Sample LabVIEW™ 5.1 VIs for Vernier LabPro®

Contents:

Overview

We created these Virtual Instrument programs (VI's) with the intention that instructors and students with some LabVIEW experience may develop VI's specific to their classroom needs. They can be used to experiment with LabVIEW data acquistion and control using our relatively inexpensive hardware. We encourage exploration, discovery, and modification to suit your needs.

One of the great things about using LabVIEW for this is that you can try out some of the powerful features built into LabVIEW. Examples include:

- Advanced data analysis features like low-pass and band-pass filters, Fourier transforms
- Web publishing
- Sharing data using data sockets
- True cross platform functionality, including Linux

These VI's were created using the LabVIEW 5.1 Full Development System for Windows. We have used them using many versions of LabVIEW 5, 6i, and 7 for Windows, Macintosh, and Linux. including the 6i Student Edition.

These VIs assume that you have LabPro connected to a computer via the serial port. You can also use them with a USB LabPro connection if you are using Macintosh OS 9 and if you have the proper USB extension installed. This extension is LabProUSB, which is installed with Logger Pro 2.2. On Windows and Linux, using LabPro via USB requires special code not supported by the 5.1 VIs. Some of the other sample VIs on our web site do support Windows via USB connection.

Using the VI's

All of the main VI's included in this collection may be used to collect data with no modifications. When you run one of the main VIs it will check for a connection to a LabPro, determine the proper com port and initialize the LabPro. The easiest way to use a VI is to open it, select the correct settings regarding number of probes, data rates, etc., and then start it with the LabVIEW run button. Try collecting data, experimenting, and modifying the VIs.

These VIs were kept very simple to make them easy to understand. The down side of this is that they do not do much in terms of data display or analysis. The RT Analog 2 Channels VI is the only one that even has a graph. You may want to add analysis features such as waveform charts, or waveform graphs, or even complex data analysis features like power spectrum displays.

There are fourteen main VIs in this collection. They can be grouped into eight types based on the type of data collection or control they are designed for.

• **RT Analog**

"RTAnalog.vi" - collect data from 1 to 4 AutoID analog sensors
"RTAnalog 2 channels.vi" - same with a graph of data (limited to 2 channels)
"RTAnalog(DIN).vi" - collect data from 1 to 4 selected non-AutoID sensors
"RTAnalog(EventsWithEntry).vi" - collect events with entry for 1 AutoID sensor

These VIs collect data from up to four analog probes in real time. If you select a sampling rate that is above LabPro's limit, the maximum attainable rate will be used instead. There are four sample variations for RT data collection.

Data in these VIs are collected one point at a time in a data collection loop. Send the data to a waveform chart or do other analysis features inside this loop.

• **NRT Analog**

"NRTAnalog.vi" - collect data from 1 to 4 AutoID analog sensors
"NRTAnalog(DIN).vi" - collect from 1 to 4 selected non-AutoID sensors

NRT Analog collection is used to collect data at speeds unattainable by real time collection. There are two sample variations for NRT data collection.

One advantage of NRT data collection is that the data is automatically returned in a format that is easy to transfer to a graphing or spreadsheet program. NRT Analog has the further advantage of being insertable into other VI's. This is useful when performing data analysis in LabVIEW.

• **NRT Fast Analog**

"NRTFastAnalog.vi" - collect data from 1 AutoID analog sensor

Fast mode data collection is a more specific form of NRT Analog collection. In fast mode you are limited to one analog channel, but you can collect data at rates up to 50,000 points per second. Everything else about this VI is the same as the "NRTAnalog.vi".

• **PhotoGateDriver**

"PhotoGateDriver.vi" - collect data from 1 or 2 photogates

The photogate driver VI is one of the most complicated VI's included with this set. Because of the command structure, there is no easy way to collect photogate data for a specific period of time. The photogate VI is a lot like the real time analog VI in that the execution stays inside a loop within the VI until data collection is complete. This loop constantly queries LabPro as to how many events have taken place. If two events happen between iterations of this loop, it is possible for more data points to be collected than had originally been requested.

Stop condition settings:

1- Stops when gate one has reached the requested number of points.
2- Stops when gate two has reached the requested number of points.
3- Stops when both gates have reached the requested number of points.
4- Stops when either gate has reached the requested number of points.

The photogate VI is often used with an analysis VI. The VI has four array outputs:

the number of seconds from start until the time when DIG 1 became unblocked
the number of seconds from start until the time when DIG 1 became blocked
the number of seconds from start until the time when DIG 2 became unblocked
the number of seconds from start until the time when DIG 2 became blocked

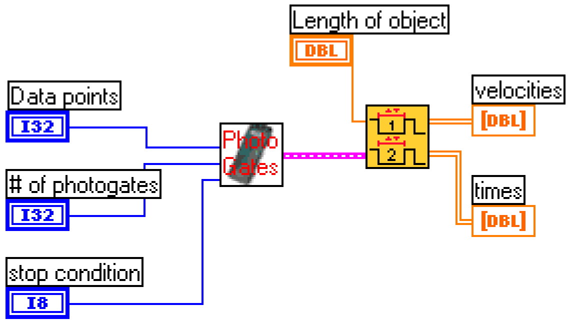
These times can be used or manipulated to handle almost any photogate timing. VI's to emulate the most popular photogate modes are in a subfolder entitled "Photogate Timing" in the "SubVIs" folder. They are: Collision Timing, Flash Timing, Gate and Pulse Timing, Gate Timing- 1 Gate, Gate Timing- 2 Gates, Motion Timing, and Pendulum Timing.

One way to use the PhotoGateDriver VI is to follow the procedure described below, starting with a new, blank VI.

1. Insert the PhotoGateDriver VI into the block diagram of the new VI.
2. Create controls for all of the inputs to the PhotoGateDriver VI, such as the number of data points, number of photogates, and stop condition.
3. Insert one of the VIs from the Photogate Timing folder onto the block diagram.
4. Wire the data output of the PhotogateDriver VI to the data input of the photogate timing VIs. In some cases, you also need to connect a control or constant to another input.
5. Create indicators for the outputs of the photogate timing VI.

As an example, if you choose to collect data with a photogate, start with the "PhotoGateDriver.vi", create controls for its inputs, insert the "Gate Timing- 2 Gates.vi" analysis function, wire its input to PhotoGate's output, and create indicators for 2 Gates' outputs.

The final product could look something like this:



Motion VI connected to the "Gate Timing-2 Gates" Timing VI

• **Motion**

"Motion.vi" - collect data from an ultrasonic Motion Detector connected to DIG/Sonic1
This VI works with a Motion Detector to collect distance data on to the object (meters), its velocity (m/s), and its acceleration (m/s/s).

• **Rotary and Analog**

"Rotary&Analog.vi" - collect data from a Rotary Motion Detector on DIG/Sonic1

The Rotary and Analog VI is like the NRT Analog VI with one key exception: while you are collecting from the four analog channels, you can simultaneously collect data from a Rotary Motion Sensor. The maximum data rates are slightly lower than those of the NRT Analog VI due to the increased workload of the LabPro. The data from the Rotary Motion Sensor is returned as position number, positive for clockwise and negative for counterclockwise. Printed on the Rotary Motion Sensor is the number of increments per rotation. Using that number and the sign of the returned data, you can calculate the net rotations of the wheel and the direction in which it was turned.

• **Digital Output and the Digital Control Unit (DCU)**

"DCU Patterns.vi" – controls a DCU on DIG/Sonic1 through a series of patterns
"DCU Warning.vi" - controls a DCU on DIG/Sonic1 depending on a analog input

There are two sample VIs to demonstrate control of the digital output lines of LabPro. They can be used for any project involving digital output, but most people will probably use them with the Vernier Digital Control Unit (DCU). The VI named DCU Patterns simply steps through a series of patterns to flash the LEDs on the DCU or control a stepper motor. The VI named DCU Warning monitors an analog input line and turns on an output line when a limit is exceeded.

• **Analog Output**

"Analog Out Triangle Wave.vi" – controls the analog out line of CH4
"Analog Out Control.vi" – allows the user to control the analog output from CH4
There are two sample VIs to demonstrate control of the analog output line of LabPro. You can connect to this line by simply plugging the standard LabPro voltage lead into the CH4 socket. This line can then be controlled by your program in the range of -4 to +4 volts (100 mA max).

The VI named Analog Out Control is a simple control for the analog output voltage. The VI named Analog Out Triangle Wave sends a triangle wave voltage pattern to the analog output. You can control the amplitude and frequency of this wave. Note: The analog output lines on your LabPro will function properly only if you have firmware (operating system) newer than 6.23 in your LabPro. Firmware upgrades are available for free at www.vernier.com. Go to www.vernier.com/calc/flash.html.

Creating Your Own VIs
The most useful tool for writing or modifying VIs that communicate directly with LabPro is the *Vernier LabPro™ Technical Reference Manual.* This manual can be downloaded from the Vernier web site (http://www.vernier.com/). It lists all of the commands that you can send to LabPro.

Non-Auto ID Sensors
It is possible to use "DIN" sensors designed for the ULI or Serial Box with LabPro and these VIs. Since the older sensors don't store their names, calibrations, or units, they are "non-AutoID" and you must input this information manually. The VIs with "(DIN)" appended to the name will not AutoID the sensors. You must select the sensor connected to each channel from one of the four pull-down menus. If you are creating your own VIs for use with not AutoID sensors, you will need to use the VI "NonBTADataConvert.vi." Put this VI in the middle of your data stream and then create a control out of the top. The default control for this VI has a list of all the compatible DIN sensors. The order of the sensors in the list is important, so make changes carefully.

Limitations

LabPro's maximum data rates are as follows:

Channels	Real Time	Non-Real Time	Fast Mode
1	385	10,000	50,000
2	300	5,000	—
3	250	3,300	—
4	200	2,500	—

Note: LabPro only stores 12,000 data points internally. LabPro beeps when asked to collect more than 12,000 points in any non-real time data (NRT) collection mode. This means that data collection at high rates is limited to very short time periods.

When large numbers of non-real time points are taken, there will be a considerable delay between the end of data collection and when the data is displayed on the screen. The VI must wait 3 milliseconds for each data point, so 10,000 data points will take 30 seconds to be retrieved. The time delay with digital data is even greater than with analog, but it is less apparent because more than a few dozen digital points are rarely taken.

Notes

- Most problems with an unresponsive LabPro can be fixed by unplugging the power to LabPro and then plugging it back in. Make sure to wait for the start-up beeps before trying to collect data. On firmware versions newer than 1.12, you can instead just hold down the Start/Stop button for 5 seconds.

- RT Analog(DIN) assumes that you are using non-autoID sensors with DIN connectors. You have to specify the sensor connected so that the voltage to correct reading conversion can be done in the program. If you use these programs with AutoID sensors and leave the type of probe setting on volts, they will read the correct numerical value.
- NRT Analog(DIN) assumes that you are using non-autoID sensors with DIN connectors. You have to specify the sensor connected so that the voltage to correct reading conversion can be done in the program. You cannot use the AutoID feature of any sensor with this program. It will always read raw voltage.
- On programs with graphs, you can right click on the graph to get options on graph scaling, zooming
- If you are using a Macintosh OS 9 with USB connection to LabPro, here are some special things to watch out for:

    -Connect your LabPro to the computer's USB connection and power up the LabPro before starting LabVIEW. If you do not, LabVIEW may not allow the USB drivers to load properly. If you lose the connection to USB as you are using LabVIEW with these programs, quit LabVIEW, disconnect the LabPro from the USB connector, and then restart LabVIEW and the program.

    - These programs may lose their USB connection if the computer powers during a period of inactivity. Adjust your computer's settings so that the computer does not power down (go into sleep mode) easily.

    - There is one known problem with using these NRT programs with Macintosh and USB connections. When using these NRT programs, the data is returned using hexadecimal data transfer because it is faster than ASCII data transfer. Unfortunately, sometimes hexadecimal data transfer when using Mac USB will fail and report incorrect values. This problem usually only shows up if you collect thousands of data points in one run. The more advanced VIs on our web site, avoid this problem.

These sample VIs were developed in a team effort at Vernier Software & Technology. All of the following people contributed: Adam Gibbs, David Gardner, Jim McBride, Sam Swartley, and Dave Vernier. We would also like to thank Ravi Marawar and Sara Seba of National Instruments for their assistance on this project.

Rev Aug 1, 2003