

Lab 3: Introduction to LabVIEW and GPIB Interfacing

Objective:

This lab provides the student with a basic introduction to the LabVIEW graphical programming interface and explores interfacing the computer to laboratory equipment via GPIB.

Reading Assignments:

- “Getting Started with LabVIEW”

PreLab Questions:

“Decode” the digital logic that is part of the .vi you are to write in lab this week. As shown in the Context Help windows in Figure 1, the “Multiply”, “Divide”, and “Decrement” functions are used. The output of the last “Multiply” operation feeds a DC voltage value to the “Voltage Level” input to the “AGE364X Example Single DC Volt” VI.

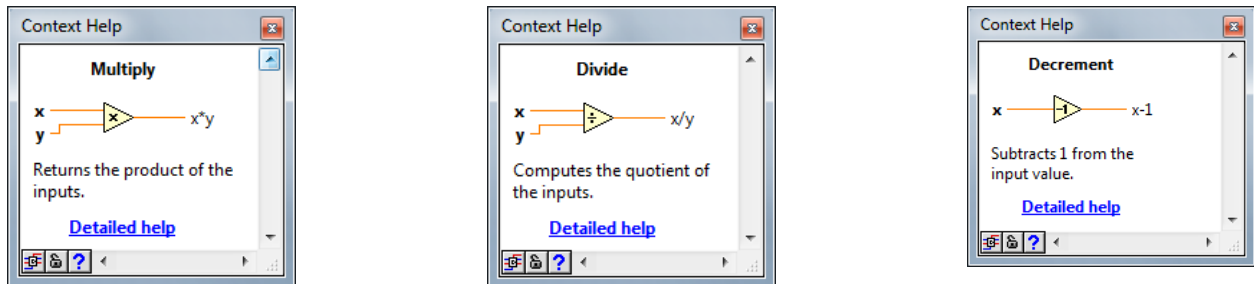


Figure 1. Context Help for Multiply, Divide, and Decrement

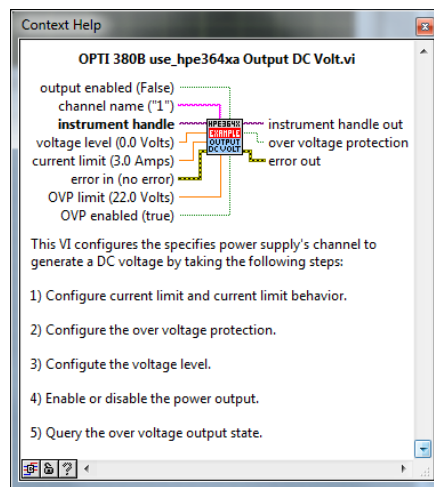


Figure 2. Context Help for Agilent E364X

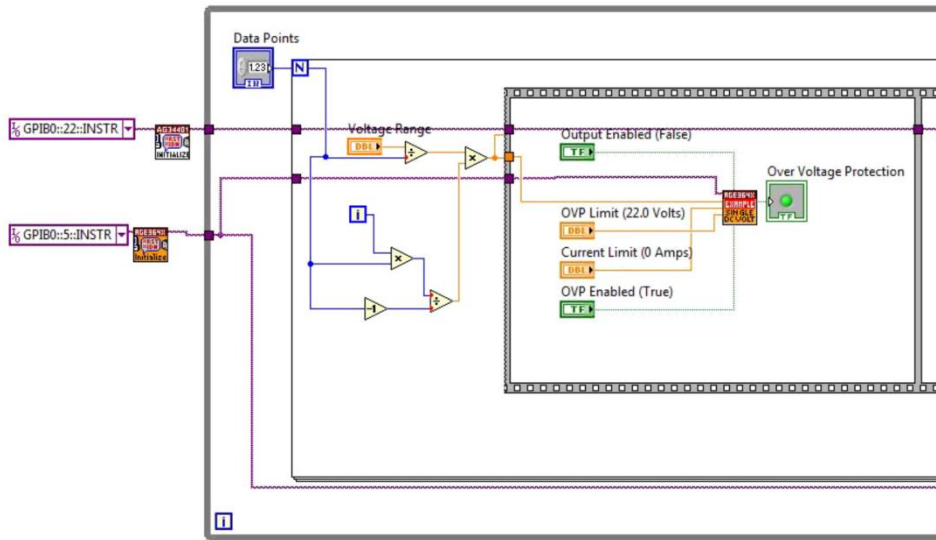


Figure 3. Example LabVIEW VI

Suppose you set the “Voltage Range” to 5V, and decide to have the power supply increment its output in 5 steps (N=5). What values of voltage appear at the output of the “Multiply” function (to the input of the “AGE364X Example Single DC Volt” VI)?

HINT: LabVIEW begins counting at $i=0$ (not $i=1$). In other words, the counter “ i ” takes on values of 0, 1, 2, 3, and 4 in this example (a total of $N=5$ values).

Lab Exercises:

A. GPIB Interface

Connect the GPIB interface (IEEE-488 cables and the Measurement Computing GPIB-488 PCI-bus computer card) to the Agilent E3640A Power Supply and the Agilent 34401 Multimeter. The interface will be used to measure the Current vs. Voltage (IV) curve of a diode in the forward bias direction

1. Turn on the power supply and multimeter. Note that upon power up an address flashes on the screen. This is the GPIB address for that instrument. Jot these in a convenient place for future reference. If necessary these addresses can be changed. The procedure for changing the GPIB address is in the user's manual.
2. Open Measurement and Automation Explorer to verify that the GPIB interface is operating properly and that it has recognized both instruments.
 - a. Click on the **Devices and Interface** line in the Configuration listing on the left.
 - b. Click on the **GPIB0** interface.
 - c. If there are no instruments listed on the right side of the screen, right click and select: **Scan for Instruments**. You may get an error. Just close the Explorer and open it back up again.
 - d. On the right side of the screen, select one of the instruments, right click and select **Communicate with Instrument**. The 488.2 communicator should pop up. This is a general GPIB communications interface where commands can be sent to the instrument. If you purchase a GPIB board it will come with its own communications software. Also, most generic type GPIB cards will work with LabVIEW. GPIB commands are sent as strings of characters. Press the Query button to send the *IDN? query. The *IDN? query directs the instrument to identify itself.
 - e. Close the Measurement and Automation Explorer.

B. Program Planning

1. In order to control instruments through the GPIB interface, the user must write a program that sends the task specific commands to the instrument. The commands and their proper formatting are generally contained in an instrument programmer's manual provided by the manufacturer. Most manufacturers will make available at additional cost their own interface program. Fortunately, these manufacturers also provide LabVIEW VI's for communications with their equipment. These instrument VI's can be downloaded from National Instruments web site free of charge, or found on the instrument driver CD provided with LabVIEW. You will use some of these VI's to write your program

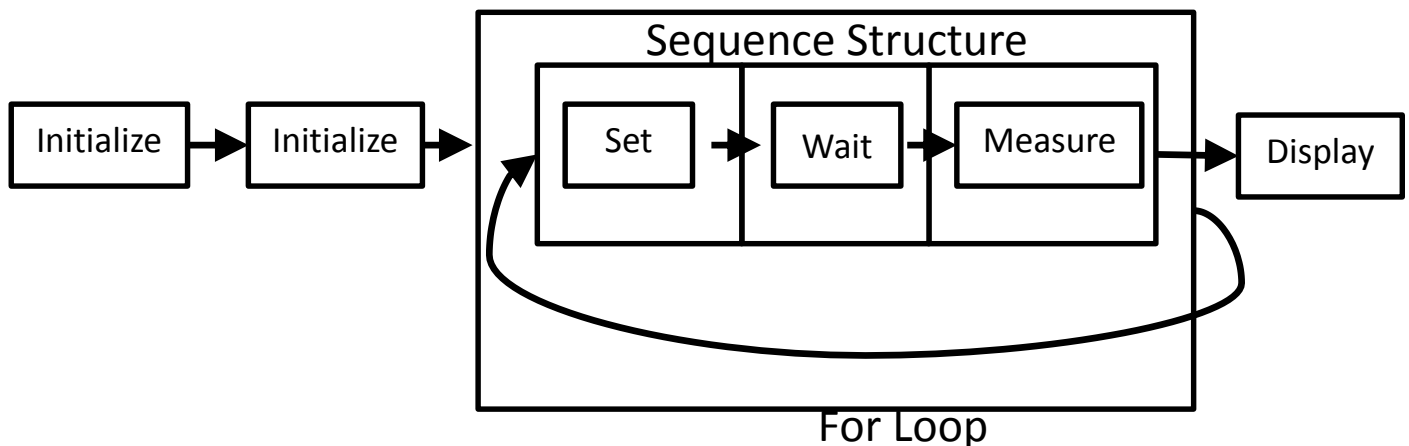


Figure 4. Program Layout

2. Figure 4 provides a schematic for the program layout. We will utilize four instrument VI's as sub VI's for our program. Two of these VI's will initialize the instruments. The third VI will set the voltage on the E3640A and the fourth will read the measured current from the 34401 multimeter.
3. Open a new VI in LabVIEW and expand the diagram window to fill the screen.

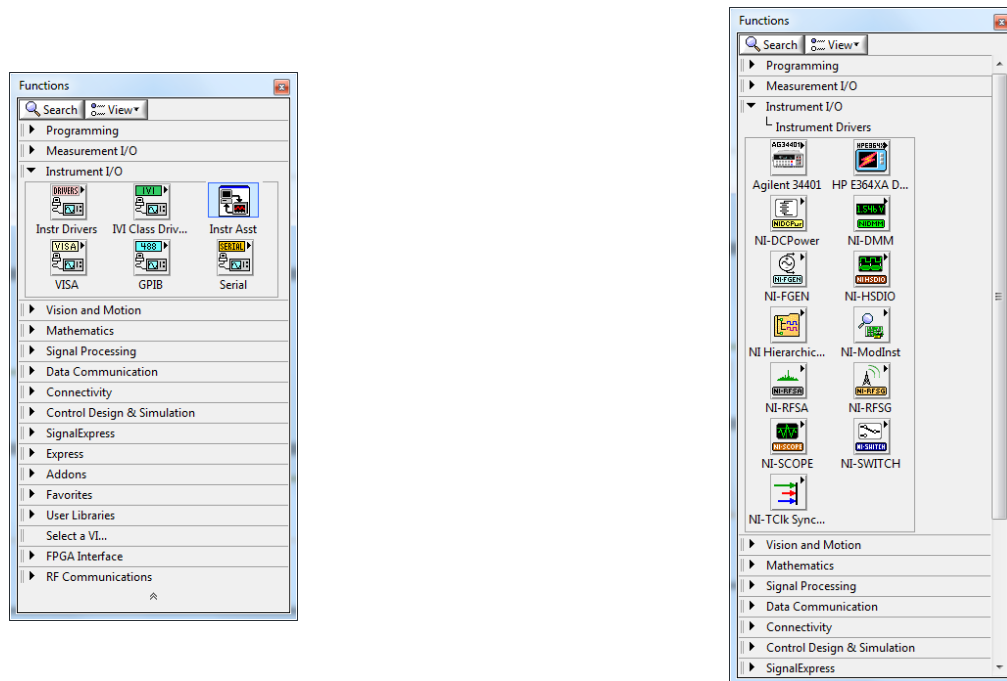


Figure 5. GPIB VI's

4. The sub VI's we will be using are found in the **Instrument IO** section of the **Functions** pallet. Using Figure 5 as a guide, locate the following VI's and place them on the diagram pallet.
 - a. Agilent 34401 Initialize
 - b. HPE364XA Initialize
 - c. Agilent 34401 Read
 - d. Agilent 34401 Configure Measurement
 - e. HPE364XA OUTPUT DC VOLT

C. Program Creation

1. Place a For Loop onto the diagram and expand it out so it covers about half of the visible screen area. The **For Loop** is located in the **Programming Structures** section of the Function pallet.
2. Place a **Flat Sequence Structure** within the **For Loop** and expand it to fill about half of the For Loop. Right click your mouse when it is placed on the border of the sequence structure. Select **Add Frame After**. Repeat the previous step to add another frame. You should now have a sequence structure with 3 sequences (0,1,2) place within a for Loop. Note: The sequence structure operates by each frame one at a time sequentially. This is one method to force the order of operation in the VI, however, usually better practice is to use the error in/out terminals in Vis to force what order they operate.
3. Drag the **OUTPUT DC VOLT VI** into frame zero of the sequence structure. Drag the **Agilent MEAS VI** into frame 2 of the sequence structure.
4. At this point your diagram should look like that shown in Figure 6.

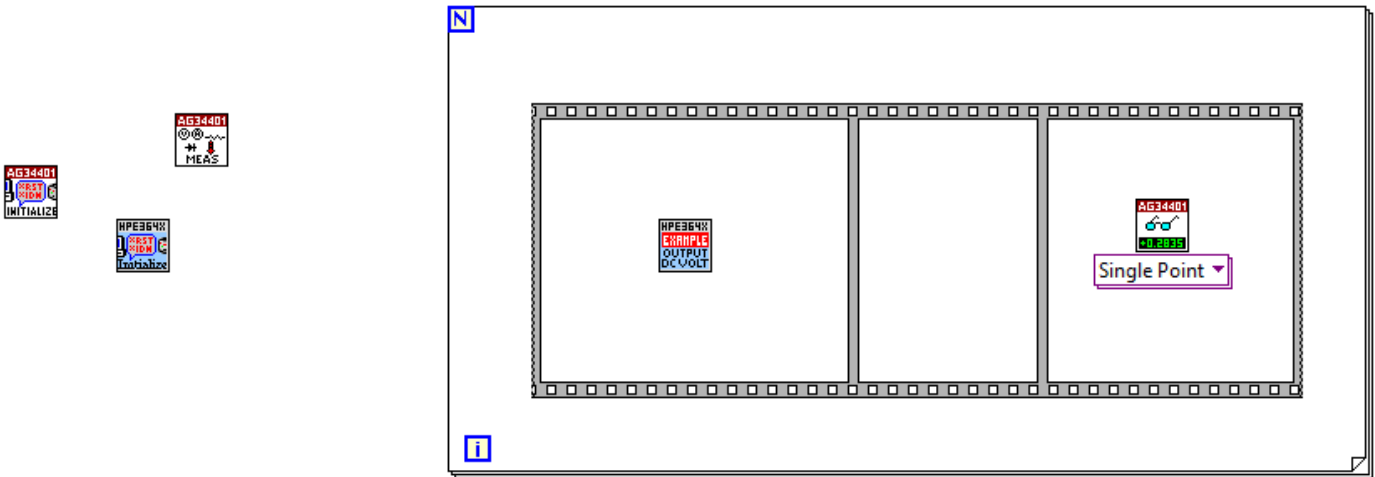


Figure 6. Intermediate Diagram Schematic

5. In the Help Menu select **Show Context Help**. Now move the Positioning tool or the Operating tool over one of the VI's placed on the diagram window. The context help should indicate the name of the VI you moved over as well as all of its inputs and outputs. Notice that when you move the wiring tool over a VI the terminal will be highlighted. Items shown in parenthesis are default values.
6. Select the wiring tool and locate the **Agilent 34401 Initialize** sub VI. Move the wiring tool over the upper left corner of this sub VI. The **resource name** input should be highlighted. Right click the mouse and select **create** and then select **constant**. The constant may be created blank or with some default value. Either way change the value of the constant by using the drop down list and selecting the proper device (GPIB::22::INSTR).
7. Repeat the above step for the **HPE364XA Initialize** sub VI. The correct channel device name for the HPE3640 is GPIB::5::INSTR.
8. Using the wiring tool, connect the error out of Agilent 34401 sub VI to the error in of the HPE364XA sub VI. The results of steps 5-8 are shown in Figure 7. Note the for loop and sequence structures should still be present in your VI block diagram they are just not show in Figure 7.

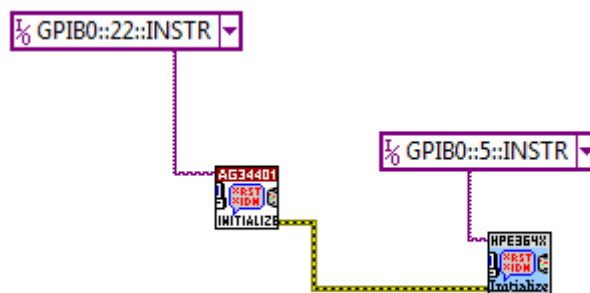


Figure 7. Diagram Schematic

9. The initialization sub VI generates an instrument handle that must be wired into the other sub VI's controlling a particular instrument.
 - a. Wire the **instrument handle** output from the **HPE3640 Initialize** sub VI into the **instrument handle** input for the **HPE64XA Output DC Volt** sub VI. This sub VI should be in frame zero of the sequence structure.

- b. Wire the **Visa Resource** output from the **Agilent 34401 Initialize** sub VI into the **instrument handle** input for the **Agilent 34401 Configuration Measurement** sub VI.
- c. Right click on the **Function** input of the **Agilent 34401 Configure Measurement** sub VI and select **Create -> Constant**. Using the drop down list of the constant value select **DC Current**.
- d. Wire the **Error Out** of the **HPE364X Initialize** Sub VI to the **Error In** of the **Agilent 34401 Configure Measurement** Sub VI.
- e. Then wire the **Error Out** of the **Agilent 34401 Configure Measurement** sub VI to the **Error In** of the **HPE364X Output DC Volt** sub VI. At this point your block diagram should look like the one in Figure 8.

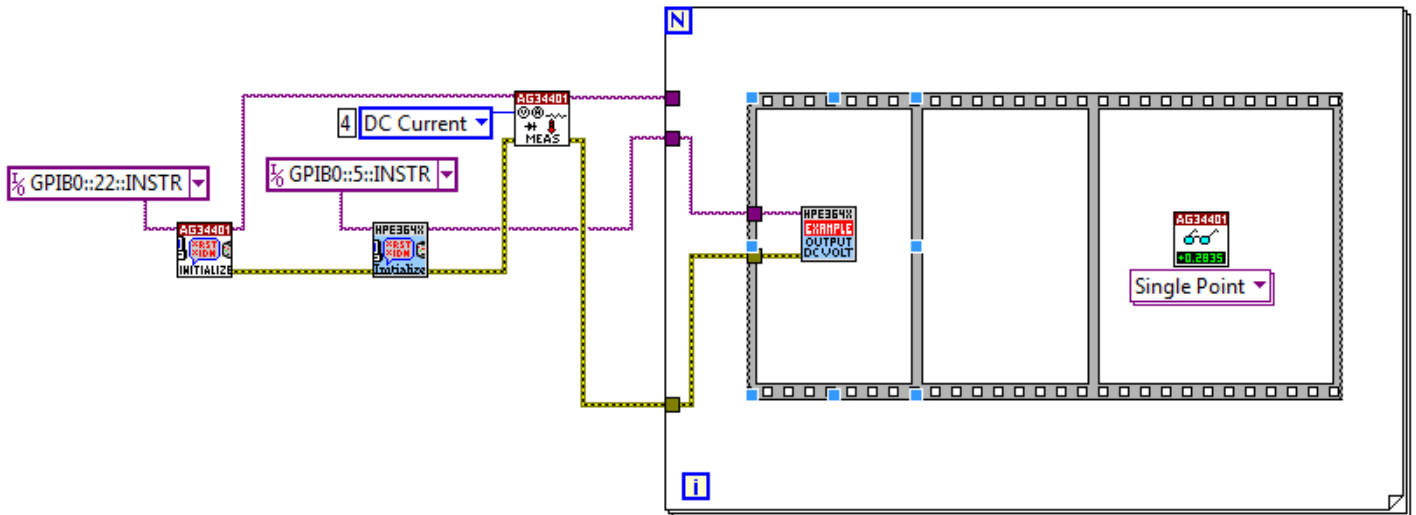


Figure 8. Diagram Schematic

10. There is an **output enabled** input for the **HPE640 Output DC Volt** sub VI. This input has a (default) value of FALSE. In order to enable the output we need to wire a TRUE constant to this input. Move the wiring tool over the **output enabled** input, right click and select Create, then select Constant. The Boolean Constant that is created will be false. To change the value to true, select the operator tool, the hand in the upper left corner, and select the "T" side of the Boolean constant.
11. There is an **OVP** input for the **HPE640 Output DC Volt** sub VI. This input sets the maximum output voltage. Create a constant at this input and set it to 10 volts. Create a constant at the **Current Limit** input and set it to 1 amp.
12. At this point your diagram should look similar to that shown in Figure 9.

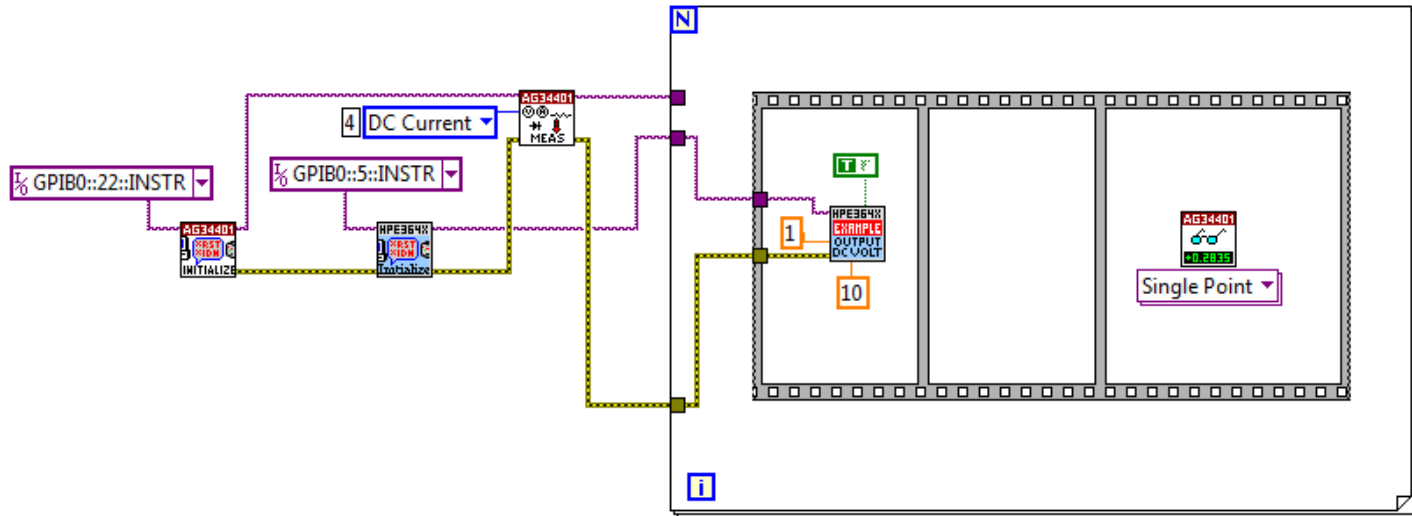


Figure 9. Diagram Schematic

13. Each time the FOR loop cycles the output voltage must be incremented by a fixed amount. For example, if we want to have 10 data points and we want the maximum output voltage to be 2 volts, then the voltage increment per step will be 2 volts / 10 steps or 0.2 volts/step. The maximum voltage and the number of data points must be selected by the user through controls placed on the front panel. Based on these two inputs, an algorithm must be generated to increment the ordered output voltage.
 - a. In the upper left corner of the FOR loop is a blue box with a capital blue **N** inside. This controls the number of cycles conducted by the loop. The value wired to this input will thus control the number of data points taken, since one data point is measured per loop cycle. Using the wiring tool, right click on the **N** and select **Create** and then select **Control**. Change the control label to read "Data Points". Switch windows to the front panel and notice that a numeric control has been created.
 - b. Switch back to the Diagram window. In the **Numeric** sub pallet, select the **Divide** function and place it within the FOR loop.
 - c. Using the wiring tool, create a control at the top input to the divide function. Label the control "Voltage Range".
 - d. Wire the **Output** of the **N** of the FOR loop to the bottom input of the **Divide** function. The output of the FOR loop is now the voltage increment.
 - e. Select the **Multiply** function and place it within the FOR loop. Wire the output of the Divide function into the top input of the **Multiply** function. Wire the iterator, **i**, of the FOR loop to the bottom input of the **Multiply** function. The output of the Multiply function will now increase by the desired voltage increment for each cycle of the FOR loop.
 - f. Finally, wire the output of the Multiply function into the voltage level input of the **HP640 Output DC Volt** sub VI.
 - g. At this point your diagram should be similar to that shown in Figure 10

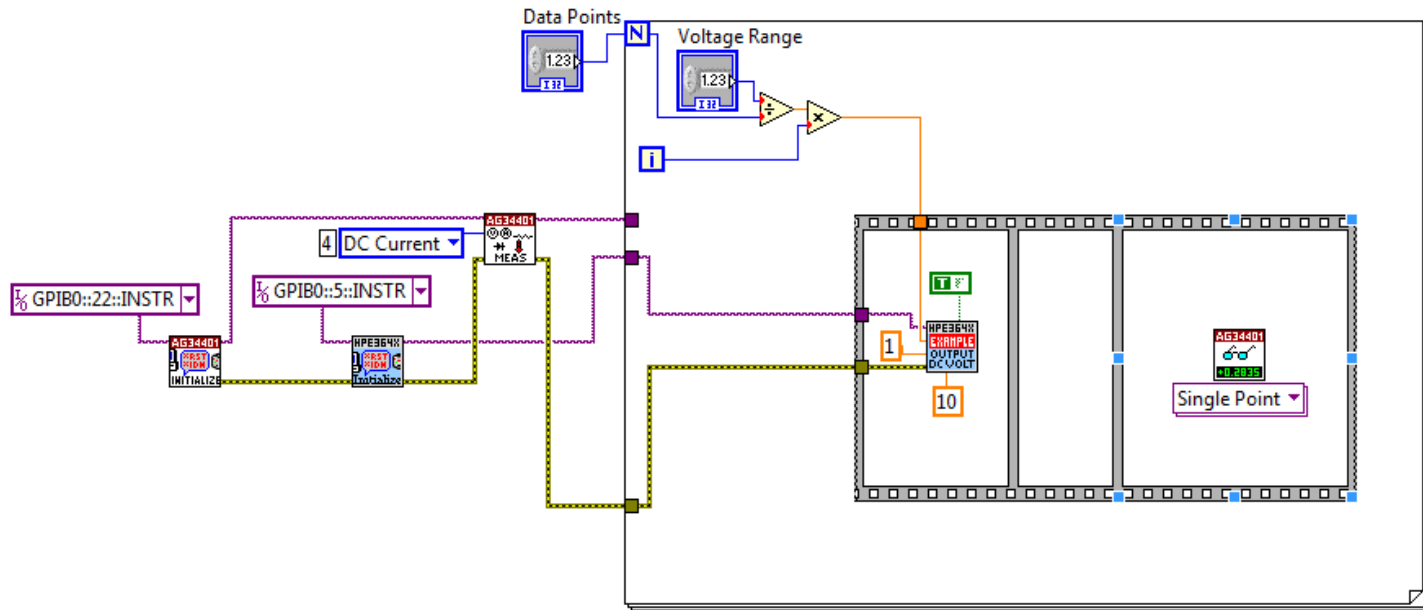


Figure 10. Diagram Schematic

14. Select frame one of the sequence structure. Using the wiring tool, create a constant at the input for the **Wait Until Next (ms) Multiple** function. Set this constant to 10.
15. Wire the **Visa Resource Name Output** of the **Agilent 34401 Configure Measurement** sub VI to the **Visa Resource Name Input** of the **Agilent 34401 Read** sub VI.
16. Wire the **Error Out** of the **HPE364X Output DC Volt** sub VI to the **Error In** of the **AG34401 Read** sub VI.
17. Each time the FOR loop cycles the voltage is incremented, and a current reading is taken. Each of these current readings must be stored in an array. Upon completion of all measurements, the data values stored in the array will be displayed on the front panel.
 - a. Select frame two of the sequence structure.
 - b. Using the wiring tool, wire the **reading** output of the **Agilent 34401 Read** sub VI to the right hand border of the FOR loop. Where the wire passes out of the sequence structure you should see a solid orange box. Where the wire intersects the border of the FOR loop you should see a solid orange box with a set of brackets inside. If you do not see the brackets then right click on the box and select **enable indexing**. Indexing means that the FOR is storing all of the values passed to its border on each cycle into an array. If indexing was disabled the FOR loop would only pass out the value obtained on the last cycle.
 - c. In addition to the current readings, we also want the associated array of voltage values. To obtain an array of voltage values simply wire voltage input value to the border of the FOR loop and ensure that indexing is enabled.
18. Create a method to handle errors that occur in the program. To do this go to the **Programming** section of the functions palette. Select the **User Interface** set of sub VIs then drag and drop a **Simple Error Handler** VI into for loop, but outside of the sequence structure (preferably toward the right side portion inside the for loop).
 - a. Wire the **Error Out** of the **Agilent 34401 Read** sub vi to the **Error in** of the **Simple Error Handler** sub VI that you just placed inside the for loop.
 - b. At this point your diagram should be similar to that shown in Figure 11.

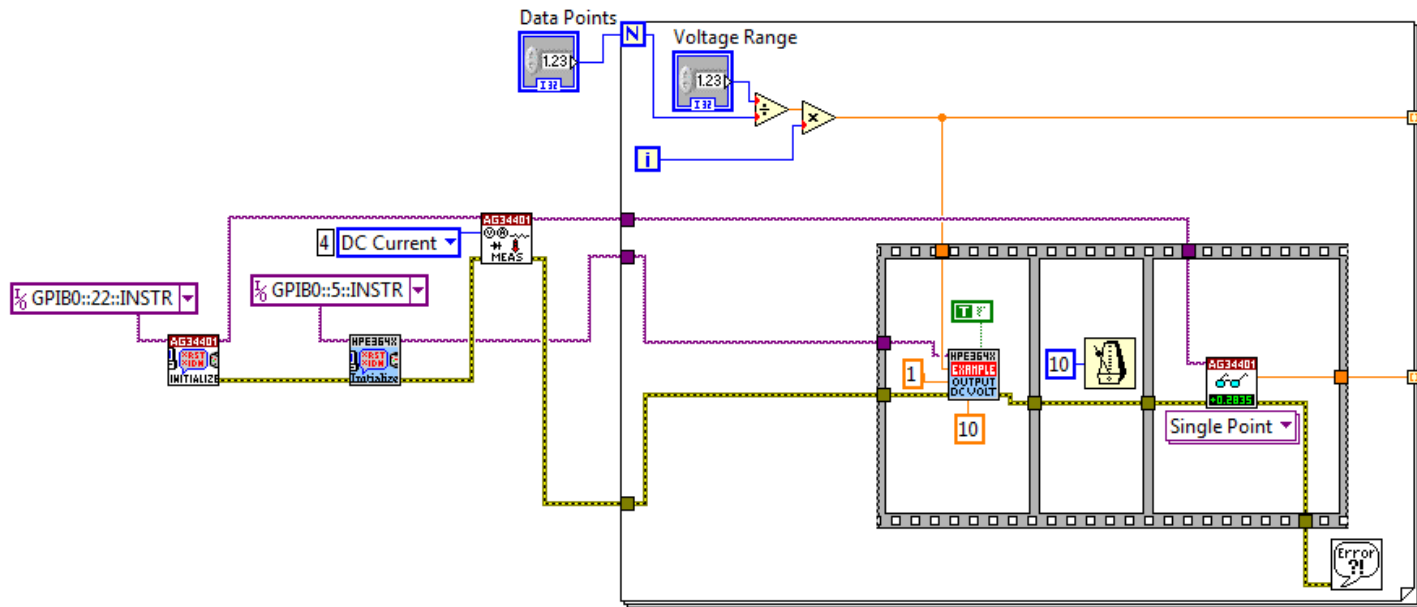


Figure 11. Diagram Schematic

19. Switch to the front panel window. From the **Graphs** section of the **Controls** pallet select the **XY Graph** and place it on the front panel.
 - a. Expand the graph to cover half the panel.
 - b. Using the edit text tool, change the X and Y axis labels of the graph to read "Voltage" and "Current (amps)" respectively.
 - c. At this point your front panel should look like that shown in Figure 12.

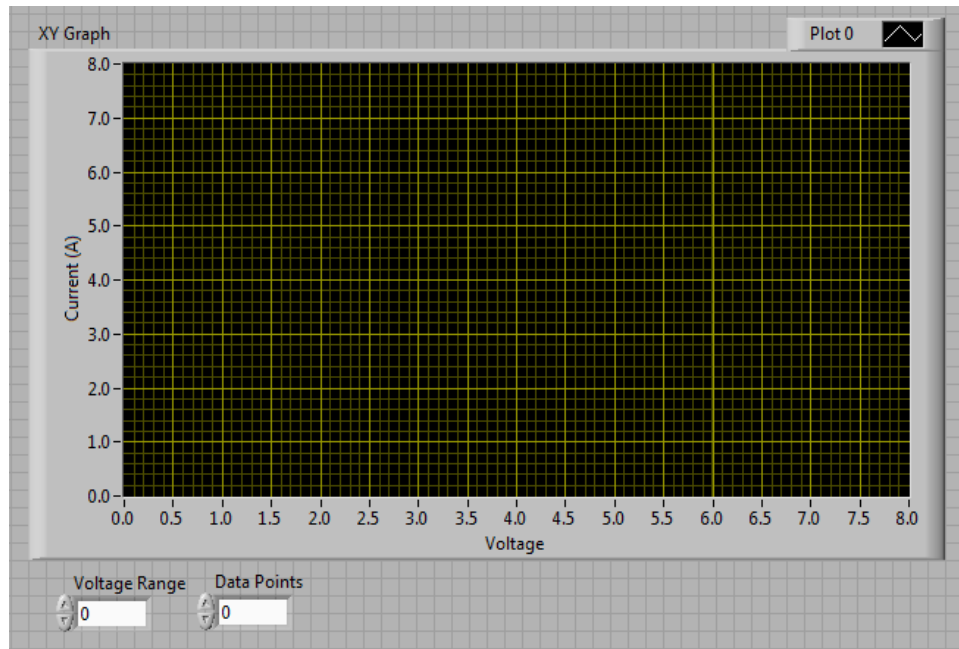


Figure 12. Front Panel Example Layout

20. Switch back to the Diagram window. Find the XY graph control that was placed on your diagram when the graph was placed on the front panel. If you are having difficulty locating the control, go back to the front panel and double click on the XY graph using the select tool. Drag the graph control to the right side of the FOR loop.

21. If this XY graph was created from the Express Palette, it is necessary to delete the “Build XY Graph” (blue box) on the block diagram.
22. The two output arrays that represent the X and Y data values must be bundled prior to input into the XY graph control. Data elements are bundled into what is called a Cluster. A Cluster is simply a grouping mechanism for data.
 - a. Select the **Bundle** function of the **Cluster** sub pallet and place it just to the right of the data output nodes on the FOR loop.
 - b. Wire the voltage array to the top cluster input.
 - c. Wire the current array to the bottom cluster input.
 - d. Wire the cluster output to the XY graph control.
23. To ensure that this saves our data, we need to take a wire from each of the two inputs into the array and build a new array (Found in the Programming-Array menu). Make sure to drag this function up to have two inputs. The output of this new array will be wired to a Write to Spreadsheet function. This can be found in the Programming-File I/O menu. You do not have to wire anything to this function; it will just ask you for a file dialog every time you run your program.
24. Upon completion of the measurements, we want to shut off the voltage source. To do this place an **HP640 Output DC Volt** sub VI outside and to the right of the FOR loop. Wire the **instrument handle out** of the same sub VI in sequence into the **instrument handle** input for this VI (make sure to **Disable Indexing** of the instrument handle wire leaving the for loop). When the FOR loop has completed cycling, this VI will shut off the output since the default value for the output enable is false. Create a constant for the current limit to be set to 1 amp. Create a constant for the output enabled, and have it set to false.
25. At this point your program should look like that shown in Figure 13, and is ready for testing. If the run arrow in the menu bar at the top of the screen is broken and dark, this means there is a problem with your VI. Double click the arrow to determine the errors and correct them.

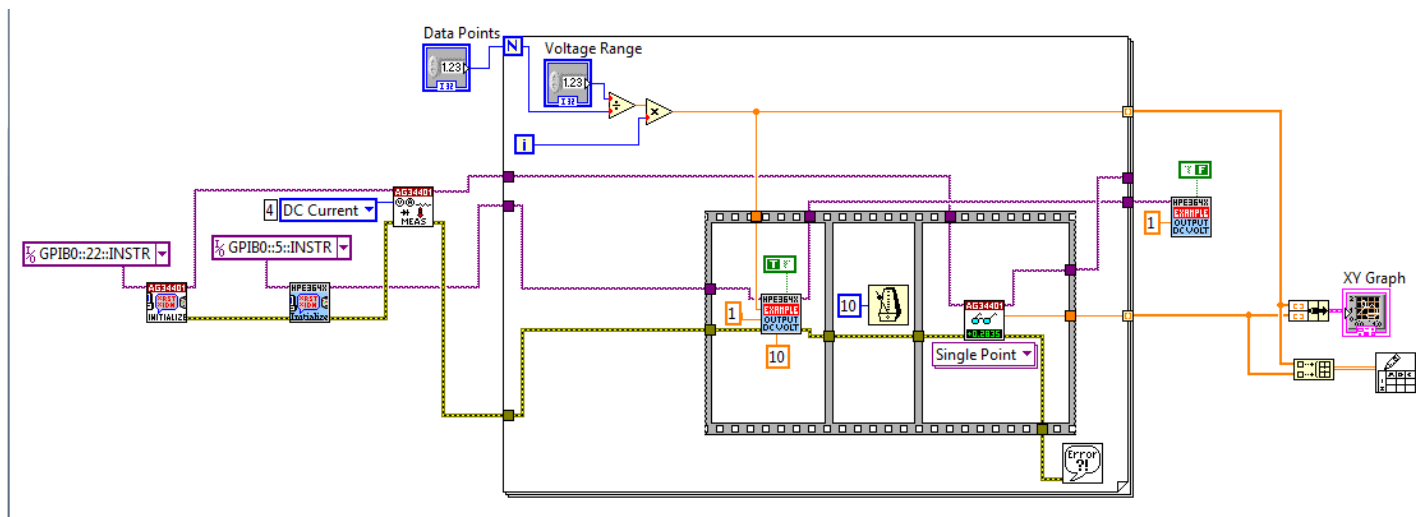


Figure 13. Final Rear Panel Example Layout

D. Program Testing

1. Turn on the HP3640A and the Agilent 34401. You should not have anything connected to the instruments for initial program testing.
2. On the front panel of your program set the voltage range at 3.0 and set the Data Points to 25.

3. Start your VI by pressing the run arrow in the menu bar at the top of the screen. One of the instruments may start to beep. If this occurs, shut off the VI by pressing the red (stop sign) abort button. Restart the VI.
4. If the program is working the voltage output of the HP3640A should be slowly stepping up to 2 volts.
5. If you receive any errors or the voltage output isn't changing check the following.
 - a. Ensure the GPIB addresses are set properly.
 - b. Ensure that the instrument handle output from the initialization VI's are wired to the same instrument sub VI's.
 - c. Try running the program again. If you still receive errors try saving your program and shut down LabVIEW and both instruments. Restart and try again.
6. Ask your TA for help if your program still will not run.
7. Set up the circuit shown in Figure 14 and run your program.

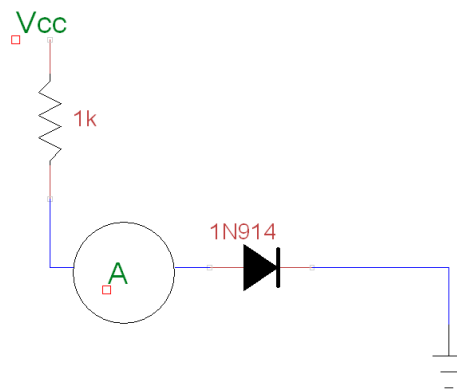


Figure 14. Circuit Schematic

- a. Adjust the scales of your graph to produce the best looking output.
 - b. Change the voltage range and number of data points a check to ensure the program behaves properly.
- [L1]** Plot this data to produce the Supply I/V curve.
- [L2]** From the Supply I/V curve, calculate and plot the diode I/V curve.
- [L3]** Compare the I/V curve that you have produced this week to the one that you made last week.
8. Change the diode out for a 5mm LED. Along with the diode change the resistor to a 200Ω.
 9. Run your program again for 1000 points, except for a maximum voltage of 10V.
- [L4]** Plot this data to produce the Supply I/V curve.
- [L5]** From the Supply I/V curve, calculate and plot the LED I/V curve.

E. Current to Voltage Converter (Transimpedance Amplifier)

The circuit found in Figure 15 is pre-built for you. This circuit uses an LF356 op-amp. It requires that you arrange the dual-output power supply to supply +15 V_{DC} to pin 7, -15 V_{DC} to pin 4, and ground to pin 3. Your TA will assist with setting this up.

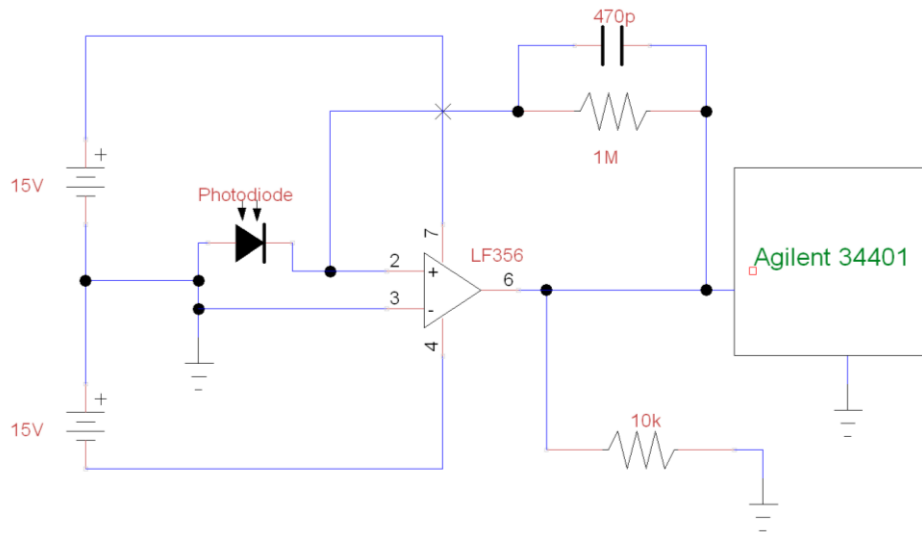


Figure 15. Transimpedance Amplifier (TIA) Circuit

1. This circuit should already be built on your proto-board. To properly work with this circuit, you will have to rewire your previous circuit to remove the Ammeter. Make sure to have your LED directly on top of the Photodiode. You will be using the LED to drive the Photodiode.
 2. With this done, wire up the Agilent DMM to measure the voltage across the 10kΩ resistor.
 3. This will also require you to modify your program from measuring the DC Current to measuring the DC Voltage.
 4. Change the maximum voltage from 3V up to 5V. Make sure to take a sufficient number of points.
- [L6]** Once you have measured the output voltage of your photodiode, based on the input LED voltage compute and plot the light response (I-Light) curve for the photodiode. Use your previous current from the LED to assist with this.