Lab 9: LabVIEW and GPIB

Physics 327

Spring 2025

Goals

Gain experience with computer-based control, automation, and data acquisition systems, and work with the widely-used LabVIEW platform and GPIB device interfaces. **Note:** See the end of this manual for important information about what to include in your lab report.

Introduction

LabVIEW

LabVIEW (National Instruments, Inc.) is a software package that allows the programming of instrument control and data acquisition by a computer. The programming is done by creating "virtual instruments" (VIs) using a graphical programming interface, where operations are connected with virtual wires. VIs can be combined into larger VIs, similar to building a conventional program out of functions or subroutines. A VI is made up of a "front panel" and "block diagram." The front panel is used to interact with the VI, for example to enter values controlling parameters, and display indicators or graphs showing the outputs. The block diagram is the "code" that specifies the steps in the VI. Within the block diagram, one can use math or logic operations, loop and conditional structures, read and write files, and interact with other VIs or instruments.

GPIB

General Purpose Interface Bus (GPIB), also called IEEE-488, is a communication protocol used between computers and instruments. GPIB hardware boards (made by National Instruments) are installed in the lab PCs. The GPIB boards may be programmed into LabVIEW VIs. A Hewlett-Packard digital multimeter (model HP34401A, referred to as HP) will be controlled via instructions through GPIB board. The Stanford Research Systems spectrum analyzer used in Physics 326 can also be controlled via GPIB.

A LabVIEW VIs

First you will inspect an existing VI, before creating your own.

1. Verify that the GPIB connectors are securely fastened, then turn on the PC and the HP. Launch LabView from the Desktop icon or Start menu.

2. Open the virtual instrument (VI) `Frequency Response.vi`. Click "Open Existing" and locate the file at `C:\Users\student\Documents\National Instruments\apps\freqresp.lib`.

   When you open the `Frequency Response.vi` file in LabVIEW, the window that opens up is the "Front Panel" of the VI. Open the Wiring Diagram (Block Diagram, the "code" of the VI) by clicking Window/Show Diagram. "Tools" and "Controls" pop up menus are used with Front Panel. "Tools" and "Functions" pop up menus are used with Block Diagram. (If you want to follow an online tutorial of basic operations, you may do so by opening the tutorial folder.)

   This VI measures the frequency response of the gain of an unknown "black box." A virtual function generator supplies a sinusoidal input to the black box, and a (virtual) digital multimeter measures the output voltage.

3. Run the VI by clicking on the arrow in the upper left corner of the Front Panel window, and play with the available controls in the Front Panel. **Question 1:** What kind of filter does the black box contain?

4. Open and examine the Block Diagram. Turn on “context help” in the Help menu to see descriptions of elements in the diagram. A double left click shows the front panel of an element; this is sometimes helpful. Note that there is no explicit element for the black box, and its function is hidden in the details of the “Simulate UUT” element.

5. Close the VI. **Do not save any changes please.**

B GPIB Multimeter

You will now see how to control the HP with the PC via GPIB communication, writing a new VI in the process. There is a library of useful VIs related to the HP 34401A multimeter located at `C:\Program Files\National Instruments\LabVIEW 2022\instr.lib\Agilent 34401`. Each instrument connected to the resident GPIB board is assigned an address (a string constant). The address of the HP is displayed immediately after its power is turned on. Note the address. If the address happens to be 21 (which is used by the PC), then it must be changed.

The process of sending information or data to an instrument is called “writing” data to the instrument. Writing is used for example to change the sensitivity of the HP. Retrieving data from the instrument to the PC is called “reading.” Functions are already provided by LabVIEW for writing and reading.

Part 1 Writing

First, you will write to the device to change the configuration.

1. Open a new VI (use “Create Project” then choose the “Blank VI” template) and go to its Block Diagram. Open the Functions Palette and Tools Palette from the View menu. Change the tool to the arrow cursor. In the Functions pop up menu, move cursor to “Instruments I/O” icon box. Select the “GPIB” icon and then “GPIB Write” (which shows an arrow going from PC to instrument indicating that data flows from PC to instrument). Place the icon into the Block Diagram.

2. From Help menu, choose Show Context Help. A pop up window appears. Then go back to the Block Diagram and place the cursor on the GPIB Write icon. The pop up help window now shows the wire connections that can be made. The “address string” and “data” terminals must be connected for the GPIB Write to operate. (Other terminals will be dealt with later.)

3. In the Functions pop up menu, under the “Programming” section select “String,” then “String Constant.” Copy the “String Constant” icon to the left side of the “GPIB Write.” Change the tool to finger cursor, click inside the “String Constant” box and type in the address of the HP. With the wiring tool, connect the string constant box to the “address string” input of the GPIB Write icon. If the wire is successfully made, it shows a solid magenta color line. If not, a dotted line (broken wire) appears; delete it and try it again.

4. Go back to the Functions menu and get another “String Constant.” Wire it to the “data” terminal of the GPIB Write. The text written into the “data” terminal will be transmitted (“written”) to the HP. Write “`conf:res`” (don’t type the quotes) into the “data” string constant. Then click the run button. Verify that the HP changes to resistance measurement mode. At this point, see the HP manual (on Canvas) to see how the configurations can be changed. Then configure the HP for DC voltage measurement. Verify the change of configuration on the HP display panel. **Question 2:** What command did you use to set the HP to DC voltage mode?

Part 2 Reading

Next, you will read data from HP. You first have to write to get the data ready, then you will read what the HP puts on the data lines.

1. To measure resistance, enter the following text into the “data” of the “GPIB write:”

```
conf:res
trig:sour imm
read?
```

The first line configures the HP for resistance measurement. The second line chooses the trigger mode for immediate measurement. The third line requests the HP to put the resistance reading on the GPIB bus. See pages 73, 114, and 119 of HP manual for details.

2. From the Functions menu, select the “Instrumentation I/O” section and open the “GPIB” folder and get a “GPIB Read” icon (which shows the flow of data from the instrument to the PC) and place the icon to the right of the “GPIB Write” icon. The Help pop-up window again identifies the wire connections. Connect the string constant box containing the address string to the “address” input of the GPIB Read icon.

   You must connect an integer number to the “byte count” input; this sets the number of bytes of data to be read from the instrument. Choose a “Numeric” input from the Function menu and select a Numeric Constant, and set it to 20.

   To display what is read from the instrument, go to the Front Panel. Open the Controls Palette via the View menu, and under the “Modern” section select “String & Path,” then “String Indicator.” Move the “String Indicator” icon into the Front Panel. Type in a label for it such as “Data Read from HP.” Now go to the Diagram. A box with the label you just made should be visible in the Diagram. Move the box and the label to the right side of the “GPIB Read” icon. Change the pointer to wiring tool and connect the indicator to the “data string” of the Read icon. Finally connect the “error out” terminal of the Write icon to the “error in” of the Read icon. This guarantees that the write process (sending commands to HP) occurs before the read.

   Now connect a thermistor to the resistance terminals of the HP (“Input” and the terminal below it, using banana plugs) and click Run. The LabVIEW Front Panel indicator should show the resistance value. **Question 3:** What is the resistance value?

3. You will now repeat the measurement N times using a *for loop*. From the Functions menu, under the “Programming” section select “Structures,” then “For Loop.” Move the For Loop into the Block Diagram. (N is called count terminal and I the iteration terminal. The variable I counts from 1 to N.) With the arrow pointer, increase the frame size of the For Loop icon.

   Now move your entire previous program into the For Loop. First select all your icons and wires (click the mouse in the upper left, drag to the bottom right to make a rectangle around your icons, then release mouse button), then move the selected items with the mouse.

   Go to the Front Panel. From the “Controls” menu, under the “Modern” section select “Numeric” and choose “Numeric Indicator.” Place the digital indicator in the Front Panel and type “Iteration Number” into the label box. Return to the Diagram. The indicator box should be visible in the Diagram. Move it inside the For Loop if it is not already. Connect a wire from the iteration terminal to the indicator. From the Functions menu, go to “Numeric” and choose “Numeric Constant.” Move the “numeric constant” icon to the block diagram and place it outside the For Loop. Type in a number around 10. Wire it to the “Count terminal” N. Go back to the Functions menu, under the “Programming” section go to “Timing” and choose “Wait Till Next ms Multiple.” Place it within the For Loop frame. From the Functions menu get a “numeric constant” as done before. Place it on the left side of the “Wait Till ...”. Type the number of ms to wait between iterations (make it a few hundred ms).

   Go back to Front Panel and click run. The indicator should change at the time interval set. Increase or decrease the time interval in the block diagram to verify its effect.

4. Next you will create an array of numbers and to store the array as a Matlab or spreadsheet file. From Functions menu, open “Array” and choose “Build Array.” Locate it outside and to the right of the For Loop. When you first create the array, it has only one row (indicated by a short bar) of elements. Additional rows can be added by pulling on a corner of the icon with the cursor. Increase the size of the array such that it contains two rows. Connect a wire from the iteration number I of the For Loop to the first (upper) row element of the array icon.

Now convert the string output of the GPIB read command to a number, using the command "Fract/Exp string to number". This command can be found in the "Programming" section by selecting "String" then "Number/String Conversion". Display this number on the Front Panel using a digital indicator, and connect it to the second (bottom) row of the array icon.

In order to write your data (iteration number and resistance, bundled in an array) to a spreadsheet file, go to the Functions menu. Under the "Programming" section, open the "File I/O" folder and choose "Write to Delimited Spreadsheet File.vi" and move it to the right of the "build array" icon. Connect the output array of the "build array" to the "2D data" input terminal of the "Write to Spreadsheet File" icon. You have to specify the file name for the spreadsheet and where to store it. Go back to "File I/O" and under the "File Constant" folder choose "Path Constant" and place it above the "Write to Spreadsheet File" icon. Write into the file path into the Path Constant: `C:\Users\student\Documents\filename.txt` (Matlab) or `filename.xls` (spreadsheet), where you replace "filename" with your own unique filename for the spreadsheet. Wire the path name to the "file path" input terminal of the "Write to Spreadsheet" icon. Also connect a logical true (found in "Function/Boolean") to the "transpose?" input in order to save columns instead of rows.

Now go back to Front Panel and run the program. If all is well, a file should have been saved in the folder you specified. Open the file with Matlab or Excel to see that it contains two columns, consisting of the iteration number and the resistance (in Ohms), respectively.

5. Using your For Loop structure, make a VI which takes a total of N (to be specified via a control on the Front Panel) measurements of resistance. The time to wait between measurements should also be specified on the Front Panel. Save the VI. Save screenshots of the VI (panel and diagram) and include them in your report.

6. Let the thermistor equilibrate to room temperature. Use your VI to measure how the resistance changes as function of time after you begin to hold the thermistor in you hand. Store this as a data file.

   (a) Use the VI to measure how the resistance returns to its room temperature value after you release it. Save this as another data file.

   (b) Plot the data files you obtained for resistance versus time using Matlab, Excel, Origin, or other plotting software. The plot should include the steady states before and after the warming and cooling. Include the graph(s) in your report.

7. Modify your VI such that the resistance is displayed on a graph in the Front Panel. This is found under the "Modern" section in the "Graph" folder. Hint: look at one of the VI's you inspected earlier. Save your VI for later modification and use.

C AD590 Temperature Sensor

The AD590 Temperature Sensor is a semiconductor device that outputs a current that is proportional to temperature in degrees K; the proportionality is 1 μA per K, see the data sheets available on Canvas. At a temperature of 300 K, the current is 300 μA. The current is independent of bias voltage in the range 4 to 30 V.

For this final part, you will embed the AD590 in an op amp circuit, read the circuit's output voltage with the HP and use your VI to plot temperature vs time; you will turn on a warning light in the front panel of your VI when the temperature is too high.

1. Op amp circuit: Connect the AD590 between +12V and the inverting input to an LM741 op amp. Ground the non-inverting input. Use a 10 kΩ resistor for negative feedback from the LM741 output to the inverting input. Don't forget the power connections to the LM741. There is no need for offset adjustment in this application. **Question 4:** What is the output voltage at a temperature of 300 K? Sketch a schematic in the report.

2. Connect the output voltage and ground of the LM741 circuit to the HP. Modify your VI from step B.2 to read a *DC voltage* from the HP (see step B.1.4).

3. Modify your VI so that the data displayed in at least one indicator and the graph on the front panel is temperature in degrees K. This can be done with the "Multiply" element and "Numeric Constant". Try your VI.

4. Modify your VI so that a "Square LED" on the front panel lights up if the temperature goes over 300 K while the VI is running. This indicator is in the "Classic Boolean" menu. The comparison of current data to 300 K can be made with the "Greater Than" element (in the "Programming" section in the "Comparison" folder) and "Numeric Constant."

5. Print your VI, including the graph showing a warm-up and cool-down of the AD590, and include this in your report.

Lab Report

When reporting a VI that you have created, give a brief description of its function and method, as well as the printout of the front panel and block diagram.

Your lab report should include the usual sections, and specifically:

1. A brief description of the LabVIEW application program. Do not exceed one page, but include brief discussions of its differences, advantages and disadvantages compared to the conventional code-based programming languages.

2. Printout of the Front Panel of the `Frequency Analysis of a Filter Design.vi`, after finding parameters that solve the design problem.

3. The printouts of the Front Panel and Block Diagram, and description, of the VI you made to measure the resistance of thermistor as function of time. The icons and wires should be neatly arranged for easy inspection.

4. The formula to convert thermistor resistance to temperature using your crude calibration.

5. The temperature vs. time graphs for both warming and cooling.

6. A discussion of how well the time dependence of the temperature can be approximated by an exponential.

7. The printout of the Front Panel and Block Diagram, and description, of the VI you made to display the resistance of the thermistor as a function to time in a graph in the front panel. The graph should show an actual plot of a measurement.

8. The printout and description of the VI for the AD590 temperature sensor in Step C.

9. An explanation and schematic of the op amp circuit with the AD590.