ME 3200 Mechatronics I Laboratory

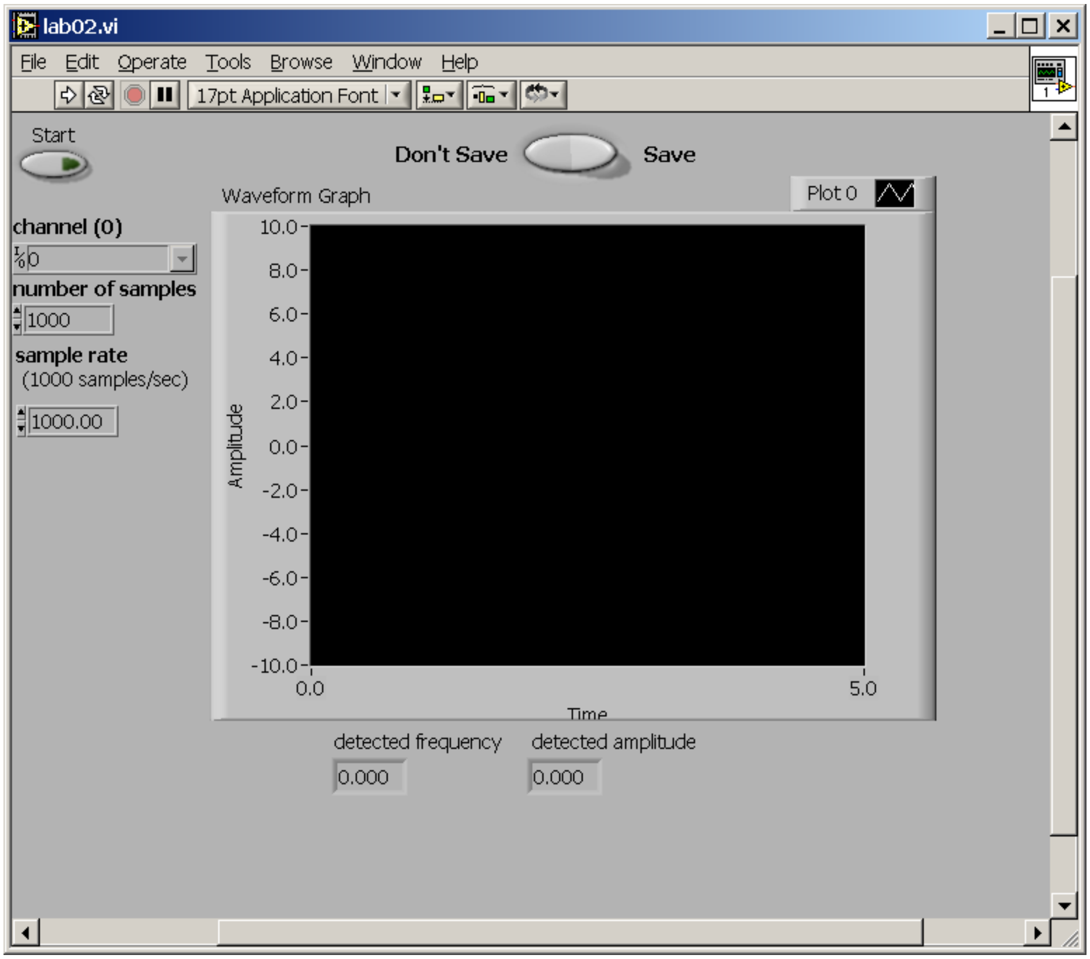
Lab 2: LabView Computer Systems

Introduction

The purpose of this lab is to introduce the basics of creating programs in LabView for data collection and analysis. LabView is a visual programming language capable of creating specialized graphical user interfaces that can collect, analyze, and export data to wide variety of applications. In this lab you will create a LabView program that will capture an AC signal, calculate the frequency and amplitude of the signal, and export the raw data to a spreadsheet file.

LabView's Basic Features

Each LabView program (also known as a VI) consists of two windows: the Panel Window and the Diagram Window. These windows are related, but serve different purposes. The Panel Window is the graphical user interface (GUI) for the program (see Figure 1 below). It holds the relevant controls for the program that allow a user to specify settings, begin data collection, initiate analysis, export data, etcetera. The Panel Window also contains indicators that display results of data analysis, the current state of a variable, or the raw data in both matrix and graphical formats. Once a VI is created, the user needs only use the Panel Window to control the intended data collection and analysis.



channel (0)
0

number of samples
1000

sample rate
(1000 samples/sec)
1000.00

detected frequency
0.000

detected amplitude
0.000

Figure 1: Panel Window of the Completed Program.

The Diagram Window (Figure 2 below) is the visual programming code for the LabView program. It prescribes the functions that the program will perform and the order of their operation. The Diagram Window may be described as block diagram representation of programming operations. Blocks placed in this window are connected with ‘wires’ in an input-output format that progresses from left to right. The ‘wires’ that connect these blocks carry different types of information to the various sub-programs that make up the whole of the current VI. The wires are color coded and assigned various line styles to signify the types of information that they carry. In the Diagram Window blocks representing the controls displayed in the Panel Window (bold outlined boxes) are the ‘sources’ of the program and the blocks representing the indicators (no bold outlines) are the ‘sinks’ of the program.

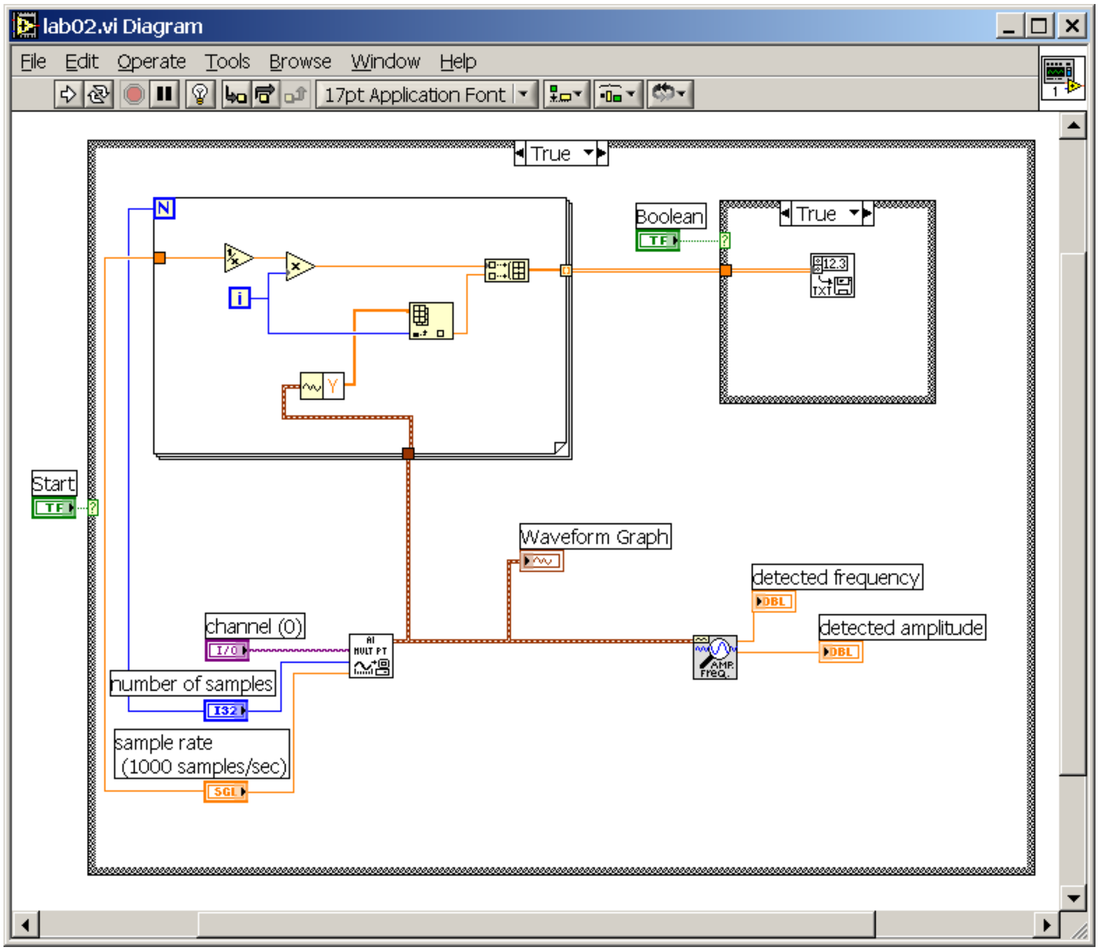


Figure 2: Diagram Window of the Completed Program

LabView is a very powerful programming tool when collecting data from a wide variety of applications and is widely utilized in many industries. In principle, LabView’s utility is only limited by the hardware it’s connected to, the skill of the programmer, and the experience of the user. Often the most common obstacle of new users is unfamiliarity with LabView’s capabilities and programming methods. Often the extensive examples included in LabView’s help system offer many examples of performing tasks that range from simple to complex. However, understanding how to implement and blend different examples to suit a specific task requires the study of the diagrams of these examples and at least a rudimentary understanding of LabView’s visual programming language. A noteworthy tool to use during the examination of LabView examples is the Context Help Window (activated by selecting ‘Help’ > ‘Context Help Window’). The Context Help Window is a window that dynamically displays a summary of the help information associated with items in the Diagram Window as the user hovers the mouse pointer over or selects the programming block of interest.

The following lab procedure will guide novice users to create a basic VI that essentially is a digital oscilloscope. It also offers the user a more complex VI to examine so as to catch a glimpse into some of the more complex and useful VI's LabView is capable of. In addition to these exercises, students are encouraged to explore LabView's help files, examples, and website to deepen their understanding of LabView's capabilities and applications.

Lab Procedure

Get Acquainted with LabView

1) Open LabView via start menu

2) Select the "New VI" button to create a new LabView program.

3) Take a moment to look around in at the different tools available to you in each window (right click in the window to show the menus for each window).

Create Your Program

4) Right click in the diagram window to show the functions menu, select 'Structures' > 'Case Structure', and draw a square that fills the whole window by clicking then dragging—you'll need the space.

5) Select the wiring tool from the tools pallet (looks like a little spool) and right click on the green square on the left of the box and select 'Create' > 'Control' and a green box labeled "Boolean" appears to the left of the case block attached with a green wire.

   Note: If the tools pallet is not visible select 'Window' > 'Show tools pallet'

6) Select the Panel Window and notice the button that LabView placed in this window. This button is the start button for your program—change its text label from 'Boolean' to 'Start'.

7) Using the pointer tool (looks like a hand), right click on the start button and select 'Mechanical Action' > 'Switch Until Released'.

8) Right click in the Diagram window and select 'Data Acquisition' > 'Analog Input' > 'AI Acquire Waveform'—place 'AI Acquire Waveform' block in the 'Case Structure' block.

9) Using the wiring tool, right-click the purple 'lead' of the 'AI Acquire Waveform' block and select Create > Control.

10) Repeat step nine for the blue and orange leads below the purple lead on the left of the 'AI Acquire Waveform' block.

11) Switch to the Panel Window and use the position/size/select tool to rearrange the three controls you just created so that their labels are easy to read (you may also want to rearrange the blocks and wires in the Diagram Window).

12) Right click in the Diagram Window, select 'Structures' > 'For Loop', and draw a square in the upper left quarter of the 'Case Structure' block.

13) Using the wire tool, left click on the number of samples block and connect it to the blue N lead of the 'For Loop' block to specify the number of iterations.

14) Right click in the Diagram Window, select 'Numeric' > 'Reciprocal' and place the 'Reciprocal' block inside the 'For Loop' block.

15) Right click in the Diagram Window, select 'Numeric' > 'Multiply', and place the block in the 'For Loop' block.

16) Right click in the Diagram Window, select 'Array' > 'Build Array', and place it into the 'For Loop' block.

17) Using the position/size/select tool, grab the bottom corner of the ‘Build Array’ block and drag it down until the block has two inputs.

18) Right click in the Diagram Window, select ‘Array’ > ‘Index Array’, and place it into the ‘For Loop’ block.

19) Right click on the brown output lead of the ‘AI Acquire Waveform’ block, select ‘Waveform Tools’ > ‘Get Waveform Components’ from the drop-down menu, and place the block that appears in the ‘For Loop’.

20) Using the wire tool, connect the input of the ‘Get Waveform Components’ to the brown wire, and the output of the ‘Get Waveform Components’ to the top input of the ‘Index Array’ block.

21) With the pointer tool (looks like a hand with an extended index finger), click on the right half of the ‘Get Waveform Components’ block and select ‘Y’ from the drop-down menu.

22) Using the wire tool, connect the (1)‘Sample Rate’ to the ‘Reciprocal’ input, (2) the ‘Reciprocal’ output to the ‘x’ input of the ‘Multiply’ block, (3) the ‘i’ block to the ‘y’ input of the ‘Multiply’ block, (4) the ‘Multiply’ output to the top input of the ‘Build Array’ block, (5) the ‘i’ block to the lower input of the ‘Index Array’ block, (6) and the output of the ‘Index Array’ block to the bottom input of the ‘Build Array’ block.

23) Right click in the Diagram Window, select ‘Structures’ > ‘Case Structure’ and make a square in the top right corner of the first ‘Case Structure’ loop.

24) Right click the green question mark input of the ‘Case Structure’ loop you just created and select Create > Control.

25) Go back to the Panel Window, use the position/size/select tool to move the switch labeled ‘Boolean’ to an aesthetically pleasing location.

26) Right click the Boolean you just moved, select Replace, and choose a rocker, toggle, or slide switch from the Boolean menu. Right click the switch again and select Visible Items > Label (the label should disappear).

27) Using the edit text tool, type the words “Don’t Save” to the left of the switch and the word “Save” to the right of the switch.

28) Right click in the Diagram Window, select “File I/O” > “Write to Spreadsheet”, and put it into the second ‘Case Structure’ block.

29) Using the wire tool, connect the output from the ‘Build Array’ block in the ‘For Loop’ to the ‘2D Data Input’ lead on the ‘Write to Spreadsheet’ block.

30) Right click in the Diagram Window, select ‘Analyze’ > ‘Waveform Measurement’ > ‘Extract Single Tone Information’, and place it into the lower right quarter of the large ‘Case Structure’ block.

31) Using the wire tool, connect the output of the ‘AI Acquire Waveform’ block (brown lead/wire) to the time signal input of the ‘Extract Single Tone Information’ block (top left lead).

32) Right click on the “Detected Frequency” output lead of the extractor block and select Create > Indicator.

33) Repeat step thirty-two for the “Detected Amplitude” output lead of the ‘Extract Single Tone Information’ block.

34) Select the panel window, right click on the gray area of the panel, select ‘Graph’ > ‘Waveform Graph’, and place it on the panel in a convenient location.

35) Go back to the Diagram Window and, using the wire tool, connect the output of the ‘AI Acquire Waveform’ block to the ‘Waveform Graph’ block.

Note: At this point your program is finished. Refer to the preceding figures if you have questions about your program's diagram.

36) Save your program to a disk that you will take with you. Make sure each lab partner gets a copy of the program.

Testing the Program

37) Locate your signal generator, turn it on, and adjust its output to a sine wave with a frequency of 100 Hz.

38) Connect the output of the signal generator to the A_CH0 of your data acquisition breakout box located on the top of the bench.

39) Select 'Run Continuously' from the tool bar on the top of the panel window—it is the button with the arrows that form a circle.

40) Click on the start button in the Panel Window of the LabView VI you just created, and, if you have done everything correctly, look at your data displayed on the waveform graph; debug your setup/program if no data is displayed.

41) Test your program with the "Don't Save"/"Save" switch in both positions and verify that your data is saved in a format that can be opened in Excel (be sure to save the file with a '.dat' file extension). Debug your program as necessary.

42) Demonstrate to your TA that your program and setup works correctly.

Learning From Other LabView VI's

43) Open the two-channel oscilloscope VI from the 'LabView' folder on the desktop ('2_channel_oscope.vi'), and open the diagram of the two-channel oscilloscope by pressing Ctrl-e on the keyboard.

44) Compare and contrast the diagram of the two-channel oscilloscope and the diagram of the program you just made—make note of the similarities and differences (the Context Help Window may be useful here).

45) Answer the lab questions on the following page.

Clean Up Your Station

46) Turn off power to the signal generator, return cables and connectors to their respective racks at the ends of the lab benches, and cabinets, exit LabView, and log off your computer.

Questions

1) What is the purpose of each the following structures in the diagram window of LabView?

    a) Case Loop

    b) For Loop

2) Explain the difference between an indicator and a control in LabView.

3) Using the help menu, list each of the possible colors of wires in the diagram window and what kind of information each symbolizes.

4) What does the ‘For Loop’ in your program do? Why is it necessary to include it in the program?

5) What are the advantages and disadvantages of the VI that you created? How does it compare with the two-channel oscilloscope VI?

6) Suggest two possible applications for LabView.