Supports Polymorphic VIs

LV-Link

Software Interface for LabVIEW™

LV-Link Features

- ***Access Data Translation hardware in LabVIEW***

— *Create test and measurement applications for Data Translation hardware using Virtual Instruments (VIs) in LabVIEW.*

- ***Stream data at full speed***

— *Use VIs to measure and control analog I/O, digital I/O, counter/timer, and quadrature decoder signals on any DT-Open Layers for .NET compliant device, and stream data at full-speed.*

- ***Integrate all your hardware***

— *Use Data Translation and National Instruments hardware in the same application.*

- ***Get up and running quickly***

— *Use the standard LabVIEW interface and the example programs to get up and running quickly.*

- ***Modeled after National Instruments DAQmx™***

— *Uses polymorphic VIs for ease of use and to speed development.*

- ***Built on the .NET architecture***

— *Easy access to the DT-Open Layers for .NET Class Library for maximum flexibility.*

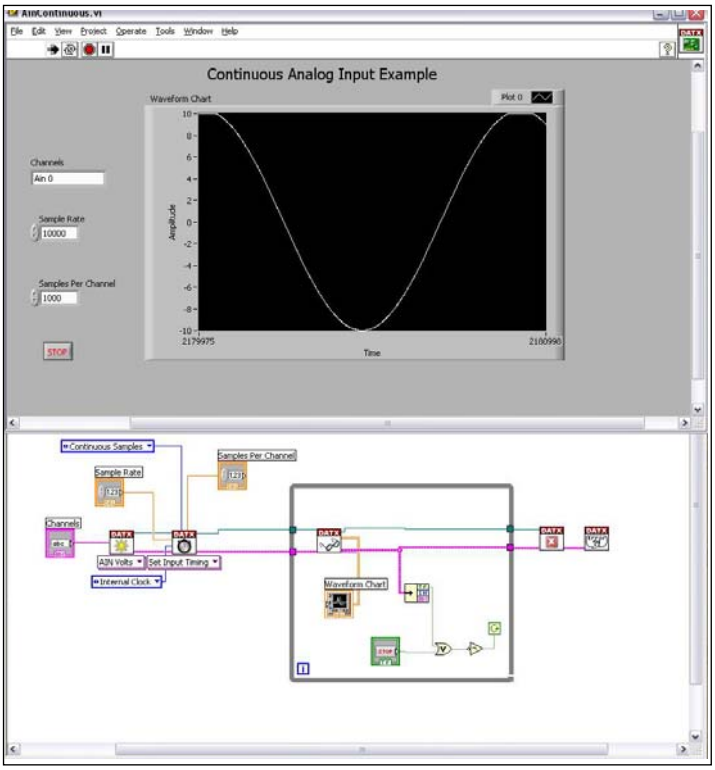

Figure 1. Use LabVIEW and LV-Link (shown above) to access the power of Data Translation DT-Open Layers for .NET compliant data acquisition devices.

Overview

LV-Link provides a collection of Virtual Instruments (VIs) that allow LabVIEW programmers to create powerful test and measurement applications using Data Translation .NET compatible boards/modules.

LV-Link supports all DT-Open Layers for .NET compliant USB and PCI hardware, providing the ability to measure and control analog I/O, digital I/O, quadrature decoders, and counter/timer signals, and stream data at full-speed.

To get up and running quickly, numerous application examples are provided with the software. Since the source code is also provided, you can easily modify the examples to speed your development time.

By using LV-Link in your LabVIEW application, you can integrate all your Data Translation and National Instruments hardware in the same application. And, by using LabVIEW's client/server functionality, you can even monitor your hardware remotely over the internet!

Summary of Features

		Types of Operations
Analog Input	✓	Single value, finite, continuous
Analog Output	✓	Single value, finite, continuous, waveform
Digital I/O	✓	Single-value read, single-value write
Counter/Timer	✓	Event counting, up/down counting, frequency measurement, period measurement, edge-to-edge measurement, continuous edge-to-edge measurement, one-shot, repetitive one-shot, rate generation
Quadrature Decoder	✓	Accepts A, B, and Index signals from a quadrature encoder

LV-Links VIs

LV-Link provides VIs for developing data acquisition tasks that work with DT-Open Layers for .NET compliant USB and PCI devices.

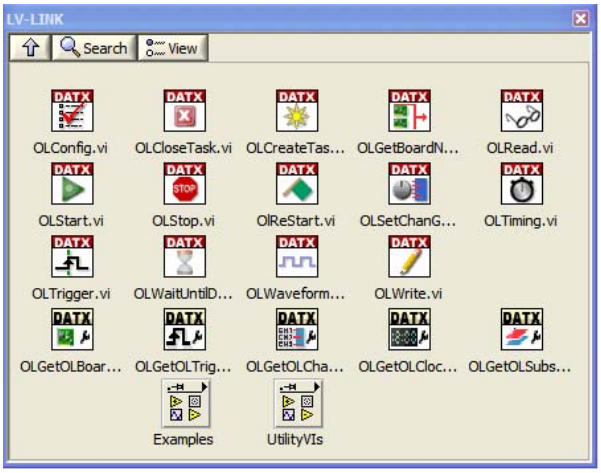

Figure 2. LV-Link provides core VIs and VIs that access the DT-Open Layers for .NET Class Library to perform data acquisition tasks on Data Translation hardware using LabVIEW.

Core VIs	
OLCloseTask	Stops the task and releases all associated resources
OLConfig	Configures the specified task
OLCreateTask	Creates virtual channels and configures the tasks with specified settings
OLGetBoardNames	Returns names of all Data Translation boards installed
OLRead	Reads samples from tasks you specify
OLReStart	Restarts a task
OLSetChanGain	Sets the gain for an analog channel
OLStart	Explicitly starts a task
OLStop	Stops a task and returns it to original state
OLTiming	Specifies the timing of analog input or output task
OLTrigger	Specifies trigger-related attributes of analog input or output tasks
OLWaitUntilDone	Waits for a specified task to be done before stopping
OLWaveformGenerate	Creates one cycle of a sine, triangle, square, or sawtooth wave
OLWrite	Outputs samples to a virtual channel

VIs that Access the DT-Open Layers Class Library	
OLGetOLBoard	Returns a .NET reference to the underlying Device class
OLGetOLChanList	Returns a .NET reference to the underlying Channel List class
OLGetOLClock	Returns a .NET reference to the underlying Clock class
OLGetOLSubsystem	Returns a .NET reference to the underlying Subsystem class
OLGetOLTrigger	Returns a .NET reference to the underlying Trigger class

Utility VI Examples	
InitDACBuffers	Initializes the values in the buffers used by analog output tasks
OLErrorParser	Converts .NET error codes into LV-Link error codes

Polymorphic VIs

LV-Link supports all DT-Open Layers for .NET compliant USB and PCI devices and many sensor and signal types, while providing a simple application programming interface (API) for developing powerful applications. To minimize the complexity of the API, LV-Link uses polymorphic VIs. Polymorphic VIs consolidate the functionality of multiple VIs into one VI that takes on different characteristics depending on its input values. The combination of VIs in one simple interface results in an easier learning curve since all operations on all devices use the same API. For example, rather than learning different ways to program different types of operations (analog input, analog output, digital I/O, counter/timer, and quadrature decoder), developers need only learn one way and then can reuse that knowledge in future applications.

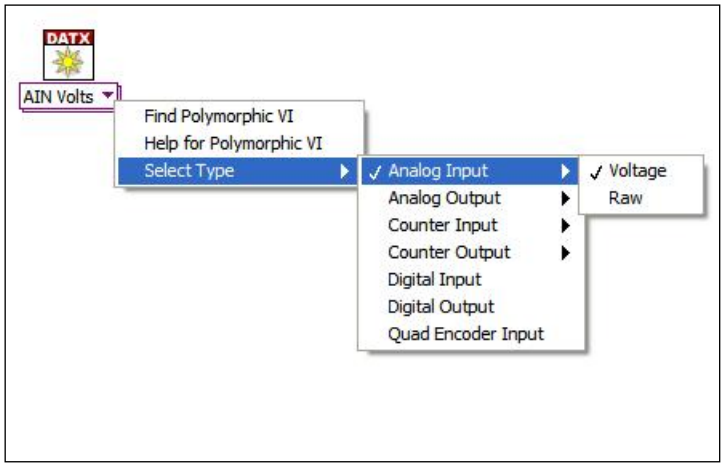

Figure 3. In this example, the polymorphic VI OLCreateTask is used to create a task for one or more analog input, analog output, counter/timer, or quadrature decoder channels.

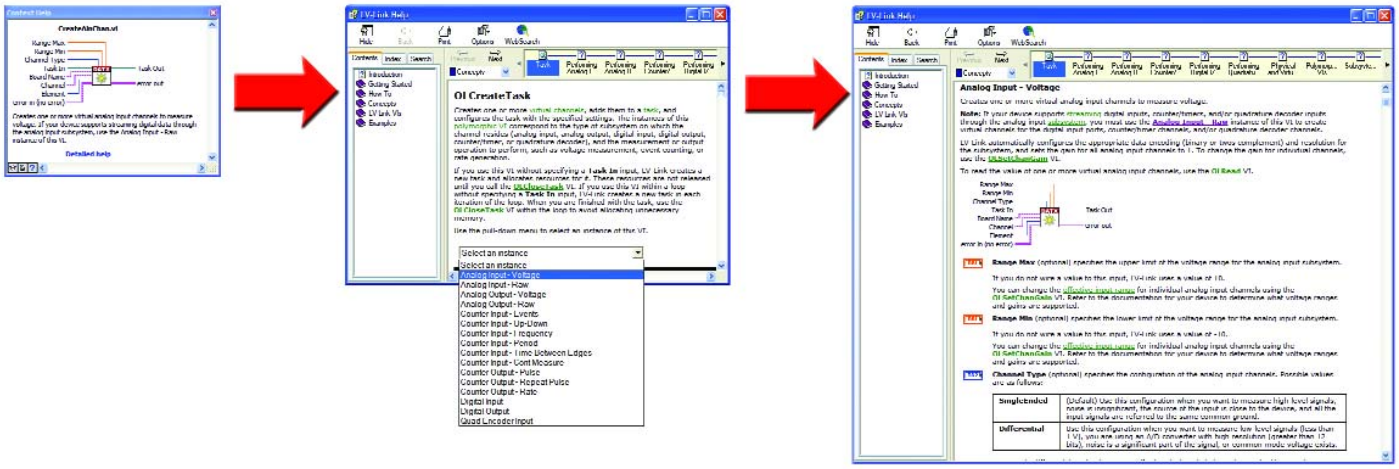

Figure 4. Extensive on-line help is available in a context-sensitive format to assist you in creating your program.

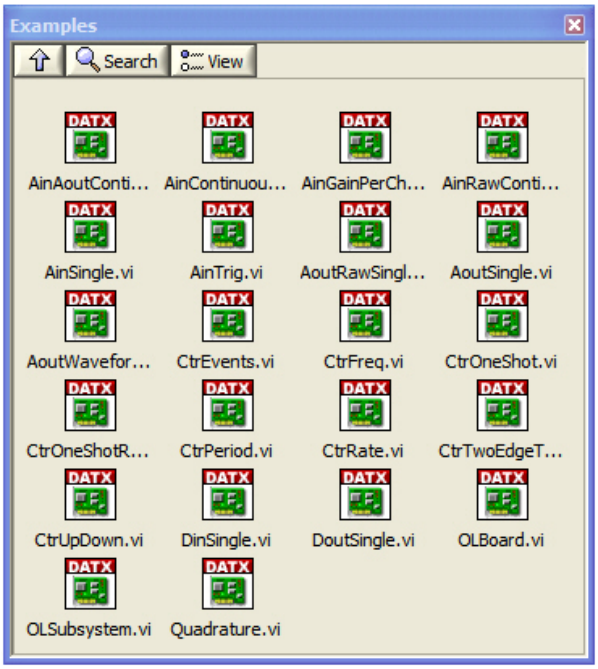

Figure 5. LV-Link provides a number of example programs that allow you to get up and running quickly.

Application Examples

LV-Link provides numerous example programs that allow you to get up and running quickly. Since the source code is included, you can easily modify these examples as you wish to speed your development time.

In addition to providing VIs that conform to the familiar LabVIEW interface, LV-Link gives users the utmost flexibility by providing VIs that access the DT-Open Layers for .NET Class Library. This approach has the advantage of providing standard LabVIEW programming while providing access to all DT-Open Layers for .NET functions.

Convert Existing Programs

Replace existing LabVIEW hardware VIs with LV-Link hardware VIs quickly and easily. Simply replace the LabVIEW VI with the corresponding LV-Link VI and all wiring and other necessary connections are made automatically.

Application Examples	
AinAoutContinuous	Demonstrates how to continuously acquire and output analog data
AinContinuous	Demonstrates how to continuously acquire data from an input channel
AinGainPerChan	Demonstrates how to set up different gains and acquire data from multiple input channels
AinRawContinuous	Demonstrates how to continuously acquire raw counts from analog input channels
AinSingle	Demonstrates how to acquire a single value from an analog input channel repetitively in a loop
ATrig	Demonstrates how to continuously acquire data when an analog trigger threshold is detected
AoutRawSingle	Demonstrates how to write a single raw count to an analog output channel repetitively in a loop
AoutSingle	Demonstrates how to write a single voltage value to an analog channel repetitively in a loop
AoutWaveform	Demonstrates how to output a waveform from an analog channel repetitively in a loop
CtrEvents	Demonstrates how to count digital pulses from an external signal
CtrFreq	Demonstrates how to measure the frequency of an external digital signal
CtrOneShot	Demonstrates how to generate a single output pulse for a specified duration
CtrOneShotRepeat	Demonstrates how to output a single pulse for a specified duration
CtrPeriod	Demonstrates how to measure the period of an external digital signal
CtrRate	Demonstrates how to generate an output pulse at a specified frequency when the C/T is enabled
CtrTwoEdgeTiming	Demonstrates how to measure the time between a specified start and stop edge
CtrUpDown	Demonstrates how to count digital pulses from an external signal that is connected to the C/T input source. If the gate input signal of the C/T is high, the counter is incremented. If the gate input signal of the C/T is low, the count is decremented
DinSingle	Demonstrates how to read a single value from a digital input port repetitively in a loop
DoutSingle	Demonstrates how to write a single value from a digital output port repetitively in a loop.
OLBoard	Demonstrates how to access methods and properties of the underlying DT-Open Layers for .NET Class Library
OLSubsystem	Demonstrates how to access properties of the underlying DT-Open Layers for .NET Class Library
Quadrature	Demonstrates how to read the value of a quadrature decode, repetitively in a loop

Technical Support

As you develop your application, technical support is available when you need it. Extensive information is available 24 hours a day on our web site at www.datatranslation.com, including drivers, examples, a searchable knowledgebase, and much more.

Support is also available from your point of purchase. You can also request complimentary support via email or fax at any time.

Ordering Summary

System Requirements

- Microsoft Windows 2000 (Service Pack 4) or XP (Service Pack 1 or higher; Service Pack 2 recommended)
- LabVIEW 8.0 or higher.
- Appropriate Data Translation hardware and device drivers

Supported Hardware for LV-Link

- Data Translation PCI boards or USB modules compliant with DT-Open Layers for .NET Class Library

A demo version is shipped on the Omni CD and can be downloaded from the website.

For pricing information, see the current price list, visit our web site, or contact your local reseller.

- SP0811-CD
  LV-Link, single license.