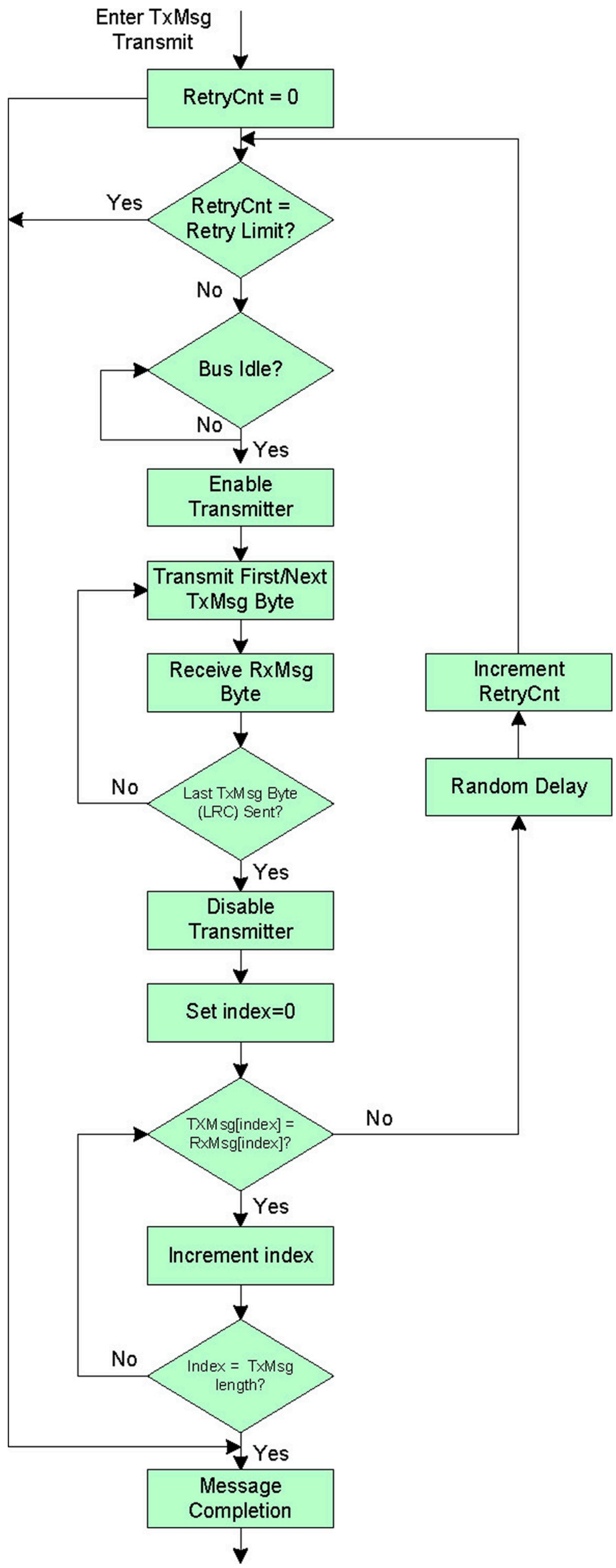


Figure 4-11. Outbound Message Transmission Flow

4.13.3 Transmitter Control

If the hardware supports the recommended Automatic Send Data Control hardware, then enabling/disabling of the transmitter is done in hardware and may be ignored.

However, if ASDC is not available in hardware, the software must ensure that the transmitter is enabled only during message transmission and disabled immediately upon transmission of the last stop bit in the last byte of the message (the LRC byte).

4.13.4 Message Transmission Retries

When message transmission fails due to a bus collision, the transmit function attempts a maximum of *N* retries (default four). *N* may optionally be made an adjustable parameter.

4.13.5 Idle Bus Detection

Message transmission does not commence until a bus idle condition is detected.

The bus idle condition is defined as the bus being in a spacing state for a period of three character transmission times (30 bit times).

Idle Time = (1/BaudRate)*30

At a baud rate of 921600 bps, the bus spacing state must exist for ~32.5 µs for the bus to be considered idle.

4.13.6 Collision Detection

NOTICE
Collision detection requires that the hardware support enabling of the receiver during data transmission.

During message transmission, the receiver circuit is enabled. This results in a character being received each time a character is transmitted. As the message is transmitted, the resulting received data should be buffered.

After the message and checksum have been transmitted, a byte-by-byte comparison is performed between the transmitted message and the resultant received message.

- If the messages are identical, then no collision occurred.

- If, however, the messages differ in any way, a collision or other data corruption event occurred.

In the event of collision or corruption, the Outbound Message Transmission Function waits for a random delay period to pass and then retries message transmission from the beginning.

Transmission retry attempts repeats until a complete uninterrupted message has been successfully transmitted without collision, or until all retry attempts have been exhausted.

4.13.7 Random Delay

If a collision or corruption is detected during message transmission, it is important that each device detecting a collision on the bus (that is, each transmitting device involved in the collision) wait a random period of time before attempting transmission.

The algorithm used to determine the back-off and retry delay is a truncated binary exponential back off algorithm. *Truncated* means that, after a certain number of increase, the exponentiation stops—that is, the retransmission timeout—reaches a ceiling and thereafter does not increase any further.

After i collisions, a random number of slot times between 0 and $2^i - 1$ is chosen. After the first collision, the sender would wait 0 or 1 slot times before retry; after the second collision it would wait 0-3 slot times, etc. After the fourth and subsequent consecutive collisions, the slot time delay is limited to 0-15 slot times. Note that there is no provision for discarding a message after many collisions; message transmission retries until it succeeds.

A slot time is defined as the time to transmit 64 characters at the selected baud rate. The slot time is computed as ceil((1/baudrate)*640). At 921600 bps, the slot time would be:

(1/921600)*640 = 0.00069, or 690 microseconds

4.14 Inbound Message Transmission

4.14.1 Inbound Message Functional Diagram

Figure 4-12 shows a functional diagram for the Inbound Message flow.

4.14.2 Inbound Message Receiving Function

4.14.2.1 Input Requirements

The Inbound Message Receiving Function is an event driven function that is kicked off when a data byte is received from the CCB.

4.14.2.2 Processing Requirements

Data bytes are read from CCB port and scanned to identify complete messages. A complete message includes the two-byte SOM sequence, all bytes that immediately follow SOM through the two-byte EOM sequence that frames the message, and the LRC byte that immediately follows the EOM sequence.

Conceptually, message reception involves the following steps, performed in sequence (some implementation details left out for clarity):

1. Scan the incoming data stream looking for a SOM sequence.
2. After SOM is discovered, buffer data, compute LRC and scan for an EOM sequence while limiting any received message to the maximum allowable length.
3. When EOM sequence found, read the LRC checksum byte that follows EOM.
4. Compare the received checksum byte to the computed LRC.
5. If checksum values are equal, a complete message was received.
6. Compare the received checksum byte to the computed LRC.
7. If checksum values are equal, a complete message was received.

Figure 4-12 shows the functional flow for an Inbound Message.

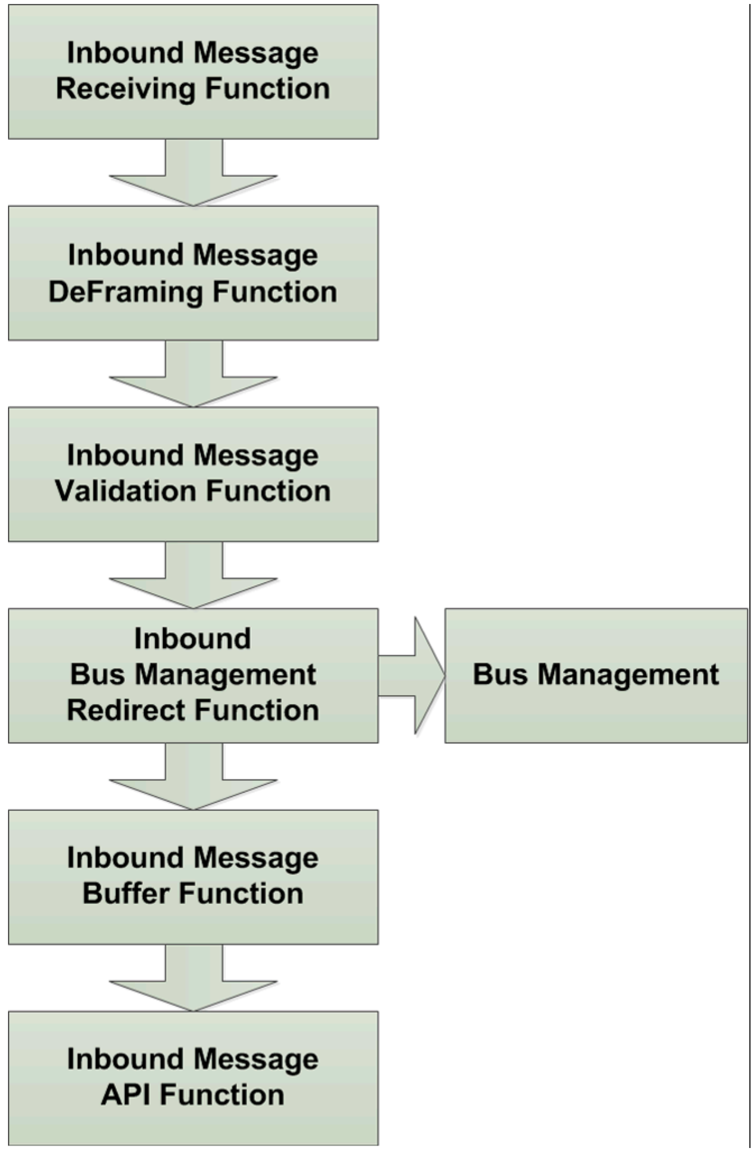


Figure 4-12. Inbound Message Flow

4.14.2.3 Output Requirements

Complete received messages passing checksum validation are passed to the Inbound Message Deframing Function.

Figure 4-13 shows the Receiving flow for an Inbound Message.

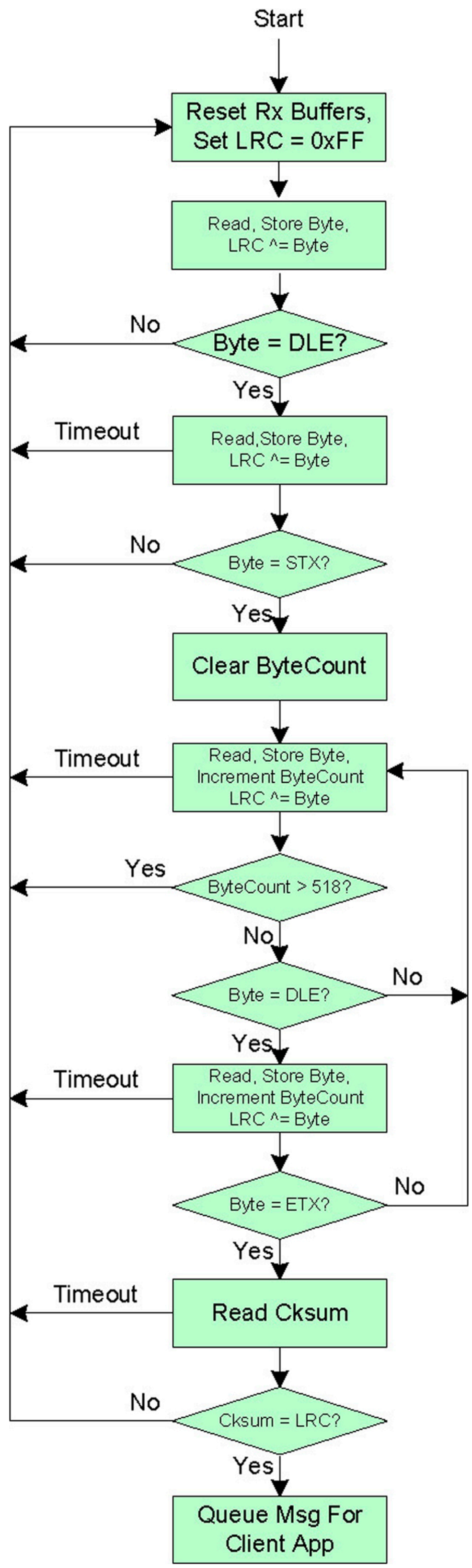


Figure 4-13. Inbound Message Receiving Flow

4.14.3 Inbound Message Deframing Function

4.14.3.1 Input Requirements

The Inbound Message Deframing Function receives complete messages that have passed checksum validation from the Inbound Message Receiving Function.

4.14.3.2 Processing Requirements

The Inbound Message Deframing Function removes all framing characters from received messages so subsequent functions can easily validate the message and buffer it for the client.

Framing characters are removed from inbound messages using the following rules:

1. Strip the two-byte SOM sequence from the beginning of the message.
2. Perform a byte-by-byte copy of the received message into a new message buffer. As the copy proceeds each byte is compared to DLE. When a DLE is encountered, it is discarded and the very next byte from the received message is taken literally and placed in the new message buffer without regard to value. Bytes placed in the message buffer are counted to determine the exact message length.
3. When the two-byte EOM sequence is encountered the scan terminates. EOM bytes are not stored in the new message buffer or counted.

Figure 4-14 depicts the deframing logic.

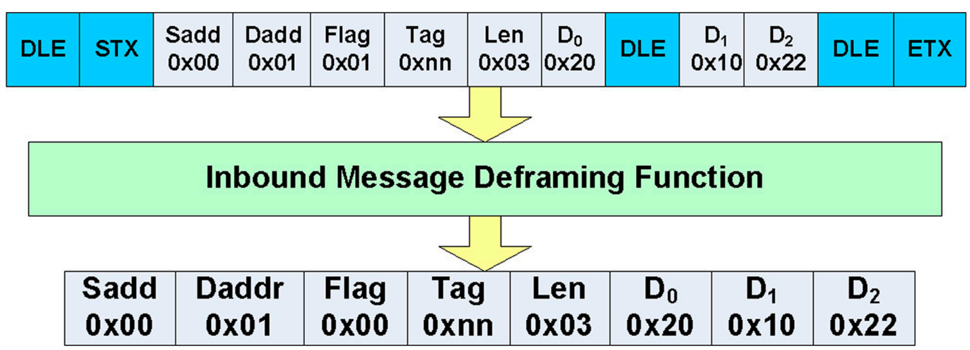


Figure 4-14. Deframing Inbound Messages

4.14.3.3 Output Requirements

The Inbound Message Deframing Function passes the new message buffer and the size of the new message to the Inbound Message Validation Function.

4.14.4 Inbound Message Validation Function

4.14.4.1 Input Requirements

The Inbound Message Validation Function receives deframed messages and their associated length from the Inbound Message Deframing Function.

4.14.4.2 Processing Requirements

Because the CCB is a bus, all messages that are sent on the bus are received by all connected devices. It is necessary to ignore messages that are not intended for the local device.

When a newly deframed message is received, the Inbound Message Validation Function accepts messages according to the following rules:

1. If the local device is a Master device (local address 0), messages are accepted only when the message destination address is 0.
2. If the local device is a Slave device, messages are accepted when either of the following statements is true:
   - The destination address equals the local device address
   - The destination address is 0xff (broadcast message)
3. All other messages are discarded.

4.14.4.3 Output Requirements

Validated Inbound messages are passed to the Inbound Bus Management Handling Function.

4.14.5 Inbound Bus Management Redirect Function

4.14.5.1 Input Requirements

The Inbound Bus Management Redirect Function receives validated inbound messages from the Inbound Message Validation Function.

4.14.5.2 Processing Requirements

The BUSMGMT bit of the Message Flags byte of each inbound message is examined to determine if the message should be buffered for the client or processed by the Bus Management Function as a bus management message:

- If the BUSMGMT bit of the Message Flags byte is set, the message is a bus management message.
- If the BUSMGMT bit of the Message Flags byte is clear, the message is a client message.

In all cases, this function updates an active device list with a timestamp indicating the last time a message was received from the device. The timestamps on the active device list are used by the Bus Management Device Disconnection Detection function to determine when to ping inactive nodes for the purpose of detecting disconnected devices.

4.14.5.3 Output Requirements

Inbound messages that are bus management messages are passed to the Bus Management Function.

Inbound messages that are not bus management messages are passed to the Inbound Message Buffer Function.

The updated timestamps for active devices are made available to the bus management function as a required part of device disconnect detection logic.

4.14.6 Inbound Message Buffer Function

4.14.6.1 Input Requirements

The Inbound Message Buffer Function receives validated, non-Bus Management messages from the Inbound Bus Management Function.

4.14.6.2 Processing Requirements

Received messages and their associated size are buffered for reading by the client application through the API in a first in, first out, fashion.

A maximum number of messages is buffered. The default value is 6, and that value can be configured. Only complete messages are buffered; if there is not enough buffer space remaining for a new message, the message is discarded.

This function maintains a count of the number of messages currently buffered.

4.14.6.3 Output Requirements

The number of buffered messages is made available to the Inbound Message API Function.

4.14.7 Inbound Message API Function

4.14.7.1 Input Requirements

Inbound messages are read when a Client Application calls the inbound message API function.

4.14.7.2 Processing Requirements

The API function is a blocking call. If no messages are currently buffered, the function does not return until a message arrives to satisfy the read request.

4.14.7.3 Output Requirements

On success, an *N* byte message is stored in the specified client read buffer.

4.15 Bus Management

4.15.1 Bus Management Overview

Each device participating on the CCB requires a unique address. Additionally, since the CCB hardware has no way to identify when devices are connected or removed from the bus, it must be done in software. Lastly, since the CCB is a "cloud" rather than a collection of point-to-point connections, it may be necessary to identify which slave device is connected to a each port. To manage these requirements the CCB Bus Management Logic implements for five specific functions:

1. Detection of newly connected slave devices on the bus
2. Assignment of unique addresses to slave devices
3. Associate master device ports to specific slave devices
4. Detection of slave devices that are no longer responding on the bus
5. Apprise client application on master device of bus status

Master and Slave devices are assigned unique bus management roles.

Slave devices perform the following bus management functions:

- Acquire a unique bus address from the master device
- Conditionally respond to port identification requests
- Respond to ping requests from the master device

The **Master device** perform the following bus management functions:

- Issue a Bus Reset command at start-up to reset connected slave devices
- Respond to slave address requests with a unique bus address assignment
- Perform port detection logic to associate slave devices to physical ports
- Issue ping requests to devices that have not been active for a period of time
- Maintain a list of all devices on the bus using serial number as unique identifier
- Allow client application to query the list of active slave devices on the bus
- Notify client application as devices are attached and detached from the bus

The bus management command and reply protocol utilizes the same message header used by the application protocol; however, bus management message packets are uniquely identified by the BUSMGMT bit in the message flags byte. When a message is received with the BUSMGMT bit set, it is passed to the bus management function for handling instead of being passed to the client application.

The following subsections provide overviews of the above functionality.

4.15.2 Bus Management Address Assignment Overview

The Master device always powers up at fixed address 0. At Power-up, the master device issues a single Bus Reset command to place any configured slave devices into an Address Acquisition mode.

Slave devices always power up at an initial and temporary address of 0xFE, and also assume address 0xFE following reception of a Bus Reset command.

The slave then assembles and transmits an address acquisition request—which includes the unique slave device serial number—to the master device at address 0. Although not enforced by the CCB stack, this is the only unsolicited message that a slave device is allowed to transmit.

When the master device receives a valid address acquisition request, it assembles an address assignment response packet—which contains the device serial number as received in the Address Request packet and also a unique bus address—and transmits it to the temporary address of 0xFE.

When a slave device receives an address assignment packet containing its unique serial number, it resets its local address to the new address contained in the message and then proceeds with normal bus operation. **NOTE:** The slave device does not explicitly acknowledge it received the address assignment packet.

Until a slave address has received an address assignment from the master device, it ignores all bus commands, except address assignment commands containing its unique serial number, and continue to transmit address acquisition requests using the defined CCB collision detection, back off, and retry method. If the address assignment packet isn't received in a timely manner, the slave device performs timeout and retries every two seconds until an address is acquired.

The address acquisition requests and reply packets are described under “Bus Management Address Acquisition Protocol” (p. 4-97).

4.15.3 Bus Management Ping Overview

The CCB provides no facility for detection of device disconnection events. The only way to know if a device is connected to the bus is when a message is received from it.

The master maintains a master list of devices that have been assigned addresses and are connected to the bus.

The CCB stack on the master performs a background polling loop of connected devices.

If a message has not been received from an assigned address for a period of time (default 10 seconds, configurable), the master CCB stack sends a ping message to the address and waits a period of time (default 2 seconds, configurable) for a response.

When a slave device receives a ping message from the master, it then assembles and transmits a ping response message. This lets the master know that the slave device is still there.

When a device has not responded to a ping transmission in three consecutive attempts, the address is marked as available and the client application on the master is informed that the device has been disconnected from the bus.

4.15.4 Bus Management Client Interface Overview

Client applications on the master device typically require the ability to know what devices are on the bus, and also when new devices are connected and old devices disconnected.

The Master device implements an API function that the client code may call to retrieve a list of all devices connected to the bus at the time of the call.

The Master device also queues a message for the client to read whenever new devices are connected to the bus (when addresses are assigned) or when old devices are disconnected from the bus (when addresses are released).

The Client application on the master would typically retrieve a list of all addresses at start-up and then use the queued bus management event messages to keep the list updated from that point on.

4.15.5 Bus Management Port Identification Overview

The Master Controller contains LED indicators to represent the current status of various slave device operational parameters. These LEDs are associated to specific physical ports on the master controller.

However, the CCB is a “cloud” architecture, where the location of specific devices on the bus is not fixed or easily identifiable. Without special logic it is impossible for software to know which physical port a slave device is connected to.

The Bus Management Port Identification logic is thus implemented to identify the physical port to which a slave device is attached. In this way, the LED indicators for that port properly represents the status of the correct device.

The Master Controller has direct control over a signal pin on the connector of each port (herein referred to as the Port Identification Pin, or PIP). For each port, it asserts the PIP and then broadcasts a bus management port identification request. Upon receipt of this message each slave device examines the state of its PIP. If PIP is asserted, the slave device responds with a port identification response. The master controller then knows which slave device is attached to that port. The PIP on the identified port is then de-asserted and this method is repeated for each port on the bus. This logic must be performed at power up and also each time a device is connected to the bus.

4.15.6 Master Device Bus Management Functional Flow

Figure 4-15 shows the flow diagram for the Master Device Bus Management.

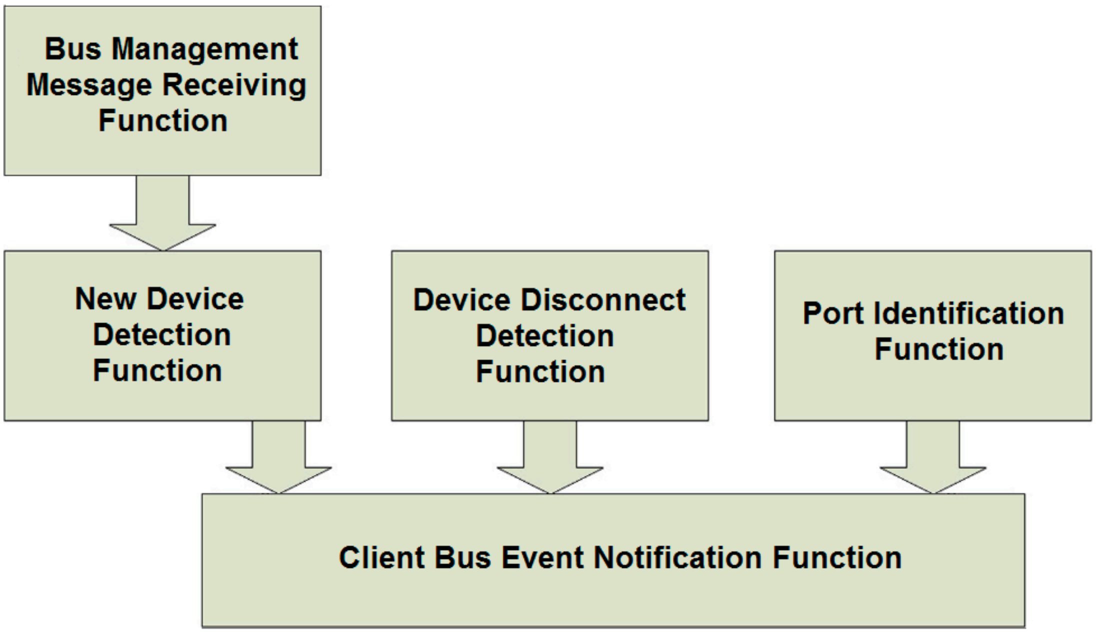


Figure 4-15. Master Device Bus Management Flow Diagram

4.15.7 Bus Management Message Receiving Function

4.15.7.1 Input Requirements

Bus Management messages are received from the Inbound Bus Management Redirect Function when the BUSMGMT bit in the message flags byte is set in a received message.

4.15.7.2 Processing Requirements

If the received bus management message is a Ping Response message from a slave device, it is discarded and no further action taken.

NOTICE

Ping Response messages have served their purpose by causing the device's last received message timestamp to be updated in the Inbound Bus Management Redirect Function.

4.15.7.3 Output Requirements

All received messages that are not slave device Ping Response messages are passed to the New Device Detection function.

4.15.8 New Device Detection Function

4.15.8.1 Input Requirements

The New Device Detection function receives bus management messages from the Bus Management Message Receiving Function.

4.15.8.2 Processing Requirements

Each received bus management message is examined to see if it is an address acquisition request message from a newly connected slave device.

If the received message is identified as an Address Acquisition Request message, the following processing steps are performed:

1. A scan of connection devices is made to remove duplicate serial numbers from list.
2. An unused address is allocated from the address pool and assigned to the device.
3. The device serial number is stored for the new address.
4. The last received message timestamp on the address is recorded.
5. An Address Assignment message is assembled as explained in the section, “Master Device Address Assignment Response” (p. 4-98) and then transmitted to the slave device.

4.15.8.3 Output Requirements

If the results of the New Device Detection Function resulted in assignment of a new device address, the event is passed to the Client Bus Event Notification Function and no further action on the message is taken.

If the message does not result in the assignment of a new device address, the message is discarded and no further action taken.

4.15.9 Device Disconnection Detection Function

4.15.9.1 Input Requirements

The Device Disconnection Detection Function is entered on a timed basis once each second.

4.15.9.2 Processing Requirements

The Device Disconnection Detection Function scans the active device list to determine the period of time that has elapsed since the last message was received from each active device.

For each active device, if the elapsed time is greater than the maximum dead time limit (default 6000 milliseconds, configurable), the device address is marked as inactive and a list of all newly inactive devices is assembled.

For each active device, if the elapsed time is less than the dead time limit, but greater than the minimum dead time limit (default 2000 milliseconds, configurable), a Master Ping Request message is assembled and transmitted to the device and no further action is taken.

4.15.9.3 Output Requirements

If any devices were newly discovered to be disconnected, the list of such devices is passed to the Client Bus Event Notification Function.

4.15.10 Port Identification Function

4.15.10.1 Input Requirements

The Port Identification Function is entered at power up and each time a slave device is connected to the master device.

4.15.10.2 Processing Requirements

For each physical port on the master device, the Port Identification Function performs a port identification sequence.

The port identification sequence operates as follows:

1. Assert PIP on one port and de-assert on all other ports.
2. Broadcast a Port Identification Request.
3. Wait for a response from a slave device for a maximum of 100 ms. If there is no response, one retry is attempted per port.

4.15.10.3 Output Requirements

On exit from this function all PIPs is de-asserted.

The master device maintains an internal “Port Association List” that associates each physical port to a specific slave device, if any. Slave devices are uniquely identified by a device serial number.

4.15.11 Client Bus Event Notification Function

4.15.11.1 Input Requirements

The Client Bus Event Notification Function receives input from two sources:

1. New Device Detection Function
2. Device Disconnection Detection Function

The New Device Detection Function supplies the address and serial number of the newly detected device.

The Device Disconnection Detection Function supplies a list of all devices that have been disconnected.

4.15.11.2 Processing Requirements

For each new device detected, the Client Bus Event Notification Function assembles a Slave Device Connection Message and queues it for the client.

For each device that has been disconnected, the Client Bus Event Notification Function assembles a Slave Device Disconnection Message and queues it for the client.

4.15.11.3 Output Requirements

This function produces no output.

4.15.12 Slave Device Bus Management Functional Flow

Figure 4-16 shows the functional flow for Slave Device Bus Management.

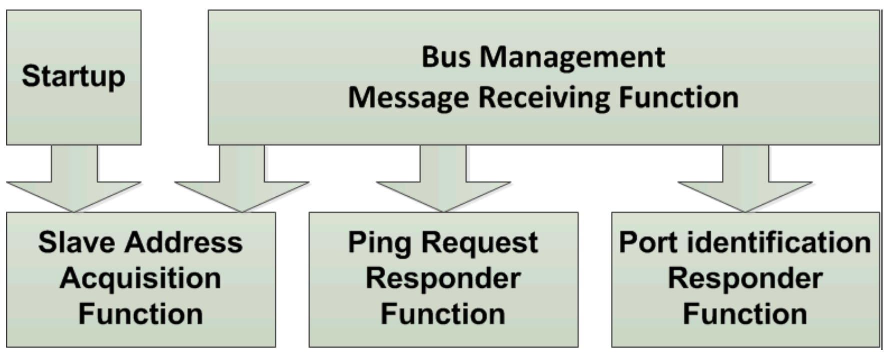


Figure 4-16. Slave Device Bus Management Functional Flow

4.15.13 Slave Address Acquisition Function

4.15.13.1 Input Requirements

This function is entered when any of the following conditions occur:

1. Initial power up of the slave device
2. Reception of a Bus Reset message from the master device
3. Timeout while waiting for an Address Assignment message from the master device
4. Arrival of an Address Assignment message from the master device

4.15.13.2 Processing Requirements

The slave CCB stack sets its local device address to 0xFE at power up, following reception of a Bus Reset command from the master device.

It then assembles and transmits an Address Request message, including the device serial number string, to the master device at address 0 and set a two second timeout.

Each time an Address Assignment message is received, the serial number on that message is compared with the local device serial number.

If an Address Assignment message is received with matching serial number, the local device address is set to the newly assigned address and the timeout canceled.

If the timeout expires before an acceptable Address Assignment message has been received, another Address Request message is sent and another two second timeout is set.

This cycle repeats until an Address Assignment message is received with matching device serial number. When an address has been successfully acquired, the Address Acquisition function is considered complete.

If an Address Assignment message with matching serial number is received subsequent to initial address assignment, the local device address is reset to the newly assigned address.

4.15.13.3 Output Requirements

The slave acquisition function is not considered complete until a unique local address has been acquired and set.

4.15.14 Slave Bus Management Message Receiving Function

4.15.14.1 Input Requirements

Bus Management messages are received from the Inbound Bus Management Redirect Function when the BUSMGMT bit in the message flags byte is set in a received message.

4.15.14.2 Processing Requirements

The message is examined and dispatched to the proper message handling function.

4.15.14.3 Output Requirements

If the received message represents an Address Assignment Message or a Bus Reset Message, the message is passed to the Slave Address Acquisition Function.

If the received message represents a Ping Request Message, the message is passed to the Slave Device Ping Responder Function.

If the received message represents a Port Identification Request Message, the message is passed to the Slave Device Port Identification Function.

4.15.15 Slave Device Ping Responder Function

4.15.15.1 Input Requirements

The Slave Device Ping Responder Function receives messages from the Inbound Bus Management Redirect Function when the BUSMGMT bit in the message flag bit is set and the message represents a master Ping Request.

4.15.15.2 Processing Requirements

The Slave Device Ping Responder function assembles and transmits a Ping Response message to the master device.

4.15.15.3 Output Requirements

The Ping Response message is transmitted to the master device.

4.15.16 Slave Device Port Identification Function

4.15.16.1 Input Requirements

The Slave Device Port Identification Function receives messages from the Inbound Bus Management Redirect Function when the BUSMGMT bit in the message flag bit is set and the message represents a Port Identification Request Message.

NOTICE
These messages are received at the broadcast address.

4.15.16.2 Processing Requirements

Upon receipt of a Port Identification Request Message, the function queries the state of the PIP pin on the CCB connector.

4.15.16.3 Output Requirements

If PIP is asserted, a Port Identification Response Message is assembled and returned to the master device.

If PIP is de-asserted, no further action is taken.

4.15.17 Bus Management Protocol Definition

4.15.18 Bus Management Address Acquisition Protocol

Figure 4-17 depicts the format for a slave Address Acquisition Request message.

Sadd 0xFE	**Dadd 0x00**	**Flag 0x01**	**Tag 0xnn**	**Len N**	**Cmd 0x00**	**D_0**	**D_1**	**.**	**.**	**.**	**D_N**

Figure 4-17. Slave Address Acquisition Request

Since the slave device only needs to acquire an address when using the temporary slave address of 0xFE, the source address is set to that value.

The destination address is always 0 for the master device.

The BUSMGMT bit of the Message Flags byte is set to indicate a bus management message.

The Length byte includes one byte for the command byte and the length of the device serial number string, including the null string terminator character.

The command byte of 0x00 indicates an Address Acquisition Request.

The slave device serial number, programmed into the device at time of manufacture, is included as data. The serial number is formatted as a null terminated text string. The serial number string and terminating null character is appended to the four-byte message header and command byte to form the complete message.

4.15.19 Master Device Address Assignment Response

Figure 4-18 depicts the format of an Address Assignment message sent from the master to a specific slave device.

Sadd 0x00	Dadd 0xFE	Flag 0x01	Tag 0xnn	Len N	Cmd 0x80	New Addr	D_0	D_1	.	.	.	D_N

Figure 4-18. Master Address Assignment Response

The source address of 0 represents the fixed address of the master device.

The destination address of 0xFE represents the temporary address of slave devices that have not yet been assigned a unique bus address.

The BUSMGMT bit of the Message Flags byte is set to indicate a bus management message.

The Len of N includes one byte for the Command byte, one byte for the newly assigned address, the length of the serial number string for the targeted slave device, and the terminating null of the string.

Appended to the four-byte message header is one command byte with value set to 0x80 representing an Address Assignment message, one byte which is the new address assigned to the slave device, the serial number string of the slave device, and the null string terminator.

When a slave receives this message, it compares the received serial number to its serial number and, if the serial number strings match, it sets its local device address to be equal to the new address sent in the received message packet.

4.15.20 Bus Management Ping Protocol

4.15.20.1 Master Ping Request

Ping requests sent by the master are formatted as shown in Figure 4-19.

Sadd 0x00	Dadd 0xN	Flag 0x01	Tag 0xnn	Len 1	Cmd 0x81

Figure 4-19. Master Ping Request

Source address of 0x00 represents the fixed master address.

Destination address is set to the unique address that is assigned to the slave device.

The BUSMGMT bit of the Message Flags byte is set to indicate a bus management message.

- The length value of 1 accommodates the single command byte.
- The command byte of 0x81 is the ping command.

4.15.20.2 Slave Ping Response

Ping responses sent by slave devices are formatted as shown in Figure 4-20.

Sadd 0xN	Dadd 0x00	Flag 0x01	Tag 0xnn	Len N	Cmd 0x01	D_0	D_1	.	.	.	D_N

Figure 4-20. Slave Ping Request

Source address of 0xN represents the unique assigned address of the slave device.

Destination address is set to the 0x00, the fixed address of the master device.

The BUSMGMT bit of the Message Flags byte is set to indicate a bus management message.

- The length value of N accommodates the single command byte and the length of the device serial number string, including null terminator.
- The command byte of 0x01 is the ping response.

The unique device serial number string—including the null string terminator—is appended to the response message.

4.15.21 Bus Management Bus Reset Message

4.15.21.1 Master Bus Reset

Bus Reset requests sent by the master are formatted as shown in Figure 4-21.

Sadd 0x00	Dadd 0xFF	Flag 0x01	Tag 0xnn	Len 1	Cmd 0x84

Figure 4-21. Master Bus Reset

Source address of 0x00 represents the fixed master address.

Destination address is set to the broadcast address of 0xFF for reception by all slave devices.

The BUSMGMT bit of the Message Flags byte is set to indicate a bus management message.

- The length value of 1 accommodates the single command byte.
- The command byte of 0x84 is the Bus Reset command.

4.15.22 Bus Management Port Identification Protocol

4.15.22.1 Master Port Identification Request

Port identification requests sent by the master are formatted as shown in Figure 4-22.

Sadd	Dadd	Flag	Tag	Len	Cmd
0x00	0xFF	0x01	0xnn	1	0x85

Figure 4-22. Master Port Identification Request

Source address of 0x00 represents the fixed master address.

Destination address is set to the broadcast address that is received by all slave devices.

The BUSMGMT bit of the Message Flags byte is set to indicate a bus management message.

- The length value of 1 accommodates the single command byte.
- The command byte of 0x85 is the Port Identification Request.

4.15.22.2 Slave Port Identification Response

When the port identification pin on the slave device is asserted, the device assembles and responds with a port identification response as shown in Figure 4-23.

Dadd	Flag	Tag	Len	Cmd	D_0	D_1	.	.	.	D_N
0x00	0x01	0xnn	N	0x02						

Figure 4-23. Slave Port Identification Response

Source address of 0xN represents the unique assigned address of the slave device.

Destination address is set to the 0x00, the fixed address of the master device.

The BUSMGMT bit of the Message Flags byte is set to indicate a bus management message.

- The length value of N accommodates the single command byte and the length of the device serial number string including null terminator.
- The command byte of 0x02 is the port identification response.

The unique device serial number string including the null string terminator is appended to the response message.

4.15.23 Bus Management Client Event Messages

4.15.23.1 Slave Device Connection Message

Each time a slave device is assigned an address the master device queues a message for its client application to indicate the event. The message is formatted as shown in Figure 4-24.

Sadd	Dadd	Flag	Tag	Len	Cmd
0xN	0x00	0x01	0xnn	1	0x82

Figure 4-24. Slave Device Connection Message

The source address is the assigned address of the newly connected slave device.

The destination address is always zero.

The BUSMGMT bit of the Message Flags byte is set to indicate a bus management message.

- The length value of 1 accommodates the single command byte.
- The command byte of 0x82 indicates that this is a connection event.

The message indicates to the client application that slave device 0xN has been newly connected to the bus and is ready for operation.

4.15.23.2 Slave Device Disconnection Message

Each time the CCB stack on the master determines that a slave device is no longer responding on the bus, master device queues a message for its client application to indicate the event. The message is formatted as shown in Figure 4-25.

Sadd	Dadd	Flag	Tag	Len	Cmd
0xN	0x00	0x01	0xnn	1	0x83

Figure 4-25. Slave Device Disconnection Message

The source address is the address of the now disconnected slave device.

The destination address is always zero.

The BUSMGMT bit of the Message Flags byte is set to indicate a bus management message.

- The length value of 1 accommodates the single command byte.
- The command byte of 0x83 indicates that this is a disconnect event.

The message indicates to the client application that slave device 0xN has been disconnected from the bus.

4.15.23.3 Slave Device "ACK" Response Message

The slave has the option of sending a general 'ACK' response when it received a message but only when there's no other specific response. The message is formatted as shown in Figure 4-26.

Sadd	Dadd	Flag	Tag	Len	Cmd
0xN	0x00	0x01	0xnn	1	0x03

Figure 4-26. Slave Device "ACK" Message

The source address is the assigned address of the slave device.

The destination address is always zero.

The BUSMGMT bit of the Message Flags byte is set to indicate a bus management message.

- The length value of 1 accommodates the single command byte.
- The command byte of 0x03 indicates that this is an acknowledgment event.

4.15.24 Bus Management Command Summary

Table 4-7 lists the commands for Bus Management.

Table 4-7. Bus Management Commands

Command Byte	Source	Function
0x00	Slave	Address Acquisition Request
0x01	Slave	Ping Response
0x02	Slave	Port Identification Response
0x03	Slave	Acknowledgment to message
0x80	Master	Address Assignment Command
0x81	Master	Ping Request
0x82	Master	Slave Connection Message
0x83	Master	Slave Disconnection Message
0x84	Master	Bus Reset Message
0x85	Master	Port Identification Request

4.15.25 Applicable Documents

- TIA-485-A Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems (ANSI/TIA/EIA-485-A-98) (R2003)
- TSB-89-A: Application Guidelines for TIA/EIA-485-A
- *Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems, Second Edition (Complete Guides series)* (Paperback) by Jan Axelson
- *RS485 Cables - Why you need 3 wires for 2 (two) wire RS485* (http://www.chipkin.com/articles/rs485-cables-why-you-need-3-wires-for-2-two-wire-rs485)
- *Guidelines for Proper Wiring of an RS-485 (TIA/EIA-485-A) Network* (http://www.maxim-ic.com/appnotes.cfm?appnote_number=763&CMP=WP-1)

INDEX

Superior Reliability & Performance

* 2 2 8 3 5 3 6 *

OBIS® LS/LX Lasers — *Operator's Manual, Part 3 of 3: Interface & Commands*

Part No. 2283536 Rev. AA