

Familiarization and Improvement of ESR Functionality Using LabView Software

J. Reed Mingione '19 & Herbert J. Sipe

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

Abstract

The Hampden-Sydney College electron spin resonance spectrometer (ESR) is operated via a student-made program which utilizes a popular computer language called LabView. The program itself has been updated several times but still needs improvement. Currently, few students at the college actually understand the language and even fewer possess the skill necessary to understand the complex ESR program. Thus, an extensive LabView familiarization was conducted with the intention of making improvements to the ESR's program.

This study pursued another update so that the program recorded more digits from the ESR output signal to produce a higher resolution spectrum. A separate program was created to display spectra previously recorded as data from the ESR spectrometer. Each newly instituted program has been carefully documented and explained within this report.

Introduction

LabView Fundamentals (1).

The LabView computer language was created by scientists for other scientists to circumvent the esoteric programming knowledge necessary for creating a program. The language particularly excels in aspects of measurement and timed operations. Although the LabView language is built to be more easily understood, its rooting in programming has led many scientists to pursue other methods of instrumentation. However, LabView is practical and reliable enough that learning the software is worth much more than its minorly inconvenient learning curve. Comprehension of LabView software and techniques will be a vital step in the deployment of LabView-based programs.

At its core, LabView is a graphical language, which is a direct contrast to text languages such as java and python. The language is "graphical" because the code appears as a set of pictures as opposed to a line of text. A typical LabView program or operation—such as the one depicted in **Figure 1**—is called a virtual instrument (VI) and consists of a set of operators and terminals connected via unique types of wires which "carry" the information. Dataflow

in LabView is determined by the connection of wires as opposed to the line of code in textual languages. It is possible and quite common to have several operations occur at once.

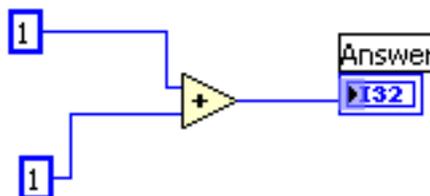


Figure 1. A simple LabView VI. 2 constants feed via wires into an addition operator to yield an integer answer (1).

A signature aspect of LabView's interface is its front and block diagram panel system. The front panel **2** displays the interface used while a VI is running; It contains all relevant buttons, controls, and meters involved in the program. The block diagram **3** shows how the front panel's button operate and communicate with each other. While developing a VI, most time is spent connecting wires and operators in the block diagram panel.

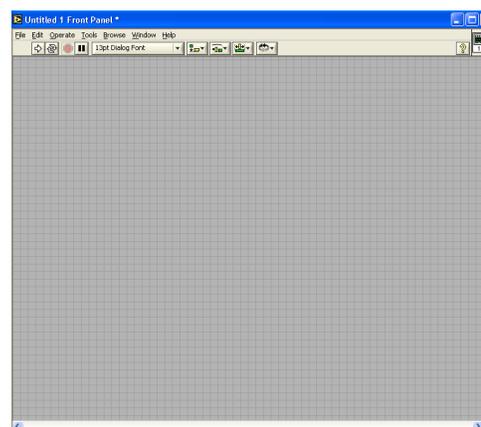


Figure 2. The front panel. This is the user interface (1).

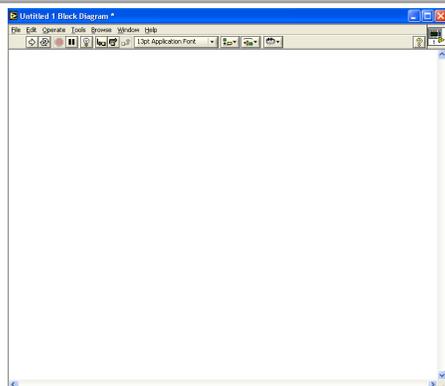


Figure 3. The block diagram. This is the raw code of the VI (1).

LabView is very appealing to scientists for not only ease of familiarization, but also its easy measurement and delay systems. For-loops and while-loops are very important in all computer languages. For-loops allow a certain operation run for a specific number of iterations. While-loops allow an operation to continue until a prechosen event occurs. Textual languages often make loops complicated endeavors, but LabView's interface allows the user to literally draw a loop around the determined operation. Several timers and dials are available in LabView as premade VIs called "express VIs" which simplify the measurement programming down to the click of a mouse.

The necessity of LabView and the ESR

At Hampden-Sydney College, the LabView program is utilized in science departments to quickly devise programs to obtain data and conduct novel experiments. Had another program been used, the familiarization would have taken much longer and most programs would be much larger. Fundamental LabView knowledge is a necessary step in the comprehension of several ongoing operations at Hampden-Sydney College. The chemistry department fully relies on the language to operate its electron spin resonance spectrometer (ESR). The physics department uses the software to constantly monitor local weather patterns. With authorization from the chemistry department, the new knowledge will be applied to the ESR.

Simply put, the ESR is an instrument used by skilled chemists to record and display spectra measured by locally synthesized free radicals. It operates by emitting a microwave frequency of 9.5 GHz from a Gunn diode through a special tube called a waveguide into a microwave cavity containing the radical sample. The cavity is specially constructed to best contain microwaves. An iris screw within the cavity can be used to manipulated the frequencies permitted in the cavity. A magnetic field is applied to the cavity via an electromagnet. Typically, the magnetic field is varied instead of the microwave

frequency. The magnetic field causes the radicals to align parallel and antiparallel to the direction, dependent on the electron's spin state. The field is varied until the difference is great enough that the energy from the microwaves causes the radicals to jump from parallel to antiparallel, which is called resonance. The resonating radicals cause an imbalance in the microwave bridge. The imbalance is recorded by the ESR spectrometer as an ESR spectrum as a function of the field values. Thus, the ESR varies from the NMR in that it is an absorption experiment rather than an emission experiment. The ESR at Hampden-Sydney College is a JEOL JES-RE1X.

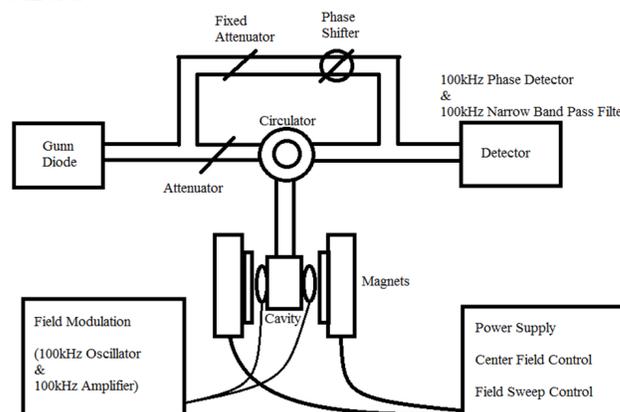


Figure 4. Simple diagram of an ESR spectrometer.

The program used to control the ESR is one of the largest functioning LabView programs on campus. It was originally devised by Chris Lea in 1998 (2) and was subsequently improved by Daniel Armata in 1999 (3). The program reached its most recent and viable state in 2004 under the research of Robert George (4). This final version is named "ESR program.vi" and works on LabView 7.1. There also have been several undocumented updates which have shifted the operation of the ESR in different ways. For instance, in 2014, a new ESR program was created and named "XY ESR 7-2-2014-USE-This-for-ESR.vi." This new version changed the DAC board channel from "0" to "1," but is unique in that the program is openable on both LabView 7.1 and LabView 8.2.1. Another version was created in 2005 and altered ESR program.vi's precision from 20 decimal places to 3. The effects of this alteration have skewed the ESR's results ever since and are a primary directive behind updating the ESR program.

LabView coding opens the possibility to several more aspects of ESR enhancement. The program records the ESR information in a file called DAT. Once saved, the information must be transferred to a separate computer to be redisplayed. Undoubtedly, a program which could read the DAT files and display them correctly on the same computer that creates the file would save a significant amount of time. Thus, another objective of the

research is to create a DAT reader program using LabView software.

Results and Discussion Enhancing ESR precision

Since the mysterious update circa Spring 2005, ESR program.vi has lost 17 decimal places worth of precision. Due to the vastness and complexity of the program, the issue has been avoided for many years. The problem was suspected to have come from the ESR's saving mechanism because the spectra displayed on the program itself showed great precision even when magnified. The ESR program's data processing was traced and the location of the save file mechanism was identified as shown in **Figure 5**.

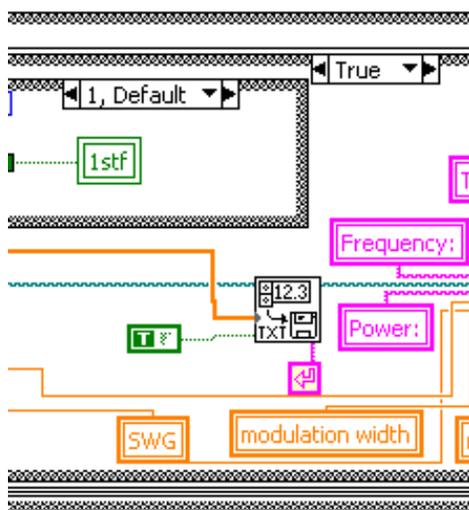


Figure 5. The Write to spreadsheet express VI within the ESR program. Its save format has skewed the precision of the ESR (1).

Upon closer analysis, none of the student-built save VIs controlled the file path of the spectral data. The spectral data was handled by an express VI created by National Instruments called "write to spreadsheet.vi." The contents of the express VI can be seen in **Figure 6**. The problematic component became evident upon analysis of the format which was set at %.3f. The "f" refers to the floating-point format of the number. The .3 refers to the permissible number of decimal points saved to the file. Regardless of its adjustability, the format always defaulted back to %.3f. A new write to spreadsheet program was created with a standard format of %.22f to resolve the issue of decreased precision. The program's efficacy was proven by running a string of numbers with 22 significant figures through the operation. The number precision was unchanged.

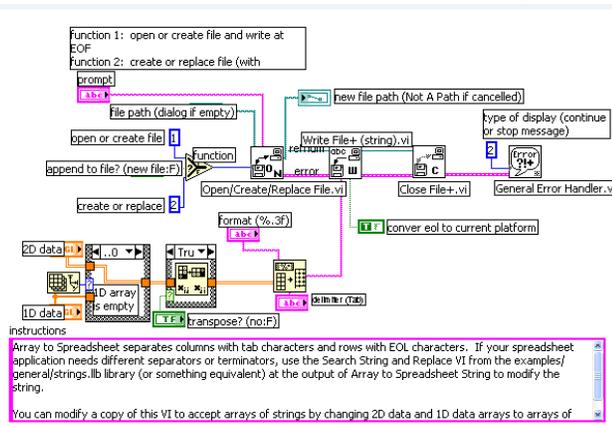


Figure 6. The contents of the stock Write to spreadsheet.vi. Note the format of %.3f (1).

Making a program for reading and identifying DAT files

The format of the DAT files created and recorded to the hard drive by the ESR program consists of 4 lines of text describing what the file is for, the number of points recorded, the centerfield of the spectrum in mT, and the range of the recorded scan in mT. Following this description is 8192 points of voltage readings. Then, 12 lines of descriptor text follow to describe the name of the operator, date, temperature, microwave frequency, power, center field, sweep width, field modulation, receiver gain, time constant, number of data points, and any comments.

```

ESRFILE
30
3382.52
8192
0.000
0.006
0.006
0.006
0.006

```

Figure 7. A sample of the beginning of a DAT file.

A program was designed using LabView to interpret and plot the DAT files according to the user's specifications. The VI was called "DAT reader.vi." The voltage points from the DAT file were systematically inserted into a ramp formula to be plotted over their respective magnetic values. The ramp formula can be described as shown in **Equation 1**.

$$B_1 = B_0 - (\Delta B / 2) + (I \Delta B / 8192)$$

Equation 1. The ramp equation. B_1 is the magnetic field of the current in mT. B_0 is the centerfield in mT. ΔB is the field width of the spectrum in mT. I is the current datapoint of the spectrum in A.

By itself, the ramp formula creates a constant-sloped graph. The slope accounts for the increasing magnetic field values of the data points. Both the data points and ramp formula points were plotted in an XY graph as intensity versus magnetic field in mT. The produced plot is a ramp with a spectrum toward the end as exhibited in **Figure 8**. A range bar was added to synthesize the spectrum from the ramp plot. User interface was added to allow the user to magnify portions of the XY spectrum and identify points with cursors and range controls.

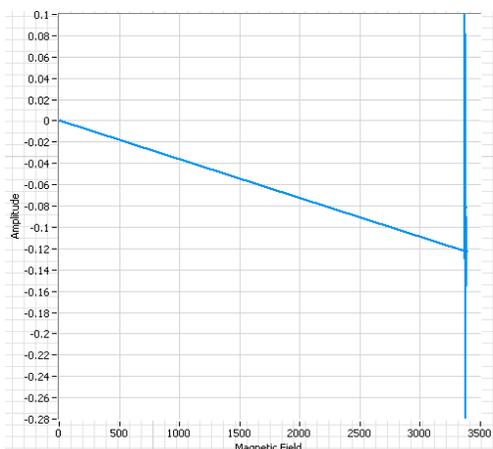
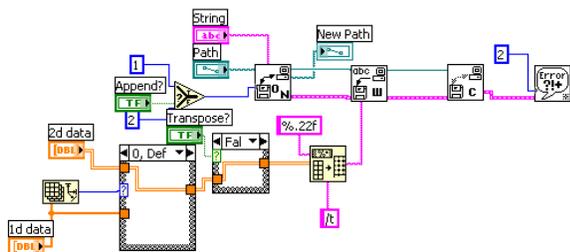


Figure 8. An example of the initial XY graph made by DAT reader.vi. The actual spectrum on this graph is located near the end (1).

In order to obtain all necessary data to make the plot, the centerfield and sweep width are obtained in a subvi popup called “DAT reader info.vi” upon initial start of the instrument. The information is fed directly into the ramp equation.

Fourier transforms and DAT descriptor data was intended to be added to the program, but troubleshooting and general VI creation delayed the process enough that the goals were never fully complete. Due to the open code, these goals are easily attainable in the near future.

VI Diagramming Custom write to spreadsheet.vi



A) Array terminals

The VI contains 2 terminals where one can insert either 1d or 2d arrays. If the data is a 2d array, it proceeds as is. If the data is a 1d array, it is converted into a column before proceeding.

B) File path

If no data is inserted, the user is asked to select an array file to be opened and processed. In the same step, The user is also allowed to choose the file destination. The files are saved as DAT file types.

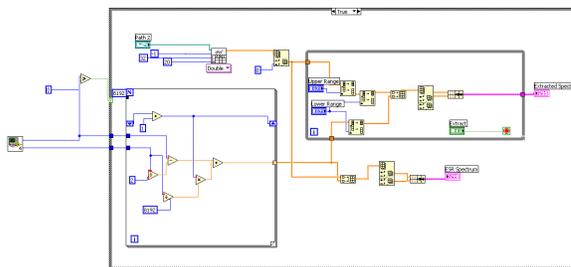
C) Array to string

Once the array is processed, the program converts it to a string. The format is permanently set to %.22f. This distinction means that the data is converted to a floating point integer with 22 decimal places. The data may be misleading because, if there aren't 22 decimal places, the program will add zeros until the decimal places are filled. The delimiter is set to /t. This distinction causes a new line to be made for the next point in the array.

D) Close file path

This portion of the program ends interaction with the new array file and attempts to fix any minor errors the program may have occurred when reading the array.

DAT reader.vi



A) Subvi

The DAT reader info vi was inserted here to obtain information relevant to the plotted spectrum. Wires containing information about a given spectrum's field width and centerfield position transport data into a case structure. In order for the case structure to accept the numbers, the field width must be a real number.

B) Ramp equation

A for loop set to iterate 8192 times for each point measured by the ESR contains the formula depicted in **equation 1**. A case structure takes the final value and adds 1 to the end of each iteration. An auto indexing component stores the final values at each iteration. Thus, the correct sweep width is assigned to the correct data points.

C) Read from DAT

A dialogue box asks the user which file to should be read. Then, a read from spreadsheet express VI extracts the data points from the file. An

index array places the array in a column form to align with the ramp equation array.

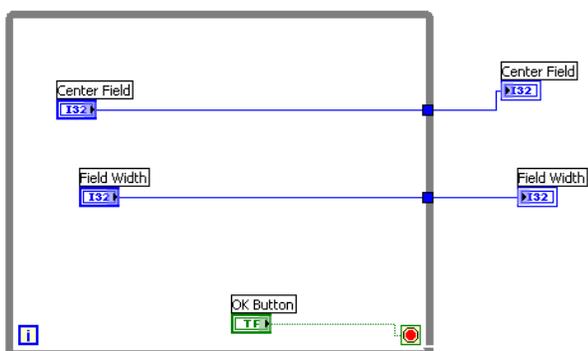
D) Ramp XY plot

The ramp XY plot is an unfiltered graph of all 8192 point of both arrays. The graph is mostly useless, but the interface is available so that a user can decide which field range he or she wants to view.

E) Extracted XY plot

This process displays an enhanced rendering of the desired spectral range. Both arrays are filtered through 2 indexes which can be controlled by the ESR user to find the best image. 2 cursors are present in the graph to compare distances and intensity.

DAT reader info.vi



A) Controls

Both the centerfield and the field width are available as controls on the front panel for the user to input. The data type is a 32 bit integer which rounds each number to the nearest integer.

B) While loop

The data is held until a custom “ok” button renders the loop false. Once the data is released, they can be used by other VIs.

REFERENCES

1. LabVIEW
2. Lea, C. R. *Summer Research 98*; Department of Chemistry: Hampden-Sydney, VA, 1998.
3. Armata, D. A. *Using LabView 5.0 to interface with a JEOL RE-1X ESR Spectrometer*, Department of Chemistry: Hampden-Sydney, VA, 1999.
4. George, R. *Rapid Scan and Signal Averaging Capability for the HSC ESR Spectrometer*, Department of Chemistry: Hampden-Sydney, 2004.