Development of LabVIEW Software with NIEHS DOS-based Software Program Electron Spin Resonance (ESR) Capabilities for Post-processing ESR Spectra

Meade C. Edmunds ’14 and Herbert J. Sipe

Department of Chemistry, Hampden-Sydney College, Hampden-Sydney, VA 23943

A variety of operating systems and programming environments have been developed since the creation of the first digital computer to analyze and manipulate electron spin resonance (ESR) spectra. Although outdated by today’s standards, a NIEHS DOS-based EPR software program contains capabilities for post-processing ESR spectra lost in the most up-to-date modern Windows-based EPR software program, WinSim, including the ability to upload and view multiple ESR spectra as the same time. The virtual instrument created in this experiment uses the LabVIEW 2011 programming environment to allow a user to upload and display multiple native ESR data files at the same time so comparisons can be more easily made between spectra.

INTRODUCTION

Electron paramagnetic resonance (EPR) spectroscopy is a technique for studying the interaction between electromagnetic radiation and magnetic moments that arise from the presence of one or more unpaired electrons. While each electron possesses an intrinsic spin similar to the spin of atomic nuclei studied in nuclear magnetic resonance spectroscopy, only a molecule containing an unpaired electron has an overall net spin moment. This net spin moment experienced by an unpaired electron is necessary for interacting with the applied electromagnetic field, which in turn produces a measurable EPR resonance signal. (1, 2)

A basic EPR spectrometer consists of six main components: a microwave source, resonator, magnetic field, detector, amplifier, and modulation system. (3) The source supplies variable radiant energy in the medium-frequency microwave region to the resonator—a cavity that contains an appropriate resonant frequency according to its dimensions. An external magnet likewise supplies a force to the resonator in the form of a static magnetic field. The frequency of the source is tuned in order to maximize the magnetic field at the location of the sample as it moves through the cavity. The absorption of an incident microwave photon by an unpaired electron within the sample occurs when the quantum energy of the photon is equal to the separation of the two energy levels of the unpaired electron. This change in energy produces absorption lines that ultimately cause a change in current within the detector, which is then amplified and recorded. The final step of the process converts the incoming electromagnetic radiation to a readable spectrum using a Fourier transform. (1)

The JEOL RE1X electron paramagnetic resonance spectrometer currently used in the Hampden-Sydney College Chemistry Department operates using software written in the programming environment Laboratory Virtual Instrument Engineering Workbench (LabVIEW). Primarily, LabVIEW is used for data acquisition, instrument control, and industrial automation through graphical notation. In terms of data acquisition, it is specifically designed to take measurements, analyze data, and present the results to the user. (4)

This paper describes the process by which the LabVIEW 2011 programming environment was used to develop a very clear and visual interface LabVIEW virtual instrument (Chart Three ESR Text Files.vi) that contains capabilities found in the DOS-based EPR program but lost in the modern WinSim program for post-processing EPR spectra. To our knowledge, no modern EPR program currently exists that contains these specific capabilities. In particular, we focused on creating a virtual instrument that is capable of retrieving the contents of multiple saved EPR spectra using their individual file paths, calculating the magnetic field values (B(I)) that correlate with the incoming spectrum intensity values (S(I)), and displaying each spectrum simultaneously on separated plots regardless of the specific number of data points contained within each file.

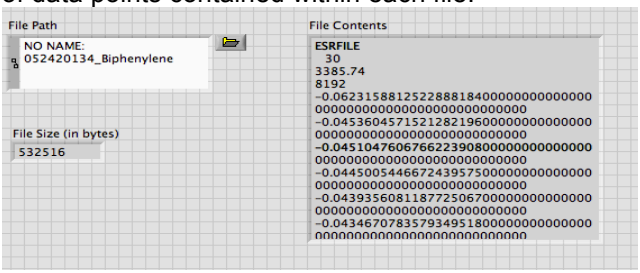

Figure 1. Virtual instrument front panel created in step one of the project. The user is able to input a native ESR file path and display the file’s contents. The contents displayed above are from a native ESR data file of biphenylene.

METHODS

The virtual instrument for plotting multiple ESR spectra simultaneously was made using the LabVIEW 2011 programming environment and a MAC OS X Version 10.6.4 Macintosh computer. All icons used to create the virtual instrument are available with the standard LabVIEW 2011 programming environment software sold by National Instruments. The created virtual instrument is compatible for both Microsoft and Apple operating systems.

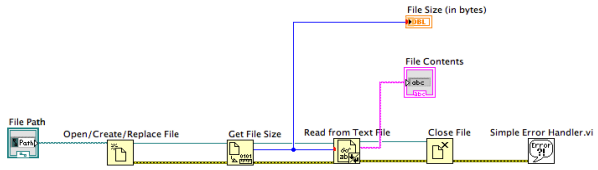

Figure 2. Virtual instrument block diagram created in step one of the project. The block diagram functions by opening a native ESR data file into a file refnum and reading the contents of the file into a string indicator.

RESULTS AND DISCUSSION

The design of the created LabVIEW program consists of four main parts, each with its own function, including the ability to upload three previously saved ASCII-formatted ESR files using individual file paths, convert the uploaded ESR spectrum intensity values into three individual arrays, calculate the magnetic field values that correspond to uploaded spectral intensity values for all three files and display the three ESR spectra within a single XY graphical window according to their individual magnetic field and spectrum intensity values, and display the three spectra as stacked plots within a single chart window. The program was originally written for a single ESR file and then copied and re-wired twice to enable up to three ESR files to be uploaded and displayed. Each section was added in a step-wise fashion and the individual steps undertaken to construct the final, functional LabVIEW program are explained in further detail below.

Step 1: Upload a native ESR data file

The virtual instrument front panel and block diagram created to upload a native ESR data file are shown in Figure 1 and Figure 2, respectively. The virtual instrument block diagram functions by opening the contents of a file into a control reference (refnum) using an Open File function, reading the contents of the file into a string indicator, and then subsequently closing the file refnum after the data within the file has been acquired. The Simple Error Handler function wired throughout the virtual instrument indicates the source of any errors if a problem occurs as the file moves through the code.

In terms of the front panel, the program allows the user to input a single native ESR data file path into the file path control, and display the contents contained within the file in the string indicator labeled “File contents.” The front panel also contains a numeric indicator that displays the file size in bytes of the uploaded ESR data file. The size directly correlates to the number of individual characters contained within the file.

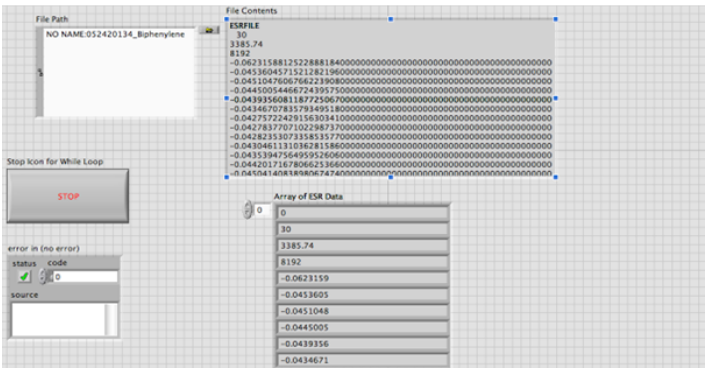

Figure 3. Virtual instrument front panel created in step two of the project. The front panel further displays the contents of a native ESR file in a floating point, one-dimensional array format. The contents displayed above are from the same native ESR data file of biphenylene displayed on the front panel from step one of the project.

Step 2: Conversion of uploaded ESR spectrum intensity values into an array format

The virtual instrument front panel and block diagram created to convert uploaded values contained within an ESR data file into an array format are shown in Figure 3 and Figure 4, respectively. Native ESR data files are often saved in a string format to save space on the computer hard drive connected to the ESR spectrometer because each character in a string format only uses one byte of memory. Although a string format is better for mass data storage, the LabVIEW programming environment is better suited to manage and manipulate data when it is in a decimal floating point

format, which uses four bytes of memory per character.

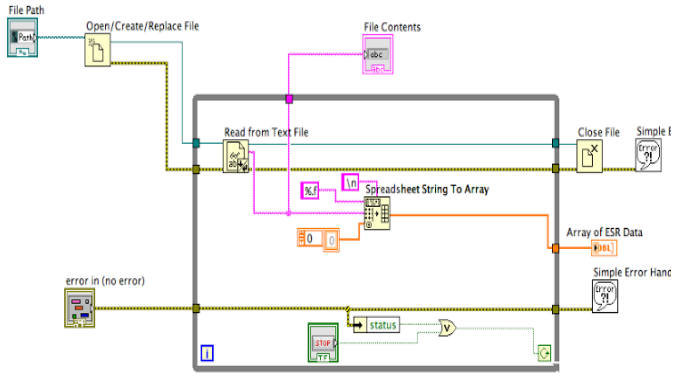

Figure 4. Virtual instrument block diagram created in step two of the project. The block diagram uses a While Structure to register each return character in the original ESR file as a new row within a floating point, one-dimensional array.

The block diagram uses the same file path function as seen in step one to open the contents of native ESR data file and return it as a file refnum. A While Structure and a Spreadsheet String To Array function contained within the loop are used to convert the incoming string-formatted data into the preferred decimal floating point format. With each iteration of the loop, a return delimiter that branches off of the delimiter input of the Spreadsheet String To Array function specifies a single line of data from the original file to be output from the loop as an individual row within an array. Once every line of string data contained within the original ESR data file exits the loop, the file refnum closes.

On the front panel, the user is able to input an ESR data file path into a file path control and display the contents of the file in the same string indicator as seen in the front panel created during step one of the project. Additionally, the front panel displays the decimal floating point values output from the While Loop in a floating point, one-dimensional array, where each line of code in the sting indicator is separated as individual rows within the array.

Step 3: Graphing calculated magnetic field values versus uploaded spectrum intensity values

A native ESR data file generally contains six main parts that correspond to different aspects of the ESR spectrum produced when the original experiment was conducted and is formatted so as to be compatible with the modern WinSim program for post-processing ESR spectrum. The first line of data—displayed as row zero within the array—specifies the file type and is often depicted as "ESRFILE" within the string indicator and a zero within the array. Row one specifies the magnetic field-sweep range (DB), which indicates the distance from the centermost magnetic field value needed to display the complete spectrum. This centermost magnetic field value is contained in the next row—row two—and is called the center of field value (B0). Row three contains the number of spectrum intensity values (S(I)) contained within the native ESR data file (NPTS). These S(I) values are then listed, starting in row four and subsequently including as many rows as there are data points within the file. The number of S(I) values contained within the native ESR data file is equal to a power of two (2x), and generally varies between 1024 to 8192 data points depending on the ESR instrument specifications. The final part typically contains twelve rows of data that specify the instrument parameters when the original experiment was conducted.

The virtual instrument front panel and block diagram constructed to calculate the corresponding B(I) values and display a plot of the B(I) versus S(I) values on a single-plot XY graph are shown in Figure 5 and Figure 6, respectively. The first portion of the block diagram not previously present in step two of this program separates the individual terms used in the above algorithm into individual arrays using two Array Subset functions and a single Decimate 1D Array function. The second portion of the block diagram uses various numeric functions (mathematical operations) to execute the above algorithm and calculate the necessary B(I) values. This portion is mainly constituted by three For Structures, in which the first For Structure outputs the integer values for I, the second For Structure multiplies each value of I with the product of (DB/NPTS), and the third For Structure adds the difference of B0 – (DB/2) to each value output by the second For Structure. The third For Structure subsequently outputs the resulting calculated B(I) values, which are then coupled with the corresponding S(I) values and wired into an XY Graph function.

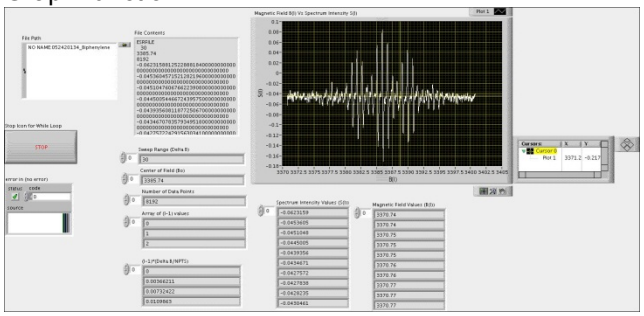

Figure 5. Virtual instrument front panel created in step three of the project. The front panel further displays the magnetic field values and spectrum intensity values in individual one-dimensional arrays and as the X- and Y-axis of a single-plot, XY graph. The contents and ESR spectrum shown above are from the same native ESR data file of biphenylene displayed on the front panel from steps one and two of the project.

The front panel still contains the controls and indicators from step two of the program. In addition, six new floating point, one-dimensional arrays are present on the front panel that indicate the original or calculated values for DB, B0, NPTS, (I – 1), (I – 1)(DB/NPTS), S(I), and B(I), respectively. The final resulting ESR spectrum of B(I) versus S(I) values is displayed in a single-plot, XY graph.

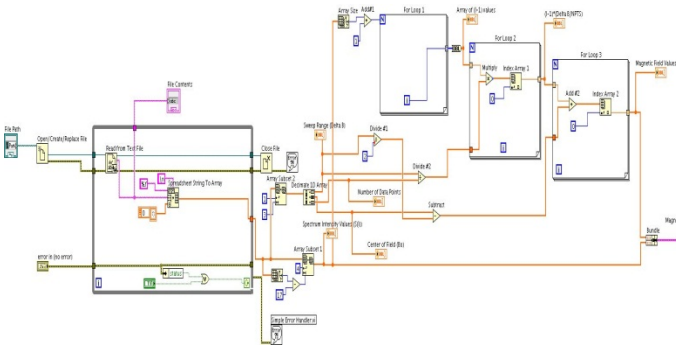

Figure 6. Virtual instrument block diagram created in step three of the project. The block diagram uses the spectrum intensity values contained within a native ESR data file to calculate the corresponding magnetic field values. The block diagram then wires both the calculated magnetic field and spectrum intensity values to an XY Graph indicator.

Step 4: Displaying three native ESR spectra in a stacked plot format within a single chart window

While graphs are extremely useful for displaying a single ESR spectrum, graphs wired to display multiple ESR spectra have many limitations within the LabVIEW programming environment. The two main downsides to graphing multiple ESR spectra within the same graphical window stem from that fact that: 1. Not all ESR spectra contain the same number of spectrum intensity values and 2. Each ESR spectrum contains its own unique center of field value and field-sweep range. When multiple ESR spectra with different values for DB, B0, or NPTS are displayed in a single graphing window, the X-axis range broadens to include all spectra magnetic field values. Typically, the resulting ESR spectra are individually too compressed to analyze and altogether too spread out from one another to adequately make comparisons between them.

A solution to the problems described above comes in the form of a chart display. In LabVIEW 2011, a chart plots a waveform data type—which includes data contained within a floating point, one-dimensional array—as its Y-axis values, and automatically sets its X-axis values to increment in evenly spaced steps for every time a new Y-value is appended to the chart. Because any ESR spectrum's magnetic field values are evenly spaced throughout the entire spectrum, the overall integrity and appearance of the spectrum is still maintained even when its magnetic field values are not used in forming the chart.

LabVIEW 2011 also contains the capabilities for displaying a multiple-plot chart in a stacked plot format, which allows the user to give each plot its own Y scale (termed duplicating) without duplicating the X scale. Thus, multiple ESR spectra can each be displayed within individual plots in the same chart window and with all spectra sharing the same X scale, which allows for easier comparisons to be made between them.

A downside to waveform charts in LabVIEW 2011 is that each plot displayed within the chart window is only capable of storing up to 1,024 data points. As an example, for an ESR spectrum containing 8,192 spectrum intensity values, only the final 1,024 values will be displayed when plotted. The determined resolution to this problem was to amend every uploaded native ESR data file to contain exactly 1,024 S(I) values, regardless of the original number of data points contained within each file.

The virtual instrument front panel and block diagram constructed to display three native ESR spectra in a stacked plot format within a single chart window are shown in Figure 7 and Figure 8, respectively. In regards to the functionality of the block diagram in Figure 8, this description will focus on the uppermost of the three identical sections of code. When only considering the uppermost section of code, no changes were made to the VI created in step three of this program. The portion of code written in this step of the project simply creates a branch off of the wire carrying the S(I) values from the original ESR data file and amends the data set to only include 1,024 data points.

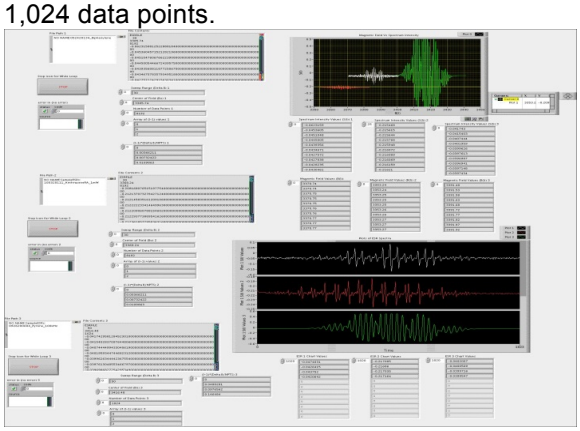

Figure 7. Finalized virtual instrument front panel created during step four of the project. The front panel displays three native ESR spectra within the same multi-plot, XY graphing window and within individual plots of a stacked plot-formatted chart window. The contents and three ESR spectra shown above are from an ESR data file of biphenylene, anthracene, and pyrene, respectively.

The portion of code created in step four is mainly constituted by a series of four Equal and Select Comparison functions, which are wired to ensure the output number of S(I) values is equal to 1,024 values in length. The Equal? Comparison functions operate by outputting either a true or false statement depending on if the numeric value output by the Array Size function is equal to the numeric constant wired to the Equal? function. The true/false statement is subsequently registered by the connected Select function, which then either outputs the numeric value wired to its true statement input if the incoming statement is true or outputs the numeric value wired to its false statement input if the incoming statement is false. Thus, the incoming array size numeric value ultimately is used to determine which rows of the original array of S(I) values should be included in the final, amended array.

For example, if the native ESR file contains 4,096 S(I) data values, the numeric value of 4,096 is output from the Array Size function. This value enters the first Equal? function wired to the numeric constant containing the number 8,192. Since these two input numeric values are unequal, a false statement is output from the first Equal? function and subsequently enters the first Select function. Since the incoming statement is false, the Select function outputs a numeric value of 4,096 because its false input is connected to the wire before the first Equal? function. The numeric value 4,096 then enters the second Equal? function wired to the numeric constant containing the number 4,096. Since these two values are equal, a true statement is output from the second Equal? function, which then subsequently causes the second Select function to output a numeric value of 4 because its true input is connected to the numeric constant containing the number 4. The remaining two pairs of Equal? and Select functions will all produce false statements due to the same principles as described above and so the fourth and final Select function will output a numeric value of four. Ultimately, this numeric value of four will be used to dictate that every fourth row within the original array of S(I) values be indexed into a sub-array containing 4,096/4, or 1,024, of the original S(I) values, which are then subsequently plotted on a waveform chart.

After the uppermost section of the VI was written, the entire portion of the code was replicated twice more and wired together to enable the user to upload up to three native ESR data files and display all three in both a single graphical window and stacked-plot chart window at the same time on the virtual instrument front panel. Thus, all controls and indicators/arrays present on the front panel from step three of the project are tripled in number. Additionally, the front panel contains three new floating point, one-dimensional arrays that indicate the 1,024 S(I) values input into the stacked plot, waveform chart from each of the three uploaded native ESR data files.

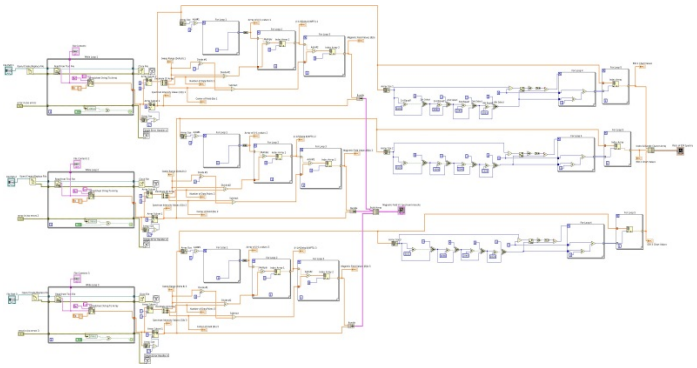

Figure 8. Finalized virtual instrument block diagram created in step four of the project. The block diagram compresses up to three native ESR data files into 1,024 data points in length and then wires each compressed file into a waveform chart indicator.

REFERENCES

1. J.A. Weil, J.R. Bolton, and J.E. Wertz. “Electron Paramagnetic Resonance.” John Wiley & Sons, Inc., 2nd Ed., New York, NY, 2007, 3-10.
2. N.M. Atherton. “Principles of Electron Spin Resonance.” Ellis Horwood Limited, Chichester, England, 1993, 6-11.
3. J.R. Bolton, and J.E. Wertz. “Electron Spin Resonance.” McGraw-Hill Book Company, New York, NY, 1972, 21-30.
4. Travis, Jeffrey, and Jim Kring. “LabVIEW For Everyone: Graphical Programming Made Easy and Fun.” 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2007. Print.