34

Using USB Through Virtual COM Ports

APPLICATION NOTE

ILX Lightwave
Laser Diode Instrumentation & Test Systems

USB 2.0 Instrument Connectivity

By: Ryan Woodham

Introduction

The Universal Serial Bus (USB) is quickly replacing GPIB as the accepted communication protocol between the PC and the instrument. While USB lowers the total cost of instrumentation ownership, upgrading legacy software from GPIB to USB can be difficult.

With this concern in mind, USB-enabled ILX Lightwave products utilize a Virtual COM Port to provide a simple programming interface.

This application note provides an overview to using Virtual COM Ports both with HyperTerminal and with various programming languages.

USB Overview

The specification for USB 1.0 was released in 1995 by a group of companies including Microsoft ®, Intel ®, IBM ® and Compaq ® [1]. The USB Implementers Forum, Inc (www.usb.org) was formed in 1995 to promote and standardize USB [2].

USB is a communications protocol which enables the user to easily connect multiple peripherals to their PC. Compared to PCI, USB allows any user to easily connect devices without opening the computer case and exposing internal circuitry. USB supports a theoretical maximum of 127 devices and data rates up to 480 Mbit/s [1] compared to GPIB's 15 devices and 63 Mb/s [5].

The USB interface allows for dissimilar devices to communicate using the same type of cables and ports. Standard USB devices such as thumb drives, keyboards, or cameras each have a standard method they use to communicate. Windows XP has a set of default drivers for each of these standard devices. One can simply plug in their new camera to the PC and retrieve pictures with ease.

The USB-enabled instruments from ILX are designed to maintain the ease of USB while providing the least amount of impact to legacy GPIB software.

Driver and Software Installation

Each USB-enabled instrument from ILX Lightwave is packaged with an installation CD-ROM. The CD-ROM contains the USB driver and the instrument control software.

The USB driver is required for the PC to recognize the instrument over USB. The control software then uses the driver to communicate with the instrument.

The actual driver and control software installation may vary from instrument to instrument. In general, the CD contains an auto-run installer which will install the driver when the CD is inserted. The control software should then be installed manually by running the setup program found within the instrument's subdirectory on the CD.

It is always recommended to install the instrument's driver before connecting the instrument to the PC. If the instrument is connected to the PC prior to installing the driver, unplug and re-plug in the instrument. In some cases, the PC might have to be rebooted before the instrument can be recognized by the PC. In these cases, unplug the device until the Windows operating system has completely rebooted then plug in the instrument.

Please refer to the instrument manual for more information on using the USB control software for your instrument.

Using Microsoft HyperTerminal

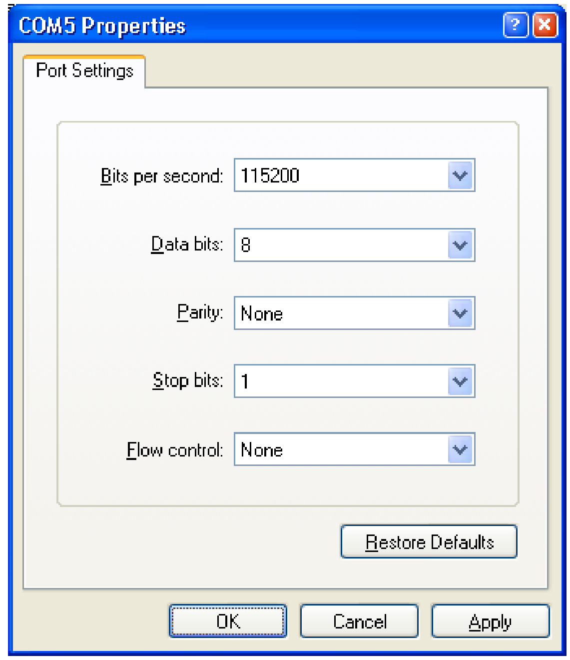
HyperTerminal is a Microsoft application packaged with Windows which allows text-based (ASCII) communications over COM ports. HyperTerminal can communicate through the Virtual COM Port driver to USB-enabled instruments.

Just like GPIB, the ILX Lightwave USB-enabled products accept ASCII commands which are very similar to SCIPI commands. Commands such as “*IDN?”, or “LAS:OUT 1” are still available using USB.

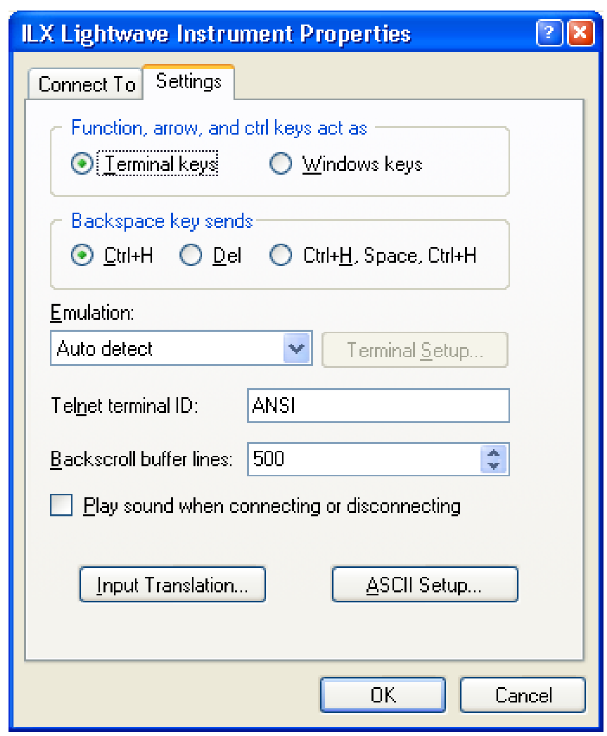
The following steps outline how to setup HyperTerminal for communicating with an ILX Lightwave USB-enabled instrument.

1) Start HyperTerminal from the Start Menu in Windows (this is usually found under *Accessories => Communications => HyperTerminal*.)
2) HyperTerminal will prompt the user to configure a new connection in the *Connection Description* dialog. Type a name, and click *OK*.
3) The *Connect To* dialog box appears. Select the COM port which is connected to the desired ILX Lightwave instrument. Please refer to the section ***Finding the VCOM Port Number*** in this document for more information.

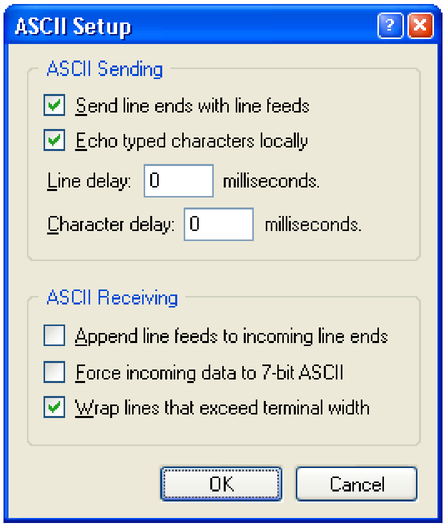
4) The *COMX Properties* dialog appears. Refer to the manual for the exact port settings as each USB enabled product will vary. Typical settings are:
   a. Baud: 115200
   b. Data Bits: 8
   c. Parity: None
   d. Stop Bits 1
   e. Flow Control: None



5) Click OK on the *COMX Properties* dialog to accept the Port Settings.
6) Select *File->Properties* in HyperTerminal.
7) The Connection Properties dialog appears. Select the *Settings* tab.



8) Click the *ASCII Setup…* button. The *ASCII Setup* dialog.
9) Check *Send line ends with line feeds*.
10) Check *Echo typed characters locally*.



11) Click OK on the *ASCII Setup* dialog.
12) Click OK on the Connection Properties dialog.
13) Type "*IDN?" without the quotation marks and hit return. The instrument will respond with an identification string.

The HyperTerminal application was originally created as a remote interface between computers. HyperTerminal will transmit to the instrument special characters such as the backspace or arrow keys. These keys are not recognized by all ILX Lightwave USB-enabled instruments and thus the unit may respond with an "E123" error. Each character is sent to the instrument as it is typed into HyperTerminal, the commands are executed when the instrument receives the new line character (when the user presses the return key).

Programming Virtual COM Ports

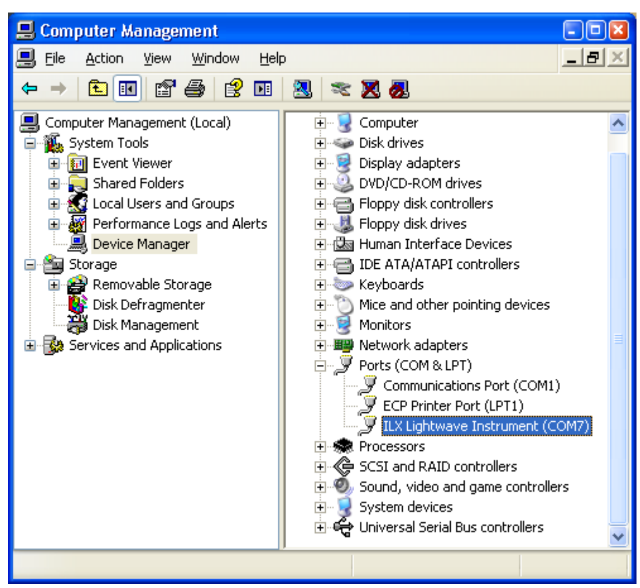
To assist in upgrading legacy software from GPIB to USB, most USB-enabled ILX Lightwave instruments are compatible with a Virtual COM Port (VCOM) driver. The VCOM driver is installed by default from the installation CD-ROM of these instruments.

A VCOM Port behaves like a standard COM port in the Windows operating system. To communicate to these instruments, the user can use any available RS-232 tools. With this leverage, the software developer does not have to learn a custom API or dig into the USB class definitions.

Finding the VCOM Port Number

Once an ILX Lightwave instrument is connected and recognized by the PC, a COM port number is assigned to the instrument. Each ILX Lightwave instrument connected to the PC will have a different COM port number. This COM port number must be determined before any program can communicate with the instrument.

The simplest method is to use Device Manager. The Device Manager, in Windows XP, can be found by right-clicking on *My Computer* and selecting *Manage*. The *Computer Management* window appears and the Device Manager can be found in the list on the left of the window. Within the *Device Manager*, select and expand the *Ports (COM & LPT)* item. If the ILX Lightwave instrument has been properly connected, it will be listed here as *ILX Lightwave Instrument (COM3)*, where COM3 is the assigned COM port number.



Some software solutions may require that the USB port number is determined dynamically. For these advanced applications, the COM port number can also be found in the Windows registry. Accessing the Windows registry is for advanced programmers only and is beyond the scope of this application note.

Command Description

Sending commands over the VCOM port is very similar to GPIB. The USB command structure is the same as GPIB command structure. Familiar commands such as **IDN?* or *LASER:LDI 10.1* are still supported. To be specific, the VCOM port expects ASCII strings terminated with a newline character (Hexadecimal 0x0A) in the same format as the IEEE 488.2 and SCPI-like standard command set.

Values queried from an ILX Lightwave USB-enabled instrument can be expected to return the data in ASCII format, terminated with the newline character.

Standard Programming APIs

Depending on the programming language, there are several methods of communication through a COM port. Programmers using the NI-VISA™ driver can modify the resource string to select between GPIB or RS-232. LabVIEW™ programmers can swap out the GPIB module for an RS-232 module. This application note focuses on a two methods which apply to programs written in either unmanaged C/C++, Visual Basic 6.0, Visual Basic.NET, managed C++, C#, or J#.

The first method describes using the Win32 API `CreateFile()` function in unmanaged C++ or Visual Basic 6.0. The second method describes the `SerialPort` class in the .NET Framework® (for the languages Visual Basic .NET, managed C++, C#, and J#).

Method 1: CreateFile( )

The `CreateFile()` function and the family of compatible support functions provide a legacy method of accessing the serial COM port. The following functions are necessary for communicating through the serial COM port:

- `CreateFile( )`
  - Opens the COM port.
- `CloseHandle( )`
  - Closes the COM port.
- `SetCommState( )`
  - Configures the COM port (baud, stop bits, parity, etc.)
- `WriteFile( )`
  - Writes bytes through the COM port.
- `ReadFile( )`
  - Reads bytes from the COM port.

These functions are compatible with Windows 98®, Windows NT®, Windows 2000®, and Windows XP®. These functions have been tested at ILX Lightwave on all four operating systems with no issues.

A full explanation of these functions is beyond the scope of this application note. However, more information can be found from MSDN™ starting here: http://msdn.microsoft.com/en-us/library/aa363858(VS.85).aspx.

Method 2: The SerialPort Class

The .NET Framework provides a number of tools for programmers. The `SerialPort` class is one such tool that simplifies serial COM port communication from the old `CreateFile`() method.

The `SerialPort` class is in the `System.IO.Ports` namespace. The class contains the expected `Open(), Close()` `Read()`, and `Write()` functions as well as a number of properties to configure the baud rate, stop bits and such.

This class was an addition to the .NET Framework as of version 2.0 [3] and should be compatible with Windows 98, Windows NT, Windows 2000, Windows XP, and Windows Vista® [4]. This method has been tested at ILX Lightwave on Windows 2000 and Windows XP with no issues.

More information on the `SerialPort` class can be found here: http://msdn.microsoft.com/en-us/library/system.io.ports.serialport.aspx.

Please keep in mind that while these software methods may be supported on Windows 98 or Windows NT, some of the ILX Lightwave USB VCOM drivers are not.

Configuring the VCOM Port
Each USB-enabled ILX Lightwave instrument might have a slightly different VCOM port configuration (baud rate, stop bits, parity, etc). The VCOM port configuration is a feature carried over from standard COM ports. Please check your manual for the exact VCOM port configuration.

Visual Studio Express Editions
For those interested in writing new software or updating legacy software, Microsoft has published free versions of the popular Visual Studio 2008 for C++, C#, and Visual Basic .NET. These free versions are categorized as "Express Editions" and offer most of the functionality found in the more costly versions. More information and the download links for the Express Editions can be found here: http://www.microsoft.com/Express/.

Conclusion

ILX Lightwave has developed USB-enabled instruments which reduce the total cost of ownership without forcing large changes to existing software packages. The switch from GPIB to USB does not have to be difficult.

References

1. Intel Corporation, *Universal Serial Bus (USB),* Retrieved March 24, 2009 from http://developer.intel.com/technology/usb/
2. International Business Machines Corporation, *Standards and Specs: The ins and outs of USB*, April 26 2005. Retrieved March 24, 2009 from http://www.ibm.com/developerworks/power/library/pa-spec7.html
3. Microsoft Corporation, *SerialPort Class (System.Io.Ports)*. Retrieved March 24, 2009 from http://msdn.microsoft.com/en-us/library/system.io.ports.serialport.aspx
4. Microsoft Corporation, *.NET Framework 2.0 Redistributable Prerequisites)*, March 2006. Retrieved March 24, 2009 from http://msdn.microsoft.com/en-us/library/aa480241.aspx
5. National Instruments, *High-Performance GPIB Interfaces for PCI Express*, Retrieved April 9, 2009 from http://www.ni.com/pdf/products/us/2004_4326_101_101_dlr.pdf

The following publications are available for download at www.ilxlightwave.com.

White Papers

- A Standard for Measuring Transient Suppression of Laser Diode Drivers
- Degree of Polarization vs. Poincaré Sphere Coverage
- Improving Splice Loss Measurement Repeatability
- Laser Diode Burn-In and Reliability Testing
- Power Supplies: Performance Factors Characterize High Power Laser Diode Drivers
- Reliability Counts for Laser Diodes
- Reducing the Cost of Test in Laser Diode Manufacturing

Technical Notes

- Attenuation Accuracy in the 7900 Fiber Optic Test System
- Automatic Wavelength Compensation of Photodiode Power
- Measurements Using the OMM-6810B Optical Multimeter
- Bandwidth of OMM-6810B Optical Multimeter Analog Output
- Broadband Noise Measurements for Laser Diode Current Sources
- Clamping Limit of a LDX-3525 Precision Current Source
- Control Capability of the LDC-3916371 Fine Temperature Resolution Module
- Current Draw of the LDC-3926 16-Channel High Power Laser Diode Controller
- Determining the Polarization Dependent Response of the FPM-8210 Power Meter
- Four-Wire TEC Voltage Measurement with the LDT-5900 Series Temperature Controllers
- Guide to Selecting a Bias-T Laser Diode Mount
- High Power Linearity of the OMM-6810B and OMH-6780/6790/6795B Detector Heads
- Large-Signal Frequency Response of the 3916338 Current Source Module
- Laser Wavelength Measuring Using a Colored Glass Filter
- Long-Term Output Drift of a LDX-3620 Ultra Low-Noise Laser Diode Current Source
- Long-Term Output Stability of a LDX-3525 Precision Current Source
- Long-Term Stability of an MPS-8033/55 ASE Source
- LRS-9424 Heat Sink Temperature Stability When Chamber Door Opens
- Measurement of 4-Wire Voltage Sense on an LDC-3916 Laser Diode Controller
- Measuring the Power and Wavelength of Pulsed Sources Using the OMM-6810B Optical Multimeter
- Measuring the Sensitivity of the OMH-6709B Optical Measurement Head
- Measuring the Wavelength of Noisy Sources Using the OMM-6810B Optical Multimeter
- Output Current Accuracy of a LDX-3525 Precision Current Source
- Pin Assignment for CC-305 and CC-505 Cables
- Power and Wavelength Stability of the 79800 DFB Source Module
- Power and Wavelength Stability of the MPS-8000 Series Fiber Optic Sources
- Repeatability of Wavelength and Power Measurements Using the OMM-6810B Optical Multimeter
- Stability of the OMM-6810B Optical Multimeter and OMH-6727B InGaAs Power/Wavehead
- Switching Transient of the 79800D Optical Source Shutter
- Temperature Controlled Mini-DIL Mount
- Temperature Stability Using the LDT-5948
- Thermal Performance of an LDM-4616 Laser Diode Mount
- Triboelectric Effects in High Precision Temperature Measurements
- Tuning the LDP-3840 for Optimum Pulse Response
- Typical Long-Term Temperature Stability of a LDT-5412 Low-Cost TEC
- Typical Long-Term Temperature Stability of a LDT-5525 TEC
- Typical Output Drift of a LDX-3412 Low-Cost Precision Current Source
- Typical Output Noise of a LDX-3412 Precision Current Source
- Typical Output Stability of the LDC-3724B
- Typical Output Stability of a LDX-3100 Board-Level Current Source
- Typical Pulse Overshoot of the LDP-3840/03 Precision Pulse Current Source
- Typical Temperature Stability of a LDT-5412 Low-Cost Temperature Controller
- Using Three-Wire RTDs with the LDT-5900 Series Temperature Controllers
- Voltage Drop Across High Current Laser Interconnect Cable
- Voltage Drop Across High Current TEC Interconnect Cable
- Voltage Limit Protection of an LDC-3916 Laser Diode Controller
- Wavelength Accuracy of the 79800 DFB Source Module

Application Notes

- App Note 1: Controlling Temperatures of Diode Lasers and Detectors Thermoelectrically
- App Note 2: Selecting and Using Thermistors for Temperature Control
- App Note 3: Protecting Your Laser Diode
- App Note 4: Thermistor Calibration and the Steinhart-Hart Equation
- App Note 5: An Overview of Laser Diode Characteristics
- App Note 6: Choosing the Right Laser Diode Mount for Your Application
- App Note 8: Mode Hopping in Semiconductor Lasers
- App Note 10: Optimize Testing for Threshold Calculation Repeatability
- App Note 11: Pulsing a Laser Diode
- App Note 12: The Differences between Threshold Current Calculation Methods
- App Note 13: Testing Bond Quality by Measuring Thermal Resistance of Laser Diodes
- App Note 14: Optimizing TEC Drive Current
- App Note 17: AD590 and LM335 Sensor Calibration
- App Note 18: Basic Test Methods for Passive Fiber Optic Components
- App Note 20: PID Control Loops in Thermoelectric Temperature Controllers
- App Note 21: High Performance Temperature Control in Laser Diode Test Applications
- App Note 22: Modulating Laser Diodes
- App Note 23: Laser Diode Reliability and Burn-In Testing
- App Note 25: Novel Power Meter Design Minimizes Fiber Power Measurement Inaccuracies
- App Note 26: ReliaTest L/I Threshold Calculations
- App Note 27: Intensity Noise Performance of Semiconductor Lasers
- App Note 28: Characterization of High Power Laser Diode Bars
- App Note 29: Accelerated Aging Test of 1310 nm Laser Diodes
- App Note 30: Measuring High Power Laser Diode Junction Temperature and Package Thermal Impedance
- App Note 31: Mounting Considerations for High Power Laser Diodes
- App Note 32: Using a Power / Wavehead for Emitter Level Screening of High Power Laser Diode Bars
- App Note 33: Estimating Laser Diode Lifetimes and Activation Energy

For application assistance or additional information on our products or services you can contact us at:

ILX Lightwave Corporation
PO Box 6310, Bozeman, MT 59771-6310
Phone: 406-556-2481 • 800-459-9459 • Fax: 406-586-9405
Email: sales@ilxlightwave.com

To obtain contact information for our international distributors and product repair centers or for fast access to product information, technical support, LabVIEW® drivers, and our comprehensive library of technical and application information, visit our website at:

www.ilxlightwave.com

ILX Lightwave
Laser Diode Instrumentation & Test Systems

 Rev01.060909