PHYC 500: Introduction to LabView

M.P. Hasselbeck, University of New Mexico

Exercise 11 (v 1.0) Reading from a DAQ device

A laboratory software platform would be of very limited value if it couldn't readily communicate with instruments. In this exercise, simulated data will be retrieved from a NI DAQ (Data Acquisition) device.

MAX

LabView installs with a separate program called MAX: Measurement and Automation Explorer. Its purpose is to configure a wide range of National Instruments peripheral interfaces and devices. If you are using hardware from another vendor, you will likely have to consult the NI website or contact the manufacturer directly for information about how or even *if* it will work with LabView.

Launch NI MAX from the Start Menu or desktop. Open “My System”. Right-click on “Devices and Interfaces” and “Create New...”. Select “Simulated NI DAQmx Device or Modular Instrument”. Click Finish. You will be presented with a list of devices for which drivers have been installed on your computer. Expand the “M Series DAQ” section and select “PCI-6220”.

The PCI-6220 (below, left) is a multi-function DAQ board with 16-bit resolution, a 250 kSample/second acquisition rate, and 16 input channels (cost ~ $900). You would need a cable and some type of collector block (right) to connect it to an experiment. This hardware is not installed on your computer, so its analog read operation will be simulated.

Expand “Devices and Interfaces” and verify that a simulated PCI-6220 is present. Select it and perform a Self-Test. If the device is recognized, open the “Test Panels...” tab. Select channel **ai0** and set the Mode to **Finite**. Press the Start button and observe a noisy (~ 3% amplitude fluctuations) sinusoidal waveform on the display chart. The next step is to retrieve a similar signal from the device using LabView. Click Close and close NI-MAX.

DAQmx

DAQmx is driver software that provides a universal interface for performing analog and digital I/O and counter functions on a wide variety of National Instruments hardware. LabView is only one method for interfacing with DAQmx. DAQmx does not come bundled with LabView, but is a free download from the NI website.

Open a blank VI and right-click on the Block Diagram. Locate the Measurement I/O palette: NI-DAQmx – Data Acquisition. Use the thumbtack in the upper left corner of the window to hold it open. It is also recommended that the Context Help (CTRL-H) of LabView be enabled to assist in wiring.

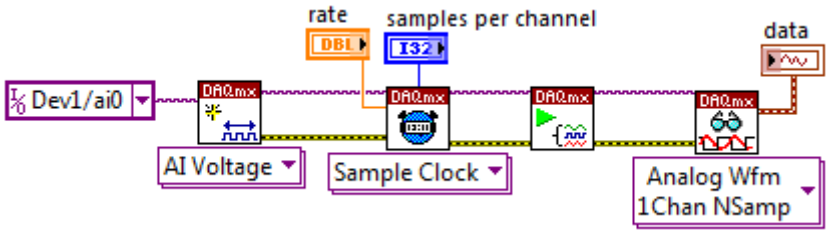
Select the Create Virtual Channel VI. On the menu immediately below the icon, select AI Voltage (i.e. analog input voltage). Find the **physical channels** input connection on the left side of the icon; right-click and create a constant. Clicking on the blank constant should display a list of 16 available analog input lines on the PCI-6220 simulated device. Select channel **ai0**. This creates a DAQmx task interface constant for the first input line.

The timing parameters for sampling the device must be set, i.e. the sampling rate and number of samples. Do this with the DAQmx Timing.vi on the open palette. Place it immediately to the right of the Create Channel VI. You may notice that LabView will try to automatically connect the **task out** and **task/channels in** terminals as well as the error terminals of adjacent VIs. If this doesn't happen, wire them together manually: Connect the **task out** terminal of the Create Channel.vi to the **task/channels in** terminal of the Timing.vi. In addition, wire the output and input error terminals. Right-click on the **rate** and **samples per channel** inputs and create Front Panel controls.

LabView must be instructed to start the DAQmx interface you have just configured. To do this, go to the Data Acquisition palette and select the Start Task VI. Place it to the right of the Timing VI and connect the task in/out and error in/out terminals as above.

The next step is to read and display the data extracted from the channel. Select and place the DAQmx Read.vi to the right of the Start Task VI and connect the task and error terminals. Configure it for Analog: Single Channel: Multiple Samples: Waveform. Right-

click on the data terminal and create an indicator. This will display a waveform cluster on the Front Panel. Replace the cluster with a Waveform Graph. The Block Diagram should look similar to this:



The DAQmx interface should be cleared after it executes, which releases computer resources. This is accomplished with the Clear Task VI on the Data Acquisition palette. Place it next in the sequence, i.e. to the right of the Read VI and connect the task and error lines. When communicating with external hardware as done here, it is good programming practice to use the Error Handling line to assist in troubleshooting problems. If an error occurs, diagnostic information can be supplied on a user interface. Connect the error line output on the Clear Task VI to Dialog & User Interface: Simple Error Handler.

Go to Waveform: Analog Wfm: Measurements: Extract Tone and place it near the data output cluster of the Read VI. Wire the **data** output terminal to the **time signal in** of the Extract Tone VI. Create a Front Panel indicator for the detected frequency (in units of Hz).

Run the VI and observe that the detected frequency is altered by changing the sampling parameters. Demonstrate operation for the instructor.

DAQ assistant

In the DAQmx palette you will find a utility called DAQ Assistant (DAQ Asst). It can simplify the setup of DAQ hardware. To use it, place DAQ Asst in the Block Diagram, double-click the icon, and follow the instructions. When finished, you will have configured an Express VI that performs essentially the same functions as in this exercise. This can be a time-saver, but the code will not be as transparent as when implemented with the individual DAQmx functions as done here. This can make troubleshooting more difficult. In addition, there can sometimes be compatibility issues when the DAQ Assistant attempts to configure using different generations of DAQmx and LabView.