UtahState UNIVERSITY

Mechanical & Aerospace Engineering

MAE 3340 *INSTRUMENTATION SYSTEMS*

Introduction to National Instruments LabVIEW and Data Acquisition (DAQ) Methods

MAE 3340 Lab 2

This lab report will be considered a formal report. You can use this *Word Document* to complete your report. To hand in your project report, you need to attach a hard copy of your code, both front panel and block diagram and your data sheets from this experiment. Answer al Questions at the end of this document. Attach extra sheets if necessary. To print your code go to File»Print Window». Your data sheet for this VI should list the battery nominal voltage and the sensed voltage. (Be sure to check for spelling and grammatical errors):

SECTION/Team Name ____

Student NAME: ____

Introduction:

This exercise is designed to acquaint you with the Labview® programming language – the basic "workhorse" for this class. The completion of this sheet will constitute the write up required for the lab. It is anticipated that this initial laboratory will take two laboratory sessions to complete. This exercise will be collected after the completion of the laboratory session in week 2 of class.

Part 1

This exercise is designed to acquaint you with the Labview® programming language – the basic "workhorse" for this class. It is recommended that you ENTIRELY complete the On-line Labview tutorials before proceeding to part 2 of this lab.

Introduction to Labview Graphical Programming Language

http://www.ni.com/academic/students/learn-labview/

View Introductory Tutorial:

1. LabVIEW Tour Videos, Section 1-5, 9.
2. Introductory Whitepaper, http://www.ni.com/white-paper/14556/en

Part 2

This lab exercise requires the students to write a simple LabVIEW program. This lab gives the students a good background in what the LabVIEW interface looks like and how to read a simple voltage from the data acquisition (DAQ) board. Simple batteries will b e used as the voltage sources to be read.

In this lab, you will write a LabVIEW virtual instrument (VI) to read voltages, display them, store values in an array, calculate the mean and standard deviations of the logged data, and write the logged data to an output file. The idea is to build a VI that works like a graphing digital multimeter. You will learn to use digital multimeters later in this class.

Objective

- To navigate the LabVIEW graphical programming language environment.
- Take voltage measurements using computer-based instrumentation.
- Understand the concept of electrical termination.

Theory

This lab covers two very broad areas of study in instrumentation: graphical computer programming and analog-to-digital conversion.

Graphical Computer Programming

Traditional computer programming involves setting down a list of tasks for the computer to execute in the given sequential order. Each instruction is executed in the order of appearance in the list. Often, the availability of data determines the order given to these instructions. *Referred to as Object Oriented Programming.* If you specify the operations and the data dependencies, the computer can execute the instructions in any order that protects the data dependencies. LabVIEW programming consists of drawing pictures that specify data dependencies. The LabVIEW programming environment includes a large set of blocks to specify operations and a Wiring tool to connect them together. As an example, *Figure 1* shows the operation of multiplying two numbers and displaying the result.

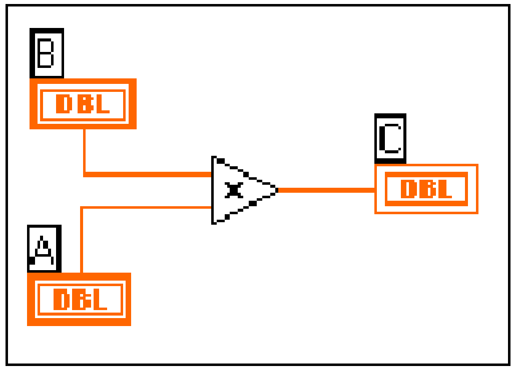


Figure 1. Program for Multiplying Two Numbers and Displaying the Result.

A LabVIEW program, called a virtual instrument (VI), is a two-window system. The code is in one window and the user interface (inputs and outputs) appears in a separate window. The program window is the block diagram window, and the user inputs and outputs are in the front panel window. *Figure 1* shows a sample program that would appear in the block diagram window. The numbers are entered into the computer and displayed in the front panel window shown in *Figure 2*. The two boxes on the left (labeled A and B) are controls, and the box on the right (labeled C) is the output or indicator. (The X and = are only displays showing the operation of the VI and not inputs or outputs.) The three boxes are associated with the similarly labeled boxes in the diagram window shown in *Figure 1.*

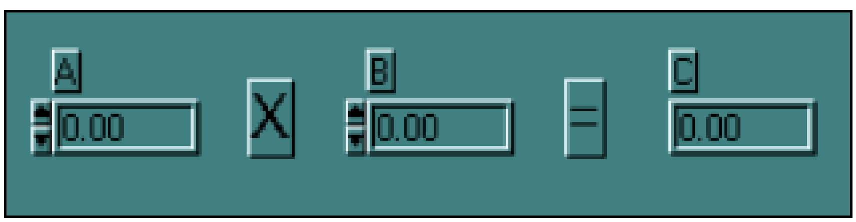


Figure 2. Front Panel for a Two-Number Multiplication Program.

Pre-Lab Preparation

- *Read through the theory and lab procedure for this experiment. Come prepared to execute the lab exercise.*
- *Bring the following to lab with you:*

  This experiment description.

  Your lab notebook and pencil.

  A virus-free thumb drive.

- *Your workstation should have the following items:*

Computer with National Instruments LabVIEW software.

National Instruments DAQ board (DMA inside the computer for this Lab).

NI-BNC 2120 Connector Block and Cable

Black BNC Connector Cable.

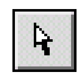
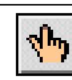
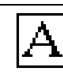
Voltage Probes for BNC Cable.

25-30 kW Resistor.

BNC Cable Adapters.

Batteries to be used as voltage sources.

Table 1. Some Useful Commands and Tools

Command/Tool	Purpose	Used When	Picture
Delete key	Deletes selected objects	There are unwanted objects in the program	<Delete>
Positioning tool	Moves and selects objects	You need to be move or delete program elements or insert new ones	
Wiring tool	Connects objects together	Program elements must be connected to allow data to flow between them	
Ctrl-B	Removes all broken wires	There are several unwanted wires in the program; use with caution	<Ctrl-B>
Operating tool	Changes values	You need to change a value in a front panel object	
Text tool	Edits text	You need to change a label or a comment	

Lab Procedure

A) Programming the Virtual Instrument

The following tutorial will help you learn the basics of LabVIEW programming. Read each step completely before executing the step. By the end of the tutorial, you will have written a VI that displays scaled random numbers on a chart. After completing this tutorial, you should be able to complete the rest of the lab. Be sure that everyone in the group gets a chance at the computer during the tutorial.

1. Setup

a. Insert your *thumb drive.*

b. Launch LabVIEW from the LabVIEW group in the Start menu of the task bar.

c. When prompted to open an existing or new VI, select New VI.

d. When the new VI windows appear, select Windows»Show Tools Palette to display the Tools palette.

e. From the Tools palette, select the Positioning tool, shown in Table 1.

f. From the File menu, select Save As and save the file to your desktop under a suitable name. The file extension must be *.vi. It is a good idea to save the file every few minutes during the development process. Save the file after making a change you want to keep. Or use <Ctrl-S> to open save menu. Save the program to your thumbdrive using a meaningful name, such as Graphing Digital Voltmeter.vi.

g. Review the commands and tools in Table 1.

2. Setting up a While Loop within your VI:

a. Click on the Block Diagram window (the window with the white background) to bring it to the front.

a. Insert a While Loop into the diagram window. First, open the Functions palette by clicking the right mouse button with the cursor in the block diagram window. Then move the pointer down to the Structures palette button (upper left button). When the cursor reaches the button, a palette of program elements will appear. Click on the While Loop (icon on the far right in the top row).

b. The program elements inside the While Loop will execute repeatedly as long as the input to the condition terminal is false. That is, as long as the variable Stop is false. Stop is the button on the front panel shown in Figure 4. When the user presses the button, Stop becomes true, and the While Loop stops executing.

c. The While Loop first appears in the Block Diagram window as a box-shaped cursor. Insert the loop by placing the cursor in the upper left corner of the block diagram window and clicking and dragging the icon to the lower right corner. Make the While Loop almost as large as the window, but don't fill the entire window.

d. Click on the Front Panel window (the window with the gray background). You will now insert a stop button to control the While Loop execution. You can find the button palette by choosing the Boolean controls from the **View>>Controls** palette. Clock on the Boolean Tab and Look for the *"Stop"* Button. Insert that button onto your front panel.

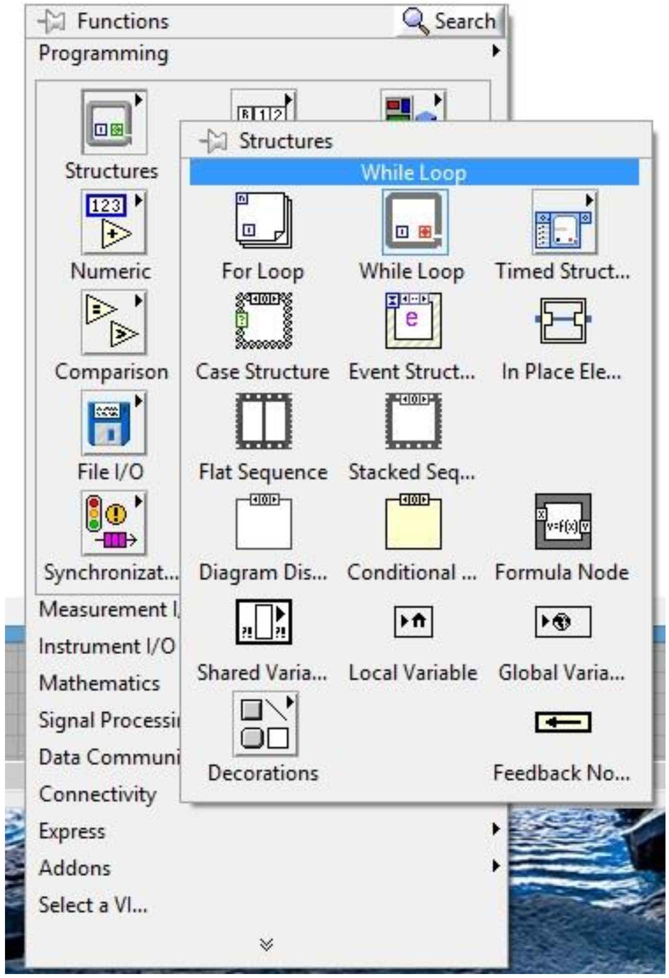


Figure 3. Palette showing location of While Loop

e. When you have the "Stop" button in place, Right Clickon the button and select Find Terminal to bring up the Block Diagram window. Make sure the boolean terminal is inside the while loop. If it isn't, use the positioning tool to drag it into the while loop.

f. Use the Wiring tool to connect the STOP terminal (small rectangle with TF written inside) to the conditional terminal that controls the While Loop. The conditional terminal should be in the lower right corner of the While Loop.

g. Insert a Waveform Chart. Right-click in the Front Panel window to bring up the Controls palette. Click on the Graph button (the right button in the second row) in the Controls palette. Choose Waveform Chart from the palette, move the cursor back to the Front Panel window, and click to insert the chart wherever you want.

h. Name the chart *Voltage Display* by typing the name and clicking on the Enter icon of the tool bar. You should see the text appear in a box near the upper left corner of the chart. If not, try pointing the cursor at the chart and clicking the right mouse button. In the menu that appears, select Visible Items»Label from the submenu and type the title.

i. Right Click on Waveform Chart and Select (from the drop down menu) "Visible Items>>Digital Display" .. this digital indicator displays the last voltage value measured.

j. Point the cursor at the chart and click and hold the right mouse button. Select Find Terminal in the pop-up menu and release the button. This should bring up the Block Diagram window, and the terminal for the chart will be highlighted. Make sure the chart terminal is inside the while loop. If it isn't, use the positioning tool to drag it into the while loop.

3. Setting up your V for Data Acquisition:

a. Right-click on block diagram, and then select *Measurement I/O>>NI-DAQmx>>DAQMXRead.VI.* Drag into your while loop.

b. "Hover" with the wiring tool near the upper left hand corner of DAQmx Block, select *Task/channels.in → Select Create>>Control.* Rockt-click on Task/channels in control, select >>Find-indicator.

c. On Front Panel click right arrow on task/channels in control floow drop down menu to TASK (see Fig 4).

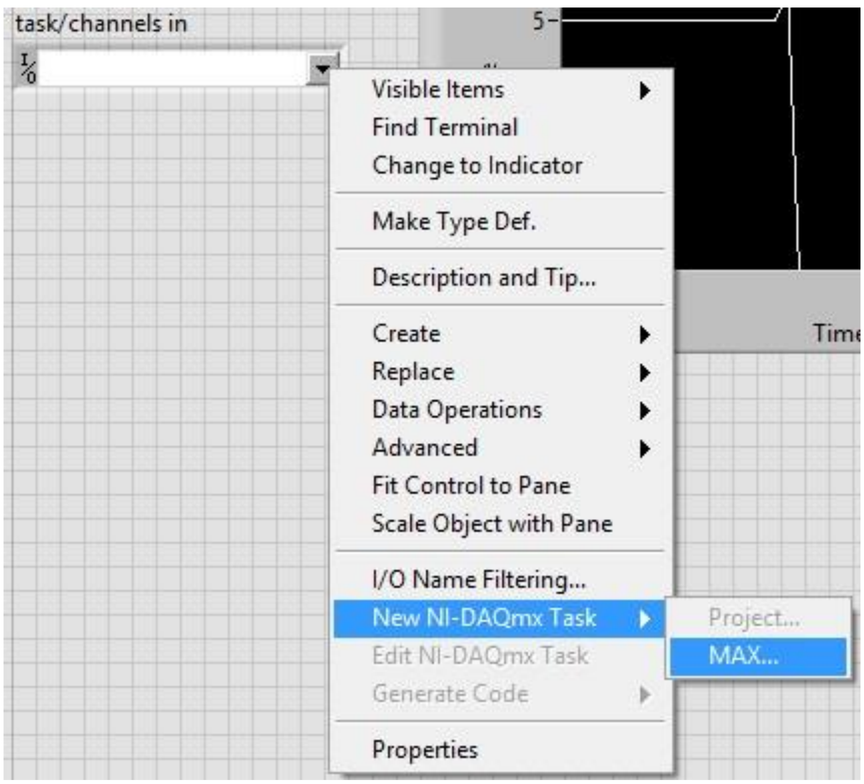


Figure 4. Palette showing location of MAX Task.

d. This action will bring up the DAQ Task Window … Select "+" *Acquire Signals>>"+" Analog Input >> Voltage*

e. This action wil bring up the *Physical channel* window showing the available Data Acquisition devices, and the available channels n that device .. see Figure 5 below.

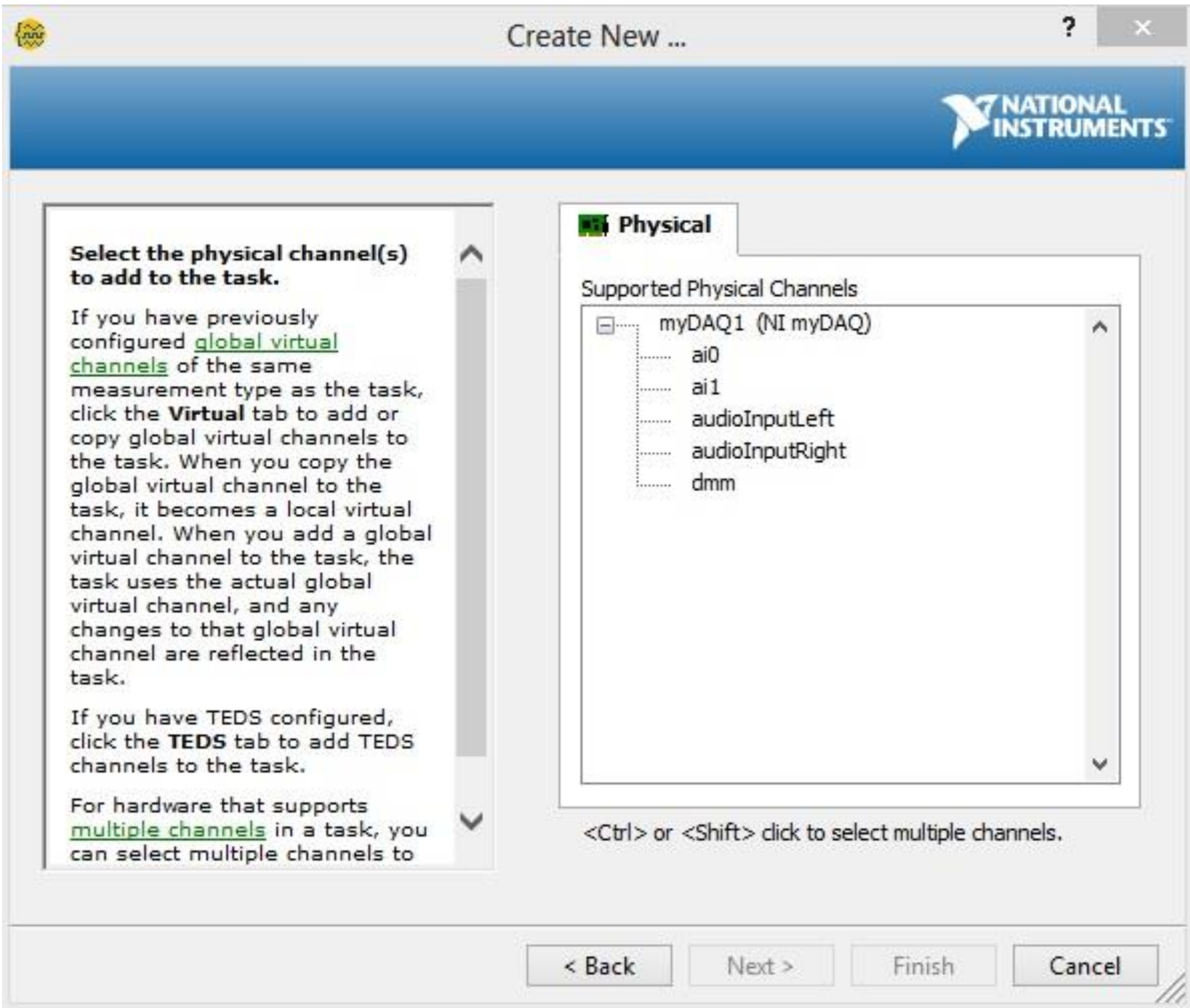


Figure 5. Palette showing location of MAX Task.

f. Click on *ai0*, when *ai0* iclon is shighlighted *click >>Next*

g. Enter a Name for the Task … something like *"Lab ~~1~~ 2 Voltage Display". .. click >>Finish*

h. When the task builds.. you will see a display like Figure 6. Set your configuration tabs to agree with this figure.

i. Wire the data output to the Waveform chart.

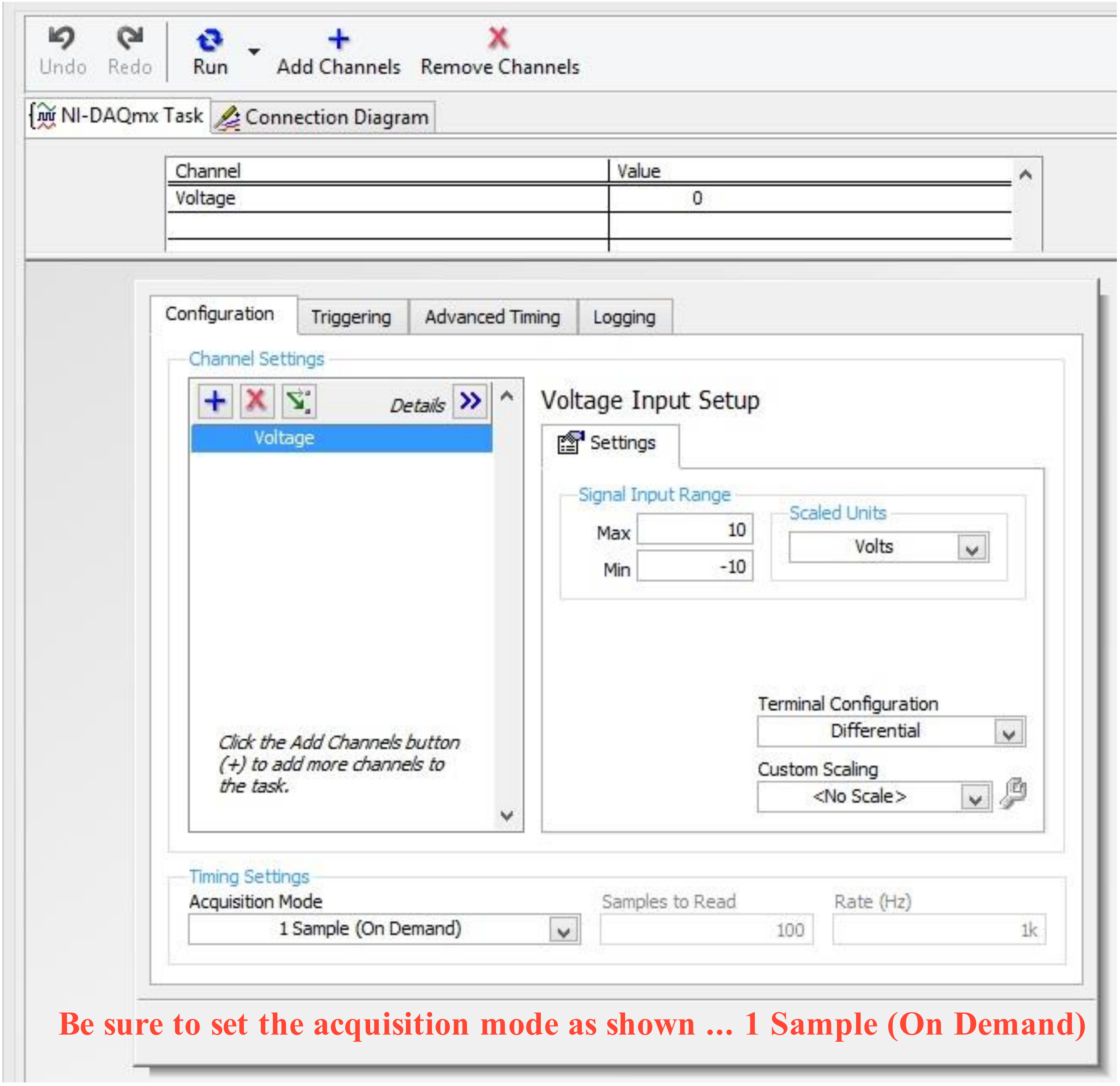


Figure 6. Palette showing Channel 0 Configuration Settings.

- *Now you can test your VI.* Click on the Run button (*The arrow in the upper left hand corner*), the single arrow in the upper left corner. To stop the execution, press the STOP button you put on the front panel. Run the VI several times. Does the VI run? How do you know? Save the VI to your desktop and drag a copy to your thumb drive. (You do not need to rename the file.)

2. *Adding a Stacked Sequence and Frame Rate Control to the VI*

a. In the next few steps, you are going to add structured frames to the VI, with each frame performing steps in sequence

b. Use the Positioning tool to arrange the controls to make your front panel look presentable.

c. Switch to the Block Diagram window. Make sure the terminals for the new controls are inside the While Loop. If they aren't, use the positioning tool to drag them into the while loop.

d. On Block Diagram right click structures>>stacked sequence. "Draw your sequence inside of the while loop and around the other objects as shown in Fig 8.

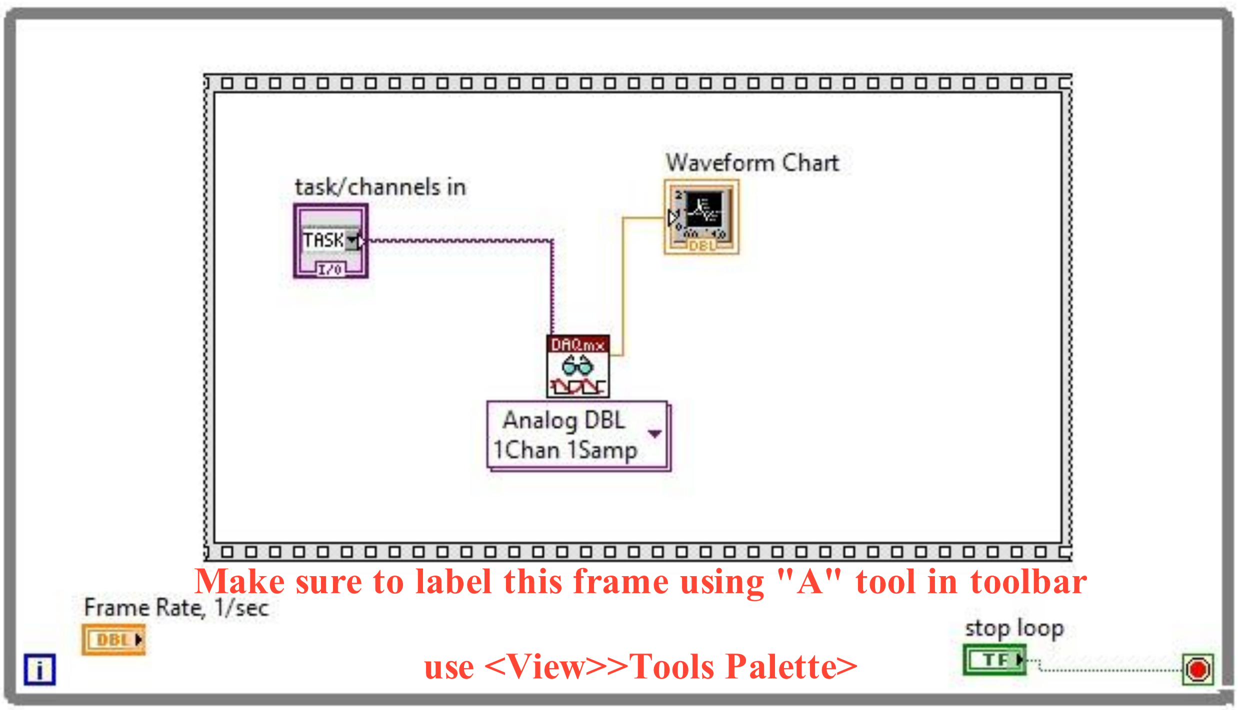


Figure 8. Stacked Sequence Around DAQ Objects of VI.

e. Right clock on center top of the sequence frame and select "Add Frame After" from pop-up Window.

f. Go to the Front Panel window and insert a digital controls (under the Controls»Numeric palette). Name the control "Frame Rate, 1/sec"

g. Click to the second frame "Frame 1"…. and drag your Frame rate icon to the frame.

h. Right click on block diagram and select "Timing>>Wait (ms)" from the Functions pallet. Select the "wiring tool" from the tools pallet, then select from top tab … "Help>> Show Context Help"

i. Hover the wiring tool over the "Wait.vi" .. and you will see help files on the operation of this object. (YOU CAN USE THIS PROCESS FOR HELP ON ANY LABVIEW BLOCK DIAGRAM OBJECT) **Select wiring tool from tools palette**

j. Convert your frame rate to mili-seconds and wire to the "Wait.vi". Figure 9 shows this calculation logic. The necessary operators are found under the "Functions>>Numeric" pallet. This logic tell the VI to wait the calculated number of milliseconds after each frame execution.

k. Switch to the front panel and run your VI with different settings for Frame rate (must be a positive value) .. what do you observe as the frame rate gets larger?

l. On the Block diagram, set the frame rate to 100 samples/second. The right click on the control and select *"Data Operations >> Make Current Value Default"*

m. Finally, add a control to determine how many time the vi loop has run when the start arrow has been hit.

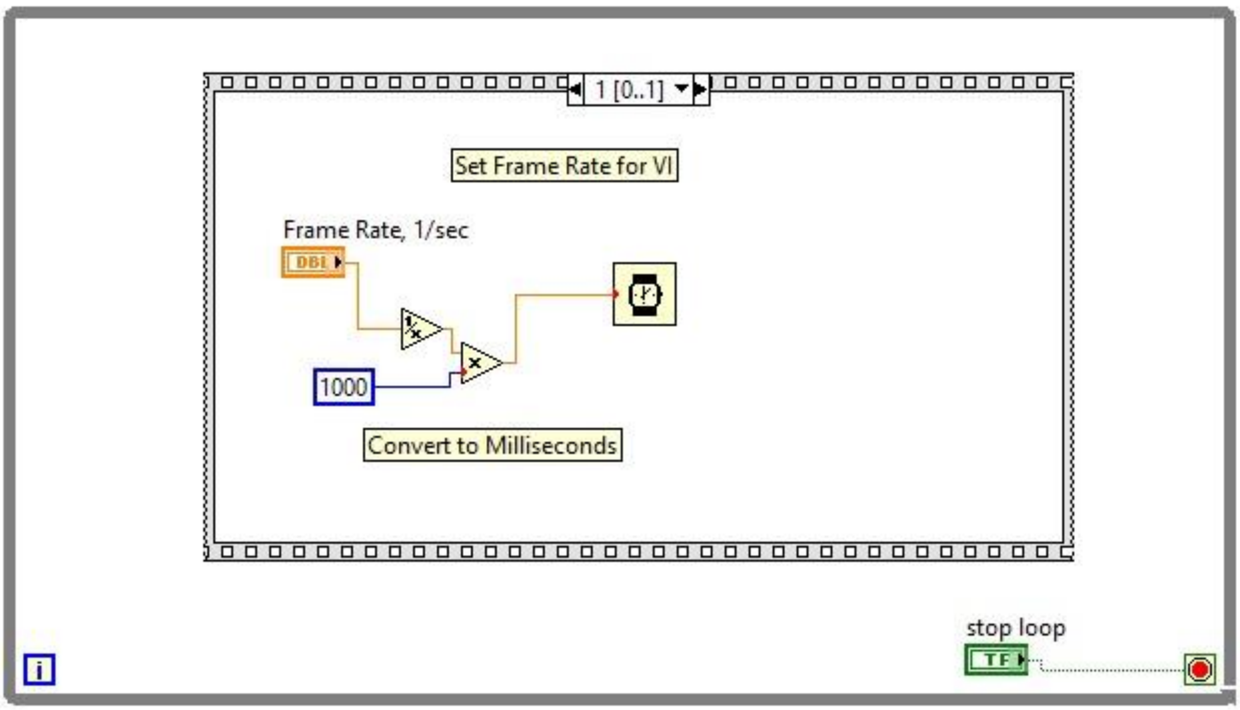


Figure 9. Logic to Control Frame Rate of VI.

3) *Have your TA check your progress.* You may want to use the Positioning tool to rearrange some of the icons to make the program clearer. In general, it is best to place input terminals on the left and output terminals on the right. Also, the wires in between should not cross unless absolutely necessary.

4) On the Front Panel, click on *"Window>>Tile Left and Right" … Your Front Panel and Block Diagram should look like Figure 10 at this point*

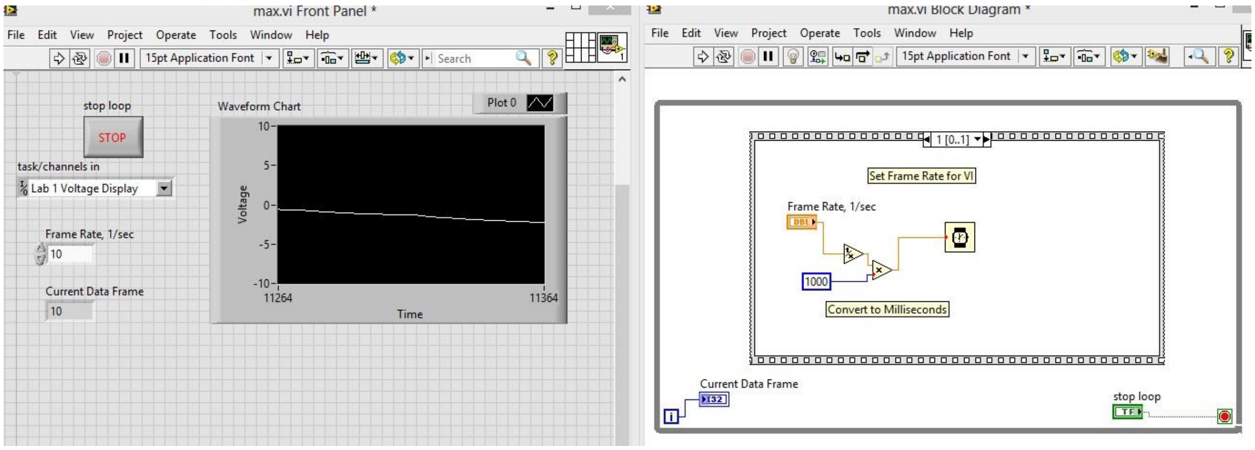


Figure 10. Front Panel and Block Diagram.

B) Taking the Lab Measurements

This part of the lab requires you to do some experimenting and use the VI you have Built in Part A

1. Connect a BNC cable to one analog input channel of the National Instruments NI-BNC 2120 Connector Block, either CH0 or CH1 (BNC connectors). *Figure 8* shows the location of these connectors. Note which channel you programmed in your VI and Make that connection on the terminal block. AI0 is the most convenient connection.

2. Be sure to configure the switches on the terminal block as shown by Figure 11.

3. Connect Voltage probes to loose end of BNC cable. If available, use red probe for positive polarity lead, black for negative polarity lead. Figure 12 shows a typical connection from the BNC cable to the Voltage Probes.

4. Have your instructor check your system and program. Save to your desktop. Drag copy to thumb drive.

5. Hold voltage probe ends to battery terminals (Alligator clips will help here

6. Set the Sample rate to 25 SPS. Run your VI.

7. Measure and record the voltages of the batteries at your workstation. Compare these values to the listed values for the batteries. Populate the columns of Table 2.

Vary the sample rate ... what happens?

"modulate" your voltage reading by tappping the "+" lead to the battery ... what happens?

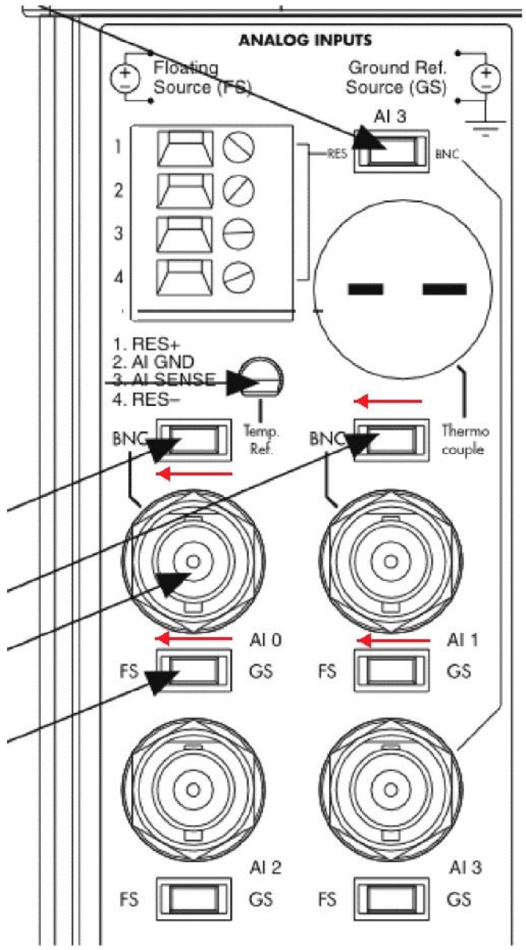


Figure 11. AI connectors on NI-BNC 2120 Connector Block.

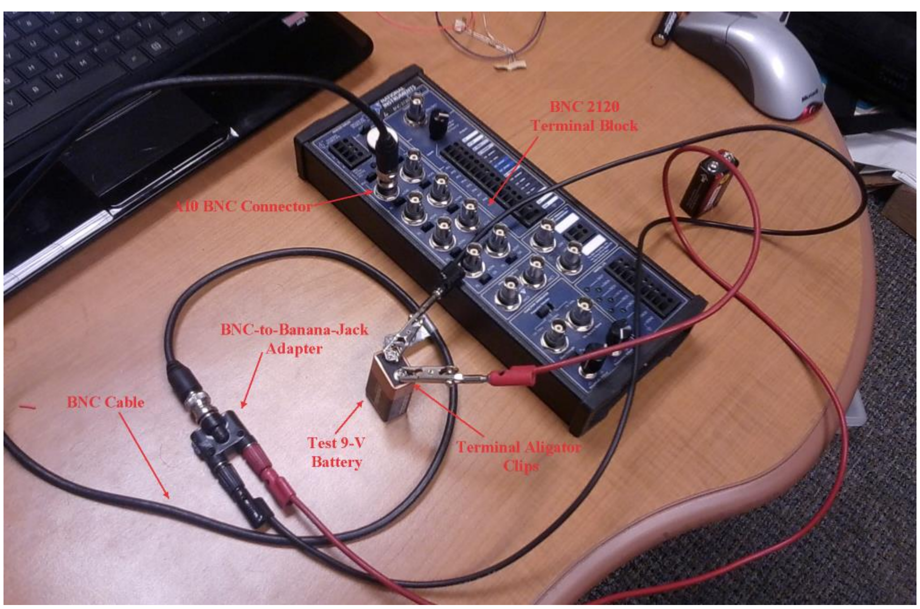


Figure 12. AI BNC Connections to Voltage Probes.

Table 2. List of Measured Voltages.

Battery	Nominal Voltage	Sensed Voltage

You may Use an .xlsx spreadsheet

8. What do you notice about your Voltage display when the probe ends not connected to the battery? Waive the loose wire ends around … what happens? .. The loose wires are acting as antenna .. and the DAQ is sensing the induced current flow along the wire. Bad for noise! Even worse if that signal is supposed to control something!

9. Now Put a *25-30 kΩ* resistor in parallel with the leads (see Figure 13). What is the effect on the readings now for the loose wires.? Repeat the measurements of Table 2 with the resistor in Parallel. Log these measurements on table 3. Any Noticeable Effect? What can you conclude?

Figure 13. Resistor in Parallel with Voltage Probe Leads.

Table 3. List of Measured Voltages with Parallel Resistor between Leads.

Battery	Nominal Voltage	Sensed Voltage

Complete Italicized Items Below (attach extra sheets if necessary)

- *Executive Summary: An introduction to your project similar to that of any other informal report. State what you have done and the objectives of the work.*

- *Submit paper copies of your VI, Both Block Diagram on Front Panel. Include an itemized description of all files that were generated. That means the filenames, where they are located, and what function they perform.*

- *Describe at least three uses for your voltmeter program.*

- *Look up and Define the Term "Electrical Termination" .. how does that apply to our Lab results?* **How did termination help your Voltmeter VI to perform better?**

- *Conclusions you have drawn about using LabVIEW to perform measurements. You should have some sort of idea about how easy or difficult LabVIEW is to use. Just give your personal thoughts.*

Complete the Quiz Below

1. Block Diagram is pictorial description or representation of a program or algorithm. ○ true ○ false
2. Indicator indicates the input value like input sine wave or temperature. ○ true ○ false
3. It is okay to stop any running VI by pressing the stop button. ○ true ○ false
4. The following are all part of LabVIEW function: numeric constant, mathematical functions, comparator, all subVIs. ○ true ○ false
5. Functions and subVIs are nodes in VIs. ○ true ○ false
6. Data acquisition is a process of acquiring data. ○ true ○ false
7. VI stand for Visual Instrument. LabVIEW is a tool for writing VI using G. ○ true ○ false
8. When construting a VI you do G programming in the front panel. ○ true ○ false