WORCESTER POLYTECHNIC INSTITUTE
MECHANICAL ENGINEERING DEPARTMENT

Engineering Experimentation

Laboratory 1: Digital Ohm Meter

COURSE No.: ME-3901, D'2012
TEXTs: Required: *Experimental methods for Engineers, 8ed*
J.P.Holman, McGraw-Hill, 2012
Recommended: *LabVIEW ™2009*, Student Edition with CD
R. H. Bishop, Prentice Hall, 2010

INSTRUCTORS: *Cosme Furlong*
Office: HL-151, x 5126
Email: cfurlong@wpi.edu

Christopher Scarpino
Office: HL-153, x 4927
Email: cscarpino@wpi.edu

WEB PAGE: http://users.wpi.edu/~cfurlong/me3901.html
LECTURES: M and W, 1:00 PM, HL-218
SECTION MTG: 11:00-12:50 PM (D01),
1:00-2:50 PM (D02),
9:00-10:50 AM (D03), Tu and Th, HL-031
SUBJECT: Course outline
DATE: 12 March 2012

TA's: *Ivo Dobrev*
Office: HL-151, x 5125
Email: ivo_d@wpi.edu

Peter White
Office: HL-151, x 5125
Email: pawhite@wpi.edu

Objectives

The objectives of this laboratory are to expose the user to LabVIEW Software, to understand how electrical resistance is measured, the technicalities of analog to digital conversion (A/D conversion), and the statistical analysis of data.

The user will create an Ohmmeter VI (Virtual Instrument) having both "analog" and digital meters on the front panel. The instrument will measure and record resistance readings that are written to Excel files.

A batch of ten carbon film resistors having the same resistance markings will be supplied to each lab group. You will compute the average and standard deviation of the sample of ten resistors and determine the bias error and the probability of a resistor exceeding the tolerance specification marked on the resistor.

A set of three resistors marked "Mixture" will also be provided. They will test the capabilities of your instrument to measure very low and very high resistances.

You will also observe the effect of temperature on a resistor using one of the ten resistors from the batch by simply holding it between your fingers and observing the change in resistance with time as the resistor temperature stabilizes.

Finally, you will verify the number of bits of the Analog to Digital convertor hardware used in the lab.

Background

The background information of this lab involves two very broad areas of study in instrumentation:

1) graphical computer programming; and
2) analog-to-digital conversion.

Both areas can be extensive, however, the following short introduction allows one to take simple measurements.

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. For example, instruction 3 in Figure 1-1a requires data calculated in instruction 2. Therefore, instruction 2 must execute before instruction 3. Therefore, instruction 3 has a data dependency on instruction 2. Note that instruction 2 has a data dependency on instruction 1. Because instruction 4 does not require the result from any other instruction in the sequence, it has no data dependencies on instructions 1, 2, or 3. Therefore, instructions 3 and 4 are data independent. Because instruction 4 is data independent with all the other instructions, it does not matter when it executes. If one plays a video clip of this action it can be seen that Instruction 1 is completed first followed by Instruction 4, Instruction 3, and Instruction 2, respectively.

1. Add **A** to **B**
2. Add **C** to Sum of **A** and **B**
3. Divide Sum of **A**, **B**, and **C** by 3
4. Subtract **A** from **C**

Fig. 1-1a. Sequence of instructions to be programmed in a computer.

Instructions shown in Figure 1-1a can be programmed into LabView in the form of a Graphical User's Interface (GUI). Figure 1-1b shows "Front Panel" of LabView with GUI corresponding to instructions shown in Figure 1-1a.

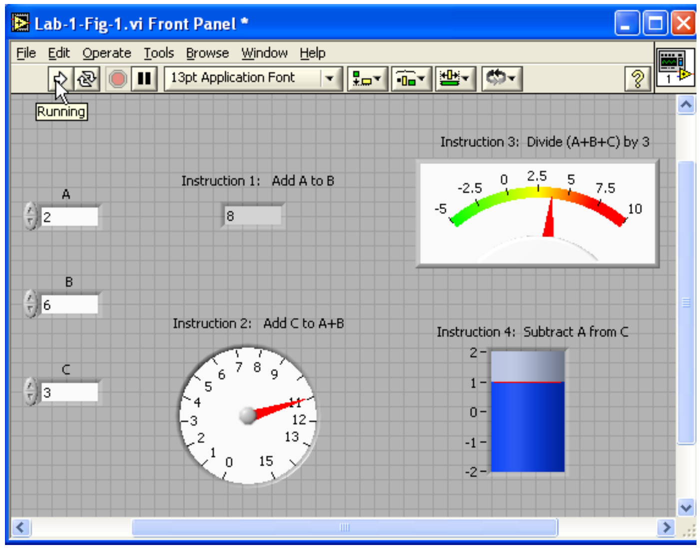

Fig. 1-1b. *Front Panel of LabView*: GUI showing the programming of instructions from Figure 1-1a.

This discussion of data dependency leads to a new way of programming. If you specify the operations and the data dependencies, the computer can execute the instructions in any order that protects the data dependencies. Now you need a way of easily specifying data dependencies. So, if you can draw a block for each operation and connect the blocks to show the dependencies, you can program the computer by drawing a picture. For most people, pictures/icons are much easier to understand than a list of instructions.

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 1-2 shows the operations associated with the set of instructions from Figure 1-1a.

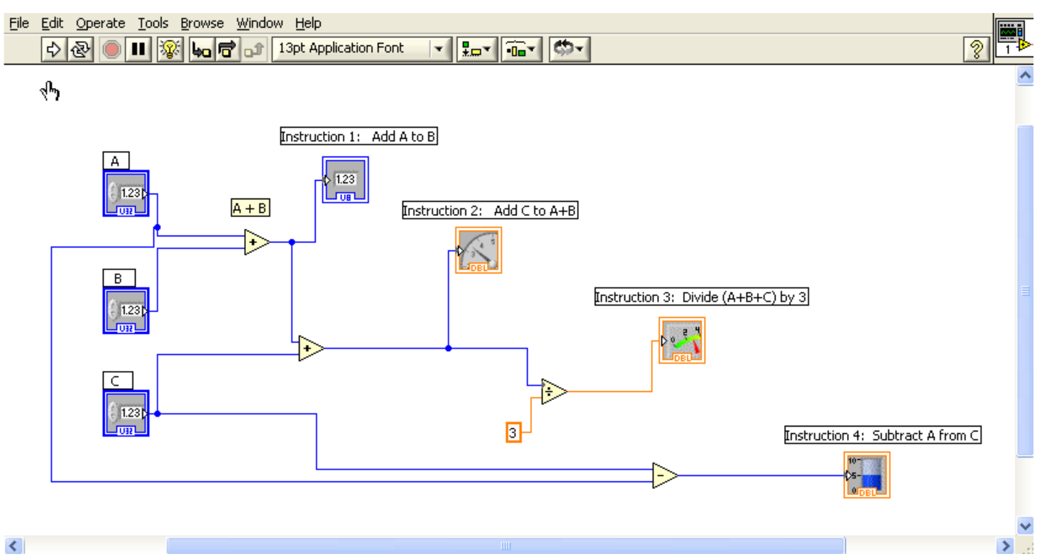

Fig. 1-2. *Block Diagram of LabView*: this is the logic diagram used to construct the GUI shown in Figure 1-1b.

A LabVIEW program, called a Virtual Instrument (VI), is a two-window system:

1) the code is in one window (block diagram), as shown in Fig. 1-2; and
2) the graphical user interface (GUI), appearing in a separate window, as shown in Fig 1-3.

In this example, the boxes on the left (labeled A, B, and C) in Figure 1-3 are CONTROLS (<u>Inputs</u>), and the various dial INDICATORS on the right are the <u>Outputs</u>.

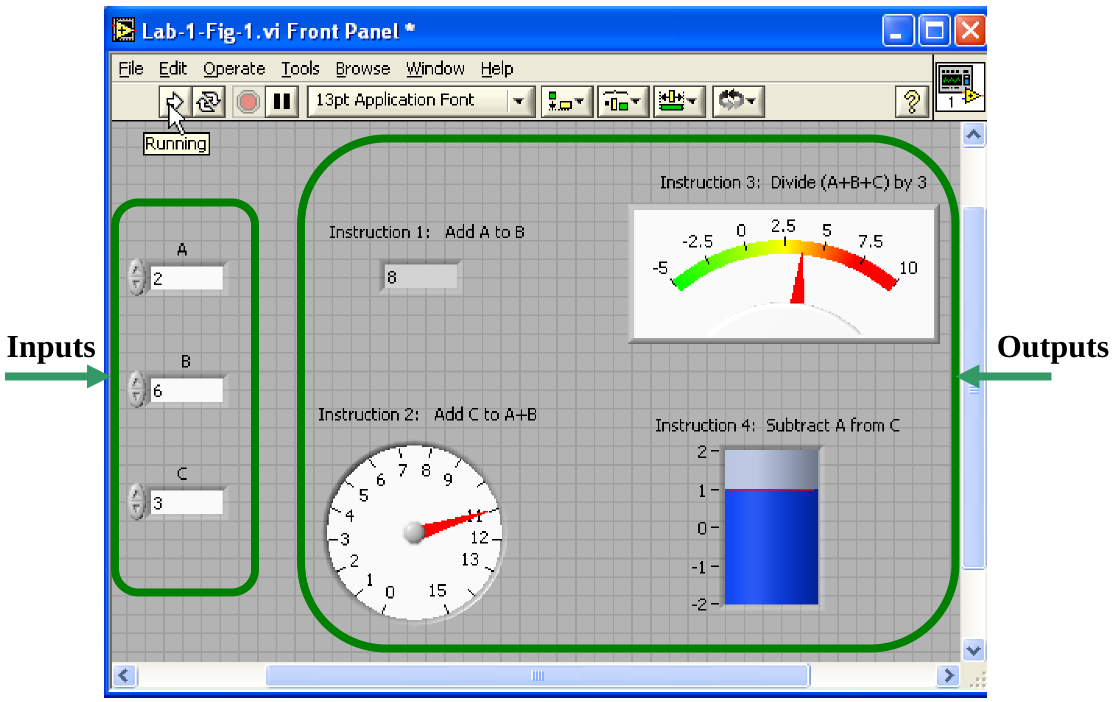

Fig. 1-3. *Front Panel of LabView*: GUI of Figure 1-1a showing *inputs* and *outputs*.

Figure 1-4 shows a more complicated program. This program reads an analog input voltage and displays it on a chart. The gray box around the program is a While Loop. The program elements inside the While Loop will execute repeatedly as long as the While Loop control is true. That is, as long as the variable stop is false. Stop is the button on the front panel shown in Figure 1-5. When the user presses the button, stop becomes true, and the value, which feeds into the While Loop control, is inverted (changed to false). When the While Loop in this example stops, there are no other program elements to execute, and the program stops running.

The "work" being done in the loop includes a block labeled AI ONE PT, which performs the operation of getting a voltage from the channel specified by Channel and the device specified by Device.

AI ONE PT reads one voltage reading from each of the specified channels on the specified device. The output of AI ONE PT is a list of voltages in a structure called an array. The terminal labeled Voltage Display is the connection point for the chart in the front panel shown in Figure 1-5. The update of the chart is such that every time a new number is input to it, the new number is graphed along with all the previous numbers. The programmer can specify a limit on the number of points that can be graphed at one time.

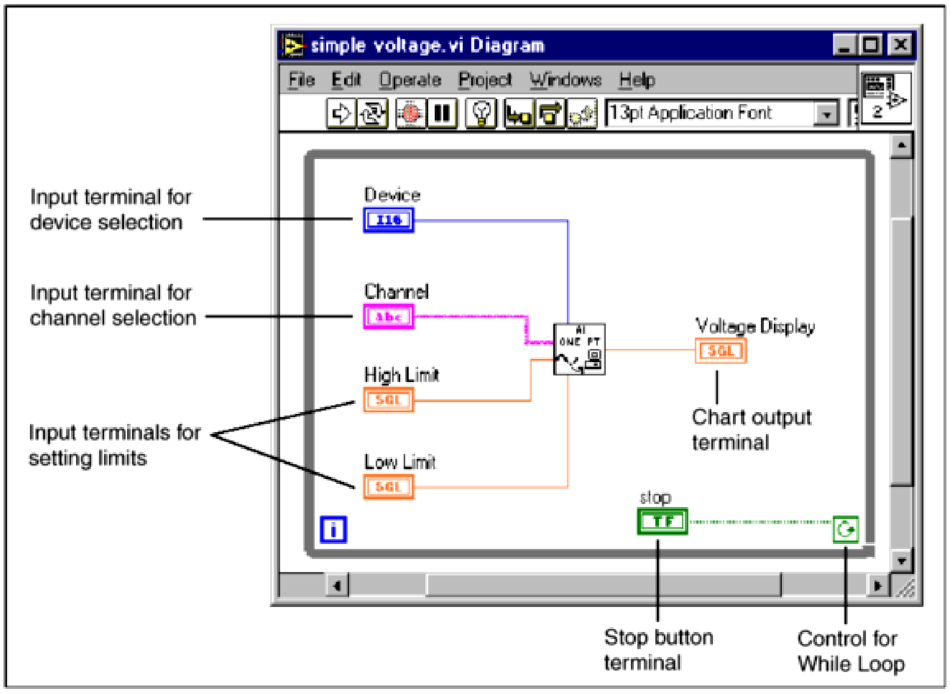

Fig. 1-4. Block diagram of LabVIEW Program to read a Voltage from a Single Channel and Display.

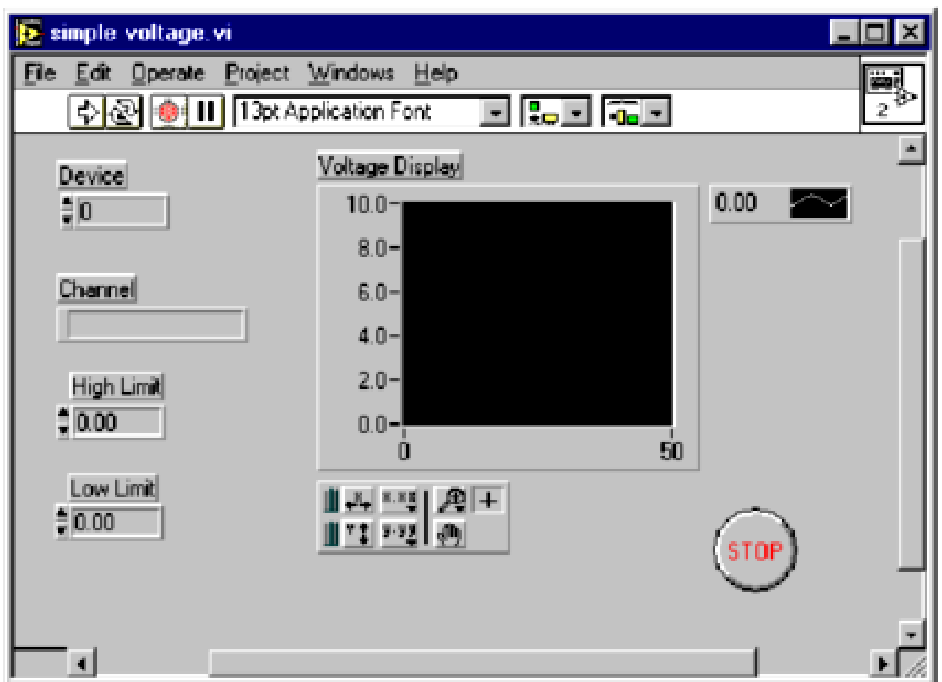

Fig. 1-5. Front Panel of a LabVIEW Program that Reads and Displays a Voltage Waveform.

Notes before staring

1) Review Chapters 1 and 2. Bishop.
2) Start an Internet Browser and go to http://www.ni.com. Find your way through "NI Developer Zone" to "LabVIEW Zone" to "Learning Center" and run the various "Interactive Tutorials"

   It does NOT require that you have LabVIEW on your system.
3) One interesting 'tutorial' can be found at: http://www.iit.edu/~labview/Dummies.html
4) Familiarize yourself with the use of "shortcuts" as well as Command/Tools in LabVIEW. See Figure 1-6.

Command/Tool	Purpose	Used When	Picture
Delete key	Deletes selected objects	There are unwanted objects in the program	<Delete>
Ctrl-S	Saves files	You want to save your changes	<Ctrl-S>
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	

Fig. 1-6. Some of the most common shortcuts and Command/Tools in LabVIEW.

Measuring Electrical Resistance

Ohm's law provides the relationship between Voltage, *V* [volts], current, *I* [amps], and resistance, *R* [Ohms] as

$$I = \frac{V}{R} \quad .$$

Our hardware is capable of generating (Analog Output) and measuring voltages (Analog Input). We will create a circuit as shown in Figure 1-7 with two resistors in series, *R1* and *R2*. *R1* will be known and *R2* will be unknown. By applying a voltage, *V1*, across the two resistors and measuring the voltage, *V2*, across the unknown resistor, *R2*, we can compute the unknown resistance.

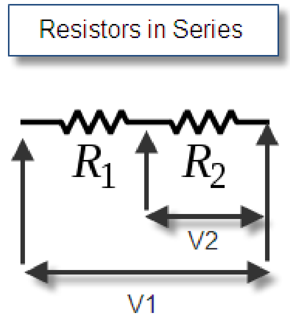

Fig. 1-7. Circuit of a voltage divider.

Figure 1-7 represents a voltage divider. The voltage *V1* is applied to an equivalent resistor of magnitude (*R1* + *R2*). The voltage *V2* is proportional to the ratio *R2*/(*R1* + *R2*). Hence you can compute *R2* from the measured values of *R1*, *V1* and *V2*.

LabVIEW Ohmmeter

An example of the front panel and block diagram for an Ohmmeter is shown in Figure 1-8. Your front panel will not necessarily look like this but it will have all the same information and controls. To build example shown in Figure 1-8, please refer to instructions given during your lab session.

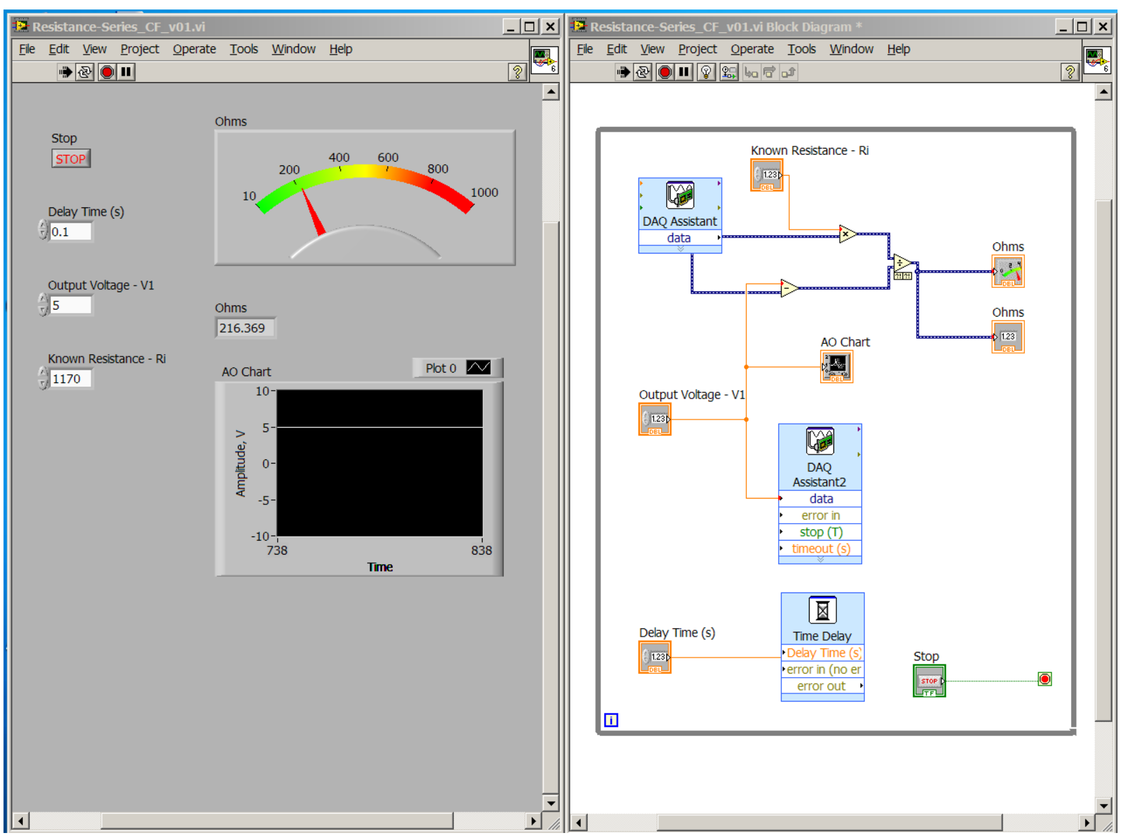

Fig. 1-8. Suggested VI to measure electrical resistance.

Checking functionality of your Ohmmeter

In order to check out your VI you need to take two resistors and connect them as illustrated in Fig. 1-7. BNC cables and alligator clips are provided for this. A good starting value to use for ***R1*** is between 1k and 10k Ohms. Resistor ***R1*** is the reference resistor and all measured resistors are compared to this value. You will measure this precisely with the precision Ohmmeter provided. A color code table is posted on the web for ME 3901. Start measuring with one of the 10 resistors in the batch. Take 5 readings and open the output file to verify that the header and readings are organized in columns.

Readings (*recommended, modify according to the functionality of your VI*):

1. Batch of 10 resistors. Settings: 1000 points, one output reading.
2. Mixture of resistors. Settings: 1000 points, one output reading.
3. Temperature experiment. Settings: 1000 points, 100 measurements at a time delay of 200 milliseconds so that it takes 20 seconds to complete the measurement.
4. Resolution measurement. Settings: 1 average, 100 measurements at 1 millisecond time delay. Make sure you know the range setting so that you get the correct answer. See details below.

Mixture readings. Take accurate readings of each of the three mixture resistors. Make sure the readings are accurate. Adjust the Max limit to give the best precision. Report the values and whether they are within the tolerance specification marked on the resistor.

Resolution measurements. Begin by setting the write to file output format to a high precision such as "%.8f" and take say 100 readings (remember just one average). Using Excel, sort the readings and plot the **Voltage** reading (not ohms). You should get a plot similar to the one below. What is the significance of the spacing between readings? From this information compute the number of bits in the A/D conversion and compare it to the label on the NI USB-6229 hardware. You need to determine the actual full scale **Voltage** range that the VI was using when you took the measurements.

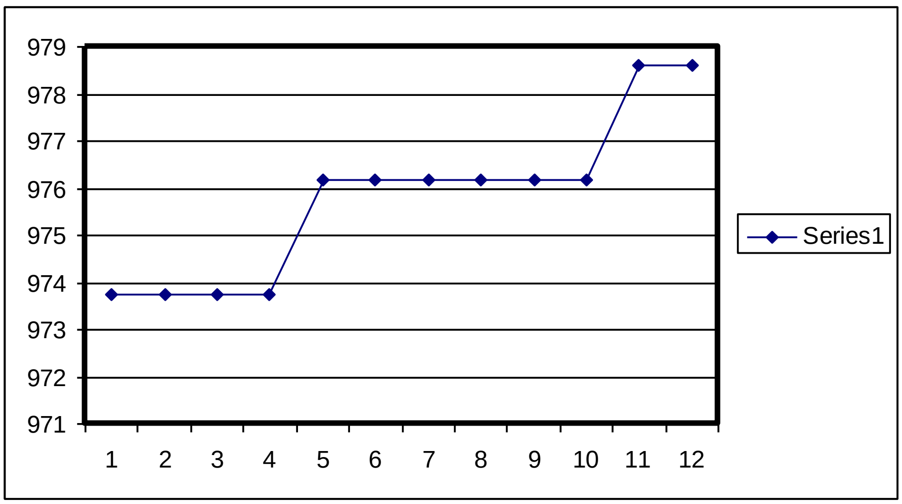

Fig. 1-9. Plot of readings to determine measuring resolution.

Careful examination of the above figure shows that the readings can only have certain values. In this case the jumps are approximately 2.4 units. This illustrates the resolution of the measurement.

Resistance vs. Temperature Observation. Take one of the Batch resistors and set the number of readings and the milliseconds between measurements so that it takes about 20 seconds to record the data. Set both "No. of AI Samples" to 1000 so that you get better resolution. Start recording and then hold the resistor between your fingers to warm it up. At about half way across the chart release the resistor so that it cools down. Capture the waveform chart using Snagit so that you can include it in your report. Does the resistance increase or decrease with temperature. Discuss the consequences of a resistor connected to a regulated voltage near the wattage rating of the resistor.

Calculations

Bias and Precision. Compute these errors for the batch of 10 resistors. Remember that the bias is the difference between the average of the measured values and the marking on the resistor. It reflects the calibration of the machine that made the resistors.

Compute the probability that a resistor in the batch could have a value exceeding the tolerance value marked on the resistor (gold band indicates +/- 5)

List of Equipment:

At the end of your report document the instrumentation used along with calibration data. Create a table as shown below. Fill in the missing information specific to your work station.

Instrument	Model No.	Manufacturer	Serial No. or ID	Calibration Date
Computer	Use label on front of Computer			
Digital Meter	hp-----	Hewlet-Packard		
BNC DAQ	NI USB-6229	National Instruments		

Since this information will be used for almost all of the labs get it right on the very first report.

When you write the report, you might find that you did not save as many screen images that you want for the report. You also might not have saved as much spreadsheet data. If so, you need to repeat some of these procedures after normal lab time.

Fully document all procedures and results!! Store your data in electronic media.

Have fun!!

References

J. M. Sullivan, *Laboratory 1: Digital Ohm meter*, Engineering Experimentation, WPI, 2009
J. Hall, *Laboratory 1: Digital Ohm meter*, Engineering Experimentation, WPI, 2010
T. G. Beckwith, R. D. Marangoni, and J. H. Lienhard, *Mechanical Measurements*, 6th ed., Prentice Hall, 2007
R. H. Bishop, *LabVIEW 2009*, Prentice-Hall, 2010

* * *