Computer Interfacing Segment Handout
AST 351 Spring 2004

1. Background

One of the critical steps in any instrumentation project is to get the information out of the instrument and into a form where you can use it. Most often, this means getting a digital or analog signal out of a piece of apparatus and displaying it graphically or numerically using a computer.

In addition to their data gathering function, computers can also play a valuable role in instrument control and data conditioning. Computers can move things around provide complex analog and digital reference signals, and substitute for many pieces of stand-alone electronics like filters, servos, and timers.

2. Goals

At the end of this segment, you should know...

(1) Something about how a computer bus handles the flow of information between the CPU, the memory, and various peripheral devices.

(2) How to program in LabView well enough to work with data gathering and instrument control peripherals.

(3) How to use a multifunction input/output board to do analog to digital conversions, digital to analog conversions, and digital input/output.

(4) How to determine the necessary sampling rate, input gain, and other setup parameters for A/D conversions.

(5) How to understand issues of dynamic range, frequency domain resolution, S/N optimization, and digitization noise in A/D and D/A applications.

(6) How to use the multifunction board to sychronize the computer with external devices.

(7) How to use motion control peripherals to command motors.

3. Reading Assignments

Background Reading: It is now possible to treat much of the topic of interfacing the real world to your computer as a black-box problem, much as you ignore the inner workings of your computer at other times. If you know a little bit about how the hardware works, however, you will understand better what people are talking about and may have a better chance at diagnosing problems when they arise. Read pp. 612-640 and Chapter 10 of Horowitz and Hill for this. (Note, this reading may be only of cultural benefit to those of you who don't get heavily into this topic).

Learning Labview: You can do this any way you want, but it is best to get some familiarity before the group starts on the exercises. The on-line tutorial will get you started but may not teach you everything you need to know. What I found best was the book, **LabView for Everyone (2nd ed.)** by J. Travis. It is probably most efficient to simply work through the book from the beginning while sitting in front of the computer. Others have found the manual that came with the software to be fine. The manual also has the advantage that there are copies available in the lab. Please do not take the books out of the lab. If you know LabView already, please take on the job of tutoring your group-mates.

Note: Any UT student can download Labview for their computer from http://www.engr.utexas.edu/ni/.

4. Tools

All four computers are loaded with LabView and have a multifunction input/output board. Each multifunction board comes equipped with a terminal block which gives you access to the interface pins on the board. Two of the computers have motion control boards which hook to the two motion control boxes. Other available equipment which might be useful for this segment include an LED array, oscilloscope (information about how oscilloscopes work can be found in Appendix A of *The Art of Electronics*, pp. 1045-1049), signal generator, digital multimeter, and DC power supply. See one of us if you need help in getting started with one of these devices.

5. Problem Set

This one doesn't have a problem set. Toward the end of the time for this segment, you will each need to demo the working of a few of the exercises for me.

6. Programming Philosophy

The whole idea of LabView is that you don't have to reinvent the wheel every time you want to do something. You build new programs or new applications out of existing programs or subprograms (VI's). In doing the assignment, you are welcome to use any ready-made VI's within LabView and to tell others in your group where to find them. Ask the previous group about ones they may have found. If you find particularly nice ones, please send mail to me at dtj@astro so that we can add notes about them to the course documentation. Your goal in this segment is to learn as much as possible about programming for instrument control. With that in mind, you may occasionally find that building your own VI's is more beneficial. Try to strike a balance between getting through parts of the assignment and getting the most out of each part.

7. Laboratory Assignment

Once you have read through whatever material you need to become an adequate LabView user, you need to go through and exercise the basic functions involved in instrument interfacing and control.

Your **group** should produce a report explaining how you dealt with each of the exercises below and what happened. I don't want a book, just a brief account of each exercise. Make sure that everyone in your group understands what you are doing and that everyone can demonstrate the effects. As **individuals**, you will need to demonstrate and explain the results of the different tasks and to have a working ability to build simple LabView programs. Arrange to meet with me if you have questions about the assignments or need explanations of any of the underlying principles. If any part of the assignment proves to be beyond your collective wisdom, check with me about an alternative. Don't be shy and don't wait until the last minute.

1.**Motion control:** Locate the VI's to control the nuDrive box. Get help from me in hooking up a motor to the motor drive box. Exercise the motor. Your goal here is to find out about the motor itself, the motor controller, and their interaction. How many steps are there in a full circle? How fast can the motor go? How accurately does it make repeated motions to and from a given starting point? Does the operating system allow you to do other work while the motor is running? (think up more questions on your own).

2. **Digital I/O and timing thereof:** Write a VI to produce a binary counter. Set up your eight-bit light emitting diode array so it flashes the binary equivalent of the number of seconds since you pressed the **START** button on your control panel. It should count so that 1 count = 1 sec. Start by using the "Wait Until Next ms Multiple" function. How accurate is it over two minutes? Are there other ways to get your clock to keep better time? If you use the canned VI for this exercise, you should also make sure you know how to send a word to the board so that you can light up any arbitrary number on the diode array.

Note: When doing this excercise, please be aware that the oscilloscopes you are using are also digital devices. Try to gauge how much of an effect this has by testing the scope using the analog signal generators rather than the D/A board as the input. If you are interested in a head-to-head comparison, we have an old analog scope available in the our lab upstairs.

3. **Digital to analog conversion:** In this exercise, you should notice non-ideal behavior

due to quantization both in the voltage and in the time of the samples. Write or find a VI to generate a sine wave. Start with a large output at a low frequency (±5 V, 20 Hz). Look at this output with the oscilloscope. Now, without changing the range setting used to determine the output gain, gradually lower the peak-to-peak voltage. At what level do you begin to see the quantum steps imposed by the digital input? What is the size of the steps and does this agree with your expectations for a 12-bit D/A operating over the range you are operating over? Now set the output voltage amplitude to some reasonable level (say, ±1 V) and start cranking up the frequency. How fast can you go? What does your top-speed waveform look like? If it differs from an ideal sinusoid, explain why. What is limiting your speed and the accurate reproduction of a sine wave? Repeat this last exercise with a square wave. How do the two results compare and what does this tell you about whether the limitations are in the analog or the digital part of the electronics?

Note: When doing this exercise, be aware that the oscilloscopes you are using are also digital devices. Try to gauge how much of an effect this has by testing the scope using the analog signal generators rather than the D/A board as the input. If you are interested in comparing digital and analog oscilloscopes, we have an old analog scope available in our lab upstairs.

4. **Analog to digital conversion:** Set up the breakout panel so that you are using the output of the DAC as the input of one of the ADC channels. Write a VI that interleaves sending voltages to the DAC and reading those voltages with an ADC. How fast can you make this VI go? By forcing the board to read one number and write one number at a time, you are making it interact with the bus to service every transfer. This technique is called "polled IO" It is the slowest form of interaction with the board.

Note: There are several different schemes for input and you must make sure that your input wiring is appropriate for the scheme you have told the software to use. The NI Daq manual and the board manual both talk about this. Briefly, you can choose between (1) "differential" input, in which the board amplifies the difference between the voltages on channels 0 and 8, 1 and 9 etc. (2) " referenced single ended input", in which the board measures the voltage difference between your input line (0..15) and an internal ground. In this mode, it is advisable to make sure that the device generating your voltage is referenced to ground independently. (3) 'non-referenced single ended input", in which the board measures the difference between your input line (0..15) and a ground you provide through the AISENSE line. Note that you need to choose the input mode when setting up the board.

5. **Learn about sampling and aliasing:** Set up the function generator to produce sine waves of roughly 1 volt amplitude. Use the generator as the input to the ADC using the differential input mode. Fix the sample rate at 10^4 samples/second. Start with a 100 Hz sine wave and plot the output. Can you display this result in the frequency domain? Now increase the signal frequency without altering the sample rate. At what point does your frequency domain display give the incorrect answer? What does this tell you about the necessary relation between the

frequency response of your analog electronics and the sampling rate of your ADC?

6. **Digital manipulation:** Input a square wave to your ADC. Devise a simple algorithm for determining the frequency and displaying it. How might one go about filtering out the higher harmonics present in this waveform while preserving the full amplitude of the underlying sine wave? Test your algorithm against a sawtooth as well.

7.**Instrument control:** Input a high frequency sine wave to the oscilloscope (either from the signal generator or from the DAC board). Set the scope to external trigger. Use the digital I/O on your computer to trigger the scope. Set up several modes: single sweep, continuous sweep, interval sweep (one sweep every t seconds), and test out the modes. Set up a switch to a 5V supply as input to the digital I/O. **Use a 1000 Ohm resistor in series with this supply to prevent accidental burnout of the board**. Use the state of this switch to initiate and stop activities in your computer.

8. Deliverables

Individual: Demo one of the 7 exercises above (each one in the group chooses a different one) for the prof or TA.

Group: Lab report about all 7 exercises