PHYS 5061 Lab 4: Data Acquisition in LabVIEW: A/D and D/A operations.

Tools developed in the previous lab may be handy here.

Chapters 5 and 6 in Essick introduce generation and measurement of signals using LabVIEW. Unlike the days of old, experimentalists can now routinely avoid worrying about the low-level aspects of communicating with data acquisition devices in many situations. (Although the rise of popular microcontrollers or microcomputers like the Arduino family or the Raspberry Pi now often allows some lower-level tinkering.) Commercially available data acquisition (DAQ) hardware typically comes with software packages or drivers that may be accessed from high level programming languages, including LabVIEW, and offers great flexibility in making measurements in a variety of circumstances.

The data acquisition (DAQ) hardware we will use is the NI USB-6211, which offers up to 16 channels of 16-bit resolution analog-to-digital conversion (ADC), 2 channels of 16-bit analog output (DAC – digital-to-analog conversion), and a small number of TTL logic level lines for digital input and output (DIO). All this is contained in a small box that is external to the PC and communicates via USB. LabVIEW makes use of driver software (NI-DAQmx drivers) to talk to the hardware. The drivers handle the work of converting your desires into configuring the hardware and transmitting or retrieving data between your program or VI and the DAQ device. The USB-6211 features hardware timing and buffers for both input and output. For example, it is possible to create a VI that loads the USB-6211 with data to produce a single cycle of a sinewave for output and arrange for the device to automatically regenerate the sinewave to produce a continuous sinewave output without further intervention by the PC or the user. Your VI can go on and make measurements or do calculations, just as if you had a stand-alone benchtop function generator to create the sinewave.

Chapter 5 provides a quick introduction to hooking up, configuring and testing the DAQ device. Chapter 6 makes use of some high-level VI's (the DAQ Assistant VI and Express VI's) that sacrifice some freedom for ease of configuring DAQ tasks, which will provide a gentle introduction to using the DAQ hardware. (Chapter 13 describes how to use the more basic DAQmx VI's as building blocks to have greater control over the individual choices needed in setting up a DAQ task and lays out more clearly what choices a user must make in setting up measurements.)

(1) Preliminaries - Chapter 5: connecting the DAQ device:
Make sure the DAQ device is NOT connected to the PC. From the Windows Desktop or under `All Programs – National Instruments …' from the Windows menu open up the NI Measurement and Automation Explorer (MAX or NI MAX). Then in MAX look in Devices and Interfaces. You may see one or more devices listed there. You may leave any GPIB devices, but if you see any USB-6211 devices listed, delete them. *Then* plug in your device to a USB port. It should be recognized, drivers load, and then it appears in the list of devices. If not, refresh the list of devices. Note how it is identified (Dev1

probably). Once it's installed and available in MAX, you can select it and run a self-test in MAX. Hereafter, that DAQ device should remain with that PC. Swapping them between PCs may cause a second DAQ device to be created and a Dev2 entry, which can lead to confusion and frustration in the not-so-distant future.

A short specifications sheet is available and an enlarged diagram with the pinouts for the device is provided. Work quickly through chapter 5. You may skip sec. 5.5 on aliasing, since we explore related issues in another lab.

(2) Fire up LabVIEW. Work through Essick chapter 6.

For analog input operations with the DAQ Assistant you encounter, use the analog inputs (ai0 …) in NRSE mode (Non-Referenced Single-Ended mode), <u>not</u> the differential mode mentioned in Essick. This appears in the DAQ Assistant as the `Terminal Configuration.' NRSE gives you 16 input channels, not that you'll need them all, but it means the wiring is a little simpler; all the channels use a common ground connection as the low side of their input, as you are likely to do when building a circuit on the breadboard. (Differential mode is preferred as a way of minimizing noise, and for small signals in a noisy environment, it's essential.) To operate in NRSE mode, connect a 10k resistor between the analog-input sense (AI SENSE) and analog-in ground (AI GND) terminals. Find a small screwdriver that fits the terminal blocks and keep it handy for making connections. You can get +5V to measure from the fancy breadboard. You also need to connect the breadboard ground to the USB-6211 AI SENSE in NRSE mode.

A note on the DIY exercise on digital I/O (DIO) before the problems: Our USB-6811 has only four general purpose digital output lines and four digital input lines. For the alternating LEDs portion, put a delay in loop to make the alternating visible to the user.

You may skip the "Use It!" section,

(3) Do Problem 6.4. in Essick. In addition, have the VI display an estimate of the "-3db" point of your filter at the end of the measurement. This is the frequency for which the "gain" = 0.707 = $2^{-1/2}$. (The gain expressed in dB (decibels) comes from: G(dB) = 20 log $[V_{out}/V_{in}]$.) Save the VI so it can be tested with filters that contain different components. Put some instructions for the user on the front panel for it, and include commentary so someone can make sense of your block diagram – a cleanly wired diagram makes data flow and the VI's operation easier to follow.